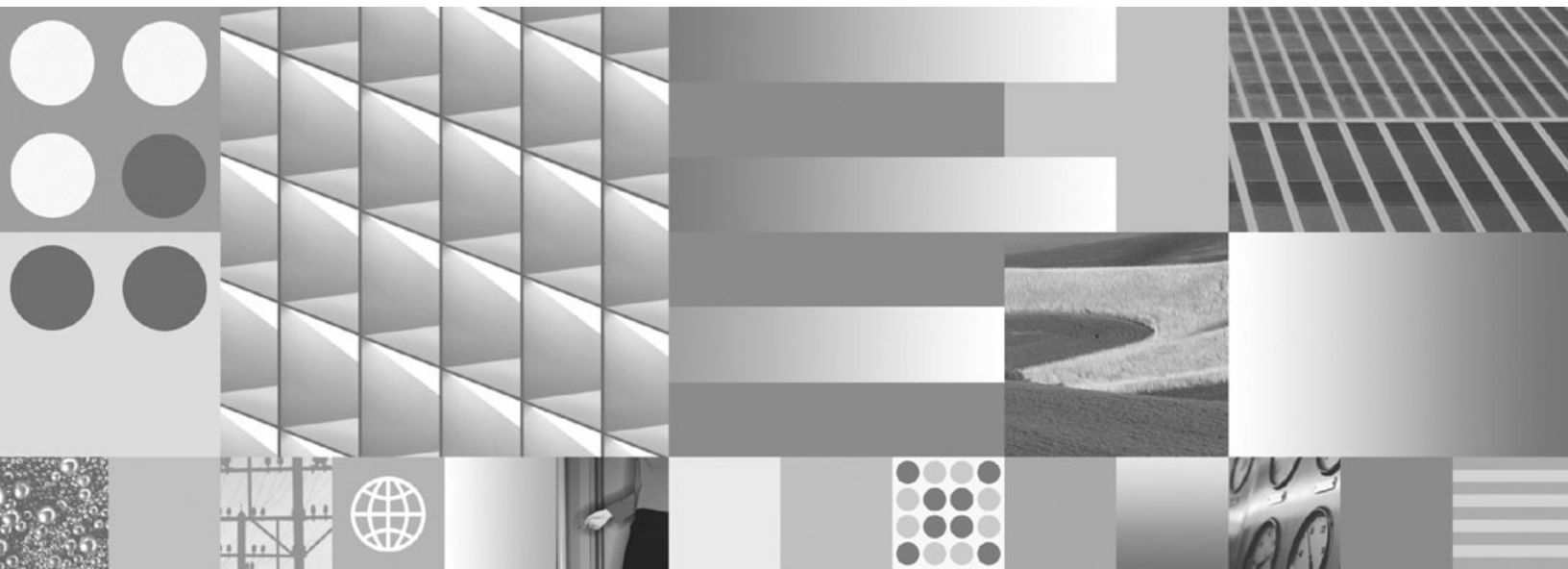


Föderierte Systeme - Verwaltung



Föderierte Systeme - Verwaltung

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 489 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM InfoSphere Federation Server Version 9.7, Administration Guide for Federated Systems,
IBM Form SC19-1020-02,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1998, 2009
© Copyright IBM Deutschland GmbH 2009

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Juli 2009

Inhaltsverzeichnis

Kapitel 1. Föderierte Systeme 1

Unterstützte Datenquellen	2
Der Server mit föderierten Datenbanken	5
Was ist eine Datenquelle?	5
Die föderierte Datenbank	6
Wrapper und Wrappermodule	6
Interaktion mit einem föderierten System	7
DB2-Steuerzentrale	8
Anwendungsprogramme	9
Tools der DB2-Produktfamilie	9
IBM Optim Data Studio	9
Web-Service-Provider	10
Der Systemkatalog der föderierten Datenbank	10
Der SQL-Compiler	11
Das Abfrageoptimierungsprogramm	11
Kompensation	12
Durchgriffssitzungen	13
Serverdefinitionen und Serveroptionen	13
Benutzerzuordnungen	14
Kurznamen und Datenquellenobjekte	15
Gültige Datenquellenobjekte	16
Kurznamenspaltenoptionen	16
Datentypzuordnungen	17
Funktionszuordnungen	18
Indexspezifikationen	18
Föderierte gespeicherte Prozeduren	19
Sortierfolgen	19
Bestimmen der Sortierreihenfolge durch die Sortierfolge	20
Definieren der lokalen Sortierfolge zum Optimieren von Abfragen	21

Kapitel 2. Modifizieren von Datenquellenkonfigurationen. 23

Ändern eines Wrappers (DB2-Steuerzentrale)	23
Ändern eines Wrappers - Beispiele	24
Ändern eines Wrappers (DB2-Befehlszeile)	24
Ändern von Serverdefinitionen und Serveroptionen	25
Einschränkungen beim Ändern von Serverdefinitionen	26
Ändern der Datenquellenversion in einer Serverdefinition (DB2-Steuerzentrale)	26
Ändern der Datenquellenversion in einer Serverdefinition (DB2-Befehlszeile)	27
Ändern aller Serverdefinitionen für einen bestimmten Datenquellentyp	28
Verwenden von Serveroptionen in Serverdefinitionen (DB2-Steuerzentrale)	28
Temporäres Ändern von Serveroptionen für relationale Datenquellen	29
Die Hierarchie bei Einstellungen von Serveroptionen	29
Verwenden von Serveroptionen in Serverdefinitionen (DB2-Befehlszeile)	30

Ändern einer Benutzerzuordnung (DB2-Steuerzentrale)	30
Ändern einer Benutzerzuordnung (DB2-Befehlszeile)	31
Ändern eines Kurznamens (DB2-Steuerzentrale)	33
Einschränkungen beim Ändern von Kurznamen	34
Ändern der Spaltennamen eines Kurznamens (DB2-Steuerzentrale)	36
Ändern der Spaltennamen eines Kurznamens (DB2-Befehlszeile)	36
Ändern von Kurznamenoptionen (DB2-Steuerzentrale)	37
Ändern von Kurznamenoptionen (DB2-Befehlszeile)	38
Ändern der Spaltenoptionen eines Kurznamens (DB2-Steuerzentrale)	39
Ändern der Spaltenoptionen eines Kurznamens (DB2-Befehlszeile)	40
Ändern eines Kurznamens (DB2-Befehlszeile)	41
Löschen eines Wrappers	42
Löschen einer Serverdefinition	43
Löschen einer Benutzerzuordnung	45
Löschen eines Kurznamens	45

Kapitel 3. Datentypzuordnungen in einem föderierten System 47

Datentypzuordnungen und der globale Katalog der föderierten Datenbank	47
Erstellen alternativer Datentypzuordnungen	48
Datentypzuordnungen für nicht relationale Datenquellen	49
Progressive und regressive Datentypzuordnungen	49
Erstellen von Datentypzuordnungen	50
Erstellen einer Typzuordnung für einen Datenquellentyp - Beispiel	51
Erstellen einer Typzuordnung für einen Datenquellentyp und eine Datenquellenversion - Beispiel	52
Erstellen einer Typzuordnung für alle Datenquellenobjekte auf einem Server - Beispiel	53
Umsetzung zwischen Datentypen	54
Unterstützung für Datentyp TIMESTAMP	55
Ändern eines lokalen Typs für ein Datenquellenobjekt (DB2-Steuerzentrale)	55
Ändern eines lokalen Typs für ein Datenquellenobjekt - Beispiele	56
Ändern eines lokalen Typs für ein Datenquellenobjekt (DB2-Befehlszeile)	58
Ändern von LONG-Datentypen in VARCHAR-Datentypen	59

Kapitel 4. Zuordnen von Funktionen und benutzerdefinierten Funktionen (UDFs) 61

Funktionszuordnungen in einem föderierten System	61
Funktionsweise von Funktionszuordnungen in einem föderierten System	61

Bedeutung von Funktionszuordnungen	62
Empfehlungen für das Erstellen von Funktionszuordnungen	62
Benutzerdefinierte Funktionen (UDFs) in Anwendungen	63
Voraussetzungen für die Zuordnung von benutzerdefinierten Funktionen	63
Erstellen von Funktionszuordnungen	64
Angabe von Funktionsnamen in der Anweisung CREATE FUNCTION MAPPING	65
Erstellen einer Funktionszuordnung für einen bestimmten Datenquellentyp	66
Erstellen einer Funktionszuordnung für einen bestimmten Datenquellentyp und eine bestimmte Datenquellentypversion	66
Erstellen einer Funktionszuordnung für alle Datenquellenobjekte auf einem bestimmten Server	67
Funktionsschablonen	68
Erstellen von Funktionsschablonen	68
Inaktivieren einer Standardfunktionszuordnung	70
Löschen einer Funktionszuordnung	70

Kapitel 5. Erstellen von Indexspezifikationen 71

Indexspezifikationen in einem föderierten System	71
Erstellen von Indexspezifikationen für Datenquellenobjekte	72
Erstellen von Indexspezifikationen für Tabellen, die neue Indizes erhalten	73
Erstellen von Indexspezifikationen für Sichten	75
Erstellen von Indexspezifikationen für Informix-Synonyme	76

Kapitel 6. Entwickeln von föderierten Prozeduren 79

Föderierte Prozeduren	79
Datenquellen und föderierte Prozeduren	81
Föderierte Prozeduren für DB2-Datenquellen	81
Föderierte Prozeduren und Microsoft SQL Server	83
Föderierte Prozeduren und Oracle	84
Föderierte Prozeduren und Sybase	88
Erstellen von föderierten Prozeduren	90
Erteilen und Widerrufen von Berechtigungen zum Aufrufen von föderierten Prozeduren	91
Anzeigen von Parameterinformationen und Aufrufen von föderierten Prozeduren	93
Ändern oder Löschen von föderierten Prozeduren	94
Verknüpfen der Ergebnismengen von föderierten Prozeduren	95
Befehl DB2FEDGENTF	98
Föderierte Prozeduren - Fehlerbehebung	100

Kapitel 7. Erstellen und Modifizieren ferner Tabellen mithilfe von transparenter DDL 105

Was ist transparente DDL?	105
Ferne LOB-Spalten und transparente DDL	106

Erstellen von fernen Tabellen und transparente DDL	106
Erstellen neuer ferner Tabellen unter Verwendung von transparenter DDL	107
Erstellen neuer ferner Tabellen unter Verwendung von transparenter DDL - Beispiele	108
Ändern ferner Tabellen unter Verwendung von transparenter DDL	110
Löschen ferner Tabellen unter Verwendung von transparenter DDL	112

Kapitel 8. Verwalten von Transaktionen in einem System föderierter Datenbanken. 113

Transaktionsunterstützung in föderierten Systemen	113
Was ist eine Aktualisierung in einem föderierten System?	114
Was ist eine Aktualisierungstransaktion in einer Durchgriffssitzung?	115
Datenquellen, die DDL-Anweisungen automatisch festschreiben	116
Benutzerdefinierte Funktionen, die zur Verarbeitung in eine Datenquelle verschoben werden	116

Kapitel 9. Ausführen von Transaktionen mit zweiphasigem Commit 117

Zweiphasiges Commit für föderierte Transaktionen	117
Planung von föderiertem zweiphasigem Commit	117
Föderierte Architektur für zweiphasiges Commit	118
Zweiphasiges Commit für föderierte Transaktionen - Beispiele	121
Verfahren zur Verarbeitung von Transaktionen mit föderiertem zweiphasigem Commit	125
Aktivieren von zweiphasigem Commit für föderierte Transaktionen	129
Anforderungen an Datenquellen und deren Konfiguration für Transaktionen mit föderiertem zweiphasigem Commit	131
Konfigurieren von DRDA-Datenquellen	131
Konfigurieren von Oracle-Datenquellen	132
Konfigurieren von Informix-Datenquellen	133
Konfigurieren von Microsoft SQL Server-Datenquellen	134
Konfigurieren von Sybase-Datenquellen	135
Recovery nach Fehlern durch föderiertes zweiphasiges Commit	136
Resynchronisation in föderierten Systemen	136
Manuelles Recovery von unbestätigten Transaktionen	137
Verfolgen der Statustypen von DUOW-Transaktionen in Datenquellen	138
Fehlerbehebung bei Problemen mit föderiertem zweiphasigem Commit	139
Leistungsverhalten bei föderiertem zweiphasigem Commit	141
Leistungssteigerung bei föderiertem zweiphasigem Commit	142

Kapitel 10. Einfügen, Aktualisieren und Löschen von Daten in einem föderierten System 143

Zugriffsrechte für INSERT-, UPDATE- und DELETE-Anweisungen	143
Einschränkungen für INSERT-, UPDATE- und DELETE-Anweisungen in föderierten Systemen	144
Nicht unterstützte Datenquellen	144
Referenzielle Integrität in einem föderierten System	144
Anweisungen INSERT, UPDATE und DELETE und große Objekte (LOBs).	145
Beibehalten der Atomizität von Anweisungen in einem föderierten System	145
Modifizieren von Daten in einem föderierten System.	146
Einfügen von Daten in Datenquellenobjekte	146
Aktualisieren von Daten in Datenquellenobjekten	146
Löschen von Daten aus Datenquellenobjekten	147
Zuordnungssemantik in einem föderierten System	148
Zuordnungssemantik in einem föderierten System - Beispiele	150
Datenquellenspezifische Einschränkungen für Datentypwerte	150
Föderierte Aktualisierungen mit Anwendungssicherungspunkten	152
Föderierte Aktualisierungen mit Anwendungssicherungspunkten - Beispiele	153
Einschränkungen bei föderierten Aktualisierungen mit Anwendungssicherungspunkten	154

Kapitel 11. Importieren und Exportieren von Daten für Kurznamen 155

Einschränkungen für den Import von Daten in Kurznamen	155
IMPORT-Befehl bei Kurznamen - Beispiele	156
Einschränkungen für den Export von Daten mit Kurznamen	157

Kapitel 12. Arbeiten mit Kurznamen 159

Kurznamen in einem föderierten System	159
Mit der Option WITH HOLD deklarierte Cursor	159
Trigger	160
Zugriff auf Daten mit Kurznamen	160
Mit Kurznamen verwendbare SQL-Anweisungen.	160
Zugriff auf neue Datenquellenobjekte	164
Erstellen von Kurznamen für relationale und nicht relationale Datenquellen	165
Namen für Kurznamenspalten und Indizes	166
Zugreifen auf Datenquellen unter Verwendung von Durchgriffssitzungen	167
Zugreifen auf heterogene Daten über föderierte Sichten	167
Erstellen von föderierten Sichten - Beispiele	168
Erstellen eines Kurznamens für einen Kurznamen	169
Datenauswahl in einem föderierten System	170
Datenauswahl in einem föderierten System - Beispiele	170

Informative Integrationsbedingungen für Kurznamen.	173
Angaben von informativen Integrationsbedingungen für Kurznamen (DB2-Steuerzentrale)	173
Angaben von informativen Integrationsbedingungen für Kurznamen (DB2-Befehlszeile)	174
Angaben von informativen Integrationsbedingungen für Kurznamen - Beispiele	175
Aktualisieren von Statistikdaten für Kurznamen	178
Aktualisierungsfunktion für Kurznamenstatistiken - Übersicht	178
Methoden für den Abruf von Kurznamenstatistik	179
Abrufen von Kurznamenstatistik	180
Einschränkungen bei HIGH2KEY- und LOW2KEY-Statistik	184
Erstellen eines DB2-Toolskatalogs.	184
Statusanzeige bei Aktualisierungen von Kurznamenstatistik (DB2-Steuerzentrale)	185
Statusanzeige bei Aktualisierungen von Kurznamenstatistik (DB2-Befehlszeile)	185
Gespeicherte Prozedur SYSPROC.NNSTAT	186
Automatisches Aktualisieren von Kurznamenstatistikdaten	188

Kapitel 13. Arbeiten mit fernen XML-Daten 191

Erstellen eines Kurznamens für ferne XML-Daten - Beispiele	191
Bearbeiten von XML-Daten - Beispiele	191
Prüfen von XML-Dokumenten anhand von XML-Schemata	192
Registrieren von XML-Schemata	192
Prüfen von XML-Dokumenten.	193
Zerlegen von XML-Dokumenten, die über mit Annotationen versehene XML-Schemata verfügen, in Kurznamen	194
Föderierte Verarbeitung ferner XML-Daten	194
Föderiertes Parsing ferner XML-Daten	194
Codepageprobleme bei fernen XML-Daten.	195
Einschränkungen beim fernen XML-Datentyp	196

Kapitel 14. Fehlertoleranz in verschachtelten Tabellenausdrücken. 197

Angaben der Fehlertoleranz in verschachtelten Tabellenausdrücken	198
Fehlertoleranz in verschachtelten Tabellenausdrücken - Beispiel	199
Datenquellenunterstützung für Fehlertoleranz in verschachtelten Tabellenausdrücken	200
Einschränkungen bei der Fehlertoleranz in verschachtelten Tabellenausdrücken	200

Kapitel 15. Überwachen von Servern mit föderierten Datenbanken und Kurznamen 201

Diagnoseanzeiger für Kurznamen und föderierte Server	201
--	-----

Aktivieren der Diagnoseanzeiger von föderierten Systemen	202
Überwachen des Status von Kurznamen und Servern für föderierte Datenbanken	202
Überwachen des Status von Kurznamen und Servern für föderierte Datenbanken - Beispiel	203
Ausführen des Snapshot Monitor für föderierte Systeme - Übersicht	204
Überwachen von föderierten Abfragen	204
Ausführen des Snapshot Monitor für föderierte Abfragen - Beispiel	206
Systemmonitorelemente der föderierten Datenbanken	207

Kapitel 16. Interaktion zwischen Clientanwendungen und Datenquellen . . . 209

Kapitel 17. Verweis auf Datenquellenobjekte in SQL-Anweisungen durch Kurznamen 211

Kurznamen in DDL-Anweisungen	211
Statistikanwendungen für Datenquellen	212
Definieren von Spaltenoptionen für Kurznamen	213
Angaben der Spaltenoption NUMERIC_STRING	213
Angaben der Spaltenoption VARCHAR_NO-TRAILING_BLANKS	214

Kapitel 18. Erstellen und Verwenden von föderierten Sichten 215

Erstellen von föderierten Sichten - Beispiele	215
---	-----

Kapitel 19. Pflegen von Datenintegrität mit Isolationsstufen. 217

Isolation auf Anweisungsebene in einem föderierten System	218
Isolation auf Verbindungsebene in einem föderierten System	219

Kapitel 20. Föderierte LOB-Unterstützung 221

LOB-Querverweise	222
Einschränkungen bei großen Objekten (LOBs)	222
Leistungsaspekte für die LOB-Verarbeitung	222

Kapitel 21. Verteilte Anforderungen für das Abrufen von Datenquellen 225

Verteilte Anforderungen für das Abrufen von Datenquellen - Beispiele	225
Optimieren von verteilten Anforderungen mit Serveroptionen	226
Abbrechen einer föderierten Abfrage	227

Kapitel 22. Direktes Abfragen von Datenquellen im Durchgriffsmodus . . . 229

Überlegungen und Einschränkungen bei der Verwendung des Durchgriffsmodus in föderierten Systemen	230
Durchgriffssitzungen bei Oracle-Datenquellen	232

Kapitel 23. Optimieren der Abfrageverarbeitung 233

Veröffentlichungen zur Leistung föderierter Datenbanken	233
Abfrageanalyse	234
Pushdown-Analyse	236
Servermerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten	237
SQL-Unterschiede	238
Sortierfolge	240
Serveroptionen des Servers mit föderierten Datenbanken	242
Faktoren der Typen- und Funktionszuordnung	243
Kurznamenmerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten	244
Lokaler Datentyp einer Kurznamenspalte	245
Spaltenoptionen der föderierten Datenbank	245
Abfragemerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten	246
Analysieren des Auswertungsortes einer Abfrage	246
Analysieren des Auswertungsortes einer Abfrage mit der Serveroption DB2_MAXIMAL_PUSHDOWN	247
Entscheidungen für die Auswertung von Zugriffsplänen	248
Warum wird dieses Vergleichselement nicht fern ausgewertet?	248
Warum wird der Operator GROUP BY nicht fern ausgewertet?	248
Warum wird der Operator SET nicht fern ausgewertet?	248
Warum wird die Operation ORDER BY nicht fern ausgewertet?	249
Warum wird eine ferne Anweisung INSERT mit Fullselect nicht vollständig fern ausgewertet?.	249
Warum wird eine ferne Anweisung INSERT mit der Klausel VALUES nicht vollständig fern ausgewertet?.	249
Warum wird eine ferne Anweisung UPDATE für eine Aktualisierung mit Suche nicht vollständig fern ausgewertet?	250
Warum wird eine Anweisung UPDATE für eine positionierte Aktualisierung nicht vollständig fern ausgewertet?	250
Warum wird eine ferne Anweisung DELETE für einen gezielten Löschvorgang nicht vollständig fern ausgewertet?	250
Aktualisieren und Anpassen von Datenquellen	251
Pushdown-Verarbeitung von Vergleichselementen mit Funktionsschablonen	251

Kapitel 24. Parallelität bei Abfragen, die auf Kurznamen verweisen 253

Partitionsinterne Parallelität bei Abfragen, die auf Kurznamen verweisen	253
Aktivieren der partitionsinternen Parallelität bei Abfragen, die auf Kurznamen verweisen	253
Partitionsinterne Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne	254

Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen	255
Aktivieren von partitionsübergreifender Parallelität bei Abfragen, die auf Kurznamen verweisen	258
Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne	259
Rechenpartitionsgruppen	261
Definieren einer Rechenpartitionsgruppe	262
Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen - erwartetes Leistungsverhalten	263
Gemischte Parallelität bei Abfragen, die auf Kurznamen verweisen	263
Aktivieren von gemischter Parallelität bei Abfragen, die auf Kurznamen verweisen	263
Gemischte Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne	264

Kapitel 25. Asynchrone Verarbeitung föderierter Abfragen 267

Asynchrone Verarbeitung föderierter Abfragen - Beispiele	267
Asynchronitätsoptimierung	268
Zugriffspläne ohne Asynchronität	268
Für Asynchronität optimierte Zugriffspläne	268
Zugriffspläne - Beispiele	269
Überwachen der Ressourcennutzung	274
Aktivieren der Asynchronitätsoptimierung	274
Optimierungsaspekte für die Asynchronitätsoptimierung	275
Einschränkungen bei der Asynchronitätsoptimierung	276
Ermitteln, ob die Asynchronitätsoptimierung für eine Abfrage angewendet wird	276

Kapitel 26. Globale Optimierung . . . 279

Servermerkmale mit Auswirkungen auf die globale Optimierung	279
Relatives Verhältnis der CPU-Geschwindigkeit	280
Relatives Verhältnis der E/A-Geschwindigkeit	281
Übertragungsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und der Datenquelle	281
Sortierfolge der Datenquelle	281
Ferne Planhinweise	282
Kurznamenmerkmale mit Auswirkungen auf die globale Optimierung	282
Indexspezifikationen	282
Statistikdaten des globalen Katalogs	283
Aktualisieren von Zeilenänderungen	285
Aktualisierung von Statistikdaten bei Spaltenänderungen	285
Analysieren der globalen Optimierung	285
Durch den Zugriffspläne bestimmte Möglichkeiten der Optimierung	286
Warum wird ein Join zwischen zwei Kurznamen derselben Datenquelle nicht fern ausgewertet?	286

Warum wird der Operator GROUP BY nicht fern ausgewertet?	287
Warum wird die Anweisung nicht vollständig fern ausgewertet?	287
Warum verfügt ein vom Optimierungsprogramm generierter und vollständig fern ausgewerteter Zugriffspläne über eine schlechtere Leistung als die ursprüngliche, direkt in der fern Datenquelle ausgeführte Abfrage?	287

Kapitel 27. Systemmonitorelemente, die die Leistung beeinflussen 289

Kapitel 28. MQTs (Materialized Query Tables) 291

MQTs (Materialized Query Tables) und föderierte Systeme - Übersicht	291
Erstellen einer MQT in föderierten Systemen	292
Datenquellenspezifische Einschränkungen für MQTs	292
Einschränkungen bei der Verwendung von MQTs mit Kurznamen	293

Kapitel 29. Cachetabellen 295

Erstellen von Cachetabellen	296
Modifizieren der Einstellungen für MQTs	298
Hinzufügen von MQTs zu einer Cachetabelle	298
Weiterleiten von Abfragen an Cachetabellen	299
Aktivieren und Inaktivieren der Cacheeinstellungen für die Replikation	300
Löschen von MQTs aus einer Cachetabelle	301
Löschen von Cachetabellen	301

Kapitel 30. Sicherheitseinrichtungen für Server mit föderierten Datenbanken 303

Kapitel 31. Gesicherte Kontexte und Verbindungen föderierter Systeme . . 305

Vorteile gesicherter Verbindungen föderierter Systeme	306
Typen gesicherter Verbindungen föderierter Systeme	307
APIs für gesicherte Verbindungen föderierter Systeme	309
Szenarios für die Implementierung gesicherter, abgehender Verbindungen bei föderierten Systemen	310
Szenario: Gesicherte End-to-End-Verbindungen föderierter Systeme ohne Benutzerzuordnungen	310
Beispielcode für Szenarios mit gesicherten End-to-End-Verbindungen föderierter Systeme	313
Szenario: Gesicherte End-to-End-Verbindungen föderierter Systeme mit Benutzerzuordnungen	314
Szenario: Gesicherte, abgehende Verbindungen föderierter Systeme	317
Benutzerzuordnungen und gesicherte Verbindungen föderierter Systeme	322

Kapitel 32. Kennsatzbasierte Zugriffssteuerung (LBAC) und Systeme mit föderierten Datenbanken	325
--	------------

Kapitel 33. Externe Repositorys für Benutzerzuordnungen	327
--	------------

Plug-in für Benutzerzuordnungen (Programmiersprache C)	327
Unterstützte Plattformen für das Plug-in für Benutzerzuordnungen (Programmiersprache C)	329
Bei der Entwicklung von Plug-ins für Benutzerzuordnungen zu beachtende Einschränkungen (Programmiersprache C)	329
Kopfdatendatei 'fsumplugin.h' für das Plug-in für Benutzerzuordnungen (Programmiersprache C)	330
Entwickeln eines Beispiel-Plug-ins für Benutzerzuordnungen (Programmiersprache C)	334
Beispiel-Plug-in für Benutzerzuordnungen (Programmiersprache C)	340
Plug-in für Benutzerzuordnungen (Programmiersprache Java)	341
Klassen für das Plug-in für Benutzerzuordnungen (Programmiersprache Java)	342
Beispiel-Plug-in für Benutzerzuordnungen (Programmiersprache Java)	347
Entwickeln eines Beispiel-Plug-ins für Benutzerzuordnungen (Programmiersprache Java)	348

Kapitel 34. Oracle-Sicherheit in einem System föderierter Datenbanken	357
--	------------

Oracle Label Security	357
Oracle-Proxyauthentifizierung und föderierter gesicherter Kontext	357

Kapitel 35. Datenquellenunterstützung für föderierte Funktionen	359
--	------------

Kapitel 36. Referenz für Datenquellenoptionen	361
--	------------

Referenz für BioRS-Optionen	361
Referenz für Optionen der DB2-Datenbank	365
Referenz für Excel-Optionen	373
Referenz für Informix-Optionen	374
Referenz für JDBC-Optionen	380
Referenz für Microsoft SQL Server-Optionen	387
Referenz für ODBC-Optionen	393
Referenz für Oracle-Optionen	399
Referenz für Scriptoptionen	405
Referenz für Sybase-Optionen	410
Referenz für Teradata-Optionen	417
Referenz für Optionen von Dateien mit Tabellenstruktur	421
Referenz für Web-Service-Optionen	424
Referenz für XML-Optionen	432

Kapitel 37. Sichten in der globalen Katalogtabelle mit Informationen föderierter Datenbanken	441
---	------------

Kapitel 38. Funktionszuordnungsoptionen für föderierte Systeme	445
---	------------

Kapitel 39. Gültige Servertypen in SQL-Anweisungen	447
---	------------

Kapitel 40. Datentypzuordnungen	449
--	------------

Standardmäßige progressive Datentypzuordnungen	449
Standardmäßige progressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen	449
Standardmäßige progressive Datentypzuordnungen bei DB2 für System i-Datenquellen	450
Standardmäßige progressive Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen	451
Standardmäßige progressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen	452
Standardmäßige progressive Datentypzuordnungen bei Informix-Datenquellen	453
Standardmäßige progressive Datentypzuordnungen bei JDBC-Datenquellen	454
Standardmäßige progressive Datentypzuordnungen bei Microsoft SQL Server-Datenquellen	456
Standardmäßige progressive Datentypzuordnungen bei ODBC-Datenquellen	457
Standardmäßige progressive Datentypzuordnungen bei Oracle NET8-Datenquellen	459
Standardmäßige progressive Datentypzuordnungen bei Sybase-Datenquellen	459
Standardmäßige progressive Datentypzuordnungen bei Teradata-Datenquellen	461
Beispiele für progressive Datentypzuordnungen	462
Standardmäßige regressive Datentypzuordnungen	464
Standardmäßige regressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen	465
Standardmäßige regressive Datentypzuordnungen bei DB2 für System i-Datenquellen	466
Standardmäßige regressive Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen	467
Standardmäßige regressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen	467
Standardmäßige regressive Datentypzuordnungen bei Informix-Datenquellen	468
Standardmäßige regressive Datentypzuordnungen bei JDBC-Datenquellen	469
Standardmäßige regressive Datentypzuordnungen bei Microsoft SQL Server-Datenquellen	470
Standardmäßige regressive Datentypzuordnungen bei ODBC-Datenquellen	471

Standardmäßige regressive Datentypzuordnungen bei Oracle NET8-Datenquellen	471	Standardmäßige regressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen	477
Standardmäßige regressive Datentypzuordnungen bei Sybase-Datenquellen	472	Standardmäßige progressive Datentypzuordnungen für Unicode bei Sybase-Datenquellen	477
Standardmäßige regressive Datentypzuordnungen bei Teradata-Datenquellen	473	Standardmäßige regressive Datentypzuordnungen für Unicode bei Sybase-Datenquellen	478
Standardmäßige regressive Datentypzuordnungen für Unicode	474	Datentypen, die für nicht relationale Datenquellen unterstützt werden	478
Standardmäßige progressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen	474	Produktdokumentation	483
Standardmäßige regressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen	475	Syntaxdiagramme lesen	485
Standardmäßige progressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen	475	Dokumentation im behinderten-gerechten Format	487
Standardmäßige regressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen	475	Bemerkungen	489
Standardmäßige progressive Datentypzuordnungen für Unicode bei NET8-Datenquellen	476	Marken	491
Standardmäßige regressive Datentypzuordnungen für Unicode bei NET8-Datenquellen	476	Index	493
Standardmäßige progressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen	477		

Kapitel 1. Föderierte Systeme

Ein *föderiertes System* ist ein spezielles Managementsystem für verteilte Datenbanken (DBMS). Es besteht aus einer DB2-Instanz, die als Server mit föderierten Datenbanken dient, aus einer Datenbank, die als föderierte Datenbank fungiert, aus einer oder mehreren Datenquellen sowie aus Clients (Benutzern und Anwendungen), die auf die Datenbank und die Datenquellen zugreifen.

In einem föderierten System können Sie mit einer einzigen SQL-Anweisung verteilte Anforderungen an mehrere Datenquellen senden. Beispielsweise können Sie mit einer einzigen SQL-Anweisung Daten verknüpfen, die sich in einer DB2-Tabelle, einer Oracle-Tabelle und einer XML-Datei befinden. In der folgenden Abbildung sind die Komponenten eines föderierten Systems und ein Beispiel für die Datenquellen dargestellt, auf die Sie zugreifen können.

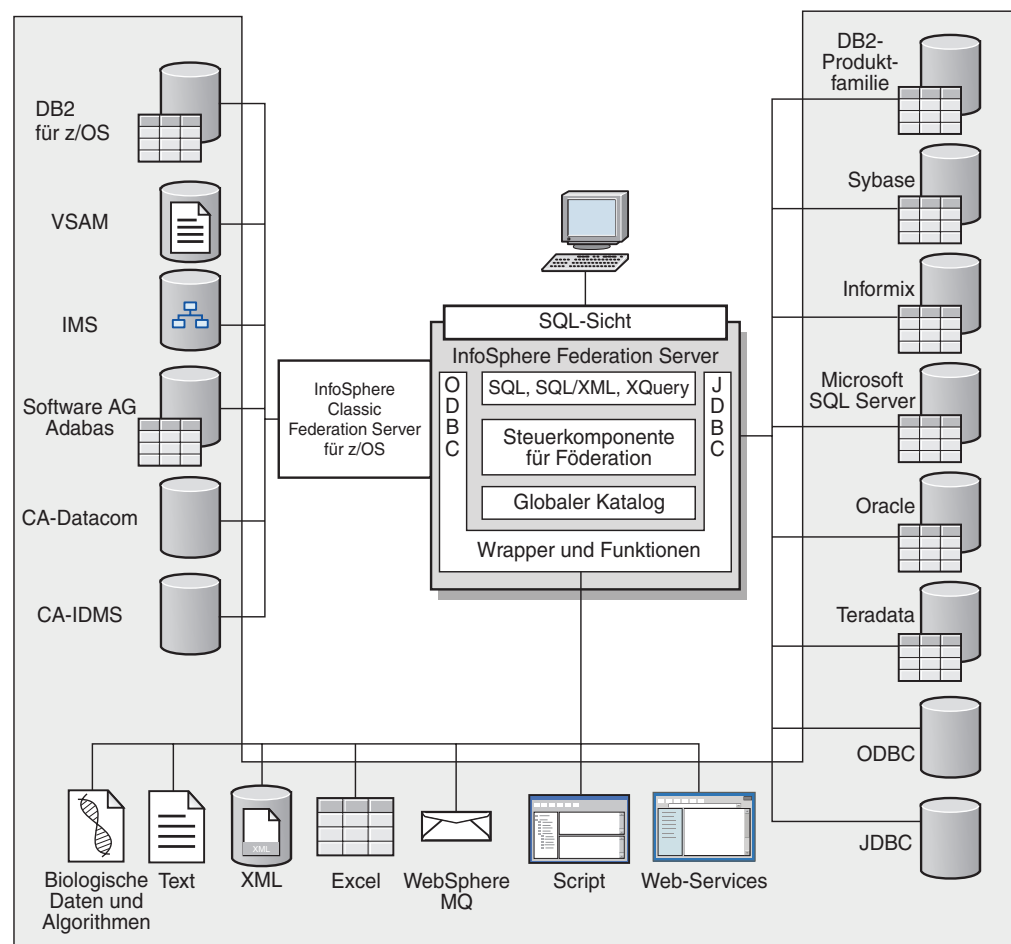


Abbildung 1. Komponenten eines föderierten Systems

Die Leistungsstärke eines föderierten Systems beruht auf folgenden Funktionsmerkmalen:

- Daten können aus lokalen Tabellen und fernen Datenquellen so korreliert werden, als ob alle Daten lokal in der föderierten Datenbank gespeichert wären.

- Daten können in relationalen Datenquellen so aktualisiert werden, als ob sie in der föderierten Datenbank gespeichert wären.
- Daten können in relationale Datenquellen und aus diesen Quellen versetzt werden.
- Die Vorteile der leistungsfähigen Datenquellenverarbeitung können genutzt werden, indem Anforderungen zur Verarbeitung an die Datenquellen gesendet werden.
- SQL-Einschränkungen bei der Datenquelle können kompensiert werden, indem Teile einer verteilten Anforderung auf dem Server mit föderierten Datenbanken verarbeitet werden.

Unterstützte Datenquellen

Mithilfe eines föderierten Systems können Sie auf eine Vielzahl von Datenquellen zugreifen.

IBM® InfoSphere Federation Server unterstützt die in den folgenden Tabellen aufgeführten Datenquellen. Die erste Tabelle enthält die Voraussetzungen für die Daten-Client-Software. Falls nicht anders angegeben, muss die Client-Software separat bezogen werden.

Sie müssen die Client-Software für die Datenquellen installieren, auf die Sie zugreifen möchten. Die Client-Software muss auf demselben System installiert werden wie IBM InfoSphere Federation Server. Darüber hinaus benötigen Sie das entsprechende Java™-Software-Development-Kit (SDK) für die Verwendung einer Reihe von Tools wie z. B. der DB2-Steuerzentrale sowie für die Erstellung und Ausführung von Java-Anwendungen, einschließlich gespeicherter Prozeduren und benutzerdefinierter Funktionen.

Die neuesten Informationen finden Sie auf der Seite mit Datenquellenvoraussetzungen im World Wide Web.

Tabelle 1. Unterstützte Datenquellen, Client-Software-Voraussetzungen und Unterstützung durch 32-Bit-Betriebssysteme

			32-Bit-Hardwarearchitektur und -Betriebssystem	
			X86-32	X86-32
Datenquelle	Unterstützte Versionen	Client-Software	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
BioRS	5.2, 5.3	Keine	J	J
DB2 für Linux, UNIX® und Windows	8.1.x, 8.2.x, 9.1, 9.5, 9.7	Keine	J	J
DB2 für z/OS	7.x, 8.x, 9.x	DB2 Connect V9.7	J	J
DB2 für System i	5.2, 5.3, 5.4, 6.1	DB2 Connect V9.7	J	J
DB2 Server für VSE und VM	7.2 , 7.4	DB2 Connect V9.7	J	J
Unstrukturierte Dateien		Keine	J	J

Tabelle 1. Unterstützte Datenquellen, Client-Software-Voraussetzungen und Unterstützung durch 32-Bit-Betriebssysteme (Forts.)

			32-Bit-Hardwarearchitektur und -Betriebssystem	
			X86-32	X86-32
Datenquelle	Unterstützte Versionen	Client-Software	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
Informix	Informix XPS 8.50, 8.51 und Informix IDS IDS 7.31, IDS 9.40, IDS 10.0, 11.5, 11.10	Informix Client SDK Version 2.81.TC2 oder höher; Version 3.0 für SLES 10 auf POWER-Systemen erforderlich	J	J
JDBC	3.0 oder höher	JDBC-Treiber, die JDBC 3.0 oder höher entsprechen	J	J
Microsoft® Excel	2000, 2002, 2003, 2007	Keine		J
Microsoft SQL Server	Microsoft SQL Server 2000/SP4, 2005, 2008	Für Windows: Microsoft SQL Server Client ODBC 3.0-Treiber (oder eine höhere Version). Für Unix: DataDirect ODBC 5.3	J	J
MQ	MQ7	MQ7	J	J
ODBC	3.0	ODBC-Treiber, die ODBC 3.0 oder höher** entsprechen	J	J
OLE DB	2.7, 2.8	OLE DB 2.0 oder höher	J	J
Oracle	10g, 10gR2, 11g, 11gR1	Oracle NET Client 10.0 - 10.1, 10.2.0.1 mit Programmkorrektur 3807408, 10.1.0.3 mit Programmkorrektur 3807408, 11.1.0.6.0	J	J
Sybase	Sybase ASE 12.5, 15.0	Sybase Open Client 12.5 - 15.0	J	J
Teradata	2.5, 2.6, 12	Shared Common Components for Internationalization for Teradata (tdicu) Version 1.01 oder höher, Teradata Generic Security Services (TeraGSS) Version 6.01 oder höher und die Software auf den folgenden Betriebssystemen: Für Windows: Teradata Client TTU 8.0 oder höher und die Teradata-API-Bibliothek CLIV2 4.8.0 oder höher Für UNIX und Linux: Teradata Call-Level Interface Version 2 CLIV2 Release 4.8.0 oder höher	J	J

Tabelle 1. Unterstützte Datenquellen, Client-Software-Voraussetzungen und Unterstützung durch 32-Bit-Betriebssysteme (Forts.)

			32-Bit-Hardwarearchitektur und -Betriebssystem	
			X86-32	X86-32
Datenquelle	Unterstützte Versionen	Client-Software	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
Web-Services	WSDL 1.0, 1.1 SOAP 1.0, 1.1	Keine	J	J
XML	XML1.0, XML1.1	Keine	J	J

** ODBC kann dazu verwendet werden, auf RedBrick 6.20cu5 und InfoSphere Classic Federation V8.2 und höher sowie auf weitere Datenquellen zuzugreifen.

Tabelle 2. Unterstützung durch 64-Bit-Betriebssysteme.

64-Bit-Hardwarearchitektur	X86-64	X86-64	Power	Itanium®	Power	Sparc	zSeries
Betriebssystem	Linux RHEL SUSE	Windows	AIX	HP-UX	Linux RHEL SUSE	Solaris	Linux RHEL SUSE
Datenquelle							
BioRS	J	J	J	J	J	J	J
DB2 für Linux, UNIX und Windows	J	J	J	J	J	J	J
DB2 für z/OS	J	J	J	J	J	J	J
DB2 für System i	J	J	J	J	J	J	J
DB2 Server für VSE und VM	J	J	J	J	J	J	J
Informix	J		J	J	J	J	J
JDBC	J	J	J	J	J	J	J
Microsoft Excel							
Microsoft SQL Server	J	J	J	J		J	
MQ	J	N	J	J	J	J	J
ODBC	J	J	J***	J		J***	J
OLE DB		J		J			
Oracle	J	J	J	J	J	J	J
Script	J	J	J	J	J	J	J
Sybase	J		J	J	J	J	
Teradata	J		J	J		J	
Web-Services	J	J	J	J	J	J	J
XML	J	J	J	J	J	J	J

*** ODBC kann dazu verwendet werden, auf RedBrick 6.20cu5 und IBM InfoSphere Classic Federation Server für z/OS mit 32-Bit- und 64-Bit-Clients zuzugreifen.

Der Server mit föderierten Datenbanken

Der DB2-Server in einem föderierten System wird als Server mit föderierten Datenbanken bezeichnet. Es kann eine beliebige Anzahl von DB2-Instanzen als Server mit föderierten Datenbanken konfiguriert werden. Sie können vorhandene DB2-Instanzen als Server mit föderierten Datenbanken verwenden oder neue DB2-Instanzen speziell für das föderierte System erstellen.

Die DB2-Instanz, die das föderierte System verwaltet, wird als Server bezeichnet, weil sie auf Anforderungen von Endbenutzern und Clientanwendungen antwortet. Der Server mit föderierten Datenbanken sendet häufig Teile der empfangenen Anforderungen zur Verarbeitung an die Datenquellen. Eine Pushdown-Operation ist eine Operation, die fern verarbeitet wird. Die DB2-Instanz, die das föderierte System verwaltet, wird als Server mit föderierten Datenbanken bezeichnet, auch wenn es beim Weiterleiten von Anforderungen im Pushdown-Modus an die Datenquellen als Client agiert.

Der Server mit föderierten Datenbanken ist wie jeder andere Anwendungsserver eine Datenbankmanagerinstanz. Anwendungsprozesse stellen eine Verbindung zur Datenbank innerhalb des Servers mit föderierten Datenbanken her und übergeben Anforderungen an die Datenbank. Der Server unterscheidet sich jedoch in zwei Hauptmerkmalen von anderen Anwendungsservern:

- Ein Server mit föderierten Datenbanken ist für den Empfang von Anforderungen konfiguriert, die ganz oder teilweise für Datenquellen bestimmt sein können. Der Server mit föderierten Datenbanken verteilt diese Anforderungen an die Datenquellen.
- Ein Server mit föderierten Datenbanken verwendet ähnlich wie andere Anwendungsserver DRDA-Kommunikationsprotokolle (über TCP/IP), um mit Instanzen der DB2-Produktfamilie zu kommunizieren. Im Gegensatz zu anderen Anwendungsservern verwendet ein Server mit föderierten Datenbanken jedoch für den Zugriff auf die Datenquelle den nativen Client der Datenquelle. Ein Server mit föderierten Datenbanken verwendet beispielsweise für den Zugriff auf Sybase-Datenquellen Sybase Open Client und für den Zugriff auf Microsoft SQL Server-Datenquellen einen ODBC-Treiber für Microsoft SQL Server.

Was ist eine Datenquelle?

In einem föderierten System kann eine *Datenquelle* eine relationale Datenbank (beispielsweise Oracle oder Sybase) oder eine nicht relationale Datenquelle (beispielsweise eine XML-Datei) sein.

Über einige Datenquellen ist auch der Zugriff auf weitere Datenquellen möglich. Mit dem ODBC-Wrapper beispielsweise können Sie auf Datenquellen des IBM InfoSphere Classic Federation Server für z/OS, z. B. auf DB2 für z/OS, IMS, CA-IDMS, CA-Datcom, Software AG Adabas und VSAM.

Die für den Zugriff auf eine Datenquelle verwendete Methode bzw. das hierzu verwendete Protokoll ist vom Datenquellentyp abhängig. So wird für den Zugriff auf Datenquellen von DB2 für z/OS beispielsweise DRDA verwendet.

Datenquellen sind autonom. Der Server mit föderierten Datenbanken kann z. B. Anfragen an Oracle-Datenquellen senden, während diese Oracle-Anwendungen auf diese Datenquellen zugreifen. Ein föderiertes System monopolisiert oder begrenzt den Zugriff auf andere Datenquellen außerhalb der Integritäts- und Verriegelungsbedingungen nicht.

Die föderierte Datenbank

Für Endbenutzer und Clientanwendungen werden die Datenquellen im DB2-Datenbanksystem als eine einzige, kollektive Datenbank dargestellt. Benutzer und Anwendungen kommunizieren mit der *föderierten Datenbank*, die vom Server mit föderierten Datenbanken verwaltet wird.

Die föderierte Datenbank enthält einen Systemkatalog, in dem Informationen zu Daten gespeichert werden. Der Katalog des föderierten Systems enthält Einträge, in denen die Datenquellen und deren Kenndaten angegeben sind. Der Server mit föderierten Datenbanken ermittelt anhand der im Systemkatalog der föderierten Datenbank und im Datenquellenwrapper gespeicherten Informationen die optimale Methode für die Verarbeitung von SQL-Anweisungen.

Das föderierte System verarbeitet SQL-Anweisungen so, als ob die Daten aus den Datenquellen gewöhnliche relationale Tabellen oder Sichten innerhalb der föderierten Datenbank wären. Dadurch ist Folgendes möglich:

- Das föderierte System kann relationale Daten mit nicht relationalen Daten korrelieren. Dies gilt sogar dann, wenn Datenquellen unterschiedliche SQL-Programmversionen verwenden oder SQL überhaupt nicht unterstützen.
- Die Kenndaten der föderierten Datenbank haben Vorrang, wenn zwischen den Kenndaten der föderierten Datenbank und den Kenndaten der Datenquellen Unterschiede bestehen. Abfrageergebnisse entsprechen der DB2-Semantik, und zwar selbst dann, wenn zum Berechnen der Abfrageergebnisse Daten aus Nicht-DB2-Datenquellen verwendet werden.

Beispiele:

- Der Server mit föderierten Datenbanken verwendet eine andere Codepage als die Datenquelle. In diesem Fall gilt: Wenn an einen Benutzer des Servers mit föderierten Datenbanken Zeichendaten von der Datenquelle zurückgegeben werden, werden diese Daten auf Basis der Codepage umgewandelt, die die föderierte Datenbank verwendet.
- Der Server mit föderierten Datenbanken verwendet eine andere Sortierfolge als die Datenquelle. In diesem Fall gilt: Alle Sortieroperationen für Zeichendaten werden auf dem Server mit föderierten Datenbanken und nicht in der Datenquelle ausgeführt.

Wrapper und Wrappermodule

Wrapper umfassen Mechanismen, mit deren Hilfe die föderierte Datenbank mit den verwendeten Datenquellen interagieren kann. Die föderierte Datenbank verwendet für die Implementierung eines Wrappers Routinen, die in einer als *Wrappermodul* bezeichneten Bibliothek gespeichert sind.

Diese Routinen ermöglichen der föderierten Datenbank das Ausführen von Operationen wie beispielsweise das Herstellen einer Verbindung zu einer Datenquelle und das interaktive Abrufen von Daten aus dieser Datenquelle. Der Eigner der föderierten Instanz verwendet normalerweise die Anweisung CREATE WRAPPER, um einen Wrapper in der föderierten Datenbank zu registrieren. Sie können einen Wrapper mit der Wrapperoption DB2_FENCED als abgeschirmt oder akzeptiert registrieren.

Sie erstellen einen Wrapper für jeden Datenquellentyp, auf den Sie zugreifen möchten. Beispiel: Sie möchten auf drei Datenbanktabellen von DB2 für z/OS, eine Tabelle von DB2 für System i, zwei Tabellen von Informix und eine Sicht von Informix zugreifen. In diesem Fall müssen Sie einen Wrapper für die DB2-Datenquellen-

objekte und einen Wrapper für die Informix-Datenquellenobjekte erstellen. Nachdem diese Wrapper in der föderierten Datenbank registriert sind, können sie für den Zugriff auf andere Objekte in diesen Datenquellen verwendet werden. Sie können beispielsweise den DRDA-Wrapper für alle Datenquellenobjekte der DB2-Produktfamilie verwenden, d. h., für DB2 für Linux, UNIX und Windows, DB2 für z/OS, DB2 für System i und DB2 Server für VM und VSE.

Verwenden Sie die Serverdefinitionen und Kurznamen zur Angabe der technischen Daten (Name, Speicherposition, usw.) jedes Datenquellenobjekts.

Ein Wrapper dient zur Ausführung zahlreicher Tasks. Hierzu gehören z. B. die im Folgenden aufgeführten Operationen:

- Er stellt die Verbindung zur Datenquelle her. Der Wrapper verwendet die Standardverbindungs-API der Datenquelle.
- Er übergibt Abfragen an die Datenquelle.
 - Bei Datenquellen, die SQL unterstützen, wird die Abfrage in SQL übergeben.
 - Bei Datenquellen, die SQL nicht unterstützen, wird die Abfrage in die native Abfragesprache der Quelle oder in eine Reihe von Quellen-API-Aufrufen umgesetzt.
- Er empfängt Ergebnismengen von der Datenquelle. Der Wrapper verwendet die Standard-APIs der Datenquellen, um Ergebnismengen zu empfangen.
- Er antwortet auf Abfragen der föderierten Datenbank zu den Standarddatentypzuordnungen für eine Datenquelle. Der Wrapper enthält die Standardtypzuordnungen, die zur Erstellung von Kurznamen für ein Datenquellenobjekt verwendet werden. Bei relationalen Wrappern werden beim Erstellen von Datentypzuordnungen die Standarddatentypzuordnungen außer Kraft gesetzt. Benutzerdefinierte Datentypzuordnungen werden im globalen Katalog gespeichert.
- Er antwortet auf Abfragen der föderierten Datenbank zu den Standardfunktionszuordnungen für eine Datenquelle. Die föderierte Datenbank benötigt die Informationen zu Datentypzuordnungen für die Planung von Abfragen. Der Wrapper enthält Informationen, die die föderierte Datenbank braucht, um festzustellen, ob und wie DB2-Funktionen den Funktionen der Datenquelle zugeordnet werden. Diese Informationen verwendet der SQL-Compiler, um festzustellen, ob die Datenquelle die Abfrageoperationen ausführen kann. Bei relationalen Wrappern werden beim Erstellen von Funktionszuordnungen die Standardfunktionszuordnungen außer Kraft gesetzt. Benutzerdefinierte Funktionszuordnungen werden im globalen Katalog gespeichert.

Wrapperoptionen werden zur Konfiguration des Wrappers verwendet, oder zum Definieren der Art und Weise, in der IBM InfoSphere Federation Server den Wrapper einsetzt.

Interaktion mit einem föderierten System

Da es sich bei der föderierten Datenbank um eine Datenbank der DB2-Produktfamilie handelt, können Sie zur Interaktion mit dieser Datenbank eine Reihe unterschiedlicher Methoden verwenden.

Sie können zur Interaktion mit einem föderierten System wahlweise folgende Methoden verwenden:

- Den DB2-Befehlszeilenprozessor (CLP)
- Die GUI der DB2-Steuerzentrale

- Anwendungsprogramme
- Tools der DB2-Produktfamilie
- Web-Service-Provider

Die Schritte in der Dokumentation der föderierten Datenbank stellen die Befehle und SQL-Anweisungen bereit, die im DB2-Befehlszeilenprozessor eingegeben werden können. Wenn über die GUI der DB2-Steuerzentrale Tasks ausgeführt werden können, ist dies in der Dokumentation angegeben. Da die GUI der DB2-Steuerzentrale intuitiv ist, sind die Schritte zum Ausführen dieser Tasks über die DB2-Steuerzentrale nicht in der vorliegenden Dokumentation enthalten.

DB2-Steuerzentrale

Mit der GUI der DB2-Steuerzentrale können die meisten erforderlichen Tasks für die Installation, Konfiguration und Änderung des föderierten Systems ausgeführt werden. Die DB2-Steuerzentrale führt den Benutzer anhand von Anzeigen (Dialogfenstern und Assistenten) durch eine Task.

Die Anzeigen der DB2-Steuerzentrale bieten interaktive Hilfe, wenn mit der Maus ein Steuerelement wie z. B. ein Listenfenster oder eine Befehlsschaltfläche berührt wird. Jede Anzeige verfügt zusätzlich über einen Hilfefknopf, der Informationen über die Task der Anzeige zur Verfügung stellt und eine Verbindung zu Informationen über zugehörige Konzepte und Referenzen herstellt.

Sie können für die Erstellung föderierter Objekte entweder einen Assistenten verwenden oder die Objekte einzeln erstellen.

Verwenden Sie die DB2-Steuerzentrale zur Konfiguration des Zugriffs auf Web-Services und XML-Datenquellen. Die in die DB2-Steuerzentrale integrierten Funktionen vereinfachen die Arbeitsschritte, die zur Konfiguration des Servers mit föderierten Datenbanken für den Zugriff auf diese Datenquellen erforderlich sind.

Die GUI der DB2-Steuerzentrale bietet die einfachste Möglichkeit, die folgenden grundlegenden Konfigurationstasks für die Datenquelle durchzuführen:

- Wrapper erstellen und die Wrapperoptionen festlegen
- Umgebungsvariablen der Datenquelle angeben
- Serverdefinitionen erstellen und Serveroptionen festlegen
- Benutzerzuordnungen erstellen und Benutzeroptionen festlegen
- Kurznamen erstellen und Kurznamen- oder Spaltenoptionen definieren

Nachdem Sie den Server mit föderierten Datenbanken für den Zugriff auf die Datenquellen konfiguriert haben, können Sie die DB2-Steuerzentrale zur Ausführung folgender Arbeitsschritte verwenden:

- Ändern der Datenquellenkonfiguration
- Überwachen des Status von Kurznamen und Servern
- Verwalten der aktuellen Kurznamenstatistikdaten
- Erstellen und Ändern von Cachetabellen
- Angeben informativer Integrationsbedingungen für Kurznamen
- Erstellen ferner Tabellen über IBM InfoSphere Federation Server mithilfe von transparenter DDL

Anwendungsprogramme

Anwendungen erfordern für die Arbeit mit Daten der föderierten Datenbank keine spezielle Codierung. Anwendungen greifen auf die gleiche Art und Weise auf das System zu wie jede andere beliebige DB2-Clientanwendung. Anwendungen verfügen über eine Schnittstelle zu der föderierten Datenbank, die sich auf dem Server befindet.

Um Daten von Datenquellen abzurufen, übergeben Anwendungen Abfragen in DB2-SQL an die föderierte Datenbank. Die föderierte Datenbank verteilt die Abfragen dann an die entsprechenden Datenquellen, ruft die angeforderten Daten ab und gibt diese Daten an die Anwendungen zurück. Da die Interaktion der föderierten Datenbank mit den Datenquellen jedoch über Kurznamen erfolgt, muss Folgendes beachtet werden:

- Die SQL-Einschränkungen bei der Verwendung von Kurznamen
- Die Art und Weise der Ausführung von Operationen für Objekte mit Kurznamen

Tools der DB2-Produktfamilie

Zur Interaktion mit einer föderierten Datenbank können Sie Host-Tools und Tools für mittlere Systeme verwenden.

Host-Tools und Tools für mittlere Systeme, die Sie zur Interaktion mit einer föderierten Datenbank verwenden können, umfassen Folgendes:

- DB2-Tools zur Verbesserung der Leistung und Verwaltung für DB2 für z/OS und DB2 für Linux, UNIX und Windows
- Interactive SQL (STRSQL) unter DB2 für System i

IBM Optim Data Studio

IBM Optim Data Studio ist eine umfassende Datenmanagementlösung, mit deren Hilfe Sie die Informationen bearbeiten können, auf die Sie mit den Funktionen föderierter Datenbanken von IBM InfoSphere Federation Server zugreifen.

Optim Data Studio bietet Tools für die Konfiguration, den Entwurf und die Entwicklung.

- Konfiguration: IBM Optim Database Administrator für DB2 Linux, UNIX und Windows

Optim Database Administrator erleichtert die Zusammenarbeit von Datenbankadministratoren, Architekten und Entwicklern und vereinfacht den Prozess für die Erkennung, Analyse und Implementierung von Änderungen des Datenbankschemas bei DB2 für Linux, UNIX und Windows.

- Entwurf: IBM InfoSphere Data Architect

InfoSphere Data Architect ist eine umfassende Entwicklungsumgebung für die Datenmodellierung, die Beziehungsdefinition und Integration von Datenressourcen sowie die Entwicklung von Datenbankanwendungen. Mit InfoSphere Data Architect können Sie Beziehungen zwischen vielen verschiedenen Datenquellen ermitteln und zuordnen, Scripts erstellen, die diese Beziehungen darstellen, und anschließend Scripts auf lokalen oder fernen Server mit oder ohne föderierten Datenbanken implementieren.

- Entwicklung: IBM Optim Development Studio

Optim Development Studio bietet eine vollständige Entwicklungs- und Testumgebung für die Erstellung von Datenbankobjekten, Abfragen, Datenbanklogik und pureQuery-Anwendungen.

Weitere Informationen zu Optim Data Studio finden Sie im Information Center von Integrated Data Management unter der folgenden Adresse:
<http://publib.boulder.ibm.com/infocenter/idm/v2r2/index.jsp>.

Web-Service-Provider

Die Interaktion mit einer föderierten Datenbank kann auch über einen Web-Service-Provider erfolgen.

In föderierten Systemen steht ein Web-Services-Wrapper zur Verfügung, um auf Web-Services zuzugreifen. Hierfür werden SQL-Anweisungen für Kurznamen und Sichten verwendet, die Web-Services aufrufen. Sie können einen Web-Services-Wrapper und Kurznamen mit Eingabe für den Web-Service erstellen und auf die Ausgabe des Web-Service mithilfe von SELECT-Anweisungen zugreifen.

Der Systemkatalog der föderierten Datenbank

Der Systemkatalog der föderierten Datenbank enthält Informationen über Objekte in der föderierten Datenbank und über Objekte in den Datenquellen.

Der Katalog in einer föderierten Datenbank wird als globaler Katalog bezeichnet, da er Informationen über das gesamte föderierte System enthält. Das DB2-Abfrageoptimierungsprogramm verwendet die Informationen des globalen Katalogs und des Datenquellenwrappers, um die beste Möglichkeit für die Verarbeitung von SQL-Anweisungen zu ermitteln. Die im globalen Katalog gespeicherten Informationen umfassen ferne und lokale Informationen wie beispielsweise Spaltennamen, Spaltendatentypen, Spaltenstandardwerte, Indexinformationen und Statistikdaten.

Ferne Kataloginformationen sind die Informationen oder Namen, die von der Datenquelle verwendet werden. Lokale Kataloginformationen sind die Informationen oder Namen, die von der föderierten Datenbank verwendet werden. Beispiel: Eine ferne Tabelle enthält eine Spalte mit dem Namen *EMPNO*. Der globale Katalog speichert den fernen Spaltennamen als *EMPNO*. Wenn Sie keinen anderen Namen vergeben, wird der lokale Spaltenname als *EMPNO* gespeichert. Sie können den lokalen Spaltennamen in *Employee_Number* ändern. Wenn Benutzer Abfragen übergeben, die diese Spalte enthalten, dann müssen sie in den Abfragen anstelle von *EMPNO* die Spalte *Employee_Number* verwenden. Verwenden Sie die Anweisung *ALTER NICKNAME*, um den lokalen Namen der Datenquellenspalten zu ändern.

Bei relationalen und nicht relationalen Datenquellen enthalten die im globalen Katalog gespeicherten Informationen sowohl ferne als auch lokale Informationen.

Zum Anzeigen der im globalen Katalog gespeicherten Tabelleninformationen der Datenquelle fragen Sie die Katalogsichten *SYSCAT.TABLES*, *SYSCAT.NICKNAMES*, *SYSCAT.TABOPTIONS*, *SYSCAT.INDEXES*, *SYSCAT.INDEXOPTIONS*, *SYSCAT.COLUMNS* und *SYSCAT.COLOPTIONS* in der föderierten Datenbank ab.

Im globalen Katalog sind auch andere Informationen über die Datenquellen enthalten. Dies sind beispielsweise Informationen, die der Server mit föderierten Datenbanken für die Verbindung zur Datenquelle und für die Zuordnung zwischen Benutzerberechtigungen für die föderierte Datenbank und Benutzerberechtigungen für die Datenquelle verwendet. Der globale Katalog enthält Attribute der Datenquelle, die Sie explizit festgelegt haben, beispielsweise Serveroptionen.

Der SQL-Compiler

Der DB2-SQL-Compiler stellt Informationen zusammen, um die Verarbeitung von Abfragen zu unterstützen.

Um Daten aus Datenquellen abzurufen, übergeben Benutzer und Anwendungen Abfragen in SQL an die föderierte Datenbank. Wenn die Abfrage übergeben wird, ruft der DB2-SQL-Compiler die Informationen im globalen Katalog und im Wrapper der Datenquelle ab, um die Verarbeitung der Abfrage zu unterstützen. Hierbei handelt es sich um Informationen über die Verbindung zur Datenquelle, Serverinformationen, Zuordnungen, Indexinformationen und Verarbeitungsstatistiken.

Das Abfrageoptimierungsprogramm

Als Teil der Verarbeitung des SQL-Compilers wird die Abfrage vom Abfrageoptimierungsprogramm analysiert. Der Compiler entwickelt für die Verarbeitung der Abfrage alternative Strategien, die als Zugriffspläne bezeichnet werden.

Die Zugriffspläne können die Abfrage unter folgenden Bedingungen aufrufen:

- Die Verarbeitung der Abfrage soll durch die Datenquellen erfolgen.
- Die Verarbeitung der Abfrage soll durch den Server mit föderierten Datenbanken erfolgen.
- Die Verarbeitung der Abfrage soll teilweise durch die Datenquellen und teilweise durch den Server mit föderierten Datenbanken erfolgen.

Das Abfrageoptimierungsprogramm wertet die Zugriffspläne in erster Linie auf der Grundlage von Informationen über das Leistungsspektrum der Datenquelle und der Daten aus. Diese Informationen sind im Wrapper und im globalen Katalog enthalten. Das Abfrageoptimierungsprogramm zerlegt die Abfrage in Segmente, die als Abfragefragmente bezeichnet werden. Normalerweise ist es effizienter, ein Abfragefragment an eine Datenquelle im Pushdown-Modus weiterzuleiten, wenn die Datenquelle das Fragment verarbeiten kann. Das Abfrageoptimierungsprogramm berücksichtigt jedoch andere Faktoren, zu denen folgende gehören:

- Die zu verarbeitende Datenmenge
- Die Verarbeitungsgeschwindigkeit der Datenquelle
- Die vom Fragment zurückgegebene Datenmenge
- Die Bandbreite der Übertragung
- Das Vorhandensein einer verwertbaren MQT auf dem Server mit föderierten Datenbanken, die die gleichen Abfrageergebnisse darstellt

Das Abfrageoptimierungsprogramm generiert alternative Zugriffspläne für die Verarbeitung eines Abfragefragments. Die Alternativpläne führen verschiedene Arbeitsmengen lokal auf dem Server mit föderierten Datenbanken sowie in den fernen Datenquellen aus. Da das Abfrageoptimierungsprogramm aufwandsbasiert ist, ordnet es die Aufwände für die Ressourcenauslastung den alternativen Zugriffsplänen zu. Anschließend wählt das Abfrageoptimierungsprogramm den Plan aus, der die Abfrage mit dem niedrigsten Ressourcenauslastungsaufwand verarbeiten kann.

Sollen Fragmente durch Datenquellen verarbeitet werden, übergibt die föderierte Datenbank diese Fragmente an die Datenquellen. Nachdem die Datenquellen die Fragmente verarbeitet haben, werden die Ergebnisse abgerufen und an die föderierte Datenbank zurückgegeben. Hat die föderierte Datenbank einen Teil der Ver-

arbeitung durchgeführt, kombiniert sie ihre Ergebnisse mit den aus der Datenquelle abgerufenen Ergebnissen. Anschließend gibt die föderierte Datenbank alle Ergebnisse an den Client zurück.

Kompensation

Die Fähigkeit von IBM InfoSphere Federation Server, einen von einer Datenquelle nicht unterstützten SQL-Code zu verarbeiten, wird als *Kompensation* bezeichnet.

Die föderierte Datenbank leitet ein Abfragefragment nicht im Pushdown-Modus weiter, wenn die Datenquelle das Fragment nicht verarbeiten kann oder wenn der Server mit föderierten Datenbanken es schneller verarbeiten kann als die Datenquelle. Beispiel: Die SQL-Programmversion der Datenquelle unterstützt keine Gruppierung CUBE in der Klausel GROUP BY. Enthält eine Abfrage eine Gruppierung CUBE und verweist diese Abfrage auf eine Tabelle in dieser Datenquelle, wird diese Abfrage an den Server mit föderierten Datenbanken übergeben. Die föderierte Datenbank leitet die Gruppierung CUBE nicht im Pushdown-Modus an die Datenquelle weiter, sondern führt die CUBE-Verarbeitung selbst durch.

Die föderierte Datenbank kompensiert die nicht ausreichende Funktionalität der Datenquelle auf zwei Arten:

- Sie kann anfordern, dass die Datenquelle mindestens eine Operation verwendet, die der DB2-Funktion in der Abfrage entspricht. Beispiel: Eine Datenquelle unterstützt nicht die Kotangensfunktion (COT(x)), aber die Tangensfunktion (TAN(x)). Die föderierte Datenbank kann anfordern, dass die Datenquelle die Berechnung (1/TAN(x)) durchführt, die der Kotangensfunktion (COT(x)) entspricht.
- Die Datensätze können an den Server mit föderierten Datenbanken zurückgegeben werden, und die Funktion wird lokal ausgeführt.

Bei relationalen Datenquellen unterstützen alle Typen von Verwaltungssystemen für relationale Datenbanken (RDBMS) eine Untergruppe des internationalen SQL-Standards. Manche RDBMS-Typen unterstützen zusätzlich SQL-Konstrukte, die über diesen Standard hinausgehen. Eine *SQL-Programmversion* ist die Gesamtheit des SQL, das ein RDBMS-Typ unterstützt. Wird ein SQL-Konstrukt in der DB2-SQL-Programmversion gefunden, nicht jedoch in der Programmversion der relationalen Datenquelle, kann der Server mit föderierten Datenbanken dieses Konstrukt für die Datenquelle implementieren.

Die föderierte Datenbank kann Unterschiede zwischen SQL-Programmversionen kompensieren. Ein Beispiel für diese Fähigkeit ist die Klausel 'allgemeiner Tabellenausdruck'. DB2-SQL schließt die Klausel 'allgemeiner Tabellenausdruck' mit ein. In dieser Klausel kann ein Name angegeben werden, mit dem alle Klauseln FROM in einem Fullselect auf eine Ergebnismenge verweisen können. Der Server mit föderierten Datenbanken verarbeitet dann einen allgemeinen Tabellenausdruck für eine Datenquelle, obwohl die SQL-Programmversion der Datenquelle allgemeine Tabellenausdrücke nicht mit einschließt.

Durch die Kompensation kann die föderierte Datenbank bei Abfragen von Datenquellen die vollständige DB2-SQL-Programmversion unterstützen. Die Vorteile dieser Kompensation gelten auch für Datenquellen, die SQL nur eingeschränkt oder überhaupt nicht unterstützen. In einem föderierten System muss - außer bei einer Durchgriffssitzung - die DB2-SQL-Programmversion verwendet werden.

Durchgriffssitzungen

Sie können SQL-Anweisungen direkt an Datenquellen übergeben, indem Sie einen speziellen Modus verwenden, der *Durchgriff* genannt wird.

Sie übergeben SQL-Anweisungen in der SQL-Programmversion, die von der Datenquelle verwendet wird. Verwenden Sie eine Durchgriffssitzung, wenn Sie eine Operation durchführen möchten, die von der DB2-SQL oder -API nicht unterstützt wird. Verwenden Sie beispielsweise eine Durchgriffssitzung, um unter Verwendung der nativen Programmversion der Datenquelle eine Prozedur oder einen Index zu erstellen oder Abfragen auszuführen.

Momentan unterstützen die Datenquellen mit Unterstützung für die Durchgriffsfunktion diese Funktion über SQL. Zukünftig wird es möglich sein, dass Datenquellen Durchgriffssitzungen auch mit anderen Sprachen unterstützen.

Sie können eine Durchgriffssitzung ebenso verwenden, um von SQL nicht unterstützte Aktionen auszuführen, wie beispielsweise bestimmte Verwaltungstasks. Allerdings können Sie nicht alle Verwaltungstasks mithilfe einer Durchgriffssitzung durchführen. Sie können beispielsweise Tabellen in der Datenquelle erstellen oder löschen. Sie können jedoch die ferne Datenbank nicht starten oder stoppen.

Sie können in einer Durchgriffssitzung sowohl statisches als auch dynamisches SQL verwenden.

Der Server mit föderierten Datenbanken stellt die folgenden SQL-Anweisungen für die Verwaltung von Durchgriffssitzungen zur Verfügung:

SET PASSTHRU

Eröffnet eine Durchgriffssitzung. Geben Sie eine weitere Anweisung SET PASSTHRU aus, um eine neue Durchgriffssitzung zu starten, wird die aktuelle Durchgriffssitzung beendet.

SET PASSTHRU RESET

Beendet die aktuelle Durchgriffssitzung.

GRANT (Server Privileges)

Erteilt einem Benutzer, einer Gruppe, einer Liste von Berechtigungs-IDs oder PUBLIC die Berechtigung, Durchgriffssitzungen zu einer bestimmten Datenquelle einzuleiten.

REVOKE (Server Privileges)

Widerruft die Berechtigung zur Einleitung von Durchgriffssitzungen.

Serverdefinitionen und Serveroptionen

Nach der Erstellung des Wrappers für die Datenquelle gibt der Instanzeigner die Datenquellen für die föderierte Datenbank an.

Der Instanzeigner stellt einen Namen zur Identifizierung der Datenquelle sowie weitere die Datenquelle betreffende Informationen zur Verfügung. Diese Informationen umfassen Folgendes:

- Typ und Version der Datenquelle
- Datenbanknamen der Datenquelle (nur RDBMS)
- Spezifische Metadaten der Datenquelle

Eine Datenquelle der DB2-Produktfamilie kann beispielsweise mehrere Datenbanken haben. In der Definition muss angegeben werden, zu welcher Datenbank der Server mit föderierten Datenbanken eine Verbindung herstellen kann. Im Gegensatz dazu hat eine Oracle-Datenquelle eine Datenbank. Der Server mit föderierten Datenbanken kann zu dieser Datenbank eine Verbindung herstellen, ohne ihren Namen zu kennen. Der Name der Datenbank ist in der Serverdefinition des Servers mit föderierten Datenbanken für eine Oracle-Datenquelle nicht mit eingeschlossen.

Der Name und die anderen Informationen, die der Instanzeigner dem Server mit föderierten Datenbanken zur Verfügung stellt, werden zusammen als *Serverdefinition* bezeichnet. Datenquellen antworten auf Datenanforderungen und sind Server mit eigenen Rechten.

Die Anweisungen CREATE SERVER und ALTER SERVER werden verwendet, um eine Serverdefinition zu erstellen oder zu ändern.

Einige der in einer Serverdefinition enthaltenen Informationen sind als *Serveroptionen* gespeichert. Für das Erstellen von Serverdefinitionen ist das Verständnis der Optionen, die für den Server angegeben werden können, wichtig.

Serveroptionen können entweder für die Dauer einer einzelnen Verbindung oder für mehrere aufeinanderfolgende Verbindungen festgelegt werden.

Benutzerzuordnungen

Eine *Benutzerzuordnung* ist eine Zuordnung zwischen einer Berechtigungs-ID auf dem Server mit föderierten Datenbanken und den Informationen, die für die Herstellung einer Verbindung zur fernen Datenquelle erforderlich sind.

Zur Herstellung einer Benutzerzuordnung verwenden Sie die Anweisung CREATE USER MAPPING. In der Anweisung geben Sie die lokale Berechtigungs-ID, den lokalen Namen des fernen Datenquellenservers entsprechend der Serverdefinition sowie die ferne ID und das zugehörige Kennwort an.

Beispiel: Angenommen, Sie haben eine Serverdefinition für einen fernen Server erstellt und 'argon' als lokalen Namen für den fernen Server angegeben. Erstellen Sie die folgende Benutzerzuordnung, um Mary für den Zugriff auf den fernen Server zu berechtigen:

```
CREATE USER MAPPING FOR Mary
SERVER argon
OPTIONS (REMOTE_AUTHID 'ferne_ID', REMOTE_PASSWORD 'fernes_kennwort')
```

Wenn Mary eine SQL-Anweisung zur Herstellung einer Verbindung zum fernen Server eingibt, führt der Server mit föderierten Datenbanken die folgenden Schritte aus:

1. Abrufen der Benutzerzuordnung für Mary
2. Entschlüsseln des fernen Kennworts 'fernes_kennwort', das dem fernen Server zugeordnet ist
3. Aufrufen des Wrappers zur Herstellung der Verbindung zum fernen Server
4. Übergeben der fernen ID 'ferne_ID' und des entschlüsselten fernen Kennworts an den Wrapper
5. Herstellen einer Verbindung zum fernen Server für Mary

Standardmäßig speichert der Server mit föderierten Datenbanken die Benutzerzuordnung in der Sicht SYSCAT.USEROPTIONS im globalen Katalog und verschlüsselt die fernen Kennwörter. Alternativ dazu können Sie ein externes Repository, z. B. eine Datei oder einen LDAP-Server, verwenden, um die Benutzerzuordnungen zu speichern. Erstellen Sie ein Benutzerzuordnungs-Plug-in, um die Schnittstelle zwischen dem Server mit föderierten Datenbanken und dem externen Repository bereitzustellen.

Unabhängig davon, wie Sie die Benutzerzuordnungen speichern, muss der Zugriff darauf sorgfältig durch die entsprechenden Berechtigungen eingeschränkt werden. Wenn die Integrität der Benutzerzuordnungen beeinträchtigt wird, besteht die Gefahr, dass die Daten in den fernen Datenbanken anfällig für unberechtigte Aktivitäten werden.

Kurznamen und Datenquellenobjekte

Ein *Kurzname* ist eine Kennung, die Sie zur Identifikation des Datenquellenobjekts verwenden, auf das Sie zugreifen möchten. Das Objekt, das durch einen Kurznamen identifiziert wird, wird als *Datenquellenobjekt* bezeichnet.

Ein Kurzname ist kein alternativer Name für ein Datenquellenobjekt in der Weise, in der dies bei einem Aliasnamen der Fall ist. Ein Kurzname ist ein Verweis, anhand dessen der Server mit föderierten Datenbanken das Objekt referenziert. Kurznamen werden normalerweise mit der Anweisung CREATE NICKNAME und bestimmten Kurznamenspaltenoptionen und Kurznamenoptionen definiert.

Übergibt eine Clientanwendung oder ein Benutzer eine verteilte Anforderung an den Server mit föderierten Datenbanken, muss die Anforderung die Datenquellen nicht angeben. Stattdessen verweist die Anforderung anhand der Kurznamen auf die Datenquellenobjekte. Kurznamen werden bestimmten Objekten in der Datenquelle zugeordnet. Durch diese Zuordnungen ist keine Qualifikation der Kurznamen durch Datenquellennamen mehr erforderlich. Die Speicherposition der Datenquellenobjekte ist für die Clientanwendung bzw. den Benutzer transparent.

Angenommen, Sie definieren den Kurznamen *DEPT*, der eine Informix-Datenbanktabelle mit dem Namen *NFX1.PERSON* darstellt. Der Server mit föderierten Datenbanken akzeptiert die Anweisung `SELECT * FROM DEPT`. Die Verwendung der Anweisung `SELECT * FROM NFX1.PERSON` über den Server mit föderierten Datenbanken ist (außer bei Durchgriffssitzungen) jedoch nicht zulässig, sofern auf diesem Server nicht eine lokale Tabelle mit dem Namen *NFX1.PERSON* vorhanden ist.

Beim Erstellen eines Kurznamens für ein Datenquellenobjekt werden dem globalen Katalog Metadaten über das Objekt hinzugefügt. Das Abfrageoptimierungsprogramm verwendet diese Metadaten und die Informationen im Wrapper, um den Zugriff auf das Datenquellenobjekt zu vereinfachen. Bezieht sich der Kurzname beispielsweise auf eine Tabelle mit einem Index, so enthält der globale Katalog Informationen über den Index. Der Wrapper enthält die Zuordnungen zwischen den DB2-Datentypen und den Datentypen der Datenquelle.

Kurznamen für Objekte, die die kennsatzbasierte Zugriffssteuerung (LBAC) verwenden, werden nicht zwischengespeichert. Daher bleibt die Sicherheit der Daten im betreffenden Objekt gewährleistet. Wenn Sie beispielsweise mit dem Oracle-Wrapper (Net8) einen Kurznamen für eine Tabelle erstellen, die Oracle Label Security verwendet, wird die Tabelle automatisch als sicher identifiziert. Der daraus resultierende Kurzname kann nicht zwischengespeichert werden. Dies bedeutet,

dass hierfür keine MQTs (Materialized Query Tables) erstellt werden können. Mit der Verwendung von LBAC wird sichergestellt, dass die Informationen nur von Benutzern gelesen werden, die über die entsprechenden Sicherheitszugriffsrechte verfügen. Für Kurznamen, die vor der Einführung der LBAC-Unterstützung erstellt wurden, müssen Sie die Anweisung ALTER NICKNAME verwenden, um kein Caching zuzulassen. LBAC wird sowohl vom DRDA-Wrapper (für Datenquellen, die DB2 für Linux, UNIX und Windows ab Version 9.1 verwenden) als auch vom Net8-Wrapper unterstützt.

Gültige Datenquellenobjekte

Mit Kurznamen werden Objekte in der Datenquelle angegeben, auf die Sie zugreifen wollen. In der folgenden Tabelle sind die Objekttypen aufgelistet, für die Sie in einem föderierten System Kurznamen erstellen.

Tabelle 3. Gültige Datenquellenobjekte

Datenquelle	Gültige Objekte
BioRS	BioRS-Datenbanken
DB2 Database für Linux, UNIX und Windows	Kurznamen, MQTs, Tabellen, Sichten
DB2 für System i	Tabellen, Sichten, physische/logische Dateien (P/L) und Tabellentypen
DB2 für VM und VSE	Tabellen, Sichten
DB2 für z/OS	Tabellen, Sichten
Informix	Tabellen, Sichten, Synonyme
JDBC	Alle Tabellentypen
Microsoft Excel	Dateien mit der Erweiterung .xls (Zugriff nur auf das erste Arbeitsblatt in der Arbeitsmappe)
Microsoft SQL Server	Tabellen, Sichten
ODBC	Tabellen, Sichten
Oracle	Tabellen, Sichten, Synonyme
Script	Scripts
Sybase	Tabellen, Sichten
Teradata	Tabellen, Sichten
Dateien mit Tabellenstruktur	Textdateien in einem bestimmten Format
Web-Services	Operationen in einer WSDL-Datei (WSDL = Web Services Description Language)
XML-Dateien	Elementgruppen in einem XML-Dokument

Kurznamenspaltenoptionen

Sie können im globalen Katalog zusätzliche Metadaten über das Objekt mit dem Kurznamen bereitstellen. Diese Metadaten beschreiben Werte in bestimmten Spalten des Datenquellenobjekts. Sie werden Parametern zugeordnet, die als *Kurznamenspaltenoptionen* bezeichnet werden.

Die Kurznamenspaltenoptionen bewirken, dass der Wrapper die Daten in einer Spalte auf andere Weise verarbeitet, als dies sonst der Fall wäre. Der SQL-Compiler

und das Abfrageoptimierungsprogramm verwenden die Metadaten, um bessere Zugriffspläne für die Daten zu entwickeln.

Kurznamenspaltenoptionen werden auch dazu verwendet, dem Wrapper weitere Informationen zur Verfügung zu stellen. Bei XML-Datenquellen gibt die Kurznamenspaltenoption z. B. an, welchen XPath-Ausdruck der Wrapper verwenden muss, um die Spalte bei einer syntaktischen Analyse aus dem XML-Dokument zu entfernen.

Bei föderierten Datenbanken behandelt der DB2-Server ein Datenquellenobjekt, auf das ein Kurzname verweist, ebenso wie eine lokale DB2-Tabelle. Daher können Sie Kurznamenspaltenoptionen für alle Datenquellenobjekte setzen, für die Sie einen Kurznamen erstellen. Manche Kurznamenspaltenoptionen sind für spezielle Datenquellentypen bestimmt und können nur auf diese Datenquellen angewendet werden.

Angenommen, eine Datenquelle weist eine andere Sortierfolge als die föderierte Datenbank auf. Normalerweise sortiert der Server mit föderierten Datenbanken in diesem Fall in der Datenquelle keine Spalten, die Zeichendaten enthalten. Stattdessen gibt er die Daten an die föderierte Datenbank zurück und führt die Sortierung lokal aus. Nehmen Sie jedoch an, dass die Spalte ein Zeichendatentyp (CHAR oder VARCHAR) ist und ausschließlich numerische Zeichen ('0', '1', ..., '9') enthält. Diese Einstellung können Sie definieren, indem Sie der Kurznamenspaltenoption NUMERIC_STRING den Wert 'Y' zuordnen. Auf diese Weise kann das Abfrageoptimierungsprogramm von DB2 die Sortierung in der Datenquelle durchführen. Wird die Sortierung fern durchgeführt, kann der Systemaufwand für die Übertragung der Daten zum Server mit föderierten Datenbanken und die lokale Sortierung vermieden werden.

Mit den Anweisungen ALTER NICKNAME können Sie Kurznamenspaltenoptionen für relationale Kurznamen definieren. Mit den Anweisungen CREATE NICKNAME und ALTER NICKNAME ist es möglich, Kurznamenspaltenoptionen für nicht relationale Kurznamen anzugeben.

Datentypzuordnungen

Die Datentypen der Datenquelle müssen den entsprechenden Datentypen von DB2 zugeordnet werden, damit der Server mit föderierten Datenbanken die Daten aus den Datenquellen abrufen kann.

Im Folgenden werden einige Beispiele für Standarddatentypzuordnungen aufgeführt:

- Der Oracle-Typ FLOAT wird dem DB2-Typ DOUBLE zugeordnet.
- Der Oracle-Typ DATE wird dem DB2-Typ TIMESTAMP zugeordnet.
- Der Typ DATE von DB2 für z/OS™ wird dem DB2-Typ DATE zugeordnet.

Für die meisten Datenquellen sind die Standardtypzuordnungen in den Wrappern enthalten. Die Standardtypzuordnungen für DB2-Datenquellen sind im DRDA-Wrapper enthalten. Die Standardtypzuordnungen für Informix sind im Informix-Wrapper enthalten, etc.

Für manche nicht relationale Datenquellen müssen die Datentypinformationen in der Anweisung CREATE NICKNAME angegeben werden. Die entsprechenden DB2-Datentypen müssen bei der Erstellung des Kurznamens für jede Spalte des

Datenquellenobjekts angegeben werden. Jede Spalte muss einem bestimmten Feld oder einer bestimmten Spalte im Datenquellenobjekt zugeordnet werden.

Für relationale Datenquellen können die Standarddatentypzuordnungen außer Kraft gesetzt werden. Der Informix-Datentyp INTEGER wird z. B. standardmäßig dem DB2-Datentyp INTEGER zugeordnet. Sie können die Standardzuordnungen jedoch außer Kraft setzen und den Informix-Datentyp INTEGER dem DB2-Datentyp DECIMAL(10,0) zuordnen.

Funktionszuordnungen

Damit der Server mit föderierten Datenbanken eine Datenquellenfunktion erkennen kann, muss die Funktion einer entsprechenden, in DB2 Database für Linux, UNIX und Windows vorhandenen Funktion zugeordnet werden.

IBM InfoSphere Federation Server stellt Standardzuordnungen zwischen vorhandenen Datenquellenfunktionen und den entsprechenden DB2-Funktionen bereit. Für die meisten Datenquellen sind die Standardfunktionszuordnungen in den Wrappern enthalten. Die Standardfunktionszuordnungen zu Funktionen von DB2 für z/OS sind im DRDA-Wrapper enthalten. Die Standardfunktionszuordnungen zu Sybase-Funktionen sind im CTLIB-Wrapper enthalten usw.

Für relationale Datenquellen können Sie eine Funktionszuordnung erstellen, wenn Sie eine Datenquellenfunktion verwenden wollen, die der Server mit föderierten Datenbanken nicht erkennt. Die Zuordnung, die Sie erstellen, besteht zwischen der Datenquellenfunktion und einer entsprechenden DB2-Funktion in der föderierten Datenbank. Funktionszuordnungen werden normalerweise verwendet, wenn eine neue integrierte oder benutzerdefinierte Funktion in der Datenquelle verfügbar wird. Funktionszuordnungen werden auch verwendet, wenn keine entsprechende Funktion in DB2 vorhanden ist. In diesem Fall muss auch eine Funktionsschablone erstellt werden.

Indexspezifikationen

Beim Erstellen eines Kurznamens für eine Datenquellentabelle werden dem globalen Katalog Informationen über alle Indizes der Datenquellentabelle hinzugefügt. Das Abfrageoptimierungsprogramm verwendet diese Informationen zur Beschleunigung der Verarbeitung von verteilten Anforderungen. Die Kataloginformationen über den Index einer Datenquelle bestehen aus einem Satz von Metadaten und werden als Indexspezifikation bezeichnet.

Das Abfrageoptimierungsprogramm verwendet diese Informationen zur Beschleunigung der Verarbeitung von verteilten Anforderungen.

Ein Server mit föderierten Datenbanken erzeugt bei der Erstellung von Kurznamen für folgende Objekte keine Indexspezifikation:

- Eine Tabelle ohne Indizes
- Eine Sicht, für die normalerweise keine Indexinformationen im fernen Katalog gespeichert sind
- Ein Datenquellenobjekt, das über keinen fernen Katalog verfügt, von dem der Server mit föderierten Datenbanken die Indexinformationen abrufen kann

Angenommen, eine Tabelle erhält zusätzlich zu den zum Zeitpunkt der Erstellung des Kurznamens vorhandenen Indizes einen neuen Index. Da die Indexinformationen zum Zeitpunkt der Erstellung des Kurznamens an den globalen Katalog gegeben werden, bemerkt der Server mit föderierten Datenbanken den neuen Index nicht. Ebenso bemerkt der Server beim Erstellen eines Kurznamens für eine Sicht nicht die zugrunde liegende Tabelle (und deren Indizes), von der die Sicht erstellt wurde. Unter diesen Umständen können Sie dem globalen Katalog die erforderlichen Indexinformationen zur Verfügung stellen. Sie können eine Indexspezifikation für Tabellen ohne Indizes erstellen. Mithilfe der Indexspezifikation kann das Abfrageoptimierungsprogramm feststellen, welche Tabellenspalte(n) durchsucht werden muss bzw. müssen, um Daten schnell zu finden.

Föderierte gespeicherte Prozeduren

Über den Zugriff auf föderierte Prozeduren können Benutzer von föderierten Systemen auf ferne gespeicherte Prozeduren in fernen Datenquellen zugreifen.

Bei einer föderierten gespeicherten Prozedur handelt es sich um eine lokale gespeicherte Prozedur, die einer gespeicherten Prozedur in einer Datenquelle zugeordnet ist. Eine föderierte gespeicherte Prozedur wird mithilfe der Anweisung `CREATE PROCEDURE` (abgeleitete Funktion) registriert.

Sortierfolgen

Die Reihenfolge, in der Zeichendaten in einer Datenbank sortiert werden, hängt von der Struktur der Daten und der Sortierfolge ab, die für die Datenbank definiert wurde.

Nehmen Sie an, die Daten in einer Datenbank bestehen nur aus Großbuchstaben und enthalten keine numerischen Werte oder Sonderzeichen. Eine Sortierung der Daten sollte unabhängig davon, ob diese an der Datenquelle oder in der föderierten Datenbank ausgeführt wird, zur gleichen Ausgabe führen. Die von den einzelnen Datenbanken verwendete Sortierfolge sollte in diesem Fall keine Auswirkungen auf die Sortierergebnisse haben. Wenn alle Daten in der Datenbank in Kleinbuchstaben vorliegen oder nur aus numerischen Zeichen bestehen, sollte eine Sortieroperation unabhängig vom System, auf dem diese ausgeführt wird, ebenso zu identischen Ergebnissen führen.

Anders liegt der Fall, wenn die Daten aus folgenden Strukturen bestehen:

- Kombination aus Buchstaben und numerischen Zeichen
- Groß- und Kleinbuchstaben
- Sonderzeichen wie z. B. @, #, €

Sortierungen solcher Daten können zu unterschiedlichen Ausgaben führen, wenn die föderierte Datenbank und die Datenquelle unterschiedliche Sortierfolgen verwenden.

Allgemein lässt sich eine Sortierfolge als definierte Anordnung von Zeichendaten beschreiben, die festlegt, ob ein bestimmtes Zeichen innerhalb der Sortierung an einer höheren, an einer niedrigeren oder an der gleichen Position wie ein anderes Zeichen angeordnet wird.

Bestimmen der Sortierreihenfolge durch die Sortierfolge

Eine Sortierfolge legt die Sortierreihenfolge der Zeichen innerhalb eines codierten Zeichensatzes fest.

Bei einem Zeichensatz handelt es sich um eine Gruppe von Zeichen, die in einem Datenverarbeitungssystem oder einer Programmiersprache eingesetzt wird. Innerhalb eines codierten Zeichensatzes ist jedem Zeichen ein bestimmter numerischer Wert innerhalb des Bereichs von 0 bis 255 (bzw. dessen hexadezimaler Entsprechung) zugeordnet. Diese numerischen Werte werden als Codepunkte, die Zuordnungen der numerischen Werte zu den einzelnen Zeichen der Gruppe werden zusammen als Codepage bezeichnet.

Zusätzlich zu seiner Zuordnung zu einem bestimmten Zeichen kann ein Codepunkt auch der Position des zugehörigen Zeichens innerhalb der Sortierreihenfolge zugeordnet sein. In der Fachsprache wird eine Sortierfolge also als die Gesamtheit der Zuordnungen der Codepunkte eines Zeichensatzes zu den Sortierreihenfolgepositionen der einzelnen Zeichen dieses Zeichensatzes bezeichnet. Die Position eines Zeichens wird durch eine Zahl dargestellt, die als Wertigkeit des Zeichens bezeichnet wird. In der einfachsten Sortierfolge, der so genannten Identitätssortierfolge, stimmen die Wertigkeiten mit den jeweiligen Codepunkten überein.

Beispiel: Datenbank ALPHA verwendet die Standardsortierfolge der EBCDIC-Codepage. Die Datenbank BETA verwendet die Standardsortierfolge der ASCII-Codepage. Die Sortierreihenfolgen für Zeichenfolgen in den beiden Datenbanken weichen voneinander ab:

```
SELECT.....  
ORDER BY COL2
```

EBCDIC-Sortierung	ASCII-Sortierung
COL2	COL2
----	----
V1G	7AB
Y2W	V1G
7AB	Y2W

Beispiel: In ähnlicher Weise hängt die Art des Zeichenvergleichs innerhalb einer Datenbank ebenfalls von der Sortierfolge ab, die für diese Datenbank definiert wurde. Die Datenbank ALPHA verwendet die Standardsortierfolge der EBCDIC-Codepage. Die Datenbank BETA verwendet die Standardsortierfolge der ASCII-Codepage. Zeichenvergleichsoperationen in diesen beiden Datenbanken erzielen unterschiedliche Ergebnisse:

```
SELECT.....  
WHERE COL2 > 'TT3'
```

EBCDIC-Ergebnisse	ASCII-Ergebnisse
COL2	COL2
----	----
TW4	TW4
X82	X82
39G	

Definieren der lokalen Sortierfolge zum Optimieren von Abfragen

Administratoren können föderierte Datenbanken mit einer bestimmten Sortierfolge erstellen, die mit der Sortierfolge einer Datenquelle übereinstimmt.

In diesem Fall wird die Serveroption `COLLATING_SEQUENCE` für alle Serverdefinitionen der Datenquellen auf den Wert `'Y'` gesetzt. Mit dieser Einstellung wird die föderierte Datenbank darüber informiert, dass die Sortierfolgen der föderierten Datenbank und der Datenquelle übereinstimmen.

Die Sortierfolge der föderierten Datenbank wird als Teil des Befehls `CREATE DATABASE` definiert. Mithilfe dieses Befehls können Sie eine der folgenden Sortierfolgen angeben:

- Eine Identitätssortierfolge
- Eine Systemsortierfolge (die Sortierfolge, die von dem Betriebssystem verwendet wird, das die Datenbank unterstützt)
- Eine angepasste Sortierfolge (eine vordefinierte Sortierfolge, die vom DB2-Datenbanksystem bereitgestellt wird, oder die vom Benutzer definiert wird)

Angenommen, es wird eine Datenquelle von DB2 für z/OS verwendet. Sortieroperationen, die in einer `ORDER BY`-Klausel definiert sind, werden von einer Sortierfolge implementiert, die auf einer EBCDIC-Codepage basiert. Wenn Sie DB2 für z/OS-Daten in einem nach `ORDER BY`-Klauseln sortierten Format abrufen wollen, müssen Sie die föderierte Datenbank so konfigurieren, dass die vordefinierte Sortierfolge auf der Basis der entsprechenden EBCDIC-Codepage verwendet wird.

Kapitel 2. Modifizieren von Datenquellenkonfigurationen

Im Laufe der Zeit kann es erforderlich werden, die Datenquellenkonfigurationen für das System föderierter Datenbanken zu modifizieren. So müssen Sie z. B., wenn sich die Datenquellen ändern, möglicherweise Serverdefinitionen, Kurznamen und Benutzerzuordnungen aktualisieren. Darüber hinaus kann es erforderlich sein, Berechtigungen für den Zugriff auf Datenquellen von Ihrem System föderierter Datenbanken aus hinzuzufügen oder zu entfernen.

Ändern von Wrappern

- Ändern eines Wrappers (DB2-Steuerzentrale)
- Ändern eines Wrappers (DB2-Befehlszeile)

Ändern von Serverdefinitionen und -optionen

- Ändern von Serverdefinitionen und Serveroptionen
- Verwenden von Serveroptionen in Serverdefinitionen (DB2-Steuerzentrale)
- Verwenden von Serveroptionen in Serverdefinitionen (DB2-Befehlszeile)

Ändern von Benutzerzuordnungen

- Ändern einer Benutzerzuordnung (DB2-Steuerzentrale)
- Ändern einer Benutzerzuordnung (DB2-Befehlszeile)

Ändern von Kurznamen

- Ändern eines Kurznamens (DB2-Steuerzentrale)
- Ändern eines Kurznamens (DB2-Befehlszeile)

Löschen von Wrappern, Serverdefinitionen, Benutzerzuordnungen und Kurznamen

- Löschen eines Wrappers
- Löschen einer Serverdefinition
- Löschen einer Benutzerzuordnung
- Löschen eines Kurznamens

Ändern eines Wrappers (DB2-Steuerzentrale)

Nach dem Konfigurieren eines Wrappers können Sie die Anweisung ALTER WRAPPER verwenden, um die Konfiguration entsprechend Ihrer Systemanforderungen zu ändern. Wrapper können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die der Anweisung zugeordnete Berechtigungs-ID muss über die Berechtigung SYSADM oder DBADM verfügen.

Einschränkungen

Die Wrapperoption DB2_FENCED kann nicht gelöscht werden.

Der Server mit föderierten Datenbanken kann die Anweisung ALTER WRAPPER in einer bestimmten Arbeitseinheit (UOW) nicht verarbeiten, wenn die UOW bereits eine der folgenden Anweisungen enthält:

- Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist, die der Wrapper einbezieht
- Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in der Datenquelle, die der Wrapper einbezieht
- Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde, die der Wrapper einbezieht

Informationen zu dieser Task

Diese Task beschreibt das Ändern eines Wrappers über die DB2-Steuerzentrale.

Vorgehensweise

Gehen Sie wie folgt vor, um einen Wrapper über die DB2-Steuerzentrale zu ändern:

1. Erweitern Sie den Ordner 'Objekte föderierter Datenbanken'. Die Wrapperobjekte werden im Inhaltsteilfenster des Fensters der DB2-Steuerzentrale angezeigt.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Wrapper an, und klicken Sie in der Liste der Aktionen anschließend **Ändern** an. Daraufhin wird das Notizbuch 'Wrapper ändern' geöffnet.
 - a. Nehmen Sie die Änderungen auf der Seite 'Einstellungen' vor.
 - b. Klicken Sie **Variablen setzen** an, um die Umgebungsvariablen der Datenquelle für den Wrapper zu definieren. Umgebungsvariablen sind nicht für alle Wrapper erforderlich.
3. Klicken Sie **OK** an, um den Wrapper zu ändern und das Notizbuch 'Wrapper ändern' zu schließen.

Ändern eines Wrappers - Beispiele

Dieser Abschnitt enthält Beispiele für das Ändern von Wrapperoptionen mit der Anweisung ALTER WRAPPER.

Setzen Sie die folgende Anweisung ab, um für den Wrapper mit dem Namen 'drda' die Option DB2_FENCED in 'Y' zu ändern:

```
ALTER WRAPPER drda OPTIONS (SET DB2_FENCED 'Y');
```

Setzen Sie die folgende Anweisung ab, um für den Wrapper mit dem Namen 'odbc' die Option MODULE in '/opt/odbc/lib/libodbc.a(odbc.so)' zu ändern:

```
ALTER WRAPPER odbc OPTIONS (SET MODULE '/opt/odbc/lib/libodbc.a(odbc.so)');
```

Ändern eines Wrappers (DB2-Befehlszeile)

Nach dem Konfigurieren eines Wrappers können Sie die Anweisung ALTER WRAPPER verwenden, um die Konfiguration entsprechend Ihrer Systemanforderungen zu ändern. Wrapper können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die der Anweisung zugeordnete Berechtigungs-ID muss über die Berechtigung SYSADM oder DBADM verfügen.

Einschränkungen

Die Wrapperoption DB2_FENCED kann nicht gelöscht werden.

Der Server mit föderierten Datenbanken kann die Anweisung ALTER WRAPPER in einer bestimmten Arbeitseinheit (UOW) nicht verarbeiten, wenn die UOW bereits eine der folgenden Anweisungen enthält:

- Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist, die der Wrapper einbezieht
- Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in der Datenquelle, die der Wrapper einbezieht
- Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde, die der Wrapper einbezieht

Informationen zu dieser Task

Diese Task beschreibt das Ändern eines Wrappers von der DB2-Befehlszeile. Die Anweisung ALTER WRAPPER kann zum Hinzufügen, Definieren oder Löschen einer oder mehrerer Wrapperoptionen verwendet werden.

Vorgehensweise

Um einen Wrapper über die DB2-Befehlszeile zu ändern, setzen Sie die Anweisung ALTER WRAPPER ab.

Ändern von Serverdefinitionen und Serveroptionen

Zum Ändern einer Serverdefinition wird die Anweisung ALTER SERVER verwendet. Einige der in einer Serverdefinition enthaltenen Informationen sind als Serveroptionen gespeichert. Beim Ändern einer Serverdefinition ist es wichtig, die Optionen zu kennen, die für den Server angegeben werden können.

Eine Serverdefinition identifiziert für die föderierte Datenbank eine Datenquelle. Die Serverdefinition besteht aus einem lokalen Namen und anderen Informationen über den Server dieser Datenquelle. Die Serverdefinition wird vom Wrapper verwendet, wenn SQL-Anweisungen mit Kurznamen an die föderierte Datenbank übergeben werden.

Für relationale Datenquellen können Serveroptionen unter Verwendung der Anweisung SET SERVER OPTION temporär definiert werden. Diese Anweisung setzt den Wert der Serveroption in der Serverdefinition für die Dauer einer einzigen Verbindung zur föderierten Datenbank außer Kraft. Bei statischem SQL hat die Verwendung der Anweisung SET SERVER OPTION nur Auswirkungen auf die Ausführung der statischen SQL-Anweisung. Die Verwendung der Option SET SERVER OPTION hat keine Auswirkungen auf die von der Optimierungsfunktion generierten Pläne.

Ändern Sie eine Serverdefinition in den folgenden Fällen:

- Wenn Sie ein Upgrade auf eine neue Version der Datenquelle durchführen

- Wenn Sie bei allen Serverdefinitionen für einen bestimmten Datenquellentyp dieselbe Änderung vornehmen möchten
- Wenn Sie eine Serveroption für eine vorhandene Serverdefinition hinzufügen oder ändern möchten

Einschränkungen beim Ändern von Serverdefinitionen

Beim Ändern von Serverdefinitionen müssen Sie einige Einschränkungen berücksichtigen.

Für das Ändern von Serverdefinitionen gelten die folgenden Einschränkungen:

- Sie können in der Anweisung ALTER SERVER keinen Wrapper angeben, der auf dem Server mit föderierten Datenbanken nicht registriert ist.
- Der Server mit föderierten Datenbanken kann die Anweisung ALTER SERVER innerhalb einer gegebenen Arbeitseinheit (UOW) unter den folgenden Bedingungen nicht verarbeiten:
 - Die Anweisung verweist auf eine einzelne Datenquelle, und die UOW enthält bereits eine der folgenden Anweisungen:
 - Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist
 - Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in der Datenquelle
 - Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde
 - Die Anweisung verweist auf eine Kategorie von Datenquellen (z. B. alle Datenquellen eines bestimmten Typs und einer bestimmten Version), und die UOW enthält bereits eine der folgenden Anweisungen:
 - Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in einer der Datenquellen verweist
 - Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in einer der Datenquellen
 - Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in einer der Datenquellen abgesetzt wurde
- Der Server mit föderierten Datenbanken prüft nicht, ob die von Ihnen angegebene Serverversion mit der Version des fernen Servers übereinstimmt. Das Angeben einer falschen Serverversion kann zu SQL-Fehlern führen, wenn Sie auf Kurznamen zugreifen, die zur DB2-Serverdefinition gehören. Diese SQL-Fehler treten am häufigsten auf, wenn eine Serverversion angegeben wird, die neuer als die tatsächliche Version des fernen Servers ist. Wenn Sie auf Kurznamen zugreifen, die zur Serverdefinition gehören, sendet der Server mit föderierten Datenbanken in diesem Fall möglicherweise SQL an den fernen Server, die dieser nicht erkennen kann. Bei der Anweisung ALTER SERVER gilt diese Situation nur für die Form der Anweisung, die die Serverversion ändert (*servername* VERSION *server-version*).
- Sie müssen den vollständigen Namen der Serveroption angeben. Sie können beispielsweise nicht die Abkürzung DB2_TWO_PHASE angeben. Sie müssen stattdessen den vollständigen Namen der Serveroption, DB2_TWO_PHASE_COMMIT, angeben.

Ändern der Datenquellenversion in einer Serverdefinition (DB2-Steuerzentrale)

Sie können eine vorhandene Serverdefinition ändern, um die Version der Datenquelle zu ändern, die vom fernen Server verwendet wird. Serverdefinitionen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, die die Anweisung ALTER SERVER absetzt, muss über die Berechtigung SYSADM oder DBADM in der föderierten Datenbank verfügen.

Einschränkungen

Informationen hierzu finden Sie unter Einschränkungen beim Ändern von Serverdefinitionen.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Serverdefinition über die DB2-Steuerzentrale zu ändern:

1. Erweitern Sie den Ordner 'Objekte föderierter Datenbanken'. Die Objekte der Serverdefinition werden im Inhaltsteilfenster des Fensters der DB2-Steuerzentrale angezeigt.
2. Klicken Sie mit der rechten Maustaste die zu ändernde Serverdefinition an, und klicken Sie in der Liste der Aktionen anschließend **Ändern** an. Daraufhin wird das Notizbuch 'Serverdefinition ändern' geöffnet.
3. Klicken Sie auf der Seite 'Server' den Pfeil **Version** an, um eine andere Version für die Datenquelle anzugeben.
4. Klicken Sie **OK** an, um die Serverdefinition zu ändern und das Notizbuch 'Serverdefinition ändern' zu schließen.

Ändern der Datenquellenversion in einer Serverdefinition (DB2-Befehlszeile)

Sie können eine vorhandene Serverdefinition ändern, um die Version der Datenquelle zu ändern, die vom fernen Server verwendet wird. Serverdefinitionen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, die die Anweisung ALTER SERVER absetzt, muss über die Berechtigung SYSADM oder DBADM in der föderierten Datenbank verfügen.

Einschränkungen

Informationen hierzu finden Sie unter Einschränkungen beim Ändern von Serverdefinitionen.

Vorgehensweise

Um eine Serverdefinition von der DB2-Befehlszeile zu ändern, setzen Sie die Anweisung ALTER SERVER ab.

Beispiel: Sie arbeiten mit einer Serverdefinition für eine Microsoft SQL Server Version 6.5-Datenquelle. Der Name, den Sie dem Server in der Anweisung CREATE SERVER zugeordnet haben, lautet `SQLSVR_ASIA`. Wenn ein Upgrade von Microsoft SQL Server auf Version 7.0 durchgeführt wird, ist die Serverdefinition mit der folgenden Anweisung zu ändern:

```
ALTER SERVER SQLSVR_ASIA VERSION 7
```

Ändern aller Serverdefinitionen für einen bestimmten Datenquellentyp

Sie können alle vorhandenen Serverdefinitionen für einen bestimmten Typ von Datenquelle mit einer einzigen Anweisung ALTER SERVER ändern. Verwenden Sie eine einzige Anweisung, wenn dieselbe Änderung bei allen Serverdefinitionen desselben Typs ausgeführt werden soll.

Vorbereitung

Die Berechtigungs-ID, die die Anweisung ALTER SERVER absetzt, muss über die Berechtigung SYSADM oder DBADM in der föderierten Datenbank verfügen.

Einschränkungen

Sie können Serveroptionen mit der Anweisung ALTER SERVER für einen ganzen Typ von Datenquellen nur definieren oder löschen, wenn die Serveroptionen zuvor durch eine ALTER SERVER-Anweisung hinzugefügt wurden.

Vorgehensweise

Setzen Sie eine einzige Anweisung ALTER SERVER ab, um alle vorhandenen Serverdefinitionen für einen bestimmten Datenquellentyp zu ändern.

Beispiel: Fünf Sybase-Server sind im globalen Katalog für Ihre Sybase-Datenquellen registriert. Wenn eine Benutzer-ID zur Authentifizierung an einen dieser Sybase-Server gesendet wird, soll der Server mit föderierten Datenbanken die Benutzer-ID stets in Großbuchstaben umsetzen. Darüber hinaus möchten Sie festlegen, wie lange der Server mit föderierten Datenbanken auf eine Antwort auf eine SQL-Anweisung von diesen Sybase-Servern wartet. Sie geben die Zeitdauer in Sekunden an. Mit der folgenden Anweisung ALTER SERVER können Sie alle fünf Serverdefinitionen gleichzeitig ändern:

```
ALTER SERVER TYPE sybase
  OPTIONS (ADD FOLD_ID 'U', ADD TIMEOUT '600')
```

Verwenden von Serveroptionen in Serverdefinitionen (DB2-Steuerzentrale)

Es gibt allgemeine Serveroptionen und Serveroptionen, die für bestimmte Datenquellentypen bestimmt sind. Serverdefinitionen können über die DB2-Steuerzentrale oder über die Befehlszeilenaufforderung geändert werden, um eine Serveroption hinzuzufügen oder zu ändern.

Vorbereitung

Die Berechtigungs-ID, die die Anweisung ALTER SERVER absetzt, muss über die Berechtigung SYSADM oder DBADM in der föderierten Datenbank verfügen.

Einschränkungen

Siehe „Einschränkungen beim Ändern von Serverdefinitionen“ auf Seite 26.

Informationen zu dieser Task

Serveroptionen werden auf Werte gesetzt, die über aufeinanderfolgende Verbindungen zur Datenquelle hinweg bestehen bleiben. Diese Werte werden im Katalog des föderierten Systems gespeichert.

Vorgehensweise

Gehen Sie wie folgt vor, um diese Task über die DB2-Steuerzentrale auszuführen:

1. Erweitern Sie den Ordner 'Objekte föderierter Datenbanken'. Die Objekte der Serverdefinition werden im Inhaltsteilfenster des Fensters der DB2-Steuerzentrale angezeigt.
2. Klicken Sie mit der rechten Maustaste die zu ändernde Serverdefinition an, und klicken Sie in der Liste der Aktionen anschließend **Ändern** an. Daraufhin wird das Notizbuch 'Serverdefinition ändern' geöffnet.
3. Wählen Sie auf der Seite 'Einstellungen' die Serveroption aus, die hinzugefügt oder entfernt werden soll.
4. Geben Sie für Optionen, die hinzugefügt oder geändert werden, den Wert einer Option an.
5. Klicken Sie **OK** an, um die Serverdefinition zu ändern und das Notizbuch 'Serverdefinition ändern' zu schließen.

Einige Serveroptionen sind erforderlich und können nicht gelöscht werden. Andere Serveroptionen können nicht hinzugefügt werden, wenn bestimmte Serveroptionen bereits definiert sind.

Temporäres Ändern von Serveroptionen für relationale Datenquellen

Mit der Anweisung SET SERVER OPTION wird der Wert der Serveroption in der Serverdefinition für die Dauer einer Verbindung zur föderierten Datenbank außer Kraft gesetzt. Der Überschreibungswert wird im globalen Katalog nicht gespeichert.

Informationen zu dieser Task

Bei statischem SQL hat die Verwendung der Anweisung SET SERVER OPTION nur Auswirkungen auf die Ausführung der statischen SQL-Anweisung. Die Verwendung der Option SET SERVER OPTION hat keine Auswirkungen auf die von der Optimierungsfunktion generierten Pläne.

Vorgehensweise

Verwenden Sie die Anweisung SET SERVER OPTION, um den Wert einer Serveroption für eine relationale Datenquelle temporär zu ändern.

Beispiel:

```
SET SERVER OPTION PLAN_HINTS TO 'Y' FOR SERVER SYB_SERVER
```

Die Hierarchie bei Einstellungen von Serveroptionen

Wenn Sie dieselbe Serveroption mit einem Wert für einen Datenquellentyp und mit einem anderen Wert für einen bestimmten Datenquellenserver definiert haben, herrscht unter den verschiedenen Einstellungen eine Hierarchie.

Beispiel: Die Serveroption PLAN_HINTS ist für den Datenquellentyp SYBASE auf 'Y' gesetzt. In der Serverdefinition für einen bestimmten Sybase-Datenquellenserver

PURNELL ist die Serveroption PLAN_HINTS jedoch auf 'N' gesetzt. Die Einstellung für den Datenquellenserver setzt hier die Einstellung für den Datenquellentyp außer Kraft. Durch diese Konfiguration wird PLAN_HINTS bei allen Sybase-Datenquellenservern mit Ausnahme von PURNELL aktiviert.

Verwenden von Serveroptionen in Serverdefinitionen (DB2-Befehlszeile)

Es gibt allgemeine Serveroptionen und Serveroptionen, die für bestimmte Datenquellentypen bestimmt sind. Serverdefinitionen können über die DB2-Steuerzentrale oder über die Befehlszeilenaufforderung geändert werden, um eine Serveroption hinzuzufügen oder zu ändern.

Vorbereitung

Die Berechtigungs-ID, die die Anweisung ALTER SERVER absetzt, muss über die Berechtigung SYSADM oder DBADM in der föderierten Datenbank verfügen.

Einschränkungen

Siehe „Einschränkungen beim Ändern von Serverdefinitionen“ auf Seite 26.

Informationen zu dieser Task

Serveroptionen werden auf Werte gesetzt, die über aufeinanderfolgende Verbindungen zur Datenquelle hinweg bestehen bleiben. Diese Werte werden im Katalog des föderierten Systems gespeichert.

Vorgehensweise

Wenn Sie diese Task von der Befehlszeilenaufforderung ausführen möchten, setzen Sie die Anweisung ALTER SERVER ab. Beispiel:

- Sie haben für einen Informix-Server eine Serverdefinition unter Verwendung des Servernamens INFMX01 erstellt. Nun möchten Sie die Option DB2_MAXIMAL_PUSHDOWN in 'Y' ändern. Die Anweisung zum Ändern der Serverdefinition lautet folgendermaßen:

```
ALTER SERVER INFMX01 OPTIONS (SET DB2_MAXIMAL_PUSHDOWN 'Y')
```

- Sie haben für einen Oracle-Server eine Serverdefinition unter Verwendung des Servernamens ORCL99 erstellt. Nun möchten Sie der Definition die Optionen FOLD_ID und FOLD_PW hinzufügen. Die Anweisung zum Ändern der Serverdefinition lautet folgendermaßen:

```
ALTER SERVER ORCL99 OPTIONS (ADD FOLD_ID 'U', FOLD_PW 'U')
```

- Sie möchten den Zeitlimitwert auf die Anzahl von Sekunden setzen, die der CTLIB-Wrapper auf eine Antwort vom Sybase-Server warten soll. Sie verwenden die Serveroption TIMEOUT zum Definieren dieses Werts. Die Anweisung zum Ändern der Serverdefinition lautet folgendermaßen:

```
ALTER SERVER SYBSERVER OPTIONS (ADD TIMEOUT '60')
```

Ändern einer Benutzerzuordnung (DB2-Steuerzentrale)

Eine Benutzerzuordnung ist die Zuordnung zwischen der Berechtigungs-ID auf dem Server mit föderierten Datenbanken und der Berechtigungs-ID in der Datenquelle. Benutzerzuordnungen sind erforderlich, damit verteilte Anforderungen an die Datenquelle gesendet werden können.

Vorbereitung

Wenn die Berechtigungs-ID, die die Anweisung absetzt, eine andere als die der Datenquelle zugeordnete ID ist, muss die die Anweisung absetzende Berechtigungs-ID entweder über die Berechtigung SYSADM oder DBADM für die föderierte Datenbank verfügen.

Einschränkungen

Der Server mit föderierten Datenbanken kann die Anweisung ALTER USER MAPPING in einer bestimmten Arbeitseinheit (UOW) nicht verarbeiten, wenn die UOW bereits eine der folgenden Anweisungen enthält:

- Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist, die Teil der Zuordnung ist
- Ein geöffneter Cursor bei einem Kurznamen für eine Tabellen oder Sicht in der Datenquelle, die Teil der Zuordnung ist
- Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde, die Teil der Zuordnung ist

Informationen zu dieser Task

Die Anweisung ALTER USER MAPPING wird zum Ändern der Berechtigungs-ID oder des Kennworts verwendet, die bzw. das in der Datenquelle für eine bestimmte Berechtigungs-ID auf dem Server mit föderierten Datenbanken verwendet wird.

Benutzerzuordnungen können über die DB2-Steuerzentrale oder über die Befehlszeile geändert werden.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Benutzerzuordnung über die DB2-Steuerzentrale zu ändern:

1. Erweitern Sie den Ordner 'Objekte föderierter Datenbanken'. Die Objekte der Benutzerzuordnung werden im Inhaltsteilfenster des Fensters der DB2-Steuerzentrale angezeigt.
2. Klicken Sie mit der rechten Maustaste die zu ändernde Benutzerzuordnung an, und klicken Sie in der Liste der Aktionen anschließend **Ändern** an. Daraufhin wird das Fenster 'Benutzerzuordnung ändern' geöffnet.
3. Ändern Sie den Wert der Option.
4. Klicken Sie **OK** an, um die Benutzerzuordnung zu ändern und das Fenster 'Benutzerzuordnung ändern' zu schließen.

Ändern einer Benutzerzuordnung (DB2-Befehlszeile)

Eine Benutzerzuordnung ist die Zuordnung zwischen der Berechtigungs-ID auf dem Server mit föderierten Datenbanken und der Berechtigungs-ID in der Datenquelle. Benutzerzuordnungen sind erforderlich, damit verteilte Anforderungen an die Datenquelle gesendet werden können.

Vorbereitung

Wenn die Berechtigungs-ID, die die Anweisung absetzt, eine andere als die der Datenquelle zugeordnete ID ist, muss die die Anweisung absetzende Berechtigungs-ID entweder über die Berechtigung SYSADM oder DBADM für die föderierte Datenbank verfügen.

Einschränkungen

Der Server mit föderierten Datenbanken kann die Anweisung ALTER USER MAPPING in einer bestimmten Arbeitseinheit (UOW) nicht verarbeiten, wenn die UOW bereits eine der folgenden Anweisungen enthält:

- Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist, die Teil der Zuordnung ist
- Ein geöffneter Cursor bei einem Kurznamen für eine Tabellen oder Sicht in der Datenquelle, die Teil der Zuordnung ist
- Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde, die Teil der Zuordnung ist

Informationen zu dieser Task

Die Anweisung ALTER USER MAPPING wird zum Ändern der Berechtigungs-ID oder des Kennworts verwendet, die bzw. das in der Datenquelle für eine bestimmte Berechtigungs-ID auf dem Server mit föderierten Datenbanken verwendet wird.

Benutzerzuordnungen können über die DB2-Steuerzentrale oder über die Befehlszeile geändert werden.

Vorgehensweise

Um eine Benutzerzuordnung von der DB2-Befehlszeile zu ändern, setzen Sie die Anweisung ALTER USER MAPPING ab:

Beispiel: Jenny verwendet den Server mit föderierten Datenbanken, um eine Verbindung zu einem Sybase-Server mit dem Namen SYBSERVER herzustellen. Sie greift mit der Berechtigungs-ID *jennifer* auf den Server mit föderierten Datenbanken zu. Die Berechtigungs-ID *jennifer* wird der Berechtigungs-ID *jenn* auf dem Sybase-Server zugeordnet. Die Berechtigungs-ID für Jenny auf dem Sybase-Server wird in *jen123* geändert. Die Anweisung ALTER USER MAPPING zum Zuordnen von *jennifer* zu *jen123* lautet folgendermaßen:

```
ALTER USER MAPPING FOR jennifer SERVER SYBSERVER
  OPTIONS (SET REMOTE_AUTHID 'jen123')
```

Tomas verwendet den Server mit föderierten Datenbanken, um eine Verbindung zu einem Oracle-Server mit dem Namen ORASERVER herzustellen. Er greift mit der Berechtigungs-ID *tomas* auf den Server mit föderierten Datenbanken zu. Die Berechtigungs-ID *tomas* wird der Berechtigungs-ID *tom* auf dem Oracle-Server zugeordnet. Das Kennwort für Tomas auf dem Oracle-Server wird geändert. Sein neues Kennwort lautet *day2night*. Die Anweisung ALTER USER MAPPING zum Zuordnen von *tomas* zum neuen Kennwort lautet folgendermaßen:

```
ALTER USER MAPPING FOR tomas SERVER ORASERVER
  OPTIONS (SET REMOTE_PASSWORD 'day2night')
```

Bei der Angabe der Benutzeroptionen REMOTE_AUTHID und REMOTE_PASSWORD ist die Groß-/Kleinschreibung zu beachten, es sei denn, Sie setzen in der Anweisung CREATE SERVER die Serveroptionen FOLD_ID und FOLD_PW auf 'U' oder 'L'.

Ändern eines Kurznamens (DB2-Steuerzentrale)

Kurznamen sind Kennungen, die für den Verweis auf ein Objekt verwendet werden, auf das Sie in einer Datenquelle zugreifen möchten. Durch das Ändern der Kurznamen können Sie die im globalen Katalog gespeicherten Spaltennamen der Datenquelle ändern und Spaltenoptionen angeben. Kurznamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Das Ändern eines Kurznamens kann für folgende Zwecke erforderlich sein:

- Ändern der lokalen Spaltennamen für die Spalten des Datenquellenobjekts
- Ändern der lokalen Datentypen für die Spalten des Datenquellenobjekts
- Hinzufügen, Definieren oder Löschen von Kurznamen- und Spaltenoptionen
- Hinzufügen oder Löschen eines Primärschlüssels
- Hinzufügen oder Löschen einer oder mehrerer eindeutiger oder referenzieller Integritätsbedingungen oder von Prüfungen auf Integritätsbedingungen
- Ändern eines oder mehrerer Attribute für eine referenzielle oder funktionale Abhängigkeitsintegritätsbedingung oder eine Prüfung auf Abhängigkeitsintegritätsbedingungen
- Vermeiden des Zwischenspeicherns des Kurznamens auf dem Server mit föderierten Datenbanken
- Aktivieren des Zwischenspeicherns des Kurznamens auf dem Server mit föderierten Datenbanken. Wenn Cachetabellen oder MQTs (Materialized Query Tables) dem zwischengespeicherten Kurznamen zugeordnet sind, müssen Sie diese Tabellen löschen, bevor Sie die Cachingoption ändern.

Vorgehensweise

Gehen Sie wie folgt vor, um einen Kurznamen über die DB2-Steuerzentrale zu ändern:

1. Wählen Sie den Ordner **Kurznamen** aus.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Kurznamen an, und klicken Sie anschließend **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
3. Ändern Sie auf der Seite 'Kurznamen' die anwendbaren Optionen.
4. Legen Sie auf der Seite 'Schlüssel' die referenziellen Integritätsbedingungen für den Kurznamen fest. Sie können eine Integritätsbedingung über Primärschlüssel, über eindeutige Schlüssel oder über Fremdschlüssel definieren.
5. Legen Sie auf der Seite 'Prüfung auf Integritätsbedingung' die Prüfungen auf Integritätsbedingung oder die Integritätsbedingungen der funktionalen Abhängigkeiten für den Kurznamen fest.
6. Geben Sie auf der Seite 'Einstellungen' die Kurznamenoptionen für den Kurznamen an.
7. Klicken Sie **OK** an, um den Kurznamen zu ändern und das Notizbuch zu schließen.

Einige Kurznamenoptionen sind erforderlich und können nicht gelöscht werden. Andere Kurznamenoptionen können nicht hinzugefügt werden, wenn bereits bestimmte Kurznamenoptionen definiert wurden.

Wenn sich die Struktur oder der Inhalt des Datenquellenobjekts grundlegend ändert, müssen Sie die Kurznamenstatistik ändern. Unter grundlegenden Änderungen ist beispielsweise das Hinzufügen oder Entfernen mehrerer Zeilen zu verstehen.

Einschränkungen beim Ändern von Kurznamen

Beim Ändern von Kurznamen müssen Sie einige Einschränkungen berücksichtigen.

Spaltenoptionen

Wenn eine der folgenden Optionen für eine Spalte definiert ist, können Sie dieser Spalte keine anderen Optionen hinzufügen:

- SOAPACTIONCOLUMN
- URLCOLUMN
- PRIMARY_KEY
- FOREIGN_KEY

Für BioRS gilt:

- Wenn Sie den Elementnamen einer Spalte unter Verwendung der Option `ELEMENT_NAME` ändern, wird der neue Name nicht auf seine Richtigkeit überprüft. Eine falsche Option kann zu Fehlern führen, wenn in einer Abfrage auf die Spalte verwiesen wird.
- Wenn Sie die Spaltenoption `IS_INDEXED` ändern, werden die Änderungen auf dem BioRS-Server nicht geprüft. Eine falsche Option kann zu Fehlern führen, wenn in einer Abfrage auf die Spalte verwiesen wird.

Datentypen

- Wenn Sie den Datentyp einer Spalte ändern, muss der neue Datentyp mit dem Datentyp der entsprechenden Spalte oder des entsprechenden Elements der Datenquelle kompatibel sein. Das Ändern des lokalen Datentyps in einen Datentyp, der inkompatibel mit dem fernen Datentyp ist, kann zu unvorhersehbaren Fehlern führen.
- Der *lokale_datentyp* darf nicht `LONG VARCHAR`, `LONG VARGRAPHIC`, `XML` oder ein benutzerdefinierter Datentyp sein.
- Der *datenquellendatentyp* darf kein benutzerdefinierter Datentyp sein.

- Bei einigen nicht relationalen Datenquellen können die vorhandenen lokalen Typen nicht überschrieben und es können keine neuen lokalen Typen erstellt werden. Die Dokumentation der jeweiligen Datenquellen-Wrapper enthält weitere Informationen zu dieser Einschränkung.
- Wenn die lokale Spezifikation des Datentyps einer Spalte geändert wird, inaktiviert der föderierte Datenbankmanager sämtliche Statistik (z. B. HIGH2KEY und LOW2KEY), die für diese Spalte erfasst wird.
- Der lokale Typ wird für das bestimmte Datenquellenobjekt definiert, wenn mit diesem Kurznamen auf das Objekt zugegriffen wird. Dasselbe Datenquellenobjekt kann einen anderen Kurznamen aufweisen, der die Standarddatentypzuordnung verwendet.

Indizes

Die Anweisung ALTER NICKNAME kann nicht zum Registrieren eines neuen Datenquellenindex in der föderierten Datenbank verwendet werden. Verwenden Sie die Anweisung CREATE INDEX mit der Klausel SPECIFICATION ONLY, um eine Indexspezifikation zu erstellen.

Parameter LOCAL NAME und LOCAL TYPE

- Die Anweisung ALTER NICKNAME kann nicht zum Ändern des lokalen Namens oder der lokalen Datentypen für die Spalten im Kurznamen verwendet werden, wenn:
 - Der Kurzname in einer Sicht, SQL-Methode oder SQL-Funktion verwendet wird
 - Sie eine informative Integrationsbedingung für den Kurznamen definieren
- Die Klausel federated_column_options muss zuletzt angegeben werden, wenn Sie darüber hinaus auch den Parameter LOCAL NAME und/oder den Parameter LOCAL in der Anweisung ALTER NICKNAME angeben müssen.

Kurznamen

Die Anweisung ALTER NICKNAME kann nicht zum Ändern des Namens der BioRS-Datenbank verwendet werden, auf die durch den BioRS-Kurznamen verwiesen oder die in einem solchen verwendet wird. Wenn sich der Name einer BioRS-Datenbank ändert, müssen Sie den Kurznamen löschen und ihn anschließend erneut erstellen.

Sie können die Anweisung ALTER NICKNAME nicht dazu verwenden, das Caching für einen Kurznamen mit Cachetabellen oder MQTs (Materialized Query Tables) zu unterbinden. Sie müssen die Cachetabellen und die MQTs löschen, wenn Sie das Caching für den Kurznamen nicht zulassen möchten.

Arbeitseinheiten (Units of Work, UOW)

Unter den folgenden Bedingungen kann der Server mit föderierten Datenbanken die Anweisung ALTER NICKNAME bei einer angegebenen Arbeitseinheit nicht verarbeiten:

- Wenn in derselben Arbeitseinheit für den Kurznamen, auf den in der Anweisung ALTER NICKNAME verwiesen wird, ein Cursor geöffnet ist.
- Wenn in derselben Arbeitseinheit eine INSERT-, DELETE- oder UPDATE-Anweisung für den Kurznamen abgesetzt wird, auf den in der Anweisung ALTER NICKNAME verwiesen wird.

Ändern der Spaltennamen eines Kurznamens (DB2-Steuerzentrale)

Sie können einen Kurznamen ändern, um die Spaltennamen zu ändern. Spaltennamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Beim Erstellen eines Kurznamens werden die Spaltennamen, die dem Datenquellenobjekt zugeordnet sind, in der föderierten Datenbank gespeichert. Bei einigen Datenquellen werden die Spaltennamen vom Wrapper angegeben. Bei anderen Datenquellen müssen Sie die Spaltennamen beim Erstellen des Kurznamens angeben.

Vorgehensweise

Gehen Sie wie folgt vor, um Spaltennamen eines Kurznamens über die DB2-Steuerzentrale zu ändern:

1. Wählen Sie den Ordner **Kurznamen** aus.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Kurznamen an, und klicken Sie anschließend **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
3. Wählen Sie im Fenster 'Kurznamen' die zu ändernde Spalte aus, und klicken Sie anschließend **Ändern** an. Das Fenster 'Spalte ändern' wird geöffnet.
4. Geben Sie den Spaltennamen ein.
5. Klicken Sie **OK** an, um den Spaltennamen zu ändern und das Fenster zu schließen.
6. Klicken Sie **OK** an, um den Kurznamen zu ändern und das Notizbuch zu schließen.

Ändern der Spaltennamen eines Kurznamens (DB2-Befehlszeile)

Sie können einen Kurznamen ändern, um die Spaltennamen zu ändern. Spaltennamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Beim Erstellen eines Kurznamens werden die Spaltennamen, die dem Datenquellenobjekt zugeordnet sind, in der föderierten Datenbank gespeichert. Bei einigen Datenquellen werden die Spaltennamen vom Wrapper angegeben. Bei anderen Datenquellen müssen Sie die Spaltennamen beim Erstellen des Kurznamens angeben.

Vorgehensweise

Um Spaltennamen eines Kurznamens von der DB2-Befehlszeile zu ändern, setzen Sie die Anweisung ALTER NICKNAME ab.

```
ALTER NICKNAME kurzname
ALTER COLUMN aktueller_name
LOCAL NAME neuer_name
```

Ändern von Kurznamenoptionen (DB2-Steuerzentrale)

Kurznamenoptionen sind Parameter, die Sie für den Kurznamen angeben, wenn Sie die Anweisungen CREATE NICKNAME und ALTER NICKNAME absetzen. Mit der Anweisung ALTER NICKNAME können Sie Kurznamenoptionen hinzufügen, definieren oder löschen. Spaltennamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Beispiel: Der Kurzname DRUGDATA1 wird für die tabellenstrukturierte Datei drugdata1.txt erstellt. Der ursprünglich in der Anweisung CREATE NICKNAME vollständig qualifizierte Pfad lautet /user/pat/drugdata1.txt.

Um die Kurznamenoption FILE_PATH zu ändern, setzen Sie die folgende Anweisung ab:

Vorgehensweise

Gehen Sie wie folgt vor, um Spaltennamen über die DB2-Steuerzentrale zu ändern:

1. Wählen Sie den Ordner **Kurznamen** aus.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Kurznamen an, und klicken Sie anschließend **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
3. Wählen Sie auf der Seite 'Einstellungen' die einzelnen Markierungsfelder neben den Optionen aus, die Sie hinzufügen oder entfernen möchten. Erforderliche Optionen können nicht entfernt werden.
4. Klicken Sie zum Angeben oder Ändern des Werts einer Option das Feld **Wert** für die Option an. Je nach Option können Sie entweder einen Wert aus der Liste auswählen, durch Anklicken mehrere Werte auswählen oder einen neuen Wert eingeben.
5. Klicken Sie **OK** an, um den Kurznamen zu ändern und das Notizbuch zu schließen.

Ändern von Kurznamenoptionen (DB2-Befehlszeile)

Kurznamenoptionen sind Parameter, die Sie für den Kurznamen angeben, wenn Sie die Anweisungen CREATE NICKNAME und ALTER NICKNAME absetzen. Mit der Anweisung ALTER NICKNAME können Sie Kurznamenoptionen hinzufügen, definieren oder löschen. Kurznamenoptionen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Vorgehensweise

Verwenden Sie zum Ändern einer Kurznamenoption über die Befehlszeilenaufforderung die Anweisung ALTER NICKNAME.

```
ALTER NICKNAME kurzname
    OPTIONS (SET optionsname 'zeichenfolgewert_der_option')
```

Beispiel: Der Kurzname DRUGDATA1 wird für die tabellenstrukturierte Datei drugdata1.txt erstellt. Der ursprünglich in der Anweisung CREATE NICKNAME vollständig qualifizierte Pfad lautet /user/pat/drugdata1.txt. Um die Kurznamenoption FILE_PATH zu ändern, setzen Sie die folgende Anweisung ab:

```
ALTER NICKNAME DRUGDATA1 OPTIONS (SET FILE_PATH '/usr/kelly/data/drugdata1.txt')
```

Ändern der Spaltenoptionen eines Kurznamens (DB2-Steuerzentrale)

Mit der Anweisung ALTER NICKNAME können Sie Spaltenoptionen von Kurznamen hinzufügen, definieren oder löschen. Spaltennamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

In der Anweisung CREATE NICKNAME oder ALTER NICKNAME können Sie Spalteninformationen angeben, indem Sie Parameter verwenden, die als Kurznamenspaltenoptionen bezeichnet werden.

Vorgehensweise

Gehen Sie wie folgt vor, um Spaltenoptionen eines Kurznamens über die DB2-Steuerzentrale zu ändern:

1. Wählen Sie den Ordner **Kurznamen** aus.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Kurznamen an, und klicken Sie anschließend **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
3. Wählen Sie im Fenster 'Kurznamen' die zu ändernde Spalte aus, und klicken Sie anschließend **Ändern** an. Das Fenster 'Spalte ändern' wird geöffnet.
4. Wählen Sie die Spaltenoption aus, die hinzugefügt oder entfernt werden soll.
5. Geben Sie für Optionen, die hinzugefügt oder geändert werden, den Wert einer Option an.

6. Klicken Sie **OK** an, um die Spaltenoption zu ändern und das Fenster zu schließen.
7. Klicken Sie **OK** an, um den Kurznamen zu ändern und das Notizbuch zu schließen.

Ändern der Spaltenoptionen eines Kurznamens (DB2-Befehlszeile)

Mit der Anweisung ALTER NICKNAME können Sie Spaltenoptionen von Kurznamen hinzufügen, definieren oder löschen. Spaltennamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

In der Anweisung CREATE NICKNAME oder ALTER NICKNAME können Sie Spalteninformationen angeben, indem Sie Parameter verwenden, die als Kurznamenspaltenoptionen bezeichnet werden. Sie können diese Werte in Großbuchstaben oder in Kleinbuchstaben angeben.

Vorgehensweise

Verwenden Sie zum Ändern einer Kurznamenspaltenoption über die Befehlszeile die Anweisung ALTER NICKNAME.

Beispiel 1: Angeben der Spaltenoption NUMERIC_STRING bei relationalen Datenquellen

Die Spaltenoption NUMERIC_STRING gilt für Zeichentypspalten (CHAR und VARCHAR). Angenommen, eine Datenquelle weist eine andere Sortierfolge als die föderierte Datenbank auf. Normalerweise sortiert der Server mit föderierten Datenbanken in diesem Fall in der Datenquelle keine Spalten, die Zeichendaten enthalten. Stattdessen gibt er die Daten an die föderierte Datenbank zurück und führt die Sortierung lokal aus. Nehmen Sie jedoch an, die Spalte weist einen Zeichendatentyp auf und enthält ausschließlich numerische Zeichen ('0', '1', ..., '9'). Sie können dies durch die Zuordnung des Wertes 'Y' zu der Spaltenoption NUMERIC_STRING angeben. Auf diese Weise kann das Abfrageoptimierungsprogramm von DB2 UDB die Sortierung in der Datenquelle ausführen. Wird die Sortierung fern durchgeführt, kann der Systemaufwand für das Sortieren der Daten auf dem Server mit föderierten Datenbanken vermieden werden.

Der Kurzname `ORA_INDSALES` gilt für eine Oracle-Tabelle mit dem Namen `INDONESIA_SALES`. Die Tabelle enthält die Spalte `POSTAL_CODE` mit dem Datentyp `VARCHAR`. Ursprünglich enthielt die Spalte nur numerische Zeichen, und die Spaltenoption `NUMERIC_STRING` war auf `'Y'` gesetzt. Nun enthält die Spalte jedoch eine Mischung von numerischen und nicht numerischen Zeichen. Verwenden Sie die folgende Anweisung, um die Spaltenoption `NUMERIC_STRING` auf `'N'` zu setzen:

```
ALTER NICKNAME ORA_INDSALES ALTER COLUMN POSTAL_CODE
  OPTIONS (SET NUMERIC_STRING 'N')
```

Beispiel 2: Angeben der Spaltenoption `VARCHAR_NO_TRAILING_BLANKS` bei relationalen Datenquellen

Die Spaltenoption `VARCHAR_NO_TRAILING_BLANKS` kann verwendet werden, um bestimmte Spalten zu identifizieren, die keine abschließenden Leerzeichen enthalten. Der SQL-Compiler berücksichtigt diese Einstellung, wenn er alle Operationen (z. B. Vergleichsoperationen) überprüft, die bei Spalten ausgeführt werden.

Der Kurzname `ORA_INDSALES` gilt für eine Oracle-Tabelle mit dem Namen `INDONESIA_SALES`. Die Tabelle enthält die Spalte `NAME` mit dem Datentyp `VARCHAR`. Die Spalte `NAME` enthält keine abschließenden Leerzeichen. Gehen Sie wie folgt vor, um die Option `VARCHAR_NO_TRAILING_BLANKS` dem Kurznamen hinzuzufügen:

```
ALTER NICKNAME ORA_INDSALES ALTER COLUMN NAME
  OPTIONS (ADD VARCHAR_NO_TRAILING_BLANKS 'Y')
```

Beispiel 3: Angeben der Spaltenoption `XPATH` bei nicht relationalen Datenquellen

Der Kurzname `EMPLOYEE` gilt für eine XML-Datenquelle. Für die Spalte `fname` wurde die Option `XPATH` angegeben. Verwenden Sie die folgende Anweisung, um die Spaltenoption `XPATH` auf einen anderen Pfad zu setzen:

```
ALTER NICKNAME EMPLOYEE ALTER COLUMN fname
  OPTIONS (SET XPATH './@first')
```

Ändern eines Kurznamens (DB2-Befehlszeile)

Kurznamen sind Kennungen, die für den Verweis auf ein Objekt verwendet werden, auf das Sie in einer Datenquelle zugreifen möchten. Durch das Ändern der Kurznamen können Sie die im globalen Katalog gespeicherten Spaltennamen der Datenquelle ändern und Spaltenoptionen angeben. Kurznamen können über die DB2-Steuerzentrale oder über die DB2-Befehlszeile geändert werden.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung `SYSADM` oder `DBADM`
- Zugriffsrecht `ALTER` für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht `CONTROL` für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht `ALTERIN` für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte `DEFINER` der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Das Ändern eines Kurznamens kann für folgende Zwecke erforderlich sein:

- Ändern der lokalen Spaltennamen für die Spalten des Datenquellenobjekts
- Ändern der lokalen Datentypen für die Spalten des Datenquellenobjekts
- Hinzufügen, Definieren oder Löschen von Kurznamen- und Spaltenoptionen
- Hinzufügen oder Löschen eines Primärschlüssels
- Hinzufügen oder Löschen einer oder mehrerer eindeutiger oder referenzieller Integritätsbedingungen oder von Prüfungen auf Integritätsbedingungen
- Ändern eines oder mehrerer Attribute für eine referenzielle oder funktionale Abhängigkeitsintegritätsbedingung oder eine Prüfung auf Abhängigkeitsintegritätsbedingungen

Vorgehensweise

Um einen Kurznamen von der DB2-Befehlszeile zu ändern, setzen Sie die Anweisung ALTER NICKNAME mit den entsprechenden Parametern ab.

Wenn sich die Struktur oder der Inhalt des Datenquellenobjekts grundlegend ändert, müssen Sie die Kurznamenstatistik ändern. Unter grundlegenden Änderungen ist beispielsweise das Hinzufügen oder Entfernen mehrerer Zeilen zu verstehen.

Löschen eines Wrappers

Das Löschen eines Wrappers kann aus verschiedenen Gründen erforderlich sein.

Vorbereitung

Zum Absetzen der Anweisung DROP WRAPPER müssen Sie über die Berechtigung SYSADM oder DBADM verfügen.

Informationen zu dieser Task

Manchmal können für den Zugriff auf eine Datenquelle mehrere Wrapper verwendet werden. Ihre Auswahl eines geeigneten Wrappers hängt möglicherweise von der von Ihnen verwendeten Version der Client-Software der Datenquelle ab. Ferner kann die Auswahl auch von dem Betriebssystem abhängig sein, das auf dem Server mit föderierten Datenbanken verwendet wird. Nehmen Sie als Beispiel an, Sie möchten auf zwei Oracle-Tabellen und eine Oracle-Sicht zugreifen. Sie verwenden Oracle Version 10 und das Betriebssystem auf Ihrem Server mit föderierten Datenbanken ist Windows. Zunächst haben Sie den Wrapper SQLNET erstellt. Da IBM InfoSphere Federation Server den Wrapper SQLNET nicht unterstützt, können Sie den Wrapper SQLNET löschen und den Wrapper NET8 erstellen.

Ein weiterer Grund für das Löschen eines Wrappers kann der Umstand sein, dass Sie keinen Zugriff auf die Datenquelle mehr benötigen, der der Wrapper zugeordnet ist. Nehmen Sie beispielsweise an, in Ihrem Unternehmen besteht die Notwendigkeit, auf Clientinformationen sowohl in Informix- als auch in Microsoft SQL Server-Datenbanken zuzugreifen. Sie erstellen einen Wrapper für die Informix-

Datenquelle und einen Wrapper für die Microsoft SQL Server-Datenquelle. Zu einem späteren Zeitpunkt wird in dem Unternehmen beschlossen, alle Informationen von Microsoft SQL Server auf Informix zu migrieren. Sie benötigen folglich den Microsoft SQL Server-Wrapper nicht mehr und können ihn löschen.

Wichtiger Hinweis: Das Löschen eines Wrappers ist mit weitreichenden Folgen verbunden. Andere Objekte, die Sie beim Server mit föderierten Datenbanken registriert haben, sind ebenfalls betroffen:

- Alle Serverdefinitionen, die vom gelöschten Wrapper abhängig sind, werden ebenfalls gelöscht.
- Alle Objekte, die von den gelöschten Serverdefinitionen abhängig sind, werden ebenfalls gelöscht.
- Alle Kurznamen, die von den gelöschten Serverdefinitionen abhängig sind, werden ebenfalls gelöscht. Das Löschen der von den Serverdefinitionen abhängigen Kurznamen betrifft die Objekte, die von diesen Kurznamen abhängig sind:
 - Alle Indexspezifikationen, die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
 - Alle Sichten, die von den gelöschten Kurznamen abhängig sind, werden als funktionsunfähig markiert.
 - Alle MQTs (Materialized Query Tables), die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
- Alle Pakete und zwischengespeicherten dynamischen SQL-Anweisungen, die von den gelöschten Kurznamen abhängig sind, werden als ungültig markiert und bleiben ungültig, bis die abhängigen Objekte erneut erstellt werden.

Vorgehensweise

Verwenden Sie zum Löschen eines Wrappers die Anweisung DROP.

Beispiel: Setzen Sie die folgende Anweisung ab, um den Microsoft SQL Server-Wrapper *MSSQLODBC3* zu löschen:

```
DROP WRAPPER MSSQLODBC3
```

Löschen einer Serverdefinition

Durch das Löschen einer Serverdefinition wird die Definition aus dem globalen Katalog gelöscht. Das Datenquellenobjekt, auf das die Serverdefinition verweist, ist nicht betroffen. Serverdefinitionen können über die DB2-Steuerzentrale oder unter Verwendung der Anweisung DROP im DB2-Befehlszeilenprozessor gelöscht werden.

Vorbereitung

Um eine Serverdefinition löschen zu können, benötigen Sie die Berechtigung SYSADM oder DBADM.

Einschränkungen

Der Server mit föderierten Datenbanken kann die Anweisung DROP SERVER innerhalb einer gegebenen Arbeitseinheit (UOW) nicht verarbeiten, wenn eine der folgenden Bedingungen gegeben ist:

- Die Anweisung verweist auf eine einzelne Datenquelle, und die UOW enthält bereits eine der folgenden Anweisungen:

- Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in der Datenquelle verweist
- Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in der Datenquelle
- Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in der Datenquelle abgesetzt wurde
- Die Anweisung verweist auf eine Kategorie von Datenquellen (z. B. alle Datenquellen eines bestimmten Typs und einer bestimmten Version), und die UOW enthält bereits eine der folgenden Anweisungen:
 - Eine Anweisung SELECT, die auf einen Kurznamen für eine Tabelle oder Sicht in einer der Datenquellen verweist
 - Ein geöffneter Cursor bei einem Kurznamen für eine Tabelle oder Sicht in einer der Datenquellen
 - Eine INSERT-, DELETE- oder UPDATE-Anweisung, die für einen Kurznamen einer Tabelle oder Sicht in einer der Datenquellen abgesetzt wurde

Informationen zu dieser Task

Wenn Sie den Zugriff auf einen Datenquellenserver nicht mehr benötigen, löschen Sie die Serverdefinition aus der föderierten Datenbank. Wenn Sie eine Serverdefinition löschen, sind auch andere Objekte betroffen, die Sie beim Server mit föderierten Datenbanken registriert haben:

- Alle benutzerdefinierten Funktionszuordnungen, benutzerdefinierten Datentypzuordnungen und Benutzerzuordnungen, die von der gelöschten Serverdefinition abhängig sind, werden ebenfalls gelöscht.
- Alle Kurznamen, die von der gelöschten Serverdefinition abhängig sind, werden ebenfalls gelöscht. Das Löschen der von der Serverdefinition abhängigen Kurznamen betrifft die Objekte, die von diesen Kurznamen abhängig sind:
 - Alle Indexspezifikationen, die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
 - Alle Sichten, die von den gelöschten Kurznamen abhängig sind, werden als funktionsunfähig markiert.
 - Alle MQTs (Materialized Query Tables), die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
- Alle Pakete und zwischengespeicherten dynamischen SQL-Anweisungen, die von den gelöschten Kurznamen abhängig sind, werden als ungültig markiert und bleiben ungültig, bis die abhängigen Objekte erneut erstellt werden.

Vorgehensweise

Setzen Sie zum Löschen einer Serverdefinition die Anweisung DROP ab:

```
DROP SERVER servername
```

Dabei gibt *servername* die zu löschende Serverdefinition an.

Beispiel: Ein Informix-Server verwendet den Servernamen INFMX01. Mit der folgenden DROP-Anweisung wird die Serverdefinition gelöscht:

```
DROP SERVER INFMX01
```

Löschen einer Benutzerzuordnung

Wenn ein Benutzer den Zugriff auf eine ferne Datenquelle nicht mehr benötigt, löschen Sie die Benutzerzuordnung zwischen dem Server mit föderierten Datenbanken und dem fernen Datenquellenserver. Wenn der Benutzer mehreren Datenquellenservern zugeordnet ist, müssen Sie jede Zuordnung einzeln löschen.

Vorbereitung

Zum Absetzen der Anweisung `DROP USER MAPPING` muss die Berechtigungs-ID der Anweisung `DROP` über die Berechtigung `SYSADM` oder `DBADM` verfügen, wenn sich diese Berechtigungs-ID von der in der Benutzerzuordnung angegebenen Benutzer-ID der föderierten Datenbank unterscheidet. Wenn die Berechtigungs-ID und die Benutzer-ID in der Benutzerzuordnung jedoch übereinstimmen, sind keine Berechtigungen oder Zugriffsrechte erforderlich.

Vorgehensweise

Setzen Sie zum Löschen einer Benutzerzuordnung die Anweisung `DROP` ab:

```
DROP USER MAPPING FOR benutzer_ID SERVER lokaler_servername
```

Dabei gilt Folgendes:

- *benutzer_ID* ist die Berechtigungs-ID für den Benutzer auf dem Server mit föderierten Datenbanken
- *lokaler_servername* ist der lokale Name, der zum Identifizieren des fernen Datenquellenservers in der Serverdefinition verwendet wird.

Löschen eines Kurznamens

Durch das Löschen eines Kurznamens wird der Kurzname aus dem globalen Katalog auf dem Server mit föderierten Datenbanken gelöscht. Das Datenquellenobjekt, auf das der Kurzname verweist, ist nicht betroffen.

Vorbereitung

Der Kurzname muss im Katalog aufgelistet sein.

Die Berechtigungs-ID der `DROP`-Anweisung muss beim Löschen von Kurznamen über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung `SYSADM` oder `DBADM`
- Zugriffsrecht `DROPIN` auf das Schema für den Kurznamen
- Definierender Benutzer des Kurznamens, der in der Spalte `DEFINER` der Katalogsicht für den Kurznamen eingetragen ist
- Zugriffsrecht `CONTROL` auf den Kurznamen

Einschränkungen

Für Kurznamen zu relationalen Datenquellen kann der Server mit föderierten Datenbanken die Anweisung `DROP NICKNAME` innerhalb einer gegebenen Arbeitseinheit (UOW) nicht verarbeiten, wenn eine der folgenden Bedingungen gegeben ist:

- Bei einem Kurznamen, auf den in der Anweisung verwiesen wird, ist in derselben UOW ein Cursor geöffnet.

- Auf einen Kurznamen, auf den in dieser Anweisung verwiesen wird, wird bereits in einer SELECT-Anweisung in derselben UOW verwiesen.
- Für den Kurznamen, auf den in der Anweisung verwiesen wird, wird in derselben UOW eine INSERT-, DELETE- oder UPDATE-Anweisung abgesetzt.

Für Kurznamen zu nicht relationalen Datenquellen kann der Server mit föderierten Datenbanken die Anweisung DROP NICKNAME innerhalb einer gegebenen Arbeitseinheit (UOW) nicht verarbeiten, wenn eine der folgenden Bedingungen gegeben ist:

- Bei einem Kurznamen, auf den in dieser Anweisung verwiesen wird, ist in derselben UOW ein Cursor geöffnet.
- Auf einen Kurznamen, auf den in dieser Anweisung verwiesen wird, wird bereits in einer SELECT-Anweisung in derselben UOW verwiesen.

Informationen zu dieser Task

Wenn Sie einen Kurznamen löschen, sind auch andere Objekte betroffen, die Sie beim Server mit föderierten Datenbanken registriert haben:

- Das Löschen von Kurznamen betrifft die Objekte, die von diesen Kurznamen abhängig sind:
 - Alle Indexspezifikationen, die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
 - Alle Sichten, die von den gelöschten Kurznamen abhängig sind, werden als funktionsunfähig markiert.
 - Alle MQTs (Materialized Query Tables), die von den gelöschten Kurznamen abhängig sind, werden ebenfalls gelöscht.
- Alle Pakete und zwischengespeicherten dynamischen SQL-Anweisungen, die von dem gelöschten Kurznamen abhängig sind, werden als ungültig markiert und bleiben ungültig, bis die abhängigen Objekte erneut erstellt werden.

Vorgehensweise

Setzen Sie zum Löschen eines Kurznamens die Anweisung DROP ab.

```
DROP NICKNAME kurzname
```

Dabei gibt *kurzname* den zu löschenden Kurznamen an.

Kapitel 3. Datentypzuordnungen in einem föderierten System

Die Datentypen der Datenquelle müssen den entsprechenden DB2-Datentypen zugeordnet sein. Diese Zuordnung ermöglicht es dem Server mit föderierten Datenbanken, Daten von der Datenquelle abzurufen.

Die föderierte Datenbank stellt für einige Datenquellen eine Reihe von Standard-datentypzuordnungen bereit. Für andere Datenquellen müssen Sie die Datenquellenzuordnungen bereitstellen, die Sie verwenden möchten. Für nicht relationale Datenquellen können die vorhandenen Datentypzuordnungen nicht überschrieben werden, und es können keine neuen Zuordnungen erstellt werden.

Im Folgenden werden einige Beispiele für Standarddatentypzuordnungen aufgeführt:

- Der Oracle-Typ FLOAT ist standardmäßig dem DB2-Typ DOUBLE zugeordnet
- Der Oracle-Typ DATE ist standardmäßig dem DB2-Typ TIMESTAMP zugeordnet
- Der Datentyp DATE von DB2 für z/OS ist standardmäßig dem DB2-Typ DATE zugeordnet

Kurznamen, die nach der Änderung einer Zuordnung erstellt wurden, verwenden die neue Typzuordnung. Kurznamen, die vor der Änderung der Zuordnung erstellt wurden, verwenden die standardmäßige Datentypzuordnung.

Wenn Sie die Kurznamen bereits erstellt haben, können Sie die vorhandenen Kurznamen auf eine der folgenden Arten aktualisieren:

- Sie können jeden einzelnen Kurznamen ändern.
- Sie können jeden Kurznamen löschen und erneut erstellen.

Server mit föderierten DB2-Datenbanken unterstützen Zuordnungen für die folgenden Datentypen nicht:

- Der lokale Datentyp darf nicht LONG VARCHAR, LONG VARGRAPHIC oder ein benutzerdefinierter Datentyp sein.
- Der ferne Datentyp darf kein benutzerdefinierter Datentyp sein.

Sie können jedoch eine Umsetzungsfunktion verwenden, um den benutzerdefinierten Datentyp in einer Sicht in der fernen Datenquelle in einen integrierten Datentyp oder einen Systemdatentyp zu konvertieren. Anschließend können Sie einen Kurznamen für die Sicht erstellen. Die auf diese Art und Weise erstellten Sichten enthalten in den meisten Datenquellen weder Statistikdaten noch Indizes, und die Sichten können nicht aktualisiert werden.

Datentypzuordnungen und der globale Katalog der föderierten Datenbank

Lokale Datentypdefinitionen werden in der Katalogsicht SYSCAT.COLUMNS des globalen Katalogs der föderierten Datenbank gespeichert.

Wenn Sie eine Anweisung CREATE NICKNAME schreiben, geben Sie ein Datenquellenobjekt an, das durch den Kurznamen dargestellt wird. In den meisten Fällen definiert der Server mit föderierten Datenbanken für jede Spalte bzw. jedes Feld in diesem Datenquellenobjekt einen von DB2 unterstützten Datentyp. Bei einigen nicht relationalen Datenquellen muss der DB2-Datentyp angegeben werden.

Um bei relationalen Datenquellen festzustellen, welcher lokale Datentyp in der Katalogsicht SYSCAT.COLUMNS zu speichern ist, sucht der Server mit föderierten Datenbanken nach Informationen zu progressiven Datentypzuordnungen in den Wrappern und in der Katalogsicht SYSCAT.TYPEMAPPINGS. Zuordnungen in der Katalogsicht SYSCAT.TYPEMAPPINGS haben dabei eine Vorrangstellung gegenüber den Standardzuordnungen in den Wrappern. Wenn Sie alternative Zuordnungen zum Überschreiben der Standarddatentypzuordnungen erstellen, verwendet der Server mit föderierten Datenbanken die alternativen Zuordnungen. Wenn für eine Spalte mehrere Zuordnungen gelten, verwendet der Server mit föderierten Datenbanken die zuletzt erstellten Zuordnung.

Um bei nicht relationalen Datenquellen festzustellen, welcher lokale Datentyp in der Katalogsicht SYSCAT.COLUMNS zu speichern ist, sucht der Server mit föderierten Datenbanken nach Informationen zu Datentypzuordnungen in den Wrappern. Der Grad, in dem vom Wrapper definierte Datentypen geändert werden können, ändert sich in Abhängigkeit von der nicht relationalen Datenquelle. Bei einigen nicht relationalen Datenquellen werden keine Spalten angegeben. Der Wrapper definiert die Datentypen. Bei anderen Datenquellen können die Datentypen überschrieben werden. Darüber hinaus gibt es Datenquellen, für die die Spaltendatentypen in der Anweisung CREATE NICKNAME angegeben werden müssen.

Wenn Sie transparente CREATE TABLE-DDL für relationale Datenquellen schreiben, geben Sie in der Anweisung DB2-Datentypen an. Der Server mit föderierten Datenbanken sucht nach Informationen über die regressiven Datentypzuordnungen zwischen der föderierten Datenbank und der Datenquelle. Der Server mit föderierten Datenbanken sucht nach diesen Informationen im Wrapper und in der Katalogsicht SYSCAT.TYPEMAPPINGS.

Wenn Werte von einer Spalte der Datenquelle an die föderierte Datenbank zurückgegeben werden, sind diese Werte vollständig konform mit dem DB2-Datentyp, dem die Datenquellenspalte zugeordnet ist. Handelt es sich bei dieser Zuordnung um eine Standardzuordnung, sind die Werte ebenfalls vollständig konform mit dem Datenquellentyp in der Zuordnung. Wenn zum Beispiel eine Oracle-Tabelle mit einer FLOAT-Spalte für die föderierte Datenbank definiert ist, gilt für diese Spalte automatisch die Standardzuordnung von Oracle-FLOAT zu DB2-DOUBLE. Die Werte, die von der Spalte zurückgegeben werden, sind sowohl mit dem Datentyp FLOAT als auch mit dem Datentyp DOUBLE konform.

Erstellen alternativer Datentypzuordnungen

Sie können alternative Datentypzuordnungen für relationale Datenquellen erstellen.

Das Erstellen alternativer Datentypzuordnungen kann in den folgenden Situationen sinnvoll sein:

- Zum Überschreiben einer Standarddatentypzuordnung

Bei einigen Wrappern können Sie das Format oder die Länge der zurückgegebenen Werte ändern. Ändern Sie Format oder Länge, indem Sie den DB2-Datentyp ändern, mit dem die Werte konform sein müssen. Der Oracle-Datentyp DATE wird beispielsweise als Zeitmarke verwendet und enthält das Jahrhundert, das Jahr, den Monat, den Tag, die Stunde, die Minute und die Sekunde. Standardmäßig ist der Oracle-Datentyp DATE dem DB2-Datentyp TIMESTAMP zugeordnet. Um nur die Informationen zu Stunde, Minute und Sekunde zurückzugeben, können Sie die Standarddatentypzuordnung überschreiben, damit der Oracle-Datentyp DATE dem DB2-Datentyp TIME zugeordnet wird.

Beim Abfragen der DATE-Spalten von Oracle wird dann nur der Zeitabschnitt der Oracle-Zeitmarkenwerte an den Server mit föderierten Datenbanken zurückgegeben.

- Wenn keine Standardzuordnung vorhanden ist

Wenn für einen Datentyp einer Datenquelle keine Standarddatentypzuordnung verfügbar ist, müssen Sie für den neuen Datentyp eine Zuordnung erstellen.

Verwenden Sie die Anweisung CREATE TYPE MAPPING, um neue Datentypzuordnungen zu definieren. Von Ihnen erstellte Zuordnungen werden in der Katalogsicht SYSCAT.TYPEMAPPINGS in der föderierten Datenbank gespeichert.

Datentypzuordnungen für nicht relationale Datenquellen

Für einige nicht relationale Datenquellen sind die Datentypzuordnungen nicht in den Wrappern enthalten. In einigen Fällen müssen Sie die Informationen des lokalen Typs in der Anweisung CREATE NICKNAME angeben.

Im folgenden Beispiel wird gezeigt, wie Spaltendatentypen in der Anweisung CREATE NICKNAME für einige der nicht relationalen Datenquellen angegeben werden:

```
CREATE NICKNAME DRUGDATA1
  (Dcode Integer NOT NULL, Drug CHAR(20), Manufacturer CHAR(20))
  FOR SERVER biochem_lab
  OPTIONS (FILE_PATH '/usr/pat/DRUGDATA1.TXT', COLUMN_DELIMITER ',',
  SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y')
```

Progressive und regressive Datentypzuordnungen

Progressive Typzuordnungen und regressive Typzuordnungen sind die beiden Arten von Zuordnungen zwischen Datenquellendatentypen und Datentypen der föderierten Datenbank.

Eine *progressive Typzuordnung* ist eine Zuordnung von einem fernen Datentyp zu einem vergleichbaren lokalen Datentyp. Progressive Typzuordnungen werden bei der Erstellung eines Kurznamens für ein Datenquellenobjekt verwendet. Der vergleichbare lokale Typ für die einzelnen Spalten im Datenquellenobjekt wird im globalen Katalog gespeichert.

Eine *regressive Typzuordnung* ist eine Zuordnung von einem lokalen Datentyp zu einem vergleichbaren fernen Datentyp. Die regressive Typzuordnung wird mit transparenter DDL verwendet.

In Abb. 2 auf Seite 50 ist die progressive und die regressive Datentypzuordnung dargestellt.

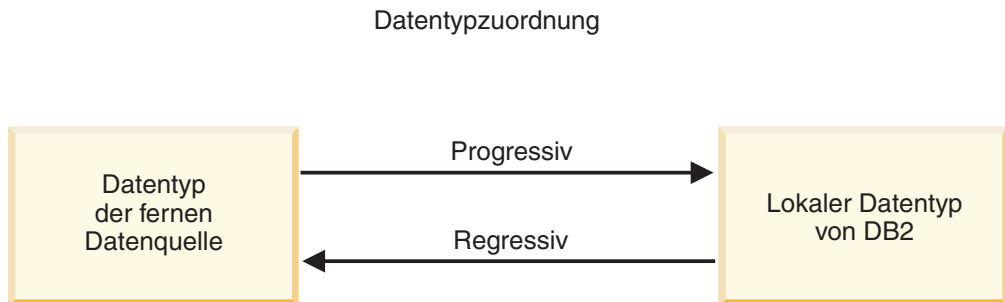


Abbildung 2. Progressive und regressive Datentypzuordnungen

Erstellen von Datentypzuordnungen

Mit der Anweisung `CREATE TYPE MAPPING` können Sie eine Datentypzuordnung erstellen. Sie können die Anweisung entweder über die DB2-Befehlszentrale oder über den Befehlszeilenprozessor ausführen oder sie in ein Anwendungsprogramm aufnehmen. Die DB2-Steuerzentrale können Sie nicht zum Erstellen oder Ändern von Datentypzuordnungen verwenden.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung `SYSADM` oder `DBADM` umfassen.

Informationen zu dieser Task

Spezielle Informationen zur Verwendung finden Sie unter Anweisung `CREATE TYPE MAPPING`.

Einschränkungen

- Der Wert *lokaler_datentyp* darf weder `LONG VARCHAR`, `LONG VARGRAPHIC` noch ein benutzerdefinierter Datentyp sein.
- Der Wert *datenquellendatentyp* darf kein benutzerdefinierter Datentyp sein.
- Für nicht relationale Datenquellen ist das Maß, in dem vorhandene Datentypzuordnungen außer Kraft gesetzt oder Zuordnungen erstellt werden können, begrenzt.

Vorgehensweise

Führen Sie die Anweisung `CREATE TYPE MAPPING` aus, um eine Datentypzuordnung zu erstellen.

Um einen Servertyp in der Anweisung `CREATE TYPE MAPPING` anzugeben, müssen Sie eine der folgenden Angaben als Wert für *servertyp* verwenden:

Tabelle 4. Gültige Servertypen

Datenquelle	Servertyp
DB2 Database für Linux, UNIX und Windows	DB2/CS DB2/UDB DB2/NT DB2/SUN DB2/HP DB2/HPUX DB2/AIX DB2/6000 DB2/PE DB2/PTX DB2/SCO DB2/LINUX DB2/EEE DB2/2
DB2 für z/OS	DB2/MVS DB2/ZOS DB2/390
DB2 Server für VSE und VM	DB2/VM DB2/VSE SQL/DS
DB2 für System i	DB2/400 DB2/ISERIES
Oracle	ORACLE
Informix	INFORMIX
ODBC	ODBC
Microsoft SQL Server	MSSQLSERVER
JDBC	JDBC
Teradata	TERADATA
Sybase	SYBASE

Erstellen einer Typzuordnung für einen Datenquellendatentyp - Beispiel

In diesem Beispiel müssen alle Oracle-Tabellen und -Sichten, die den Oracle-Datentyp NUMBER verwenden, dem DB2-Datentyp DECIMAL(8,2) zugeordnet werden. Der Oracle-Datentyp NUMBER wird standardmäßig dem DB2-Datentyp DOUBLE, einem Gleitkommadatentyp, zugeordnet.

Verwenden Sie die Anweisung ALTER NICKNAME, um die lokalen Typen vorhandener Kurznamen zu ändern. Sie müssen jeden Kurznamen gesondert ändern, um den lokalen Datentyp in DECIMAL(8,2) zu ändern. Falls die Kurznamen nicht vorhanden sind, erstellen Sie eine Datentypzuordnung, die den Datenquellentyp angibt. Stellen Sie sicher, dass der Datenquellenwrapper erstellt wurde, bevor Sie die Anweisung CREATE TYPE MAPPING ausführen. Um die Typzuordnung vom Oracle-Datentyp NUMBER zum DB2-Datentyp DECIMAL(8,2) zu erstellen, führen Sie die Anweisung CREATE TYPE MAPPING aus. Beispiel:

```
CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(8,2)
TO SERVER TYPE ORACLE TYPE NUMBER
```

MY_ORACLE_DEC

Der Name, den Sie der Typzuordnung geben. Der Name darf nicht identisch mit einem Namen einer Datentypzuordnung sein, die im Katalog bereits vorhanden ist.

FROM *SYSIBM.DECIMAL(8,2)*

Das lokale DB2-Schema und der lokale Datentyp. Wenn die Länge oder Genauigkeit und die Anzahl der Kommastellen nicht angegeben sind, werden diese Werte vom Quellendatentyp bestimmt.

TO SERVER TYPE *ORACLE*

Der Typ der Datenquelle.

TYPE *NUMBER*

Der Datentyp der Datenquelle, den Sie dem lokalen Datentyp zuordnen. Benutzerdefinierter Datentypen sind nicht zulässig.

Der DB2-Datentyp *DECIMAL(8,2)* wird für die Oracle-Spalten lokal definiert. Wenn Sie Kurznamen für Oracle-Tabellen und -Sichten erstellen, die *NUMBER*-Spalten enthalten, wird der Oracle-Datentyp *NUMBER* dem DB2-Datentyp *DECIMAL(8,2)* zugeordnet.

Erstellen einer Typzuordnung für einen Datenquellendatentyp und eine Datenquellenversion - Beispiel

Bei diesem Beispiel sind Oracle-Tabellen und -Sichten auf unterschiedlichen Versionen des Oracle-Servers vorhanden. Für alle Tabellen und Sichten auf Oracle Version 8.0.3-Servern müssen Spalten, die den Oracle-Datentyp *NUMBER(23,3)* verwenden, dem DB2-Datentyp *DECIMAL(8,2)* zugeordnet werden.

Der Oracle-Datentyp *NUMBER(23,3)* wird standardmäßig dem DB2-Datentyp *DECIMAL(23,3)* zugeordnet. Verwenden Sie die Anweisung *ALTER NICKNAME*, um die lokalen Typen vorhandener Kurznamen zu ändern. Sie müssen jeden Kurznamen gesondert ändern, um den lokalen Datentyp in *DECIMAL(8,2)* zu ändern. Falls die Kurznamen nicht vorhanden sind, erstellen Sie eine Datentypzuordnung, die den Datenquellentyp angibt. Stellen Sie sicher, dass der Datenquellenwrapper erstellt wurde, bevor Sie die Anweisung *CREATE TYPE MAPPING* ausführen. Um für Oracle-Server, die Version 8.0.3 verwenden, den Oracle-Datentyp *NUMBER(23,3)* dem DB2-Datentyp *DECIMAL(8,2)* zuzuordnen, führen Sie die Anweisung *CREATE TYPE MAPPING* aus. Beispiel:

```
CREATE TYPE MAPPING ORA_DEC FROM SYSIBM.DECIMAL(8,2)
  TO SERVER TYPE ORACLE VERSION 8.0.3 TYPE NUMBER(23,3)
```

ORA_DEC

Der Name, den Sie der Typzuordnung geben. Der Name darf nicht identisch mit einem Namen einer Datentypzuordnung sein, die im Katalog bereits vorhanden ist.

FROM *SYSIBM.DECIMAL(8,2)*

Das lokale DB2-Schema und der lokale Datentyp. Wenn die Länge oder Genauigkeit und die Anzahl der Kommastellen nicht angegeben sind, werden diese Werte vom Quellendatentyp bestimmt.

TO SERVER TYPE *ORACLE*

Der Typ der Datenquelle.

VERSION 8.0.3

Die Version des Datenquellenservers. Die Version muss angegeben werden. Sie können, wie im vorliegenden Beispiel dargestellt, auch das Release und die Änderung angeben.

TYPE NUMBER(23,3)

Der Datentyp der Datenquelle, den Sie dem lokalen Datentyp zuordnen. Benutzerdefinierter Datentypen sind nicht zulässig.

Die föderierte Datenbank definiert den DB2-Datentyp DECIMAL(8,2) für die Oracle-Spalten auf Version 8.0.3-Servern lokal. Für Oracle-Tabellen und -Sichten auf Servern, die nicht Version 8.0.3 verwenden, wird stattdessen die standardmäßige Datentypzuordnung verwendet. Wenn Sie Kurznamen für Oracle-Tabellen und -Sichten erstellen, die NUMBER-Spalten enthalten, wird der Oracle-Datentyp NUMBER dem DB2-Datentyp DECIMAL(8,2) zugeordnet.

Erstellen einer Typzuordnung für alle Datenquellenobjekte auf einem Server - Beispiel

In diesem Beispiel ist der Server für die föderierte Datenbank als ORA2SERVER definiert. Jede Tabelle enthält eine Spalte mit dem Oracle-Datentyp DATE.

Der Oracle-Datentyp DATE enthält das Jahrhundert, das Jahr, den Monat, den Tag, die Stunde, die Minute und die Sekunde. Standardmäßig wird der Oracle-Datentyp DATE dem lokalen DB2-Datentyp TIMESTAMP zugeordnet. Wenn Sie jedoch eine Abfrage eines Objekts auf diesem Server ausführen, darf die Ergebnismenge nur die Zeitinformationen (Stunde, Minute und Sekunde) zurückgeben.

Verwenden Sie die Anweisung ALTER NICKNAME, um die lokalen Typen vorhandener Kurznamen zu ändern. Sie müssen jeden Kurznamen gesondert ändern, um den lokalen Datentyp in TIME zu ändern.

Falls die Kurznamen nicht vorhanden sind, erstellen Sie eine Datentypzuordnung, die den Datenquellentyp angibt.

Um den Oracle-Datentyp DATE dem DB2-Datentyp TIME für den ORA2SERVER zuzuordnen, setzen Sie die folgende Anweisung ab:

```
CREATE TYPE MAPPING ORA2_DATE FROM SYSIBM.TIME  
TO SERVER ORA2SERVER TYPE DATE
```

ORA2_DATE

Der Name, den Sie der Typzuordnung geben. Der Name darf nicht identisch mit einem Namen einer Datentypzuordnung sein, die im Katalog bereits vorhanden ist.

FROM SYSIBM.TIME

Das lokale DB2-Schema und der lokale Datentyp. Wenn die Länge oder Genauigkeit und die Anzahl der Kommastellen nicht angegeben sind, werden diese Werte vom Quellendatentyp bestimmt.

TO SERVER ORA2SERVER

Der lokale Name des Datenquellenservers.

TYPE DATE

Der Datentyp der Datenquelle, den Sie dem lokalen Datentyp zuordnen. Benutzerdefinierter Datentypen sind nicht zulässig.

Die föderierte Datenbank definiert den DB2-Datentyp TIME für die Oracle-Spalten vom Datentyp DATE lokal.

Wenn Sie Kurznamen für Oracle-Tabellen und -Sichten erstellen, die DATE-Spalten enthalten, wird der Oracle-Datentyp DATE dem DB2-Datentyp DECIMAL(8,2) zugeordnet.

Datenquellenobjekte auf anderen Oracle-Servern werden durch die Datentypzuordnung nicht beeinflusst.

Umsetzung zwischen Datentypen

Sie können einen Datentyp aus dem Quelldatentyp in den Zieldatentyp umsetzen.

Die Umsetzung zwischen Datentypen, die auch *Casting* genannt wird, kann implizit oder explizit stattfinden.

- Als *implizites Casting* wird die automatische Konvertierung von Daten eines Datentyps in Daten eines anderen Datentyps bezeichnet, die auf einer implizierten Gruppe von Konvertierungsregeln basiert. Diese automatische Konvertierung findet zur Unterstützung der schwachen Typprüfung statt.
- Das *explizite Casting* unterstützt die starke Typprüfung. Die starke Typprüfung setzt übereinstimmende Datentypen voraus. Sie müssen einen oder beide Datentypen explizit in einen gemeinsamen Datentyp konvertieren, bevor Sie Vergleiche ausführen oder Zuordnungen vornehmen.

Die Federation-Unterstützung für das Casting zwischen Datentypen ermöglicht es föderierten Abfragen für Kurznamen, auf Server zuzugreifen, die sowohl die schwache als auch die starke Typprüfung unterstützen.

Umsetzungsfunktionen können nicht im Pushdown-Modus an ferne Server weitergeleitet werden, falls auf dem fernen Server keine entsprechende Funktion vorhanden ist. Ein übermäßiger Einsatz des Castings kann zu Leistungsproblemen führen.

Beispiele: Explizites Casting

```
UPDATE nickname SET varcharcol = CAST(intcol AS varchar(10))  
  
SELECT REAL(varchar_col) FROM nickname1;  
  
SELECT VARCHAR(double_col) FROM nickname1;
```

Beispiele: Implizites Casting

Beispiel 1:

```
UPDATE nickname SET varcharcol = intcol;
```

In dieser Zuordnungsoperation ist die an den fernen Server im Pushdown-Modus weitergeleitete Anweisung äquivalent zur folgenden Anweisung:

```
UPDATE nickname SET varcharcol = varchar(intcol);
```

Beispiel 2:

```
INSERT INTO nickname (varcharcol) SELECT intcol FROM nickname1;
```

In dieser Zuordnungsoperation ist die an den fernen Server im Pushdown-Modus weitergeleitete Anweisung äquivalent zur folgenden Anweisung:

```
INSERT INTO nickname (varcharcol) SELECT varchar(intcol) FROM nickname1;
```

Beispiel 3:

```
SELECT * SELECT nickname SELECT intcol = varcharcol;
```

In dieser Vergleichsoperation ist die an den fernen Server im Pushdown-Modus weitergeleitete Anweisung äquivalent zur folgenden Anweisung:

```
SELECT * SELECT nickname SELECT intcol = CAST(varcharcol AS decfloat)
```

Unterstützung für Datentyp **TIMESTAMP**

Der Datentyp **TIMESTAMP** wurde zur Steuerung der Genauigkeit bei Sekundenbruchteilen parametrisiert.

Bei der DB2 für Linux, UNIX und Windows-Datenquelle werden ferne Zeitmarken dem Datentyp **TIMESTAMP(*p*)** zugeordnet. Hierbei steht *p* für die Genauigkeit und gibt die Anzahl der Sekundenbruchteile an. Für *p* ist der Bereich 0 bis 12 (einschließlich) zulässig.

Bei anderen Datenquellen werden ferne Zeitmarken dem Datentyp **TIMESTAMP** mit der Genauigkeit 6 zugeordnet. Bei diesen Datenquellen können Sie die Unterstützung von **TIMESTAMP(*p*)** nutzen, indem Sie ähnliche Zuordnungen wie bei den Beispielen verwenden, die für die standardmäßige progressive Typzuordnung bereitgestellt werden.

Wenn die Kurznamenspalte **TIMESTAMP** eine geringere Genauigkeit als ihre entsprechende Spalte in der fernen Tabelle hat und die ferne Tabellenspalte bei den überzähligen Dezimalstellen Ziffern ungleich null enthält, sind die Ergebnisse von Vergleichselementen, die diese Kurznamenspalte verwenden, nicht vorhersehbar. Falls das Vergleichselement nicht im Pushdown-Modus weitergeleitet wird, ist das Ergebnis der Vergleichselemente, die diese Kurznamenspalte verwenden, einwandfrei.

Ändern eines lokalen Typs für ein Datenquellenobjekt (DB2-Steuerzentrale)

Verwenden Sie die Anweisung **ALTER NICKNAME** anstatt der Anweisung **CREATE TYPE MAPPING**, um einen lokalen Datentyp zu ändern. Sie können den Datentyp über die DB2-Steuerzentrale oder über die DB2-Befehlszeile ändern.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung **SYSADM** oder **DBADM**
- Zugriffsrecht **ALTER** für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht **CONTROL** für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht **ALTERIN** für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte **DEFINER** der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Beim Erstellen eines Kurznamens werden die Datentypen, die dem Datenquellenobjekt zugeordnet sind, in der föderierten Datenbank gespeichert. Bei einigen Datenquellen werden die Datentypen vom Wrapper angegeben. Bei anderen Datenquellen müssen Sie die Datentypen beim Erstellen des Kurznamens angeben.

Sie können für eine Spalte eines bestimmten Datenquellenobjekts einen lokalen Typ angeben.

Wichtiger Hinweis: Das Ändern des lokalen Datentyps kann zu Fehlern oder Datenverlust führen, wenn Sie den lokalen Datentyp für eine Spalte in einen Typ ändern, der erheblich von seinem fernen Typ abweicht.

Vorgehensweise

Gehen Sie wie folgt vor, um den lokalen Datentyp über die DB2-Steuerzentrale zu ändern:

1. Wählen Sie den Ordner **Kurznamen** aus.
2. Klicken Sie mit der rechten Maustaste den zu ändernden Kurznamen an, und klicken Sie anschließend **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
3. Wählen Sie im Fenster 'Kurznamen' die zu ändernde Spalte aus, und klicken Sie anschließend **Ändern** an. Das Fenster 'Spalte ändern' wird geöffnet.
4. Wählen Sie den Datentyp aus.
5. Klicken Sie **OK** an, um den Datentyp zu ändern und das Fenster zu schließen.
6. Klicken Sie **OK** an, um den Kurznamen zu ändern und das Notizbuch zu schließen.

Um den Inhalt einer lokalen Spalte, die einen Zeichendatentyp aufweist, als Bitdaten (binär) zu behandeln, verwenden Sie die Klausel FOR BIT DATA in der Anweisung ALTER NICKNAME. Wenn Sie diese Klausel zum Ändern des lokalen Datentyps einer Spalte verwenden, werden die Codepagekonvertierungen nicht ausgeführt, wenn Daten mit anderen Systemen ausgetauscht werden. Abgleiche werden unabhängig von der Sortierfolge der fernen Datenbank im binären Modus ausgeführt.

Ändern eines lokalen Typs für ein Datenquellenobjekt - Beispiele

In diesem Abschnitt werden Beispiele beschrieben, die das Ändern der Datentypen für ein Datenquellenobjekt veranschaulichen.

Beispiel: Eine numerische Datentypzuordnung

In einer Oracle-Tabelle für Mitarbeiterdaten ist die Spalte BONUS mit dem Datentyp NUMBER(32,3) definiert. Der Oracle-Datentyp NUMBER(32,3) ist standardmäßig dem DB2-Datentyp DOUBLE, einem Datentyp für Gleitkommazahlen mit doppelter Genauigkeit, zugeordnet. Bei einer Abfrage, die die Spalte BONUS umfasst, werden möglicherweise Werte zurückgegeben, die folgendermaßen aussehen:

```
5.0000000000000E+002  
1.0000000000000E+003
```

Die Exponentialschreibweise gibt die Anzahl von Dezimalstellen an sowie die Richtung, in der das Dezimalzeichen zu bewegen ist. Im vorliegenden Beispiel bedeutet die Angabe +002, dass das Dezimalzeichen um zwei Stellen nach rechts zu verschieben ist. Die Angabe +003 bedeutet, dass das Dezimalzeichen um drei Stellen nach rechts zu verschieben ist.

Bei Abfragen, die die Spalte BONUS einbeziehen, können Werte zurückgegeben werden, die wie Dollar-Beträge aussehen. Sie ändern die lokale Definition für die Spalte BONUS in der Tabelle vom Datentyp DOUBLE in den Datentyp DECIMAL. Verwenden Sie eine Genauigkeit und eine Kommastellenanzahl, die das Format tatsächlicher Bonusbeträge widerspiegeln. Wenn der Dollar-Teil der Bonusbeträge beispielsweise nicht länger als sechs Stellen ist, ordnen Sie den Datentyp NUMBER(32,3) dem Typ DECIMAL(8,2) zu. Unter Berücksichtigung der Einschränkungen dieses neuen lokalen Typs werden bei Abfragen, die die Spalte BONUS einbeziehen, beispielsweise folgende Werte zurückgegeben:

```
500.00  
1000.00
```

Der Kurzname für die Oracle-Tabelle lautet ORASALES. Um die Spalte BONUS in der Tabelle ORASALES dem DB2-Datentyp DECIMAL (8,2) zuzuordnen, setzen Sie die folgende ALTER NICKNAME-Anweisung ab:

```
ALTER NICKNAME ORASALES ALTER COLUMN BONUS  
LOCAL TYPE DECIMAL(8,2)
```

ORASALES

Der Kurzname, den Sie für die Oracle-Tabelle definiert haben.

ALTER COLUMN BONUS

Der Name der Spalte, der in der Katalogsicht SYSCAT.COLUMNS der föderierten Datenbank lokal definiert ist.

LOCAL TYPE DECIMAL(8,2)

Gibt den neuen lokalen Typ für die Spalte an.

Diese Zuordnung gilt nur für die Spalte BONUS in der Oracle-Tabelle, die durch den Kurznamen ORASALES identifiziert wird. Alle anderen Oracle-Datenquellenobjekte, die die Spalte BONUS enthalten, verwenden die Standarddatentypzuordnung für den Oracle-Datentyp NUMBER.

Beispiel: Eine Zuordnung von Datumsdatentypen

Der Kurzname für eine Oracle-Tabelle mit dem Namen SALES lautet ORASALES. Die Tabelle SALES enthält eine Spalte mit dem Oracle-Datentyp DATE. Standardmäßig ist der Oracle-Datentyp DATE dem DB2-Datentyp TIMESTAMP zugeordnet. Sie möchten jedoch, dass beim Abrufen von Daten von dieser Spalte nur der Datumswert angezeigt wird. Sie können den Kurznamen für die Tabelle SALES ändern, um den lokalen Typ in den DB2-Datentyp DATE zu ändern.

```
ALTER NICKNAME ORASALES ALTER COLUMN ORDER_DATE  
LOCAL TYPE DATE
```

Beispiel: Eine Datentypzuordnung für eine nicht relationale Datenquelle

Der Kurzname für eine tabellenstrukturierte Datei namens drugdata1.txt lautet DRUGDATA1. Die Datei drugdata1.txt enthält eine Spalte, in der die Namen von Arzneimitteln aufgelistet sind. Der Spaltenname ist DRUG. Die Spalte DRUG war ursprünglich als CHAR(20) definiert.

Die Länge der Spalte muss in CHAR(30) geändert werden. Sie können den Kurznamen für die Datei drugdata1.txt ändern, um die Zuordnung auf die richtige Länge zu ändern.

```
ALTER NICKNAME DRUGDATA1 ALTER COLUMN DRUG  
LOCAL TYPE CHAR(30)
```

Ändern eines lokalen Typs für ein Datenquellenobjekt (DB2-Befehlszeile)

Verwenden Sie die Anweisung ALTER NICKNAME anstatt der Anweisung CREATE TYPE MAPPING, um einen lokalen Datentyp zu ändern. Sie können den Datentyp über die DB2-Steuerzentrale oder über die DB2-Befehlszeile ändern.

Vorbereitung

Die Berechtigungs-ID, von der die Anweisung abgesetzt wird, muss mindestens eines der folgenden Zugriffsrechte aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht ALTER für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht CONTROL für den in der Anweisung angegebenen Kurznamen
- Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
- Definierender Benutzer des Kurznamens, der in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen ist

Einschränkungen

Siehe „Einschränkungen beim Ändern von Kurznamen“ auf Seite 34.

Informationen zu dieser Task

Beim Erstellen eines Kurznamens werden die Datentypen, die dem Datenquellenobjekt zugeordnet sind, in der föderierten Datenbank gespeichert. Bei einigen Datenquellen werden die Datentypen vom Wrapper angegeben. Bei anderen Datenquellen müssen Sie die Datentypen beim Erstellen des Kurznamens angeben.

Sie können für eine Spalte eines bestimmten Datenquellenobjekts einen lokalen Typ angeben.

Wichtiger Hinweis: Das Ändern des lokalen Datentyps kann zu Fehlern oder Datenverlust führen, wenn Sie den lokalen Datentyp für eine Spalte in einen Typ ändern, der erheblich von seinem fernen Typ abweicht.

Vorgehensweise

Verwenden Sie zum Ändern des lokalen Datentyps über die Befehlszeilenaufforderung die Anweisung ALTER NICKNAME. Beispiel:

```
ALTER NICKNAME kurzname ALTER COLUMN spaltenname  
LOCAL TYPE datentyp
```

Um den Inhalt einer lokalen Spalte, die einen Zeichendatentyp aufweist, als Bitdaten (binär) zu behandeln, verwenden Sie die Klausel FOR BIT DATA in der Anweisung ALTER NICKNAME.

Wenn Sie diese Klausel zum Ändern des lokalen Datentyps einer Spalte verwenden, werden die Codepagekonvertierungen nicht ausgeführt, wenn Daten mit anderen Systemen ausgetauscht werden. Abgleiche werden unabhängig von der Sortierfolge der fernen Datenbank im binären Modus ausgeführt.

Ändern von LONG-Datentypen in VARCHAR-Datentypen

Um Einfügungs- und Aktualisierungsoperationen bei LONG-Datentypen zu ermöglichen, können Sie die LONG-Datentypen in den Datentyp VARCHAR ändern.

In Tabelle 5 sind die LONG-Datentypen nach der Datenquelle aufgelistet, die geändert werden kann.

Tabelle 5. LONG-Datentypen nach Datenquelle, die in den Datentyp VARCHAR geändert werden kann

Datenquelle	Ferner Datentyp	Länge	Lokaler Standard-datentyp	ALTER in VARCHAR
DRDA	LONG VARCHAR	1–32672	CLOB	VARCHAR
	LONG VARCHAR FOR BIT DATA	1–32672	BLOB	VARCHAR FOR BIT DATA
Informix	BYTE	1–32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1–32672	CLOB	VARCHAR
Microsoft SQL Server	IMAGE	1–32672 Hostvariablen; 1–8000 Literal	BLOB	VARCHAR FOR BIT DATA
	TEXT	1–32672 Hostvariablen; 1–8000 Literal	CLOB	VARCHAR
Oracle NET8	LONG	1–32672 Hostvariablen; 1–4000 Literal	CLOB	VARCHAR
	LONG RAW	1–32672 Hostvariablen; 1–4000 Literal	BLOB	VARCHAR FOR BIT DATA
Sybase CTLIB	IMAGE	1–32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1–32672	CLOB	VARCHAR
Teradata	BYTE	32673–64000	BLOB	VARCHAR FOR BIT DATA(32672)
	CHAR	32673–64000	CLOB	VARCHAR(32672)
	VARBYTE	32673–64000	BLOB	VARCHAR FOR BIT DATA(32672)
	VARCHAR	32673–64000	CLOB	VARCHAR(32672)

Kapitel 4. Zuordnen von Funktionen und benutzerdefinierten Funktionen (UDFs)

Funktionszuordnungen ordnen Funktionen von Servern mit föderierten Datenbanken und benutzerdefinierte Funktionen (UDFs) vorhandenen Funktionen in der Datenquelle zu.

Funktionszuordnungen in einem föderierten System

IBM InfoSphere Federation Server stellt Standardzuordnungen zwischen vorhandenen Datenquellenfunktionen und den entsprechenden DB2-Funktionen bereit.

Damit der Server mit föderierten Datenbanken eine Datenquellenfunktion verwenden kann, ist eine Zuordnung einer DB2-Funktion oder einer Funktionsschablone zur entsprechenden Datenquellenfunktion erforderlich.

Die Standardfunktionszuordnungen sind in den Wrappermodulen vorhanden.

Für nicht relationale Datenquellen können keine vorhandenen Funktionszuordnungen außer Kraft gesetzt und keine neuen Zuordnungen erstellt werden.

Funktionsweise von Funktionszuordnungen in einem föderierten System

Wenn Sie Abfragen an den Server mit föderierten Datenbanken übergeben, die eine Funktion oder mehrere Funktionen beinhalten, sucht der Server mit föderierten Datenbanken nach Informationen über die Zuordnungen zwischen den DB2-Funktionen und den Datenquellenfunktionen.

Der Server mit föderierten Datenbanken sucht an den folgenden beiden Bereichen nach Zuordnungsinformationen:

- Im Wrapper. Der Wrapper der Datenquelle enthält die Standardfunktionszuordnungen.
- In der Katalogsicht SYSCAT.FUNCMAPPINGS. Diese Sicht enthält erstellte Einträge, die die Standardfunktionszuordnungen im Wrapper überschreiben oder ergänzen. Sie enthält ebenfalls neue Zuordnungen, die Sie erstellen, wenn keine Standardfunktionszuordnung vorhanden ist. Wenn auf eine Funktion mehrere Zuordnungen angewendet werden können, wird die zuletzt erstellte angewendet.

Optionen für Funktionszuordnungen geben Informationen zu der Funktion und dem potenziellen Aufwand an, der zur Verarbeitung einer Funktion in der Datenquelle erforderlich wäre. Optionen für Funktionszuordnungen stellen beispielsweise folgende Informationen zur Verfügung:

- Name der fernen Datenquellenfunktion
- Geschätzte Anzahl der beim ersten und letzten Aufruf der Datenquellenfunktion verarbeiteten Anweisungen
- Geschätzte Anzahl der beim ersten und letzten Aufruf der Datenquellenfunktion ausgeführten Eingaben und Ausgaben
- Geschätzte Anzahl der pro Aufruf der Datenquellenfunktion verarbeiteten Anweisungen.

Beim Erstellen einer Funktionszuordnung ordnen Sie eine DB2-Funktion oder Funktionsvorlage einer entsprechenden Funktion in der Datenquelle zu. Ist in DB2 keine entsprechende Funktion vorhanden, oder soll der Server mit föderierten Datenbanken die Datenquellenfunktion verwenden, können Sie eine Funktionsvorlage als entsprechende Funktion erstellen.

Bedeutung von Funktionszuordnungen

Funktionszuordnungen gehören zu den wichtigen Eingaben für die vom Abfrageoptimierungsprogramm ausgeführte Pushdown-Analyse.

Bei der Auswahl des am besten geeigneten Abfragezugriffsplans faktorisiert das Abfrageoptimierungsprogramm die Funktionalität der Datenquelle, um einen bestimmten Typ SQL-Funktion oder -Operation auszuführen. Wenn die Funktion keine Zuordnung hat, wird die Funktion nicht zur Verarbeitung an die Datenquelle gesendet. Funktionen und andere Operationen, die im Pushdown-Modus an die Datenquelle weitergeleitet werden können, steigern die Leistung.

Wenn die Datenquelle eine einer DB2-Funktion entsprechende Funktion enthält, die jedoch leicht abweichende Ergebnisse liefert, kann durch das Erstellen einer Funktionszuordnung die Leistung gesteigert werden. Die Informix-Funktion STDEV (standard deviation = Standardabweichung) beispielsweise liefert für einige Eingabedatenmengen andere Ergebnisse als die DB2-Funktion STDDEV. Aus diesem Grund enthält der Informix-Wrapper keine Standardzuordnung zwischen diesen beiden Funktionen. Wenn für Ihre Zwecke die Abweichungen in den Ergebnissen unerheblich sind, können Sie die Leistung der Abfragen steigern, die auf Informix-Datenquellen zugreifen und die DB2-Funktion STDDEV verwenden. Indem Sie eine Zuordnung zwischen der Informix-Funktion STDEV und der DB2-Funktion STDDEV erstellen, bieten Sie dem Abfrageoptimierungsprogramm die Möglichkeit, die Verarbeitung dieser Funktion an die Datenquelle weiterzuleiten.

Empfehlungen für das Erstellen von Funktionszuordnungen

Wenn keine Standardfunktionszuordnung für eine Datenquellenfunktion vorhanden ist, können Sie eine Funktionszuordnung erstellen.

Ein Grund, warum eine Funktionszuordnung nicht verfügbar ist, kann darin bestehen, dass die föderierte Datenbank keine Funktion aufweist, die der Datenquellenfunktion entspricht.

Wenn für eine Datenquellenfunktion zwar eine entsprechende DB2-Funktion vorhanden ist, aber trotzdem keine Standardzuordnung zur Verfügung steht, kann dies daran liegen, dass diese Funktion nicht dasselbe Ergebnis zurückgibt. Wenn die Datenquelle leicht oder deutlich abweichende Ergebnisse für bestimmte Eingabedatenmengen liefert, stellen die Wrapper normalerweise keine Zuordnung zu diesen Funktionen her. Wenn Ihnen die Unterschiede in den Ergebnissen jedoch nicht so wichtig sind, können Sie eine Zuordnung zwischen den Funktionen erstellen. Das Erstellen einer Zuordnung kann zu einer Leistungssteigerung führen.

Verwenden Sie Funktionszuordnungen in folgenden Situationen:

- In der Datenquelle wird eine neue integrierte Funktion verfügbar.
- In der Datenquelle wird eine neue benutzerdefinierte Funktion verfügbar.
- Es ist keine entsprechende DB2-Funktion verfügbar.
- Es ist eine entsprechende Funktion verfügbar, die aber abweichende Ergebnisse liefert, was für Ihre Zwecke jedoch unerheblich ist.

Die Anweisungen für Funktionszuordnungen werden in der Katalogsicht SYSCAT.FUNCMAPPINGS gespeichert.

Beim Erstellen einer Funktionszuordnung kann es vorkommen, dass sich die Rückgabewerte einer in der Datenquelle ausgewerteten Funktion von den Rückgabewerten einer in der föderierten Datenbank ausgewerteten kompatiblen Funktion unterscheiden. IBM InfoSphere Federation Server verwendet dann zwar die Funktionszuordnung, die jedoch zu einem SQL-Syntaxfehler oder unerwarteten Ergebnissen führen kann.

Benutzerdefinierte Funktionen (UDFs) in Anwendungen

Anwendungsentwickler müssen häufig eine eigene, speziell auf ihre Anwendung oder ihre Domäne abgestimmte Funktionsgruppe erstellen. Dabei können sie benutzerdefinierte Skalarfunktionen verwenden.

Ein Einzelhandelsunternehmen könnte beispielsweise für die Überwachung der Preise der verkauften Waren einen Datentyp PRICE definieren. Möglicherweise möchte dieses Unternehmen auch eine Funktion SALES_TAX definieren. Diese Funktion würde einen gegebenen Wert für einen Preis als Eingabe verwenden, die entsprechende Verkaufssteuer berechnen und diese Daten an den anfordernden Benutzer oder die anfordernde Anwendung zurückgeben.

Diese Funktionen sind auf alle Datenbanktypen anwendbar, einschließlich LOB-Datentypen und einzigartige Datentypen. Benutzerdefinierte Funktionen erlauben in Abfragen leistungsfähige Berechnungs- und Suchvergleichselemente für die Filterung von irrelevanten Daten in der Umgebung der Datenquelle, wodurch die Antwortzeit reduziert wird. Das SQL-Optimierungsprogramm behandelt benutzerdefinierte Funktionen genauso wie integrierte Funktionen, wie z. B. SUBSTR und LENGTH. Anwendungen können in unterschiedlichen Umgebungen für Anwendungssprachen wie z. B. C, C++ und COBOL entwickelt werden. Die Anwendungen können eine Reihe benutzerdefinierter SQL-Funktionen gemeinsam benutzen, obwohl sie in unterschiedlichen Umgebungen für Anwendungssprachen entwickelt wurden.

Benutzerdefinierte Funktionen können Daten bearbeiten und Aktionen ausführen. Eine benutzerdefinierte Funktion kann z. B. für das Versenden einer elektronischen Nachricht oder das Aktualisieren einer Flachdatei aktiviert werden.

In DB2 können benutzerdefinierte Funktionen folgende Funktionen einschließen:

- Neu definierte Funktionen
- Im Schema SYSFUN enthaltene Funktionen. Die Beispiele enthalten mathematische Funktionen wie SIN, COS und TAN, wissenschaftliche Funktionen wie RADIANS, LOG10 und POWER sowie allgemeine Funktionen wie LEFT, DIFFERENCE und UCASE.

Voraussetzungen für die Zuordnung von benutzerdefinierten Funktionen

Bevor Sie eine benutzerdefinierte Datenquellenfunktion in einem föderierten System aufrufen können, muss die föderierte Datenbank die Datenquellenfunktion einer Funktionsspezifikation zuordnen, die im globalen Katalog auf dem Server gespeichert ist.

Es gibt zwei Bedingungen, unter denen die föderierte Datenbank eine Funktionspezifikation einer Datenquellenfunktion zuordnen kann:

- Die föderierte Datenbank enthält eine Funktion, deren Kennung der Kennung der Datenquellenfunktion entspricht. Eine *Kennung* setzt sich aus dem Funktionsnamen und den Funktionseingabeparametern zusammen. Kennungen *entsprechen einander*, wenn die beiden folgenden Bedingungen erfüllt sind:
 - Sie enthalten dieselben Namen und dieselbe Anzahl von Parametern.
 - Der Datentyp jedes Parameters in der einen Kennung stimmt mit dem Datentyp des entsprechenden Parameters in der anderen Kennung überein (oder kann in diesen umgesetzt werden).
- Wenn die föderierte Datenbank keine Funktion mit der erforderlichen Kennung enthält, können Sie eine Funktionsschablone definieren, die diese Kennung enthält. Ordnen Sie anschließend die Funktionsschablone der Datenquellenfunktion zu, die Sie aufrufen wollen.

Die Anweisungen für Funktionszuordnungen werden in der Katalogsicht SYSCAT.FUNCMAPPINGS gespeichert.

Erstellen von Funktionszuordnungen

Verwenden Sie die Anweisung CREATE FUNCTION MAPPING, um alternative Funktionszuordnungen anzugeben, die die Standardfunktionszuordnungen außer Kraft setzen.

Wenn Sie alternative Funktionszuordnungen erstellen, werden die Einträge in der Katalogsicht SYSCAT.FUNCMAPPINGS angezeigt.

Sie können die Anweisung CREATE FUNCTION MAPPING auch zum Angeben von Funktionszuordnungsoptionen verwenden. Wenn Sie Funktionszuordnungsoptionen angeben, werden die Informationen in der Katalogsicht SYSCAT.FUNCMAPOPTIONS angezeigt.

Mit der Anweisung CREATE FUNCTION MAPPING können Sie Folgendes ausführen:

- Erstellen einer Funktionszuordnung für alle Datenquellen eines bestimmten Typs. Dies können beispielsweise alle Datenquellen von Informix sein.
- Erstellen einer Funktionszuordnung für alle Datenquellen eines bestimmten Typs und einer bestimmten Version. Dies können beispielsweise alle Datenquellen von Informix 9 sein.
- Erstellen einer Funktionszuordnung für einen spezifischen Server.
- Bereitstellen von statistischen Informationen zu Funktionszuordnungen für das Optimierungsprogramm.
- Inaktivieren einer Standardfunktionszuordnung oder einer benutzerdefinierten Funktionszuordnung.

Sie können die Anweisung CREATE FUNCTION MAPPING im Befehlszeilenprozessor eingeben. Sie können die Anweisung CREATE FUNCTION MAPPING auch in ein Anwendungsprogramm einbetten. Die DB2-Steuerzentrale unterstützt das Erstellen oder Ändern von Funktionszuordnungen nicht.

Angeben von Funktionsnamen in der Anweisung CREATE FUNCTION MAPPING

Die Werte, die Sie in der Anweisung CREATE FUNCTION MAPPING angeben, hängen davon ab, ob die Funktionen, die Sie einander zuordnen, identische oder unterschiedliche Namen haben.

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Zuordnen von Funktionen mit identischem Namen

Sie können eine Zuordnung zwischen zwei Funktionen (oder zwischen einer DB2-Funktionsschablone und einer Datenquellenfunktion) mit demselben Namen erstellen.

Vorgehensweise

Um zwei Funktionen mit identischem Namen zuzuordnen, setzen Sie die Anweisung CREATE FUNCTION MAPPING ab:

Beispiel: Sie möchten eine benutzerdefinierte Funktion mit dem Namen MYFUN in einer Informix-Datenquelle der benutzerdefinierten DB2-Funktion TINA.MYFUN zuordnen. Der Informix-Datenquellenserver hat den Namen INFORMIX2. Mit der folgenden Anweisung wird die Funktion zugeordnet:

```
CREATE
FUNCTION MAPPING FOR TINA.MYFUN(SYSTEM.INTEGER) SERVER
INFORMIX2
```

Zuordnen von Funktionen mit unterschiedlichen Namen

Sie können eine Zuordnung zwischen zwei Funktionen (oder zwischen einer DB2-Funktionsschablone und einer Datenquellenfunktion) mit unterschiedlichen Namen erstellen.

Um eine Zuordnung zwischen zwei Funktionen mit unterschiedlichen Namen zu erstellen, setzen Sie die Anweisung CREATE FUNCTION MAPPING ab:

1. Ordnen Sie den Namen der DB2-Funktion bzw. der Funktionsschablone dem Parameter 'function_name' für den Funktionsnamen zu.
2. Geben Sie eine Funktionszuordnungsoption mit dem Namen REMOTE_NAME an, und ordnen Sie den Namen der Datenquellenfunktion dieser Option zu. REMOTE_NAME muss weniger als 255 Zeichen lang sein.

Beispiel: Sie möchten eine benutzerdefinierte Funktion mit dem Namen UPPER-CASE in einer Oracle-Datenquelle der DB2-Funktion UCASE(CHAR) zuordnen. Der Oracle-Datenquellenserver hat den Namen ORACLE2. Sie entscheiden sich dazu, diese Funktionszuordnung ORACLE_UPPER zu nennen. Die Syntax dann lautet wie folgt:

```
CREATE FUNCTION MAPPING ORACLE_UPPER FOR SYSFUN.UCASE(CHAR)
SERVER ORACLE2 OPTIONS
(REMOTE_NAME 'UPPERCASE')
```

Erstellen einer Funktionszuordnung für einen bestimmten Datenquellentyp

Sie können eine Funktionszuordnung für alle Datenquellen eines bestimmten Typs erstellen.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Einschränkungen

Für nicht relationale Datenquellen können Sie keine neuen Funktionszuordnungen erstellen, und Sie können die vorhandenen Funktionszuordnungen nicht außer Kraft setzen.

Vorgehensweise

Um eine DB2-Funktionsschablone einer Datenquellenfunktion zuzuordnen, verwenden Sie die Anweisung CREATE FUNCTION MAPPING.

Beispiel: Zuordnen einer DB2-Funktionsschablone zu einer benutzerdefinierten Oracle-Funktion für alle Oracle-Datenquellen

```
CREATE
FUNCTION MAPPING MY_ORACLE_FUN1
  FOR
NOVA.STATS
( DOUBLE, DOUBLE )
  SERVER TYPE
ORACLE
  OPTIONS (REMOTE_NAME
'STAR.STATISTICS')
```

Die Schablone hat den Namen STATS und gehört zu einem Schema mit dem Namen NOVA. Die benutzerdefinierte Oracle-Funktion hat den Namen STATISTICS und gehört zu einem Schema mit dem Namen STAR.

Erstellen einer Funktionszuordnung für einen bestimmten Datenquellentyp und eine bestimmte Datenquellentypversion

Sie können eine Funktionszuordnung für alle Datenquellen erstellen, die eine bestimmte Version des Datenquellentyps verwenden.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Einschränkungen

Für nicht relationale Datenquellen können Sie keine neuen Funktionszuordnungen erstellen, und Sie können die vorhandenen Funktionszuordnungen nicht außer Kraft setzen.

Vorgehensweise

Um eine Zuordnung für einen bestimmten Datenquellentyp und eine bestimmte Datenquellentypversion zu erstellen, verwenden Sie die Anweisung CREATE FUNCTION MAPPING.

Beispiel: Zuordnen einer DB2-Funktionsschablone zu einer benutzerdefinierten Sybase-Funktion für alle Sybase-Datenquellen, die Version 15 verwenden

Die Schablone hat den Namen SYB_STATS und gehört zu einem Schema mit dem Namen EARTH. Die benutzerdefinierte Sybase-Funktion hat den Namen STATISTICS und gehört zu einem Schema mit dem Namen MOON. Die Anweisung CREATE FUNCTION MAPPING lautet wie folgt:

```
CREATE FUNCTION MAPPING SYBASE_STATS
  FOR EARTH.SYB_STATS ( DOUBLE, DOUBLE )
  SERVER TYPE SYBASE VERSION 15
  OPTIONS (REMOTE_NAME
'MOON.STATISTICS')
```

Erstellen einer Funktionszuordnung für alle Datenquellenobjekte auf einem bestimmten Server

Sie können eine Funktionszuordnung für alle Datenquellenobjekte erstellen, die sich auf einem bestimmten fernen Server befinden.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Einschränkungen

Für nicht relationale Datenquellen können Sie keine neuen Funktionszuordnungen erstellen, und Sie können die vorhandenen Funktionszuordnungen nicht außer Kraft setzen.

Vorgehensweise

Um eine Funktionszuordnung für alle Datenquellenobjekte auf einem bestimmten Server zu erstellen, verwenden Sie die Anweisung CREATE FUNCTION MAPPING.

Beispiel: Zuordnen einer Funktionsschablone mit dem Namen BONUS zu einer benutzerdefinierte Funktion mit dem Namen BONUS

Sie wollen, dass die Funktion nur auf den Oracle-Datenquellenserver ORA_SALES angewendet wird. Da die Funktionsnamen identisch sind, müssen Sie die Funktionszuordnungsoption REMOTE_NAME nicht angeben.

```
CREATE FUNCTION MAPPING BONUS_CALC FOR BONUS()
  SERVER ORA_SALES
```

Funktionsschablonen

Der Server mit föderierten Datenbanken erkennt eine Datenquellenfunktion, wenn eine Zuordnung zwischen der Datenquellenfunktion und einer entsprechenden DB2-Funktion in der föderierten Datenbank vorhanden ist.

Ist keine entsprechende DB2-Funktion vorhanden, können Sie eine Funktionsschablone erstellen, die als Entsprechung fungiert.

Eine *Funktionsschablone* ist eine DB2-Funktion, die Sie erstellen, um das Aufrufen einer Datenquellenfunktion durch den Server mit föderierten Datenbanken zu erzwingen. Eine Funktionsschablone hat jedoch im Gegensatz zu einer regulären Funktion keinen ausführbaren Code. Empfängt der Server mit föderierten Datenbanken Abfragen, bei denen die Funktionsschablone angegeben wird, ruft er die Datenquellenfunktion auf.

Die Funktionsschablone wird mit der Anweisung `CREATE FUNCTION` unter Verwendung des Parameters `AS TEMPLATE` erstellt.

Nach der Erstellung einer Funktionsschablone muss die Funktionszuordnung zwischen der Schablone und der Datenquellenfunktion erstellt werden. Eine Funktionszuordnung wird mit der Anweisung `CREATE FUNCTION MAPPING` erstellt.

Erstellen von Funktionsschablonen

Der Server mit föderierten Datenbanken erkennt eine Datenquellenfunktion, wenn eine Zuordnung zwischen der Datenquellenfunktion und einer entsprechenden Funktion in der föderierten Datenbank besteht. Ist keine entsprechende Funktion vorhanden, können Sie stattdessen eine Funktionsschablone erstellen.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung `SYSADM` oder `DBADM`
- Berechtigung `IMPLICIT_SCHEMA` für die Datenbanken, wenn der implizite oder der explizite Schemaname der Funktion nicht vorhanden ist
- Zugriffsrecht `CREATEIN` für das Schema, wenn der Schemaname der Funktion vorhanden ist

Einschränkungen

Wenn die Datenquellenfunktion über Eingabeparameter verfügt:

- Die entsprechende Funktion in DB2 muss dieselbe Anzahl Eingabeparameter haben wie die Datenquellenfunktion.
- Die Datentypen der Eingabeparameter für die entsprechende DB2-Funktion müssen mit den zugehörigen Datentypen der Eingabeparameter für die Datenquellenfunktion kompatibel sein. Der Datentyp kann nicht `LONG VARCHAR`, `LONG VARGRAPHIC` oder ein benutzerdefinierter Typ sein.

Wenn die Datenquellenfunktion über keine Eingabeparameter verfügt, kann auch die entsprechende DB2-Funktion über keine Eingabeparameter verfügen.

Vorgehensweise

Um eine Funktionsschablone zu erstellen, gehen Sie wie folgt vor:

1. Verwenden Sie die Anweisung `CREATE FUNCTION` mit dem Parameter `AS TEMPLATE`. Beispiel:

```
CREATE FUNCTION BONUS ()  
  RETURNS DECIMAL(8,2)  
  AS TEMPLATE  
  DETERMINISTIC  
  NO EXTERNAL ACTION
```

BONUS ()

Der Name, den Sie der Funktionsschablone geben.

RETURNS *DECIMAL(8,2)*

Der Datentyp der Ausgabe.

AS TEMPLATE

Gibt an, dass es sich hierbei um eine Funktionsschablone und nicht um eine Funktion handelt.

DETERMINISTIC

Gibt an, dass die Funktion immer dieselben Ergebnisse für eine bestimmte Argumentwertmenge zurückgibt.

NO EXTERNAL ACTION

Gibt an, dass die Funktion keine externen Auswirkungen auf Objekte hat, die nicht vom Datenbankmanager verwaltet werden.

In Abhängigkeit davon, ob die Funktion selbst deterministisch ist und ob sie externe Aktionen auslöst, müssen Sie die Klauseln `DETERMINISTIC` und `NO EXTERNAL ACTION` angeben. Andernfalls werden den SQL-Operationen, die von dieser Funktionsschablone unterstützt werden, Einschränkungen auferlegt.

2. Nach der Erstellung einer Funktionsschablone muss die Funktionszuordnung zwischen der Schablone und der Datenquellenfunktion erstellt werden. Eine Funktionszuordnung wird mit der Anweisung `CREATE FUNCTION MAPPING` erstellt. Beispiel:

```
CREATE  
FUNCTION MAPPING  
MY_INFORMIX_FUN  
FOR BONUS()  
  SERVER TYPE INFORMIX OPTIONS (REMOTE_NAME 'BONUS()')
```

MY_INFORMIX_FUN

Der Name, den Sie der Funktionszuordnung geben. Der Name darf nicht mit dem Namen einer im DB2-Katalog bereits beschriebenen Funktionszuordnung identisch sein. Er muss eindeutig sein.

FOR *BONUS*()

Der Name der lokalen DB2-Funktionsschablone. Schließen Sie die Parameter für die Datentypeingabe in runde Klammern ein.

SERVER TYPE *INFORMIX*

Gibt den Datenquellentyp an, der die Funktion enthält, zu der die Zuordnung erfolgen soll.

OPTIONS (*REMOTE_NAME* '*BONUS*()')

Eine Option für die Angabe des Namens der fernen Datenquellenfunktion, die der lokalen DB2-Funktionsschablone zugeordnet wird.

Inaktivieren einer Standardfunktionszuordnung

Standardfunktionszuordnungen können nicht gelöscht werden. Sie können diese Zuordnungen jedoch "funktionsunfähig" machen, indem Sie sie inaktivieren.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Vorgehensweise

Beim Inaktivieren einer Standardfunktionszuordnung gibt die Anweisung CREATE FUNCTION MAPPING den Namen der DB2-Funktion an und setzt die Option DISABLE auf 'Y'.

Beispiel: Inaktivieren einer Standardfunktionszuordnung zwischen der DB2-Funktion SIN und einer ähnlichen Funktion für Oracle-Datenquellen

Wenn eine Abfrage verarbeitet wird, die Oracle-Daten anfordert und auf SIN verweist, können beide Funktionen aufgerufen werden. Welche Funktion aufgerufen wird, hängt davon ab, für welche Funktion die Schätzung des Abfrageoptimierungsprogramms einen geringeren Bedarf an Systemaufwand ergibt.

Um sicherzustellen, dass die DB2-Funktion SIN und nicht die Oracle-Funktion SIN aufgerufen wird, müssen Sie die Standardfunktionszuordnung inaktivieren. Verwenden Sie die folgende Syntax:

```
CREATE FUNCTION MAPPING FOR SYSFUN.SIN(INT)
TYPE ORACLE OPTIONS (DISABLE 'Y')
```

Löschen einer Funktionszuordnung

Wenn Sie eine selbst erstellte Funktionszuordnung nicht mehr benötigen, können Sie diese durch Ausführen der Anweisung DROP löschen.

Vorbereitung

Die Zugriffsrechte der Berechtigungs-ID der Anweisung müssen die Berechtigung SYSADM oder DBADM umfassen.

Informationen zu dieser Task

Wenn Sie eine Funktionszuordnung löschen, mit der eine Standardfunktionszuordnung außer Kraft gesetzt wurde, wird wieder die Standardfunktionszuordnung verwendet. Die Funktionszuordnungen sind in der Katalogsicht SYSCAT.FUNCTION MAPPINGS aufgelistet.

Vorgehensweise

Verwenden Sie zum Löschen einer selbst erstellten Funktionszuordnung die Anweisung DROP:

Beispiel: Löschen einer Funktionszuordnung mit dem Namen BONUS_CALC

```
DROP
FUNCTION MAPPING BONUS_CALC
```

Kapitel 5. Erstellen von Indexspezifikationen

Erstellen Sie für relationale Datenquellen Indexspezifikationen, um Informationen über die Spalten von fernen Indizes im globalen Katalog zu speichern und eine optimale Leistung bei Suchabfragen sicherzustellen.

Indexspezifikationen in einem föderierten System

In einem föderierten System verwenden Sie die Anweisung `CREATE INDEX` mit einem Kurznamen, um im globalen Katalog Informationen über die Verfügbarkeit eines Index für das ferne Objekt zu speichern. Das Abfrageoptimierungsprogramm verwendet diese Informationen zum Optimieren von Abfragen.

Beim Absetzen einer Anweisung `CREATE INDEX` gilt:

- Wird ein Kurzname für eine Tabelle erstellt, erfasst die Anweisung `CREATE INDEX` Indexierungsinformationen über den Index, der für die ferne Tabelle erstellt wurde.
- Wird ein Kurzname für eine Sicht erstellt, verweist die Anweisung `CREATE INDEX` auf den Kurznamen für die Sicht und enthält Informationen über den Index für die Tabelle, die der Sicht zugrunde liegt.

Die Indexspezifikation teilt dem Server mit föderierten Datenbanken Informationen über die Spalten und deren Eindeutigkeitsmerkmale mit, auf deren Basis ein ferner Index erstellt wird. Informationen in den Statistikmerkmalen des Indexes (z. B. zur Anzahl der eindeutigen Werte des Indexschlüssels) werden in dieser Spezifikation hingegen nicht an den Server mit föderierten Datenbanken übergeben.

Sie müssen keine Indexspezifikationen angeben, wenn der ferne Index zum Zeitpunkt der Erstellung des Kurznamens bereits definiert war.

Ein Server mit föderierten Datenbanken erzeugt bei der Erstellung von Kurznamen für folgende Objekte keine Indexspezifikation:

- Eine Tabelle ohne Indizes
- Eine Sicht, für die normalerweise keine Indexinformationen im fernen Katalog gespeichert sind
- Ein Datenquellenobjekt, das über keinen fernen Katalog verfügt, von dem der Server mit föderierten Datenbanken die Indexinformationen abrufen kann

Angenommen, eine Tabelle erhält zusätzlich zu den zum Zeitpunkt der Erstellung des Kurznamens vorhandenen Indizes einen neuen Index. Da die Indexinformationen zum Zeitpunkt der Erstellung des Kurznamens an den globalen Katalog gegeben werden, bemerkt der Server mit föderierten Datenbanken den neuen Index nicht. Ebenso bemerkt der Server beim Erstellen eines Kurznamens für eine Sicht nicht die zugrunde liegende Tabelle (und deren Indizes), von der die Sicht erstellt wurde. Unter diesen Umständen können Sie dem globalen Katalog die erforderlichen Indexinformationen zur Verfügung stellen. Sie können eine Indexspezifikation für Tabellen ohne Indizes erstellen. Mithilfe der Indexspezifikation kann das Abfrageoptimierungsprogramm feststellen, welche Tabellenspalte(n) durchsucht werden muss bzw. müssen, um Daten schnell zu finden.

Verwenden Sie Indexspezifikationen mit relationalen Datenquellen. Bei nicht relationalen Datenquellen bewirkt das Erstellen einer Indexspezifikation keine Leistungssteigerung.

Erstellen von Indexspezifikationen für Datenquellenobjekte

Wenn für eine Datenquellentabelle ein Kurzname erstellt wird, stellt der Server mit föderierten Datenbanken für den globalen Katalog Informationen über alle Indizes der Datenquellentabelle zur Verfügung. Das Optimierungsprogramm verwendet diese Informationen zur Beschleunigung der Verarbeitung von verteilten Anforderungen. Diese Informationen bestehen aus einem Satz von Metadaten und werden als *Indexspezifikation* bezeichnet.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung SYSADM oder DBADM
- Entweder das Zugriffsrecht CONTROL oder das Zugriffsrecht INDEX für das Objekt. Außerdem muss die ID entweder über die Berechtigung IMPLICIT_SCHEMA für die Datenbank verfügen, wenn der implizite oder der explizite Schemaname des Indexes nicht vorhanden ist, oder über das Zugriffsrecht CREATEIN für das Schema, wenn sich der Schemaname des Indexes auf ein bestehendes Schema bezieht.

Einschränkungen

Bei der Erstellung einer Indexspezifikation für einen Kurznamen gibt es einige Einschränkungen.

- Wenn die Bindeoption DYNAMICRULES BIND zutrifft, kann die Anweisung nicht dynamisch vorbereitet werden. Außerdem können Sie die Parameter INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS und ALLOW REVERSE SCANS in der Anweisung CREATE INDEX nicht verwenden.
- UNIQUE sollte nur dann angegeben werden, wenn die Daten für den Indexschlüssel für jede Zeile der Datenquellentabelle eindeutige Werte enthalten. Die Eindeutigkeit wird nicht überprüft.
- Die Summe der gespeicherten Längen der angegebenen Spalten darf nicht größer als 1024 sein.
- Keine LOB-Spalte bzw. keine Spalte eines einzigartigen Datentyps auf der Basis von LOB kann für einen Index verwendet werden. Diese Einschränkung wird auch dann durchgesetzt, wenn das Längenattribut der Spalte klein genug ist, damit die Begrenzung von 1024 Byte nicht überschritten wird.

Informationen zu dieser Task

Der Server mit föderierten Datenbanken erstellt in folgenden Fällen keine Indexspezifikationen:

- Ein Kurzname wird für eine Tabelle erstellt, die keinen Index hat.
- Ein Kurzname wird für ein Datenquellenobjekt erstellt, das keine Indizes enthält, wie beispielsweise eine Sicht oder ein Informix-Synonym.
- Ein Kurzname wird für ein nicht relationales Objekt erstellt, wie beispielsweise eine Datei mit Tabellenstruktur, ein Excel-Arbeitsblatt oder eine XML-Datei.

- Der ferne Index besteht für eine LOB- oder eine XML-Spalte.
- Der ferne Index enthält eine Gesamtschlüssellänge von mehr als 1024 Byte.
- Die maximale Anzahl an Schlüsselkomponenten beträgt mehr als 16.

Unter diesen Umständen speichert der Server mit föderierten Datenbanken keine Indexspezifikationen für die Datenquellenobjekte. Allerdings können Sie dem globalen Katalog die erforderlichen Indexinformationen für die ersten beiden Einträge in der vorherigen Liste zur Verfügung stellen. Mithilfe der Anweisung CREATE INDEX können Sie die Indexinformationen angeben.

Vorgehensweise

Um einen Index zu erstellen, können Sie die Anweisung CREATE INDEX in ein Anwendungsprogramm integrieren oder als dynamische SQL-Anweisung über die Steuerzentrale oder die Befehlszeile absetzen.

Wenn die Anweisung CREATE INDEX mit Kurznamen verwendet wird, erstellt sie im globalen Katalog der föderierten Datenbanken eine Indexspezifikation. Für die Datenquellentabelle erstellt sie keinen Index.

Verwenden Sie die folgende Syntax, um eine Indexspezifikation zu erstellen:

```
CREATE
INDEX indexname
ON kurzname
(spaltenname) SPECIFICATION ONLY
CREATE UNIQUE INDEX indexname ON nickname
(spaltenname DESC)
SPECIFICATION ONLY
```

Bei einer Indexspezifikation ist *spaltenname* der Name, mit dessen Hilfe der Server mit föderierten Datenbanken auf eine Spalte einer Datenquellentabelle verweist.

Erstellen von Indexspezifikationen für Tabellen, die neue Indizes erhalten

Für den Fall, dass eine Tabelle einen neuen Index erhält, sollten Sie eine Indexspezifikation für den Kurznamen erstellen, der der Tabelle entspricht.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung SYSADM oder DBADM
- Entweder das Zugriffsrecht CONTROL oder das Zugriffsrecht INDEX für das Objekt. Außerdem muss die ID entweder über die Berechtigung IMPLICIT_SCHEMA für die Datenbank verfügen, wenn der implizite oder der explizite Schemaname des Indexes nicht vorhanden ist, oder über das Zugriffsrecht CREATEIN für das Schema, wenn sich der Schemaname des Indexes auf ein bestehendes Schema bezieht.

Einschränkungen

Bei der Erstellung eines Indexes für einen Kurznamen gibt es einige Einschränkungen.

- Wenn die Bindeoption DYNAMICRULES BIND zutrifft, kann die Anweisung nicht dynamisch vorbereitet werden. Außerdem können Sie die Parameter INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS und ALLOW REVERSE SCANS in der Anweisung CREATE INDEX nicht verwenden.
- UNIQUE sollte nur dann angegeben werden, wenn die Daten für den Indexschlüssel für jede Zeile der Datenquellentabelle eindeutige Werte enthalten. Die Eindeutigkeit wird nicht überprüft.
- Die Summe der gespeicherten Längen der angegebenen Spalten darf nicht größer als 1024 sein.
- Keine LOB-Spalte bzw. keine Spalte eines einzigartigen Datentyps auf der Basis von LOB kann für einen Index verwendet werden. Diese Einschränkung wird auch dann durchgesetzt, wenn das Längenattribut der Spalte klein genug ist, damit die Begrenzung von 1024 Byte nicht überschritten wird.

Informationen zu dieser Task

Es gibt mehrere Situationen, in denen eine Tabelle einen neuen Index erhält:

- Sie erstellen einen Kurznamen für eine Tabelle, die keinen Index hat, zu einem späteren Zeitpunkt jedoch einen Index erhält.
- Sie erstellen einen Kurznamen für eine Tabelle, die einen Index hat, zu einem späteren Zeitpunkt jedoch einen anderen Index erhält.

In solchen Situationen sollten Sie eine Indexspezifikation für die Tabelle erstellen, sodass der SQL-Compiler diese Informationen bei der Verarbeitung von Abfragen, die auf diese Tabelle verweisen, verwenden kann.

Vorgehensweise

Anhand der folgenden Beispiele wird beschrieben, wie eine Indexspezifikation für einen Kurznamen erstellt wird, der einer Tabelle entspricht, die einen neuen Index erhält.

Beispiel: Eine Tabelle hat keinen Index, erhält aber später einen Index

Angenommen, Sie erstellen den Kurznamen *EMPLOYEE* für eine Datenquellentabelle mit dem Namen *CURRENT_EMP*, die keine Indizes hat. Einige Zeit nach der Erstellung dieses Kurznamens wird für *CURRENT_EMP* unter Verwendung der Spalten *WORKDEPT* und *JOB* für den Indexschlüssel ein Index definiert.

Die Syntax zur Erstellung einer Indexspezifikation, die diesen Index beschreibt, lautet wie folgt:

```
CREATE UNIQUE INDEX JOB_BY_DEPT ON EMPLOYEE
(WORKDEPT, JOB) SPECIFICATION ONLY
```

Hierbei gilt: *JOB_BY_DEPT* ist der Indexname.

Beispiel: Eine Tabelle erhält einen neuen Index

Angenommen, Sie erstellen den Kurznamen *JP_SALES* für eine Tabelle mit dem Namen *JAPAN_SALES*. Der Tabelle wird später zusätzlich zu den Indizes, die sie zum Zeitpunkt der Erstellung des Kurznamens hatte, ein neuer Index hinzugefügt. Der neue Index verwendet die Spalte *MARKUP* für den Indexschlüssel.

Die Syntax zur Erstellung einer Indexspezifikation, die diesen Index beschreibt, lautet wie folgt:

```
CREATE UNIQUE INDEX  
JP_MARKUP  
ON JP_SALES  
(MARKUP)  
SPECIFICATION ONLY
```

Hierbei gilt: *JP_MARKUP* ist der Indexname.

Erstellen von Indexspezifikationen für Sichten

Der Server mit föderierten Datenbanken bemerkt beim Erstellen eines Kurznamens für eine Sicht nicht die zugrunde liegende Tabelle (und deren Indizes), aus der die Sicht generiert wurde. Erstellen Sie eine Indexspezifikation für die Sicht, sodass der SQL-Compiler diese Informationen bei der Verarbeitung von Abfragen, die auf diese Sicht verweisen, verwenden kann.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung SYSADM oder DBADM
- Entweder das Zugriffsrecht CONTROL oder das Zugriffsrecht INDEX für das Objekt. Außerdem muss die ID entweder über die Berechtigung IMPLICIT_SCHEMA für die Datenbank verfügen, wenn der implizite oder der explizite Schemaname des Indexes nicht vorhanden ist, oder über das Zugriffsrecht CREATEIN für das Schema, wenn sich der Schemaname des Indexes auf ein bestehendes Schema bezieht.

Einschränkungen

Bei der Erstellung eines Indexes für einen Kurznamen gibt es einige Einschränkungen.

- Wenn die Bindeoption DYNAMICRULES BIND zutrifft, kann die Anweisung nicht dynamisch vorbereitet werden. Außerdem können Sie die Parameter INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS und ALLOW REVERSE SCANS in der Anweisung CREATE INDEX nicht verwenden.
- UNIQUE sollte nur dann angegeben werden, wenn die Daten für den Indexschlüssel für jede Zeile der Datenquellentabelle eindeutige Werte enthalten. Die Eindeutigkeit wird nicht überprüft.
- Die Summe der gespeicherten Längen der angegebenen Spalten darf nicht größer als 1024 sein.
- Keine LOB-Spalte bzw. keine Spalte eines einzigartigen Datentyps auf der Basis von LOB kann für einen Index verwendet werden. Diese Einschränkung wird auch dann durchgesetzt, wenn das Längenattribut der Spalte klein genug ist, damit die Begrenzung von 1024 Byte nicht überschritten wird.

Vorgehensweise

Um eine Indexspezifikation für eine Sicht zu erstellen, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass die Spalte(n), auf der/denen der Tabellenindex basiert, Teil der Sicht ist/sind.

- Wenn Sie für alle Indizes der zugrunde liegenden Tabelle Indexspezifikationen erstellen möchten, muss jede Indexspezifikation separat erstellt werden.

Beispiel: Erstellen einer Indexspezifikation, die den Index REGION beschreibt

Angenommen, Sie erstellen den Kurznamen *JP_SALES2007* für eine Sicht mit dem Namen *JAPAN_SALES2007*. Die zugrunde liegende Tabelle für diese Sicht ist die Tabelle *JAPAN_SALES*, die mehrere Indizes enthält: *REGION*, *AMOUNT*, *SALES_REP*. Die Anweisung *CREATE INDEX*, die Sie erstellen, verweist auf den Kurznamen für die Sicht und enthält Informationen zum Index der zugrunde liegenden Tabelle für die Sicht.

```
CREATE UNIQUE INDEX JP_2007_REGION ON JP_SALES2007
(REGION) SPECIFICATION ONLY
```

Hierbei gilt: *JP_2007_REGION* ist der Indexname, und *JP_SALES2007* ist der Kurzname für die Sicht *JAPAN_SALES2007*.

Erstellen von Indexspezifikationen für Informix-Synonyme

In diesem Abschnitt wird die Aktion beschrieben, die der Server mit föderierten Datenbanken für Informix-Synonyme ausführt, die auf einer Tabelle oder Sicht basieren.

Vorbereitung

Die Berechtigungs-ID der Anweisung muss zumindest über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung *SYSADM* oder *DBADM*
- Entweder das Zugriffsrecht *CONTROL* oder das Zugriffsrecht *INDEX* für das Objekt. Außerdem muss die ID entweder über die Berechtigung *IMPLICIT_SCHEMA* für die Datenbank verfügen, wenn der implizite oder der explizite Schemaname des Indexes nicht vorhanden ist, oder über das Zugriffsrecht *CREATEIN* für das Schema, wenn sich der Schemaname des Indexes auf ein bestehendes Schema bezieht.

Einschränkungen

Bei der Erstellung eines Indexes für einen Kurznamen gibt es einige Einschränkungen.

- Wenn die Bindeoption *DYNAMICRULES BIND* zutrifft, kann die Anweisung nicht dynamisch vorbereitet werden. Außerdem können Sie die Parameter *INCLUDE*, *CLUSTER*, *PCTFREE*, *MINPCTUSED*, *DISALLOW REVERSE SCANS* und *ALLOW REVERSE SCANS* in der Anweisung *CREATE INDEX* nicht verwenden.
- *UNIQUE* sollte nur dann angegeben werden, wenn die Daten für den Indexschlüssel für jede Zeile der Datenquellentabelle eindeutige Werte enthalten. Die Eindeutigkeit wird nicht überprüft.
- Die Summe der gespeicherten Längen der angegebenen Spalten darf nicht größer als 1024 sein.
- Keine *LOB*-Spalte bzw. keine Spalte eines einzigartigen Datentyps auf der Basis von *LOB* kann für einen Index verwendet werden. Diese Einschränkung wird auch dann durchgesetzt, wenn das Längenattribut der Spalte klein genug ist, damit die Begrenzung von 1024 Byte nicht überschritten wird.

Informationen zu dieser Task

In Informix können Sie für eine Tabelle oder Sicht ein Synonym erstellen. Der Server mit föderierten Datenbanken ermöglicht zwar die Erstellung von Kurznamen für Informix-Synonyme, welche Aktion der Server mit föderierten Datenbanken jedoch ausführt, hängt davon ab, ob das Synonym auf einer Tabelle oder auf einer Sicht basiert:

- Angenommen, für ein Synonym wird ein Kurzname erstellt, und das Synonym basiert auf einer Informix-Tabelle. Wenn der Server mit föderierten Datenbanken ermittelt, dass die Tabelle, auf die sich das Synonym bezieht, einen Index hat, wird für das Synonym eine Indexspezifikation erstellt. Wenn die Tabelle, auf die sich das Synonym bezieht, keinen Index hat, wird für das Synonym keine Indexspezifikation erstellt. Sie können jedoch unter Verwendung der Anweisung `CREATE INDEX` manuell eine Indexspezifikation erstellen.
- Angenommen, für ein Synonym wird ein Kurzname erstellt, und das Synonym basiert auf einer Informix-Sicht. Der Server mit föderierten Datenbanken kann nicht ermitteln, auf welcher/welchen zugrunde liegende(n) Tabelle(n) die Sicht basiert. Daher wird für das Synonym keine Indexspezifikation erstellt. Sie können jedoch unter Verwendung der Anweisung `CREATE INDEX` manuell eine Indexspezifikation erstellen.

Vorgehensweise

Anhand der folgenden Beispiele wird beschrieben, wie eine Indexspezifikation für einen Kurznamen erstellt wird, der einem Informix-Synonym entspricht.

Beispiel: Ein Kurzname wird für ein Informix-Synonym erstellt, das auf einer Tabelle basiert

Wenn das Synonym auf einer Informix-Tabelle basiert, die keinen Index enthält, können Sie für das Synonym eine Indexspezifikation erstellen, mit der dem Optimierungsprogramm mitgeteilt wird, welche Spalte(n) durchsucht werden muss/müssen, um Daten rasch zu finden. Die Anweisung, die Sie erstellen, gibt den Kurznamen für das Synonym an, und Sie stellen Informationen zu der Spalte/den Spalten in der Tabelle bereit, auf der/denen das Synonym basiert.

In diesem Beispiel erstellen Sie den Kurznamen *CONTRACTS* für ein Synonym mit dem Namen *SALES_CONTRACTS*. Die Tabelle, auf der dieses Synonym basiert, heißt *SALES2006_TABLE* und enthält mehreren Indizes: *REGION*, *AMOUNT*, *SALES_REP*. Die Anweisung `CREATE INDEX`, die Sie erstellen, verweist auf den Kurznamen für das Synonym und enthält Informationen zum Index der zugrunde liegenden Tabelle für das Synonym.

Die Syntax zur Erstellung einer Indexspezifikation, die den Index *REGION* beschreibt, lautet wie folgt:

```
CREATE UNIQUE INDEX NORTHWEST_2006_REGION ON CONTRACTS (REGION) SPECIFICATION ONLY
```

Hierbei gilt: *NORTHWEST_2006_REGION* ist der Indexname, und *CONTRACTS* ist der Kurzname für das Synonym *SALES_CONTRACTS*.

Beispiel: Ein Kurzname wird für ein Informix-Synonym erstellt, das auf einer Sicht basiert

Sie erstellen den Kurznamen *JP_SALES2007* für eine Sicht mit dem Namen *JAPAN_SALES2007*. Die zugrunde liegende Tabelle für diese Sicht ist die Tabelle

JAPAN_SALES, die mehrere Indizes enthält: REGION, AMOUNT, SALES_REP. Die Anweisung CREATE INDEX, die Sie erstellen, verweist auf den Kurznamen für das Synonym und enthält Informationen zum Index der zugrunde liegenden Tabelle für die Sicht.

Wenn Sie eine Indexspezifikation für ein Synonym erstellen, das auf einer Sicht basiert, müssen Sie sich vergewissern, dass die Spalte(n), auf der/denen der Tabellenindex basiert, Teil der Sicht ist/sind. Wenn Sie für alle Indizes der zugrunde liegenden Tabelle Indexspezifikationen erstellen möchten, muss jede Indexspezifikation separat erstellt werden.

Die Syntax zur Erstellung einer Indexspezifikation, die den Index REGION beschreibt, lautet wie folgt:

```
CREATE UNIQUE INDEX JP_2007_REGION ON JP_SALES2007 (REGION) SPECIFICATION ONLY
```

Hierbei gilt: *JP_2007_REGION* ist der Indexname, und *JP_SALES2007* ist der Kurzname für die Sicht JAPAN_SALES2007.

Kapitel 6. Entwickeln von föderierten Prozeduren

Mit föderierten Prozeduren können Sie Prozeduren in einer Datenquelle so aufrufen, also ob die ferne Prozedur eine lokale Prozedur wäre.

Föderierte Prozeduren

Eine *föderierte Prozedur* ist ein föderiertes Datenbankobjekt, das auf eine Prozedur in einer Datenquelle verweist.

Bei föderierten Prozeduren handelt es sich nicht um alternative Namen für Datenquellenprozeduren im Sinne von Aliasnamen. Eine föderierte Prozedur wird auf der föderierten Datenbank definiert; bei Aufruf der föderierten Prozedur wird jedoch eine Datenquellenprozedur aufgerufen. Da die föderierte Prozedur ein föderiertes Datenbankobjekt ist, können Benutzer und Clientanwendungen die Logik der Datenquellenprozedur durch Aufruf einer föderierten Prozedur aufrufen. Die Ergebnisse der Datenquellenprozedur, wie beispielsweise die Ausgabeparameter, werden von der föderierten Prozedur zurückgegeben. Durch die Verwendung einer föderierten Prozedur wird Benutzern und Clientanwendungen die Position der Datenquellenprozedur transparent gemacht. Zum Aufrufen der Datenquellenprozedur verwenden Sie den Namen der föderierten Prozedur.

Eine föderierte Prozedur ist für eine ferne Prozedur dasselbe wie ein Kurzname für eine ferne Tabelle. Kurznamen und föderierte Prozeduren sind Objekte auf der föderierten Datenbank. Ein Kurzname ist ein Objekt, das in der Datenquelle auf ein Objekt (z. B. eine Tabelle oder Sicht) verweist. Mit einem Kurznamen fragen Sie ein Datenquellenobjekt ab. Mit einer föderierten Prozedur rufen Sie eine Datenquellenprozedur auf.

Mithilfe der Anweisung `CREATE PROCEDURE` (Quellenfunktion) können Sie eine föderierte Prozedur registrieren, mithilfe der Anweisung `CALL` eine Prozedur aufrufen. Sie können die Anweisung `CREATE PROCEDURE` (Quellenfunktion) in ein Anwendungsprogramm einbetten oder die Anweisung mit dynamischen SQL-Anweisungen absetzen.

Parameter und Datentypen

Federation verwendet drei Parametertypen: `IN`, `OUT` und `INOUT`. Bei der Erstellung einer föderierten Prozedur werden die Datentypen für die Parameter der Datenquellenprozedur den föderierten Datentypen unter Verwendung der standardmäßigen progressiven Datentypzuordnungen zugeordnet. Diese Zuordnung umfasst Datentypen für die Eingabe- und Ausgabeparameter sowie Datentypen für die Spalten in der Ergebnismenge. Mit der Anweisung `CREATE TYPE MAPPING` kann die Standardtypzuordnung für die Datenquellenparameter überschrieben werden. Die Typzuordnungen der Ergebnismenge sind von benutzerdefinierten Typzuordnungen jedoch nicht betroffen. Sie können alle Datentypen verwenden, die für Kurznamenspalten unterstützt werden, mit Ausnahm von `LOB`-Datentypen. Komplexe Datentypen werden von föderierten Prozeduren nicht unterstützt.

Benutzerzuordnungen und Berechtigung

Zum Aufrufen einer föderierten Prozedur müssen Sie über die entsprechenden Berechtigungen für die föderierte Prozedur und die Datenquellenprozedur verfügen. Beim Aufrufen einer föderierten Prozedur werden die Benutzerzuordnung und die Zugriffsrechte der Berechtigungs-ID, die zur Erstellung der föderierten Prozedur verwendet wurden, für den Zugriff auf die Datenquellentabellen verwendet.

Beispiel: Der Benutzer ZELLER erstellt die föderierte Prozedur FP1. Die Prozedur FP1 verweist auf eine Sybase-Prozedur, die auf eine Sybase-Tabelle zugreift. Die ferne Benutzer-ID in der Benutzerzuordnung für ZELLER verfügt über die Berechtigung zum Aktualisieren der Sybase-Tabelle. Der Benutzer ZELLER erteilt dem Benutzer BHATIA das Zugriffsrecht EXECUTE für die Prozedur FP1. Der Benutzer BHATIA muss über eine gültige Benutzerzuordnung für eine ferne Benutzer-ID verfügen, die das Zugriffsrecht EXECUTE für die Sybase-Prozedur aufweist, auf die Prozedur FP1 verweist. Die ferne Benutzer-ID, der Benutzer BHATIA zugeordnet wird, muss nicht über das Zugriffsrecht SELECT für die Sybase-Prozedur verfügen. Wenn der Benutzer BHATIA die Prozedur FP1 aufruft, kann er die Tabelle in Sybase aktualisieren.

Prozeduraufrufe und Zugriffsebenen

Bei Prozeduraufrufen und Zugriffsebenen können die folgenden Probleme auftreten:

- Bei Ausgabe des Befehls PRECOMPILE wird für föderierte Prozeduren standardmäßig die Klausel CALL RESOLUTION IMMEDIATE verwendet. Die Klausel CALL RESOLUTION DEFERRED wird für föderierte Prozeduren nicht unterstützt.
- Wenn eine föderierte Prozedur eine Datenquellenprozedur aufruft und die Ausgabe in einen Puffer oder an die Standardausgabe erfolgt, wird keine Ausgabe angezeigt.
- Wenn Sie eine föderierte Prozedur über eine externe benutzerdefinierte Funktion aufrufen, darf als Zugriffsebene für die föderierte Prozedur nicht READS SQL DATA oder MODIFIES SQL DATA gesetzt sein. Der föderierte Zugriff wird innerhalb von externen Funktionen blockiert.
- Die Einschränkungen bezüglich der aufrufenden lokalen Prozeduren im Durchgriffsmodus gelten auch für föderierte Prozeduren.
- Jedes Argument in der Anweisung CALL muss mit dem entsprechenden Parameter in der Prozedur kompatibel sein. Für föderierte Prozeduren gelten dieselben Parameterzuordnungsregeln wie für lokale Prozeduren.

Transaktionen

Bei Verwendung von föderierten Prozeduren mit Transaktionen können die folgenden Probleme auftreten:

- Die Datenquellenprozedur, auf die die föderierte Prozedur verweist, darf keine Anweisung COMMIT oder ROLLBACK absetzen. Diese Einschränkung wird von Federation nicht umgesetzt. Deshalb kann es zu Dateninkonsistenzen kommen, wenn die Datenquellenprozedur eine Anweisung COMMIT oder ROLLBACK ausgibt.
- Föderierte Prozeduren mit der Zugriffsebene MODIFIES SQL DATA können innerhalb von Triggern, dynamischen Compound-Anweisungen, SQL-Skalarfunktionen, Tabellen, Zeilenfunktionen oder Methoden nicht aufgerufen werden.

Nach dem Absetzen einer Anweisung SAVEPOINT kann eine föderierte Prozedur mit der Zugriffsebene MODIFIES SQL DATA nicht aufgerufen werden.

- Mit WITH HOLD definierte Cursor, verschiebbare oder aktualisierbare Cursor werden von föderierten Prozeduren nicht unterstützt. Wenn die Datenquellenprozedur einen dieser Cursortypen verwendet, wird keine Warnung oder Fehlermeldung ausgegeben. Anwendungen, die mit Datenquellenprozeduren auf Ergebnismengen zugreifen, die diese Cursortypen verwenden, verhalten sich möglicherweise nicht erwartungsgemäß. Normalerweise werden Cursor ohne Holdfunktionalität zurückgegeben. Aktualisierbare Cursor werden als Nur-Lese-/Nur-Vorwärtscursor zurückgegeben.

Katalogsichten

In den folgenden Systemkatalogsichten werden Informationen zu föderierten Prozeduren gespeichert:

- SYSCAT.ROUTINES
- SYSCAT.ROUTINESFEDERATED
- SYSCAT.ROUTINEOPTIONS
- SYSCAT.ROUTINEPARMS
- SYSCAT.ROUTINEPARMOPTIONS

Datenquellen und föderierte Prozeduren

Lesen Sie vor der Erstellung föderierter Prozeduren die Informationen zur Unterstützung von Datenquellenprozeduren durch Federation.

Föderierte Prozeduren für DB2-Datenquellen

Lesen Sie vor der Erstellung einer föderierten Prozedur die Informationen zur Angabe von Optionen in der Anweisung CREATE PROCEDURE.

Bitte beachten Sie die folgenden Aspekte bei der Erstellung föderierter Prozeduren:

- DB2-Prozeduren unterstützen die Parameter IN, OUT und INOUT.
- Sie müssen dieselben Optionen angeben, die die DB2-Prozedur für die Klauseln SQL DATA ACCESS, DETERMINISTIC und EXTERNAL ACTION der Anweisung CREATE PROCEDURE angibt.
- Wenn zwei ferne DB2-Prozeduren denselben Namen haben, verwenden Sie die Option NUMBER OF PARAMETERS, um die Prozedur anzugeben, die Sie verwenden möchten.
- Die föderierte Datenbank setzt keine COMMIT-Anweisung für föderierte Prozeduren ab, die für Prozeduren in DB2 Universal Database für z/OS erstellt wurden und die die Klausel COMMIT ON RETURN YES enthalten.

Beispiel

In diesem Beispiel wird verdeutlicht, wie die Anweisung CREATE PROCEDURE zur Erstellung einer föderierten Prozedur für eine Datenquellenprozedur bei DB2 verwendet wird.

```
CREATE PROCEDURE PROC1 SOURCE KELLER.PROC1_DB2
    NUMBER OF PARAMETERS 3 FOR SERVER DB2_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Erforderlich. Gibt den Namen der föderierten Prozedur an.

SOURCE KELLER.PROC1_DB2

Erforderlich. Gibt das Schema und den Namen für die DB2-Prozedur an. Für DB2-Prozeduren wird in der Anweisung CREATE PROCEDURE ein zweiteiliger Name angegeben. Das Format für diesen zweiteiligen Namen ist *name_des_quellschemas.name_der_quellenprozedur*.

NUMBER OF PARAMETERS 3

Gibt die Gesamtzahl der von der DB2-Prozedur verwendeten IN-, OUT- und INOUT-Parameter an. Verwenden Sie diesen Parameter, wenn mehrere Prozeduren mit demselben Schema- und Prozedurnamen vorhanden sind. Wenn das Schema beispielsweise KELLER lautet und eine PROC1-Prozedur mit drei Parametern und eine weitere PROC1-Prozedur mit einem Parameter vorhanden ist, lautet der Name für beide Prozeduren KELLER.PROC1. Der Wert für NUMBER OF PARAMETERS in der Datenquellenprozedur gibt an, auf welche Prozedur in der Anweisung CREATE PROCEDURE verwiesen wird.

FOR SERVER DB2_SERVER

Erforderlich. Gibt eine Serverdefinition für die Erstellung der föderierten Prozedur an.

SPECIFIC MYPROC1

Gibt einen eindeutigen Namen für die zu erstellende föderierte Prozedur an. Dieser Parameter wird nur für föderierte Prozeduren verwendet und keinen Datenquellenprozeduren zugeordnet. Wenn Sie keinen eindeutigen Namen angeben, wird vom Datenbankmanager der föderierten Datenbank ein Name generiert.

WITH RETURN TO CLIENT ALL

Gibt an, dass die Ergebnismenge an die Clientanwendung zurückgegeben wird. Federation gibt maximal eine Ergebnismenge zurück. Wenn dieser Parameter nicht angegeben wird, lautet die Standardeinstellung WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Gibt die Stufe des Datenzugriffs für SQL-Anweisungen an, die in der föderierten Prozedur enthalten sind. Wenn die angegebene Klausel nicht mit der DB2-Prozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die DB2-Prozedur verwendet.

DETERMINISTIC

Gibt an, ob die föderierte Prozedur immer dieselben Ergebnisse für eine bestimmte Argumentwertmenge zurückgibt. Dieser Parameter kann die Interaktionsleistung zwischen dem Server mit föderierten Datenbanken und der Datenquelle verbessern. Wenn die angegebene Klausel nicht mit der DB2-Prozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die DB2-Prozedur verwendet.

EXTERNAL ACTION

Gibt an, ob die föderierte Prozedur eine Aktion ausführt, die den Status eines Objekts ändert, das nicht vom Datenbankmanager verwaltet wird. Wenn die angegebene Klausel nicht mit der DB2-Prozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die DB2-Prozedur verwendet.

Föderierte Prozeduren und Microsoft SQL Server

Lesen Sie vor der Erstellung einer föderierten Prozedur die Informationen dazu, welche Parameter unterstützt werden, wie Ergebnismengen zurückgegeben werden und welche Einschränkungen bestehen.

Parameter

Microsoft SQL Server-Prozeduren unterstützen die Verwendung optionaler INPUT- und OUTPUT-Parameter. Der SQL Server-Wrapper ordnet jeden INPUT-Parameter einem föderierten IN-Parameter und jeden OUTPUT-Parameter einem föderierten INOUT-Parameter zu. Wenn Sie optionale Parameter in einer SQL Server-Prozedur verwenden, müssen Sie sie zählen, wenn Sie die Klausel NUMBER OF PARAMETERS der Anweisung CREATE PROCEDURE angeben.

Ergebnismengen

Der SQL Server-Wrapper kann eine Ergebnismenge und einen Ausgabeparameter (OUTPUT) zurückgeben. Wenn eine SQL Server-Prozedur einen Ausgabeparameter und eine Ergebnismenge zurückgibt, wird nur der Parameter zurückgegeben. Die Ergebnismenge wird verworfen, und Sie empfangen die Fehlermeldung SQL0464W. Der Wert von DB2_RETURN_STATUS wird für Prozeduren abgerufen, die Ergebnismengen zurückgeben. Unabhängig vom tatsächlichen Rückgabewert der Prozedur wird jedoch stets der Wert null (0) zurückgegeben.

Einschränkungen

Der SQL Server-Wrapper kann in den folgenden Situationen keine Prozedur aufrufen:

- Wenn bereits eine andere Prozedur aufgerufen wurde.
- Wenn während einer einzelnen Verbindung eine andere Anweisung ausgeführt wird.

Zur Umgehung dieser Probleme kann die föderierte Prozedur auf einem anderen Server definiert werden. In diesem Beispiel werden die folgenden Anweisungen erfolgreich ausgeführt, wenn der Kurzname *sql-kurzname* und die Prozedur *sql-prozedur* auf verschiedenen Servern definiert sind. Wenn der Kurzname und die Prozedur auf demselben Server definiert werden, schlagen die Anweisungen fehl.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM sql-kurzname OPEN clientcur; CALL sql_proc();
```

Beispiel

In diesem Beispiel wird verdeutlicht, wie die Anweisung CREATE PROCEDURE zur Erstellung einer föderierten Prozedur für eine Datenquellenprozedur bei Microsoft SQL Server verwendet wird. Bitte beachten Sie, dass Microsoft SQL Server die Klausel UNIQUE ID und den Wert für den Quellenpaketnamen (source_package_name) der Klausel SOURCE nicht unterstützt.

```
CREATE PROCEDURE PROC1 SOURCE BHATIA.PROC1_MSSQL  
    NUMBER OF PARAMETERS 5 FOR SERVER MSSQL_SERVER  
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL  
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Erforderlich. Gibt den Namen der föderierten Prozedur an.

SOURCE BHATIA.PROC1_MSSQL

Erforderlich. Gibt den Namen des Schemas und der Prozedur in der Datenquelle an.

FOR SERVER MSSQL_SERVER

Erforderlich. Gibt eine Serverdefinition für die Erstellung der föderierten Prozedur an.

NUMBER OF PARAMETERS 5

Gibt die Gesamtzahl der von der Datenquellenprozedur verwendeten IN- und OUTPUT-Parameter an.

SOURCE BHATIA.PROC1_SQL

Gibt das Schema und den Namen für die Datenquellenprozedur an. Das Format für diesen zweiteiligen Namen ist *name_des_quellenschemas.name_der_quellenprozedur*.

SPECIFIC MYPROC1

Gibt einen eindeutigen Namen für die zu erstellende föderierte Prozedur an. Dieser Parameter wird nur für föderierte Prozeduren verwendet und keinen Datenquellenprozeduren zugeordnet. Wenn Sie keinen eindeutigen Namen angeben, wird vom Datenbankmanager der föderierten Datenbank ein Name generiert.

WITH RETURN TO CLIENT ALL

Gibt an, dass die Ergebnismenge an die Clientanwendung zurückgegeben wird. Federation gibt maximal eine Ergebnismenge zurück. Wenn dieser Parameter nicht angegeben wird, lautet die Standardeinstellung WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Gibt die Stufe des Datenzugriffs für SQL-Anweisungen an, die in der föderierten Prozedur enthalten sind. Wenn die angegebene Klausel nicht mit der Datenquellenprozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die Datenquellenprozedur verwendet.

DETERMINISTIC

Gibt an, ob die föderierte Prozedur immer dieselben Ergebnisse für eine bestimmte Argumentwertmenge zurückgibt. Dieser Parameter kann die Interaktionsleistung zwischen dem Server mit föderierten Datenbanken und der Datenquelle verbessern.

EXTERNAL ACTION

Gibt an, ob die föderierte Prozedur eine Aktion ausführt, die den Status eines Objekts ändert, das nicht vom Datenbankmanager verwaltet wird.

Föderierte Prozeduren und Oracle

Für eine Oracle-Funktion kann eine föderierte Prozedur und eine Funktionszuordnung erstellt werden.

Verwenden Sie eine Funktionszuordnung, wenn die Oracle-Funktion in SQL als Skalarfunktion eingesetzt wird. Verwenden Sie eine föderierte Prozedur, wenn die Oracle-Funktion in CALL-Anweisungen eingesetzt wird.

Oracle gibt für Funktionen nur einen einzigen Wert zurück. Der Rückgabewert für die föderierte Prozedur wird oben in der Parameterliste als zusätzlicher OUT-Parameter aufgelistet. Der Name des Parameters lautet stets DEFAULT. Wenn Sie die Klausel NUMBER OF PARAMETERS in der Anweisung CREATE PROCEDURE (Quellenfunktion) angeben, werden die Rückgabewerte nicht gezählt.

Bei einigen Oracle-Datentypen werden Informationen zu Genauigkeit, Länge und Anzahl der Kommastellen nicht im Oracle-Katalog gespeichert, wenn die Parameter einer Prozedur deklariert werden. Bei der Erstellung einer föderierten Prozedur werden die Informationen zur Oracle-Prozedur aus dem Oracle-Katalog abgerufen.

Da die Informationen zu Genauigkeit, Länge und Anzahl der Kommastellen nicht im Oracle-Katalog gespeichert sind, verhalten sich föderierte Prozeduren wie folgt:

- Für die Parameterdatentypen wird die maximale Länge verwendet.
- Der Oracle-Datentyp NUMBER wird dem föderierten Datentyp DOUBLE zugeordnet. Sie können diese Zuordnung ändern, indem Sie die standardmäßige progressive Datentypzuordnung für den Datentyp NUMBER überschreiben.

Tipp: Vom Überschreiben der progressiven Standarddatentypzuordnungen sind andere föderierte DDL-Operationen betroffen, zum Beispiel CREATE NICKNAME. Aus diesem Grund sollten Sie die Typzuordnung vor der Erstellung der Prozedur ändern. Erstellen Sie die Prozedur für die Oracle-Prozedur mit der neuen Typzuordnung, und löschen Sie anschließend die neue Typzuordnung. Nachfolgend erstellte Kurznamen und Prozeduren verwenden die Standardtypzuordnung.

Beispiel

In diesem Beispiel wird verdeutlicht, wie die Anweisung CREATE PROCEDURE zur Erstellung einer föderierten Prozedur für eine Datenquellenprozedur bei Oracle verwendet wird.

```
CREATE PROCEDURE PROC2 SOURCE ZELLER_SCHEMA.ORACLE_PKG9.PROC2
    NUMBER OF PARAMETERS 5 UNIQUE_ID '2' FOR SERVER ORA_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION;
```

PROC2

Erforderlich. Gibt den Namen der föderierten Prozedur an.

SOURCE ZELLER_SCHEMA.ORACLE_PKG9.PROC2

Erforderlich. Gibt das Schema, das Paket und den Namen für die Oracle-Prozedur oder -Funktion an. Wenn die Oracle-Prozedur oder -Funktion in einem Paket enthalten ist, müssen Sie in der Anweisung CREATE PROCEDURE einen dreiteiligen Namen angeben. Das Format für diesen dreiteiligen Namen ist *name_des_quellenschemas.name_des_quellenpakets.name_der_quellenprozedur*. Wenn die Oracle-Prozedur oder -Funktion nicht in einem Paket enthalten ist, müssen Sie in der Anweisung CREATE PROCEDURE einen zweiteiligen Namen angeben. Das Format für diesen zweiteiligen Namen ist *name_des_quellenschemas.name_der_quellenprozedur*.

NUMBER OF PARAMETERS 5

Gibt die Gesamtzahl der von der Oracle-Prozedur verwendeten IN-, OUT- und INOUT-Parameter an. Verwenden Sie diesen Parameter, wenn mehrere Prozeduren mit demselben Schema- und Prozedurnamen vorhanden sind. Wenn das Schema beispielsweise ZELLER lautet und eine PROC1-Prozedur mit zwei Parametern und eine weitere PROC1-Prozedur mit drei Parametern vorhanden ist, lautet der Name für beide Prozeduren ZELLER.PROC1. Der Wert für NUMBER OF PARAMETERS in der Datenquellenprozedur gibt an, auf welche Prozedur in der Anweisung CREATE PROCEDURE verwiesen wird. Oracle-Parameter vom Typ REF_CURSOR müssen in den Wert von NUMBER OF PARAMETERS einbezogen werden.

UNIQUE_ID '2'

Gibt die eindeutige Kennung für die Oracle-Prozedur an. Verwenden Sie den Parameter UNIQUE_ID nur dann, wenn der Name des Schemas, der Name der Prozedur und die Anzahl der Parameter eine Oracle-Prozedur nicht eindeutig identifizieren. Der Wert für UNIQUE ID steht in der Spalte ALL_ARGUMENTS.OVERLOAD im Oracle-Systemkatalog. Wenn Sie den Parameter UNIQUE ID nicht angeben, erkennt der Server mit föderierten

Datenbanken die überladenen Prozeduren und gibt einen Fehler zurück. Verwenden Sie diese Option nur mit Oracle-Prozeduren.

FOR SERVER ORA_SERVER

Erforderlich. Gibt eine Serverdefinition für die Erstellung der föderierten Prozedur an.

SPECIFIC MYPROC1

Gibt einen eindeutigen Namen für die zu erstellende föderierte Prozedur an. Dieser Parameter wird nur für föderierte Prozeduren verwendet und keinen Datenquellenprozeduren zugeordnet. Wenn Sie keinen eindeutigen Namen angeben, wird vom Datenbankmanager der föderierten Datenbank ein Name generiert. Die Angabe dieses Parameters ist optional.

WITH RETURN TO CLIENT ALL

Gibt an, dass die Ergebnismenge an die Clientanwendung zurückgegeben wird. Federation gibt maximal eine Ergebnismenge zurück. Wenn dieser Parameter nicht angegeben wird, lautet die Standardeinstellung WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Gibt die Stufe des Datenzugriffs für SQL-Anweisungen an, die in der föderierten Prozedur enthalten sind. Wenn die angegebene Klausel nicht mit der Oracle-Prozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die Oracle-Prozedur verwendet.

DETERMINISTIC

Gibt an, ob die föderierte Prozedur immer dieselben Ergebnisse für eine bestimmte Argumentwertmenge zurückgibt. Dieser Parameter kann die Interaktionsleistung zwischen dem Server mit föderierten Datenbanken und der Datenquelle verbessern.

NO EXTERNAL ACTION

Gibt an, ob die föderierte Prozedur eine Aktion ausführt, die den Status eines Objekts ändert, das nicht vom Datenbankmanager verwaltet wird.

Überladene Prozeduren in föderierten Systemen

Überladene Prozeduren sind Prozeduren mit identischen Namen und Schemata. Überladene Prozeduren können eine unterschiedliche Anzahl Parameter oder unterschiedliche Parametersignaturen aufweisen. Überladene Prozeduren werden verwendet, um ähnliche Versionen einer Prozedur zu erstellen.

Der Server mit föderierten Datenbanken lässt überladene Prozeduren nur dann zu, wenn jede Prozedur über eine unterschiedliche Anzahl Parameter verfügt.

Oracle lässt überladene Prozeduren nur dann zu, wenn jede Prozedur über eine unterschiedliche Anzahl Parameter verfügt oder wenn die Parametertypen unterschiedlich sind. Für überladene Oracle-Prozeduren können föderierte Prozeduren erstellt werden.

Bei der Erstellung der föderierten Prozedur müssen Sie eine eindeutige ID (UNIQUE ID) angeben, um Prozeduren voneinander zu unterscheiden, die identische Namen und Schemanamen verwenden und dieselbe Parameteranzahl aufweisen.

Die eindeutige ID kann über die Discoveryfunktion in der Steuerzentrale oder durch Abfragen des Oracle-Katalogs ermittelt werden.

Verwendung der Discoveryfunktion in der Steuerzentrale

Wenn Sie eine föderierte Prozedur in der Steuerzentrale erstellen, stellt die Discoveryfunktion fest, ob die Oracle-Prozedur überladen ist. Der betreffende Wert wird für jede Oracle-Prozedur im Feld UNIQUE ID angezeigt.

Abfrage des Oracle-Katalogs

Sie können die Spalte OVERLOAD im Oracle-Katalog SYS.ALL_ARGUMENTS abfragen, um die eindeutige ID zu ermitteln. Der Wert für die eindeutige ID ist ein Zeichenliteral, das eine Zahl enthält (z. B. 1). Sie können den Durchgriffsmodus auf dem Server mit föderierten Datenbanken verwenden oder den Oracle-Client direkt abfragen.

Verwenden Sie beispielsweise die folgende Anweisung SELECT, um den Oracle-Katalog abzufragen und die Signatur und die Spalte OVERLOAD für eine mit HJZ beginnende Prozedur anzuzeigen:

```
SELECT owner, package_name, object_name, overload, position, argument_name,
in_out, data_type FROM all_arguments aa
WHERE object_name like 'HJZ%'
ORDER BY owner, package_name, object_name, overload, position;
```

Die folgende Ausgabe aus der obigen Abfrage zeigt, dass das Paket HJZ_PACK1 drei Prozeduren mit dem Namen HJZTEST1 enthält. Die Anzahl der Prozeduren kann mithilfe der Spalten OBJECT_NAME und OVERLOAD ermittelt werden. Die erste Prozedur verfügt über einen IN-Parameter mit einem numerischen Datentyp. Die zweite Prozedur verfügt über einen IN-Parameter mit einem Zeichendatentyp. Die dritte Prozedur verfügt über einen OUT-Parameter mit einem Zeichendatentyp einen IN-Parameter mit einem numerischen Datentyp. Die Ausgabe zeigt, dass im Paket HJZ_PACK1 zwei Prozeduren den Namen HJZTEST3 verwenden. Ferner gibt es eine Prozedur mit dem Namen HJZTEST1, die nicht im Paket enthalten ist. Diese letzte Prozedur verfügt über einen IN-Parameter, der einen numerischen Datentyp verwendet.

OWNER	PACKAGE_NAME	OBJECT_NAME	OVERLOAD	POSITION	ARGUMENT_NAME	IN_OUT	DATA_TYPE
J15USER1	HJZ_PACK1	HJZTEST1	1	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST1	2	1	A	IN	CHAR
J15USER1	HJZ_PACK1	HJZTEST1	3	0	-	OUT	CHAR
J15USER1	HJZ_PACK1	HJZTEST1	3	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	1	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	1	2	B	OUT	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	2	1	A	IN	CHAR
J15USER1	HJZ_PACK1	HJZTEST3	2	2	B	OUT	CHAR
J15USER1	-	HJZTEST1	-	1	A	IN	NUMBER

9 record(s) selected.

Setzen Sie die folgende Anweisung CREATE PROCEDURE ab, um für die zweite überladene Prozedur mit einem IN-Parameter des Datentyps CHAR eine föderierte Prozedur zu erstellen:

```
CREATE PROCEDURE HJZTEST1 SOURCE J15USER1.HJZ_PACK1.HJZTEST1
NUMBER OF PARAMETERS 1 UNIQUE ID '2'
FOR SERVER ORA_SERVER WITH RETURN TO CLIENT ALL;
```

Wichtig: Im obigen Beispiel wird die Prozedur durch die Klausel NUMBER OF PARAMETERS nicht eindeutig identifiziert. In der Tabelle gibt es zwei Prozeduren mit dem Namen HJZTEST1, und jede Prozedur verfügt über einen Parameter. Sie müssen die überladene Prozedur, die Sie verwenden möchten, mithilfe der Klausel UNIQUE ID angeben. Geben Sie den Wert aus der Spalte OVERLOAD als Wert für

die Klausel `UNIQUE_ID` an. Bei Angabe der Klausel `UNIQUE ID` ist die Klausel `NUMBER OF PARAMETERS` optional. Stellen Sie mithilfe der Klausel `NUMBER OF PARAMETERS` fest, ob die Datenquellenprozedur über die erwartete Anzahl Parameter verfügt.

Föderierte Prozeduren und Sybase

Lesen Sie vor der Erstellung einer föderierten Prozedur die Informationen dazu, welche Parameter unterstützt werden, wie Ergebnismengen zurückgegeben werden und welche Einschränkungen bestehen.

Parameter

Sybase-Prozeduren verwenden Eingabe- und Ausgabeparameter. Der Sybase-Wrapper ordnet einem föderierten `IN`-Parameter einen Sybase-Eingabeparameter und einem föderierten `INOUT`-Parameter einen Sybase-Ausgabeparameter zu. In Sybase-Prozeduren können optionale Parameter verwendet werden, in föderierten Prozeduren ist die Verwendung dieser Parameter jedoch nicht möglich. Aus diesem Grund müssen Sie beim Absetzen einer Anweisung `CALL` alle Parameter angeben.

Für Sybase Version 12.0 sind alle Parameter Eingabeparameter. Föderierte Prozeduren können keine Ausgabeparameterwerte zurückgeben. Hierbei handelt es sich um eine Einschränkung des Sybase-Katalogs, die für neuere Sybase-Versionen nicht zutrifft.

Ergebnismengen

Bei Sybase-Prozeduren, die einen Ausgabeparameter und eine Ergebnismenge zurückgeben, wird die Ergebnismenge verworfen. Wenn die Sybase-Prozedur eine Ergebnismenge und einen Rückgabewert zurückgibt, wird unabhängig vom tatsächlichen Rückgabewert der Datenquellenprozedur ein Rückgabewert von 0 angegeben. In beiden Fällen wird keine Warnung ausgegeben.

Die Ergebnismengen werden jedoch verworfen, und die Nachricht `SQL0464W` wird zurückgegeben, wenn beide der folgenden Bedingungen zutreffen:

- Die föderierte Prozedur wird für eine Sybase-Prozedur definiert, die Ergebnismengen zurückgibt.
- Die föderierte Prozedur wird in einem Trigger oder in einer benutzerdefinierte Funktion aufgerufen.

Einschränkungen

Der Sybase-Wrapper kann in den folgenden Situationen keine Prozedur aufrufen:

- Wenn bereits eine andere Prozedur aufgerufen wurde.
- Wenn während einer einzelnen Verbindung eine andere Anweisung ausgeführt wird.

Zur Umgehung dieser Probleme kann die föderierte Prozedur auf einem anderen Server definiert werden. In diesem Sybase-Beispiel werden die folgenden Anweisungen erfolgreich ausgeführt, wenn der Kurzname *sybase-kurzname* und die Prozedur *sybase-prozedur* auf unterschiedlichen Servern definiert sind. Wenn der Kurzname und die Prozedur auf demselben Server definiert werden, schlagen die Anweisungen fehl.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM sybase-kurzname OPEN clientcur; CALL syb_proc();
```

Beispiel

In diesem Beispiel wird verdeutlicht, wie die Anweisung CREATE PROCEDURE zur Erstellung einer föderierten Prozedur für eine Datenquellenprozedur bei Sybase verwendet wird. Bitte beachten Sie, dass Sybase die Klausel UNIQUE ID der Anweisung CREATE PROCEDURE nicht unterstützt.

```
CREATE PROCEDURE PROC1 SOURCE BHATIA.PROC1_SYBASE
    NUMBER OF PARAMETERS 3 FOR SERVER SYBASE_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Erforderlich. Gibt den Namen der föderierten Prozedur an.

SOURCE BHATIA.PROC1_SYBASE

Erforderlich. Gibt das Schema und den Namen für die Sybase-Prozedur an. Für Sybase-Prozeduren wird in der Anweisung CREATE PROCEDURE ein zweiteiliger Name angegeben. Das Format für diesen zweiteiligen Namen ist *name_des_quellenschemas.name_der_quellenprozedur*.

NUMBER OF PARAMETERS 3

Gibt die Gesamtzahl der von der Sybase-Prozedur verwendeten IN-, OUT- und INOUT-Parameter an. Verwenden Sie diesen Parameter, wenn mehrere Prozeduren mit demselben Schema- und Prozedurnamen vorhanden sind. Wenn das Schema beispielsweise BHATIA lautet und eine PROC1-Prozedur mit drei Parametern und eine weitere PROC1-Prozedur mit einem Parameter vorhanden ist, lautet der Name für beide Prozeduren BHATIA.PROC1. Der Wert für NUMBER OF PARAMETERS in der Datenquellenprozedur gibt an, auf welche Prozedur in der Anweisung CREATE PROCEDURE verwiesen wird.

FOR SERVER SYBASE_SERVER

Erforderlich. Gibt eine Serverdefinition für die Erstellung der föderierten Prozedur an.

SPECIFIC MYPROC1

Gibt einen eindeutigen Namen für die zu erstellende föderierte Prozedur an. Dieser Parameter wird nur für föderierte Prozeduren verwendet und keinen Datenquellenprozeduren zugeordnet. Wenn Sie keinen eindeutigen Namen angeben, wird vom Datenbankmanager der föderierten Datenbank ein Name generiert.

WITH RETURN TO CLIENT ALL

Gibt an, dass die Ergebnismenge an die Clientanwendung zurückgegeben wird. Federation gibt maximal eine Ergebnismenge zurück. Wenn dieser Parameter nicht angegeben wird, lautet die Standardeinstellung WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Gibt die Stufe des Datenzugriffs für SQL-Anweisungen an, die in der föderierten Prozedur enthalten sind. Wenn die angegebene Klausel nicht mit der Sybase-Prozedur übereinstimmt, wird eine Fehlermeldung zurückgegeben. Wenn Sie diese Klausel nicht angeben, wird die Klausel für die Sybase-Prozedur verwendet.

DETERMINISTIC

Gibt an, ob die föderierte Prozedur immer dieselben Ergebnisse für eine bestimmte Argumentwertmenge zurückgibt. Dieser Parameter kann die Interaktionsleistung zwischen dem Server mit föderierten Datenbanken und der Datenquelle verbessern.

EXTERNAL ACTION

Gibt an, ob die föderierte Prozedur eine Aktion ausführt, die den Status eines Objekts ändert, das nicht vom Datenbankmanager verwaltet wird.

Erstellen von föderierten Prozeduren

Für jede Datenquellenprozedur, die über den Server mit föderierten Datenbanken aufgerufen werden soll, muss eine föderierte Prozedur erstellt werden.

Vorbereitung

- Die Datenquellenprozedur muss bereits vorhanden sein.
- Der Wrapper für die Datenquelle wurde registriert, und die Serverdefinition wurde erstellt.
- Alle erforderlichen Benutzerzuordnungen wurden erstellt.
- Die Zugriffsrechte, über die die Berechtigungs-ID der Anweisung verfügt, umfassen mindestens eines der folgenden:
 - Zugriffsrecht IMPLICIT_SCHEMA für die Datenbank, wenn der Schemaname der Prozedur nicht auf ein vorhandenes Schema verweist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Prozedur auf ein vorhandenes Schema verweist
 - Berechtigung SYSADM oder DBADM
- Zu den Zugriffsrechten, über die die Berechtigungs-ID in der Datenquelle verfügt, muss darüber hinaus das Zugriffsrecht zur Auswahl der Beschreibung der Prozedur in den fernen Katalogtabellen gehören.

Informationen zu dieser Task

Zur Erstellung einer föderierten Prozedur können Sie die SQL-Anweisung CREATE PROCEDURE (Quellenfunktion) oder die Steuerzentrale verwenden. Bei der Verwendung der Steuerzentrale wählen Sie eine Datenquellenprozedur aus, und die Steuerzentrale ruft automatisch die zur Erstellung der föderierten Prozedur erforderlichen Informationen ab.

Gehen Sie wie folgt vor, um die Steuerzentrale zu verwenden:

1. Klicken Sie mit der rechten Maustaste den Ordner **Föderierte gespeicherte Prozeduren** an, und klicken Sie **Erstellen** an. Der Ordner **Föderierte gespeicherte Prozeduren** befindet sich unter dem Wrapper und der Serverdefinition, die für die Datenquellen registriert wurden.
2. Klicken Sie im Fenster **Föderierte gespeicherte Prozeduren erstellen** die Option **Aufspüren** an.
3. Geben Sie im Fenster **Aufspüren** Kriterien zum Filtern der Suchoperationen nach Schema, Paket-, Prozedur- oder Funktionsnamen an. Verwenden Sie die Operatoren BETWEEN, IN, NOT IN und LIKE zur Angabe von Kriterien, und setzen Sie die angegebenen Werte in einfache Anführungszeichen. Mithilfe des Knopfs **Anzahl** im Fenster **Aufspüren** können Sie feststellen, wie viele Prozeduren zurückgegeben werden. Ist die Anzahl hoch, geben Sie zusätzliche Kriterien an, um die Suche einzugrenzen.

4. Optional: Wählen Sie im Fenster **Föderierte gespeicherte Prozeduren erstellen** das Markierungsfeld neben der Prozedur aus, die Sie erstellen möchten. Klicken Sie anschließend **Merkmale** an, um die Informationen zu überprüfen, und stellen Sie sicher, dass alle erforderlichen Felder ausgefüllt sind, bevor Sie die föderierte Prozedur erstellen.
5. Wählen Sie im Fenster **Föderierte gespeicherte Prozeduren erstellen** das Markierungsfeld neben einer oder mehreren Prozeduren aus, die Sie erstellen möchten, und klicken Sie anschließend **OK** an.

Erteilen und Widerrufen von Berechtigungen zum Aufrufen von föderierten Prozeduren

Der Administrator der föderierten Datenbank muss anderen Benutzern Berechtigungen zum Aufrufen der föderierten Prozeduren erteilen.

Vorbereitung

Der Benutzer, der die föderierte Prozedur aufruft, muss über eine gültige Benutzerzuordnung zwischen dem Server mit föderierten Datenbanken und der Datenquelle verfügen. Die ferne Benutzer-ID muss über eine Berechtigung für die Datenquelle verfügen, die der Berechtigung EXECUTE auf dem Server mit föderierten Datenbanken entspricht. Die Berechtigung EXECUTE kann einem Benutzer für die föderierte Prozedur erteilt werden. Entspricht die Benutzerberechtigung für die Datenquelle jedoch nicht der Berechtigung EXECUTE auf dem Server mit föderierten Datenbanken, schlagen Aufrufe an die Datenquellenprozedur fehl.

Die Berechtigungs-ID für die Anweisung GRANT muss mindestens eine der folgenden Berechtigungen aufweisen:

- WITH GRANT OPTION für EXECUTE für die föderierte Prozedur
- Berechtigung SYSADM oder DBADM

Vorgehensweise

Gehen Sie wie folgt vor, um die Berechtigung zum Aufrufen von föderierten Prozeduren zu erteilen oder zu widerrufen:

Die Berechtigungen können über die Steuerzentrale oder die Befehlszeile angegeben werden:

Methode	Beschreibung
Steuerzentrale	<ol style="list-style-type: none">1. Klicken Sie den Namen der föderierten Prozedur mit der rechten Maustaste an, und klicken Sie Zugriffsrechte an.2. Wählen Sie den Benutzer oder die Gruppe aus, für den bzw. die Sie Zugriffsrechte festlegen möchten. Klicken Sie Benutzer hinzufügen an, um einen neuen Benutzer hinzuzufügen. Wählen Sie die Indexzunge Gruppe aus, und klicken Sie Gruppe hinzufügen an, um eine neue Gruppe hinzuzufügen.3. Wählen Sie in der Dropdown-Liste Zugriffsrechte: EXECUTE Folgendes aus:<ul style="list-style-type: none">• Wählen Sie Ja aus, um nur das Zugriffsrecht EXECUTE zu erteilen.• Wählen Sie Erteilen aus, um die Zugriffsrechte EXECUTE und WITH GRANT OPTION zu erteilen.• Wählen Sie Nein aus, um das Zugriffsrecht EXECUTE für den Benutzer oder die Gruppe zu entfernen.4. Zugriffsrechte können gleichzeitig für mehrere Benutzer erteilt oder widerrufen werden. Wählen Sie die Benutzer aus der Liste aus, und klicken Sie Alle erteilen oder Alle entziehen an, um das Zugriffsrecht EXECUTE für die ausgewählten Benutzer zu erteilen oder zu widerrufen.5. Klicken Sie OK an.

Methode	Beschreibung
Befehlszeile	<p>Geben Sie die Zugriffsrechte in der Anweisung GRANT an.</p> <p>Beispiel 1: Wenn Sie den Benutzern in der Gruppe HR_DEPT das Zugriffsrecht EXECUTE für alle Prozeduren im Schema BHATIA (einschließlich aller zukünftig erstellten Prozeduren) erteilen möchten, verwenden Sie die folgende Anweisung: GRANT EXECUTE ON PROCEDURE BHATIA.* TO HR_DEPT</p> <p>Beispiel 2: Wenn Sie dem Benutzer ZELLER das Zugriffsrecht EXECUTE für die Prozedur PROC 1 erteilen und festlegen möchten, dass dieser Benutzer anderen Benutzern das Zugriffsrecht EXECUTE für diese Prozedur erteilen kann, verwenden Sie die folgende Anweisung: GRANT EXECUTE ON PROCEDURE PROC1 TO ZELLER WITH GRANT OPTION</p> <p>Beispiel 3: Wenn Sie dem Benutzer ERFAN das Zugriffsrecht EXECUTE für die Prozedur PROC2 erteilen möchten, die mit dem spezifischen Namen MY_PROC2 erstellt wurde, verwenden Sie die folgende Anweisung: GRANT EXECUTE ON SPECIFIC PROCEDURE MY_PROC2 TO ERFAN</p>

Anzeigen von Parameterinformationen und Aufrufen von föderierten Prozeduren

Zum Anzeigen von Parameterinformationen können Sie die Steuerzentrale verwenden oder eine Abfrage für die Katalogsicht SYSCAT.ROUTINESFEDERATED ausführen. Rufen Sie dann mit der Anweisung CALL eine föderierte Prozedur auf.

Vorbereitung

Der Benutzer, der eine föderierte Prozedur aufruft, muss über das Zugriffsrecht EXECUTE für die Datenquelle sowie über eine gültige Benutzerzuordnung für eine ferne Datenquellenbenutzer-ID verfügen, die für den Zugriff auf die Datenquellentabellen berechtigt ist.

Verwenden Sie zum Anzeigen von Parameterinformationen und zum Aufrufen einer föderierten Prozedur eine der folgenden Methoden:

Methode	Vorgehensweise
Steuerzentrale	<ol style="list-style-type: none"> 1. Erweitern Sie den Ordner Föderierte Objekte, den Ordner Serverdefinitionen und den Ordner Föderierte gespeicherte Prozeduren in der Objektbaumstruktur. 2. Wählen Sie den Namen der föderierten Prozedur aus, um Informationen, wie z. B. den Datentyp und Modus der einzelnen Parameter, anzuzeigen. 3. Klicken Sie ToolsBefehlseditor an, um den Befehlseditor zu öffnen, und geben Sie eine Anweisung CALL ein.
Befehlszeile	<ol style="list-style-type: none"> 1. Verwenden Sie die Anweisung SELECT, um eine Abfrage für die Sicht SYSCAT.ROUTINEPARMS im Katalog der föderierten Datenbank auszuführen. 2. Setzen Sie eine Anweisung CALL ab.

In diesem Beispiel wird eine Anweisung SELECT verwendet, um Informationen zur föderierten Prozedur abzurufen. Wenn sich zum Beispiel die föderierte Prozedur FEDPROC1 im föderierten Schema BOB befindet, setzen Sie die folgende Anweisung SELECT ab:

```
SELECT ordinal, char(parmname,30)
AS name,
rowtype, char(typename,30) AS type
FROM syscat.routineparms
WHERE routinename='FEDPROC1' AND
routineschema = 'BOB'
ORDER BY ordinal;
```

Die Parameter sind in den Ergebnissen der Abfrage aufgelistet:

```
ORDINAL NAME ROWTYPE TYPE
-----
1      P1      P      INTEGER
2      P2      O      VARCHAR
```

Der Zeilentyp P gibt einen Eingabeparameter an. Der Zeilentyp O gibt einen Ausgabeparameter an. Ein Zeilentyp B, der einen Parameter INOUT angibt, ist in diesem Beispiel nicht dargestellt.

Ändern oder Löschen von föderierten Prozeduren

Sie können eine vorhandene föderierte Prozedur ändern, indem Sie den Datentyp von mindestens einem Parameter der föderierten Prozedur ändern. Sie können die föderierte Prozedur nicht direkt ändern, um andere Änderungen vorzunehmen.

Vorbereitung

Die Berechtigungs-ID für die Anweisung DROP PROCEDURE muss eine der folgenden Berechtigungen aufweisen:

- Berechtigung SYSADM oder DBADM
- Zugriffsrecht DROPIN für das Schema der föderierten Prozedur
- Definierender Benutzer der Prozedur, wie in der Spalte DEFINER der Katalogsicht für die föderierte Prozedur eingetragen
- Zugriffsrecht CONTROL für die föderierte Prozedur

Informationen zu dieser Task

Neben dem Ändern des Datentyps eines Parameters sind möglicherweise noch weitere Änderungen an einer föderierten Prozedur erforderlich. Sie müssen eine föderierte Prozedur zum Beispiel möglicherweise ändern, wenn die ferne Prozedur geändert wird. In diesem Fall können Sie die föderierte Prozedur nicht direkt ändern. Sie müssen die Prozedur zuerst löschen und sie anschließend mit den neuen Einstellungen erneut erstellen. Andernfalls müssen Sie die vorhandene Prozedur mithilfe der Anweisung `CREATE OR REPLACE PROCEDURE` ersetzen.

Wenn Sie eine föderierte Prozedur löschen, wird die Prozedur aus dem Systemkatalog in der föderierten Datenbank gelöscht, dies hat jedoch keine Auswirkungen auf die Datenquellenprozedur. Bitte beachten Sie, dass beim Löschen einer föderierten Prozedur Anwendungen, die von den gelöschten Prozeduren abhängig sind, beeinträchtigt werden können.

Zum Löschen einer föderierten Prozedur können Sie die Steuerzentrale oder die Befehlszeile verwenden.

Vorgehensweise

Verwenden Sie zum Löschen einer föderierten Prozedur eine der folgenden Methoden:

Methode	Vorgehensweise
Steuerzentrale	<ol style="list-style-type: none">1. Erweitern Sie den Ordner Föderierte Objekte, den Ordner Serverdefinitionen und den Ordner Föderierte gespeicherte Prozeduren in der Objektbaumstruktur.2. Klicken Sie mit der rechten Maustaste die föderierte Prozedur an, die gelöscht werden soll, und klicken Sie Löschen an.
Befehlszeile	Verwenden Sie die Anweisung <code>DROP</code> . Beispiel: <code>DROP PROCEDURE name_der_föderierten_prozedur</code>

Verknüpfen der Ergebnismengen von föderierten Prozeduren

Sie können die von einer föderierten Prozedur zurückgegebenen Ergebnismengen mithilfe des Befehls `DB2FEDGENTF` verknüpfen.

Vorbereitung

- Für die Verwendung des Parameters `-create` des Befehls `DB2FEDGENTF` müssen Sie entweder über die Berechtigung `DBADM` für die föderierte Datenbank oder über eine Kombination der folgenden Berechtigungen oder Zugriffsrechte für die föderierte Datenbank verfügen:
 1. Sie müssen über eine der folgenden Berechtigungen verfügen:
 - `CREATE_EXTERNAL_ROUTINE`
 - `CREATETAB`
 2. Darüber hinaus müssen Sie über eine der folgenden Berechtigungen bzw. eines der folgenden Zugriffsrechte verfügen:

- Schreibrecht für das Verzeichnis *installationsverzeichnis/sqllib/function*, wobei *installationsverzeichnis* das Verzeichnis ist, in dem der Server mit föderierten Datenbanken installiert ist
 - Berechtigung IMPLICIT_SCHEMA, wenn der implizite oder explizite Schemaname der Funktion nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Funktion vorhanden ist
- Für die Verwendung des Parameters **-drop** des Befehls DB2FEDGENTF müssen Sie über mindestens eine der folgenden Berechtigungen bzw. über eines der folgenden Zugriffsrechte für die föderierte Datenbank verfügen:
 - Zugriffsrecht DROPIN für das Schema des Objekts
 - Zugriffsrecht OWNER für das Objekt
 - Berechtigung DBADM

Informationen zu dieser Task

Von föderierten Prozeduren zurückgegebene Ergebnismengen werden verknüpft, um Daten von lokalen und fernen Tabellen zu verknüpfen. Dies gilt insbesondere in Systemen, die nur über föderierte Prozeduren den Zugriff ermöglichen.

Vorgehensweise

Gehen Sie wie folgt vor, um die Ergebnismengen von föderierten Prozeduren zu verknüpfen:

1. Erstellen und registrieren Sie eine Tabellenfunktion mit dem Parameter **-create** des Befehls DB2FEDGENTF. Beispiel:

```
DB2FEDGENTF -db sample -u user1 -p password1
  -create
  -stpn S1_INVENTORY
  -tfn S1_INVTRY_TF
  -c 'PART_NUM CHAR(10), S1_QTY INT'
```

2. Verwenden Sie einen Join, um die Daten in den Ergebnismengen der föderierten Prozedur zu kombinieren. Sie können die Ergebnismenge einer föderierten Prozedur mit einer lokalen Tabelle verknüpfen oder die Ergebnismengen von zwei föderierten Prozeduren verknüpfen.

Beispiele

In den folgenden Beispielen sind die gespeicherten Prozeduren ProINVENTORY und ProINVENTORY2 vorhanden und jede Prozedur gibt den Lagerbestand von zwei Lieferanten als Ergebnismenge zurück. Ferner ist die Tabelle PARTS vorhanden mit dem Lagerbestand des Herstellers und den folgenden Spaltennamen und -typen:

Tabelle 6. Spalten der Tabelle PARTS

Spaltenname	Spaltentyp
PART_NUM	CHAR(10)
PART_DESC	VARCHAR(400)
INVENTORY	INT

Verknüpfen von Ergebnismengen mit lokalen Tabellen - Beispiel

Im folgenden Beispiel wird der Lagerbestand von einem Lieferanten und dem Hersteller kombiniert, um den verfügbaren Gesamtlagerbestand zu ermitteln:

1. Erstellen Sie mithilfe der Anweisung CREATE PROCEDURE die föderierte Prozedur S1_INVENTORY aus der gespeicherten Prozedur ProINVENTORY:

```
CREATE PROCEDURE S1_INVENTORY
SOURCE ProINVENTORY
FOR SERVER ORA1;
```

2. Erstellen Sie mithilfe des Befehls DB2FEDGENTF die Tabellenfunktion S1_INVTRY_TF, die die Ergebnismengen PART_NUM und S1_QTY beinhaltet:

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
-stpn S1_INVENTORY
-tfn S1_INVTRY_TF
-c 'PART_NUM CHAR(10), S1_QTY INT'
```

3. Verwenden Sie einen Join, um die Daten der Tabelle PARTS mit der Ergebnismenge der Tabellenfunktion S1_INVTRY_TF zu kombinieren:

```
SELECT a.PART_NUM, a.PART_DESC, a.INVENTORY + b.S1_QTY as MAX_QTY
FROM PARTS a, TABLE(S1_INVTRY_TF()) b
WHERE a. PART_NUM = b. PART_NUM
```

Verknüpfen von zwei Ergebnismengen - Beispiel

Im folgenden Beispiel kombiniert der Hersteller die Lagerbestände von beiden Lieferanten, um deren verfügbaren Gesamtlagerbestand zu ermitteln:

1. Erstellen Sie die Tabellenfunktion S1_INVTRY_TF für den ersten Lieferanten:

- a. Erstellen Sie mithilfe der Anweisung CREATE PROCEDURE die föderierte Prozedur S1_INVENTORY aus der gespeicherten Prozedur ProINVENTORY:

```
CREATE PROCEDURE S1_INVENTORY
SOURCE ProINVENTORY
FOR SERVER ORA1;
```

- b. Erstellen Sie mithilfe des Befehls DB2FEDGENTF die Tabellenfunktion S1_INVTRY_TF, die die Ergebnismengen PART_NUM und S1_QTY beinhaltet:

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
-stpn S1_INVENTORY
-tfn S1_INVTRY_TF
-c 'PART_NUM CHAR(10), S1_QTY INT'
```

2. Erstellen Sie die Tabellenfunktion S2_INVTRY_TF für den zweiten Lieferanten:

- a. Erstellen Sie mithilfe der Anweisung CREATE PROCEDURE die föderierte Prozedur S2_INVENTORY aus der gespeicherten Prozedur ProINVENTORY2:

```
CREATE PROCEDURE S2_INVENTORY
SOURCE ProINVENTORY2
FOR SERVER SYBA1;
```

- b. Erstellen Sie mithilfe des Befehls DB2FEDGENTF die Tabellenfunktion S2_INVTRY_TF, die die Ergebnismengen PART_NUM und S2_QTY beinhaltet:

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S2_INVENTORY
  -tfn S2_INVTRY_TF
  -c 'PART_NUM CHAR(10), S2_QTY INT'
```

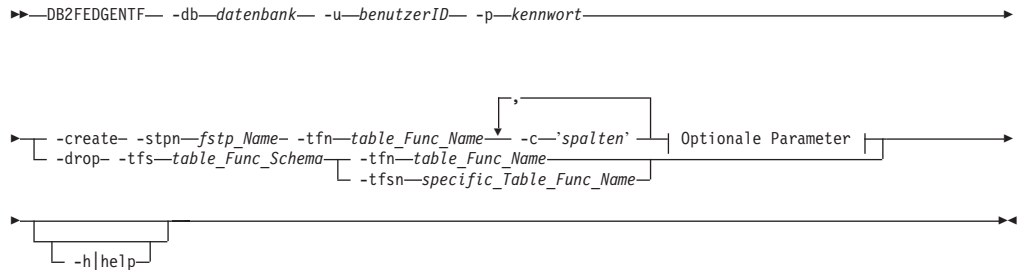
- Verwenden Sie einen Join, um die Ergebnismengen der Tabellenfunktionen S1_INVTRY_TF und S2_INVTRY_TF zu kombinieren:

```
SELECT a.PART_NUM, a.PART_DESC, b.S1_QTY + c.S2_QTY as MAX_SUPP_QTY
FROM PARTS a, TABLE(S1_INVTRY_TF()) b, TABLE(S2_INVTRY_TF()) c
WHERE a. PART_NUM = b. PART_NUM
AND a. PART_NUM = c. PART_NUM
```

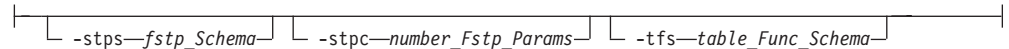
Befehl DB2FEDGENTF

Erstellt und registriert Tabellenfunktionen, die Ergebnismengen von föderierten Prozeduren zurückgeben.

Syntax



Optionale Parameter:



Parameter

-db *datenbank*

Gibt den Namen der Datenbank an, zu der Sie eine Verbindung herstellen möchten.

-u *benutzerID*

Gibt die Benutzer-ID der föderierten Datenbank an.

-p *kennwort*

Gibt das Kennwort für die Benutzer-ID an.

-create

Erstellt und registriert eine Tabellenfunktion im aktuellen Schema, wenn der Parameter **-tfs** nicht angegeben ist. Die Tabellenfunktion gibt die angegebenen Spalten der Ergebnismenge von der föderierten Prozedur zurück.

-stpn *fstp_Name*

Gibt den Namen der föderierten Prozedur an.

-tfn *table_Func_Name*

Gibt den Namen der Tabellenfunktion an. Wenn der Name der Tabellenfunktion nicht angegeben ist, wird der Name der föderierten Prozedur verwendet.

-c *'spalten'*

Gibt eine durch Kommas begrenzte Liste an, die die Paare aus Spaltenname und Spaltentyp in der Signatur der von der föderierten Prozedur zurückgegebenen Ergebnismenge enthält.

Beispiel

Die Spaltennamen lauten PID, PRICE und QTY und die Spaltentypen sind CHAR(10), DOUBLE und INT:

```
'PID CHAR(10), PRICE DOUBLE, QTY INT'
```

-stps *fstp_Schema*

Gibt das Schema der föderierten Prozedur an. Die Angabe dieses Parameters ist optional. Wenn ein Name nicht angegeben ist, wird das im Sonderregister CURRENT SCHEMA definierte SQL-Schema verwendet.

-stpc *number_Fstp_Params*

Gibt die Anzahl der Eingaben für die föderierte Prozedur an. Die Angabe dieses Parameters ist optional. Wenn die Anzahl der Eingaben nicht angegeben ist, wird die föderierte Prozedur durch den Namen der angegebenen föderierten Prozedur festgelegt. Wenn die föderierten Prozeduren überladen sind, erhalten Sie eine Fehlermeldung.

-tfs *table_Func_Schema*

Gibt das Schema der Tabellenfunktion an. Die Angabe dieses Parameters ist optional. Wenn das Schema der Tabellenfunktion nicht angegeben ist, wird das im Sonderregister CURRENT SCHEMA definierte SQL-Standard-schema verwendet.

- drop

Löscht die von Ihnen angegebene Tabellenfunktion. Die Beschreibung aus dem Katalog wird ebenfalls gelöscht und alle Pakete, die auf die angegebene Tabellenfunktion verweisen, werden ungültig.

-tfs *table_Func_Schema*

Gibt das Schema der Tabellenfunktion an, die Sie löschen möchten. Wenn das Schema der Tabellenfunktion nicht angegeben ist, wird das im Sonderregister CURRENT SCHEMA definierte SQL-Standardschema verwendet.

-tfn *table_Func_Name*

Gibt den Namen der Tabellenfunktion an, die Sie löschen möchten.

-tfsn *specific_Table_Func_Name*

Gibt bei überladenen Funktionen den bestimmten Namen der Tabellenfunktion an, die Sie löschen möchten. Dieser Parameter und der Parameter **-tfn** *table_Func_Name* schließen sich gegenseitig aus. Sie müssen diese Option nicht angeben, wenn die Tabellenfunktion durch ihren Namen und ihr Schema eindeutig identifiziert ist. Die Angabe dieses Parameters ist optional.

-h | help

Stellt Informationen zur Verwendung des Befehls DB2FEDGENTF bereit.

Beispiele

In diesem Beispiel wird der Befehl DB2FEDGENTF für die föderierte Prozedur EAST_INVTRY ausgeführt, um die Tabellenfunktion S1_INVTRY_TF zu erstellen, die die Spalten PRODID, PRICE und QTY zurückgibt:

```

DB2FEDGENTF -db sample -u user1 -p password1
-Create
  -stpn EAST_INVTRY
  -tfn E_INVTRY_TF
  -c 'PRÖDID INT, PRICE DOUBLE, QTY INT'

```

Föderierte Prozeduren - Fehlerbehebung

Treten im Zusammenhang mit föderierten Prozeduren Fehler auf, gibt es mehrere Möglichkeiten zur Fehlerbehebung.

Mit den folgenden Abfragen und Diagnosetools können Sie Informationen zu föderierten Prozeduren aufrufen. Diese Informationen unterstützen Sie bei der Behebung von Fehlern, die im Zusammenhang mit föderierten Prozeduren auftreten können.

Überprüfen der Informationen zu Datenquellenprozeduren

Wird beim Absetzen der Anweisung CREATE PROCEDURE der Fehler SQL1253N zurückgegeben, können Sie die folgenden Abfragen für die Katalogtabellen in der Datenquelle absetzen, um die Informationen zur Datenquellenprozedur zu überprüfen. Der Fehler SQL1253N gibt an, dass die in der Anweisung CREATE PROCEDURE (Quellenfunktion) angegebene Quellenprozedur in der Datenquelle nicht gefunden wurde. Sie können den Oracle-Server entweder direkt abfragen oder eine Durchgriffssitzung verwenden.

Bei Prozeduren unter DB2 für Linux, UNIX und Windows:

```

SELECT parm_count, result_sets,
       sql_data_access, deterministic, external_action
FROM syscat.routines
WHERE routineschema = ''
AND routinename = ''
AND routinetype = 'P'
AND parm_count = '' <-- optional

```

Bei Prozeduren unter DB2 für iSeries:

```

SELECT in_parms+out_parms+inout_parms,
       number_of_results, sql_data_access, deterministic,
       external_action
FROM qsys2.sysroutines
WHERE routine_schema = ''
and routine_name = ''
and routine_type = 'PROCEDURE'
and in_parms+out_parms+inout_parms = ''; <-- optional

```

Bei Prozeduren unter DB2 für z/OS:

```

SELECT parm_count, result_sets,
       sql_data_access, deterministic, external_action
FROM sysibm.sysroutines
WHERE schema = ''
AND name = ''
AND routinetype = 'P'

```

Bei Microsoft SQL Server

```

SELECT id
FROM dbo.sysobjects
WHERE id = object_id AND
      (TYPE = 'P' or TYPE = 'X')

```


Für Oracle-Prozeduren, die in einem Paket enthalten sind:

```
SELECT owner, package_name, object_name, overload, parm_count
FROM (
  SELECT owner, package_name, object_name, overload,
  SUM(case
    WHEN data_type IS NULL
    THEN 0
    ELSE 1
  END)
  AS parm_count
  FROM sys.all_arguments
  WHERE data_level = 0
  GROUP BY owner, package_name, object_name, overload
) aa
WHERE object_name = '' AND
package_name = '' AND
owner = '' AND
overload = '' AND <-- optional
parm_count =; <-- optional
```

Für Oracle-Prozeduren, die nicht in einem Paket enthalten sind:

```
SELECT object_name, object_type, status
FROM sys.all_objects
WHERE owner = '' AND
object_name = '' AND
object_type IN ('PROCEDURE', 'FUNCTION')
```

Für Sybase-Prozeduren:

```
SELECT id
FROM dbo.sysobjects
WHERE id = object_id('.') AND
(TYPE = 'P' OR TYPE = 'XP')
```

Diagnosetools

Verwenden Sie zum Diagnostizieren von Fehlern bei föderierten Prozeduren das Dienstprogramm EXPLAIN, den Befehl DESCRIBE oder den Befehl 'db2audit'.

Beispiel: Die Prozedur FED_PROC1 verfügt über drei Ausgabeparameter. Setzen Sie den folgenden Befehl ab, um den Befehl DESCRIBE für die Prozedur FED_PROC1 zu verwenden:

```
DESCRIBE CALL FED_PROC1(?,?,?);
```

Systemmonitor

Die Systemmonitorelemente in der föderierten Datenbank enthalten Informationen zu föderierten Prozeduren. Die folgenden Monitorelemente sind vorhanden:

- Das Monitorelement für die Zeit für gespeicherte Prozeduren (stored_proc_time) gibt die Zeit an, die von der Datenquelle benötigt wurde, um auf Anweisungen föderierter Prozeduren zu antworten.
- Das Monitorelement für die von gespeicherten Prozeduren zurückgegebenen Zeilen (sp_rows_selected) enthält die Anzahl Zeilen, die von der Datenquelle an den Server mit föderierten Datenbanken gesendet werden. Mit diesem Element können Sie die durchschnittliche Anzahl Zeilen berechnen, die von der Datenquelle für jede föderierte Prozedur an den Server mit föderierten Datenbanken gesendet wird; Sie können auch die durchschnittliche Zeit für die Rückgabe einer Zeile von der Datenquelle an den Server mit föderierten Datenbanken berechnen.

- Das Monitorelement für gespeicherte Prozeduren (stored_procs) gibt die Gesamtzahl der Prozeduren an, die der Server mit föderierten Datenbanken von dieser Datenquelle aufgerufen hat.

SQL-Fehler SQL30090N mit Rückkehrcode 21

Es können mehrere Situationen auftreten, in denen der Fehler SQL30090N mit dem Rückkehrcode 21 zurückgegeben wird. Eine der häufigsten Situationen ist die Erstellung einer föderierte Prozedur mit einem abgeschirmten Wrapper. Föderierte Prozeduren können nur mit einem gesicherten Wrapper erstellt werden.

Ergebnismenge wird nicht zurückgegeben

Eine Ergebnismenge kann aus einem der folgenden Gründe nicht an den Client oder das aufrufende Programm zurückgegeben werden:

- Die Klausel für die Rückgabe von Ergebnismengen wurde in der föderierten Prozedur nicht richtig angegeben.
- Einige Datenquellen geben Ergebnismengen bei Aufruf einer Prozedur nicht immer in derselben Reihenfolge zurück. Da föderierte Prozeduren nur die erste Ergebnismenge zurückgeben, wird bei Aufruf der föderierten Prozedur möglicherweise eine andere Ergebnismenge von der Datenquelle zurückgegeben.

Beispiel: Es sind zwei Prozeduren für die Datenquelle vorhanden, Prozedur A (PROCEDURE A) und Prozedur B (PROCEDURE B). Prozedur B ruft Prozedur A auf. Diese Prozeduren werden mithilfe der folgenden Anweisungen erstellt:

```
CREATE PROCEDURE A ()
BEGIN
    DECLARE cur1 CURSOR WITH RETURN TO CLIENT
    FOR SELECT * FROM t;
    OPEN cur1
END
CREATE PROCEDURE B (arg1 INT)
BEGIN
    DECLARE cur2 CURSOR WITH RETURN TO CLIENT
    FOR SELECT * FROM t;
    IF arg1<10) THEN
        CALL A();
    END IF;
    OPEN cur2
END;
```

Die föderierte Prozedur FEDPROC1 verweist auf Prozedur B der Datenquelle. Die Anweisung für FEDPROC1 lautet wie folgt:

```
CREATE PROCEDURE FEDPROC1
SOURCE newton.B
FOR SERVER s1
NUMBER OF PARAMETERS 1
WITH RETURN TO CLIENT 1;
```

Die föderierte Prozedur FEDPROC1 wird von einer lokalen Prozedur aufgerufen. Die Anweisung für die lokale Prozedur lautet wie folgt:

```
CREATE PROCEDURE local (arg1 INT)
BEGIN
    CALL FEDPROC1 (arg1)
END;
```

Wenn Sie die Anweisung `CALL LOCAL(1)` absetzen, wird die Ergebnismenge 'cur1' von Prozedur A zurückgegeben. Die Ergebnismenge 'cur2' wird nicht zurückgegeben.

Wenn Sie jedoch die Anweisung `CALL LOCAL(20)` absetzen, wird die Ergebnismenge 'cur2' von Prozedur B zurückgegeben.

Durchgriffssitzung (nur Oracle)

Wenn Sie eine Datenquellenprozedur oder -funktion bzw. ein Datenquellenpaket in einer Durchgriffssitzung erstellen, wird eine Nachricht über den Erfolg zurückgegeben, auch wenn die Objektdefinition fehlerhaft ist. Das Objekt wird auf dem Oracle-Server erstellt, aber als ungültig (INVALID) markiert. Für ungültige Objekte können keine föderierte Prozeduren erstellt werden. Wenn Sie eine föderierte Prozedur erstellen, die auf ein ungültiges Oracle-Objekt verweist, schlägt die Anweisung `CREATE PROCEDURE` (Quellenfunktion) fehl.

Verwenden Sie eine der folgenden Methoden, um festzustellen, warum ein Objekt ungültig ist:

- Verwenden Sie den Befehl `SHOW ERRORS` im Dienstprogramm SQL*Plus von Oracle.
- Führen Sie eine Abfrage für die Oracle-Katalogtabelle 'sys.all_errors' aus.

Kapitel 7. Erstellen und Modifizieren ferner Tabellen mithilfe von transparenter DDL

Mit transparenter DDL können Sie Prozeduren, mit denen Sie vertraut sind, verwenden, um ferne Tabellen mithilfe der föderierten Datenbank ohne Durchgriffs-sitzung zu erstellen und zu modifizieren.

Was ist transparente DDL?

Transparente DDL bietet die Möglichkeit, ferne Tabellen über die föderierte Datenbank zu erstellen und zu ändern, ohne Durchgriffssitzungen zu verwenden.

Die mit transparenter DDL verwendeten SQL-Anweisungen sind CREATE TABLE, ALTER TABLE und DROP TABLE.

Eine transparente DDL-Anweisung CREATE TABLE erstellt eine ferne Tabelle in der Datenquelle und einen Kurznamen für diese Tabelle auf dem Server mit föderierten Datenbanken. Sie ordnet die von Ihnen angegebenen DB2-Datentypen mithilfe der standardmäßigen regressiven Typzuordnungen den fernen Datentypen zu. Im Allgemeinen stellen die Wrapper die Typzuordnungen bereit. Sie können auch benutzerdefinierte regressiv Typzuordnungen erstellen, um die Standardzuordnungen zu überschreiben.

Der Vorteil bei der Verwendung von transparenter DDL besteht darin, dass Datenbankadministratoren sowohl zum Erstellen von lokalen als auch von fernen Tabellen Prozeduren verwenden können, die ihnen vertraut sind. Transparente DDL zentralisiert die Tabellenverwaltung und vereinfacht das Erteilen von Berechtigungen.

Transparente DDL wird von den folgenden Datenquellen unterstützt:

- DB2 für z/OS
- DB2 für System i
- DB2 Database für Linux, UNIX und Windows
- DB2 Server für VM und VSE
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle
- Sybase
- Teradata

Der Datenbankadministrator kann zum Erstellen der Tabellen entweder die DB2-Steuerzentrale oder DDL-Anweisungen im DB2-Befehlszeilenprozessor (Command Line Processor, CLP) verwenden. Bei Verwendung von transparenter DDL muss nicht die unterschiedliche DDL-Syntax für die einzelnen Datenquellen erlernt werden.

Bevor Sie ferne Tabellen in einer Datenquelle über die föderierte Datenbank erstellen können, müssen Sie den Zugriff auf die Datenquelle konfigurieren:

- Der Wrapper für diese Datenquelle muss im globalen Katalog registriert sein.
- Die Serverdefinition muss für den Server erstellt werden, auf dem sich die ferne Tabelle befinden soll.
- Die Benutzerzuordnungen müssen bei Bedarf zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver erstellt werden.

Verwenden Sie zum Erstellen von fernen Tabellen den Tabellenassistenten in der DB2-Steuerzentrale.

Zu den Zugriffsrechten der Berechtigungs-ID der transparenten DDL-Anweisungen muss mindestens eine der folgenden Berechtigungen gehören:

- Berechtigung SYSADM oder DBADM
- Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen:
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder der explizite Schemaname der Tabelle nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist

Zum Absetzen transparenter DDL-Anweisungen muss Ihre Berechtigungs-ID über die notwendigen Zugriffsrechte für den Kurznamen verfügen (damit der Server mit föderierten Datenbanken die Anforderung akzeptiert) sowie über die vergleichbaren Zugriffsrechte für den Server der fernen Datenquelle (damit die Datenquelle die Anforderung akzeptiert).

Ferne LOB-Spalten und transparente DDL

Bei der Verwendung von transparenter DDL geben Sie die Länge einer LOB-Spalte an.

Einige Datenquellen, z. B. Oracle und Informix, speichern die Längen von LOB-Spalten in ihren Systemkatalogen nicht. Wenn Sie einen Kurznamen für eine Tabelle erstellen, werden Informationen vom Systemkatalog der Datenquelle einschließlich der Spaltenlänge abgerufen. Da für die LOB-Spalten keine Länge vorhanden ist, geht die föderierte Datenbank von der maximalen Länge einer LOB-Spalte unter DB2 Database für Linux, UNIX oder Windows aus. Die föderierte Datenbank speichert die maximale Länge im Katalog der föderierten Datenbank als Länge der Kurznamenspalte.

Wenn Sie eine ferne Tabelle unter Verwendung transparenter DDL erstellen, müssen Sie jedoch die Länge der LOB-Spalte angeben. Wenn der Server mit föderierten Datenbanken einen Kurznamen in der fernen Tabelle erstellt, wird die von Ihnen im Katalog der föderierten Datenbank angegebene Länge als Länge der Kurznamenspalte gespeichert. Die maximale Länge einer LOB-Spalte beträgt 2 Gigabyte.

Erstellen von fernen Tabellen und transparente DDL

Das Erstellen einer fernen Tabelle über die föderierte Datenbank unter Verwendung von transparenter DDL geht mit einigen anderen Aktionen einher.

Beim Erstellen der fernen Tabelle gilt Folgendes:

- Für die ferne Tabelle wird automatisch ein Kurzname erstellt. Der Kurzname ist identisch mit dem Tabellennamen, der in der Anweisung CREATE TABLE ange-

geben wird. Die ferne Tabelle weist denselben Namen wie die Tabelle auf, es sei denn, Sie geben über die Option `REMOTE_TABNAME` einen anderen Namen an.

- Das Schema der fernen Tabellen entspricht dem Schema des Kurznamens, es sei denn, Sie geben über die Option `REMOTE_SCHEMA` ein anderes Schema an.
- Der unter Verwendung von transparenter DDL erstellte Kurzname kann wie jeder andere Kurzname verwendet werden. Darüber hinaus kann die ferne Tabelle mithilfe von `ALTER` und `DROP` geändert bzw. gelöscht werden (dies ist bei einem unter Verwendung von `CREATE NICKNAME` erstellten Kurznamen nicht möglich).
- In der Katalogsicht `SYSCAT.TABOPTIONS` wird eine Zeile mit dem Optionsnamen `TRANSPARENT` und dem Wert `Y` hinzugefügt.

Erstellen neuer ferner Tabellen unter Verwendung von transparenter DDL

Zum Erstellen einer fernen Tabelle unter Verwendung von transparenter DDL können Sie entweder den Assistenten der DB2-Steuerzentrale oder die Anweisung `CREATE TABLE` verwenden.

Vorbereitung

Vor dem Erstellen einer fernen Tabelle müssen Sie den Server mit föderierten Datenbanken für den Zugriff auf diese Datenquelle konfigurieren. Das Konfigurieren umfasst folgende Aufgaben:

- Erstellen des Wrappers für diesen Datenquellentyp
- Angeben der Serverdefinition für den Server, auf dem sich die ferne Tabelle befinden soll
- Erstellen der Benutzerzuordnungen zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver

Zum Absetzen transparenter DDL-Anweisungen muss Ihre Berechtigungs-ID über die notwendigen Zugriffsrechte für den Kurznamen verfügen (damit der Server mit föderierten Datenbanken die Anforderung akzeptiert) sowie über die vergleichbaren Zugriffsrechte für den Server der fernen Datenquelle (damit die Datenquelle die Anforderung akzeptiert).

Die Berechtigungs-ID, die die transparenten DDL-Anweisungen absetzt, muss mindestens über eines der folgenden Zugriffsrechte verfügen:

- Berechtigung `SYSADM` oder `DBADM`
- Berechtigung `CREATETAB` für die Datenbank und Zugriffsrecht `USE` für den Tabellenbereich sowie eine der folgenden Berechtigungen:
 - Berechtigung `IMPLICIT_SCHEMA` für die Datenbank, wenn der implizite oder der explizite Schemaname der Tabelle nicht vorhanden ist
 - Zugriffsrecht `CREATEIN` für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist

Einschränkungen

Beim Erstellen einer fernen Tabelle unter Verwendung von transparenter DDL gelten die folgenden Einschränkungen:

- Tabellen, die ursprünglich in der fernen Datenquelle erstellt wurden, können nicht geändert oder gelöscht werden.

- MQTs (Materialized Query Tables) können nicht auf fernen Datenquellen erstellt werden.
- Sie können in der Tabellendefinition grundlegende Spalteninformationen, aber keine Tabellen- oder Spaltenoptionen angeben. Die LOB-Optionen (LOGGED und COMPACT) werden beispielsweise nicht unterstützt.
- Sie können keine Kommentare für eine Spalte angeben.
- Sie können keine Spalteninhalte generieren.
- Sie können einen Primärschlüssel, aber keinen Fremdschlüssel oder Prüfungen auf Integritätsbedingungen angeben. Die für einen Primärschlüssel verwendeten Spalten müssen NOT NULL sein und dürfen keine Spalten mit LOBs enthalten.
- Die Parameter vorhandener Spalten, z. B. der Datentyp oder die Länge, können nicht geändert werden.
- Die Klausel DEFAULT wird in der Anweisung CREATE TABLE nicht unterstützt.

Informationen zu dieser Task

Verwenden Sie den Assistenten für ferne Tabellen in der DB2-Steuerzentrale. Auf diese Weise können Sie das Angeben nicht unterstützter Parameter oder Optionen vermeiden. Mithilfe des Assistenten können Sie Spalten angeben, indem Sie sie aus einer Liste vordefinierter Spalten auswählen oder die Attribute für eine neue Spalte angeben.

Vorgehensweise

Setzen Sie zum Erstellen einer fernen Tabelle über die Befehlszeilenaufforderung die Anweisung CREATE TABLE mit den entsprechenden Parametern ab.

Verwenden Sie zum Erstellen einer fernen Tabelle in der DB2-Steuerzentrale den Assistenten 'Ferne Tabelle erstellen':

1. Erweitern Sie den Ordner **Objekte föderierter Datenbanken**.
2. Erweitern Sie die Wrapperobjekte und die Objekte der Serverdefinition für die Datenquelle, für die eine ferne Tabelle erstellt werden soll.
3. Klicken Sie mit der rechten Maustaste den Ordner **Ferne Tabellen** an, und klicken Sie anschließend **Erstellen** an. Der Assistent 'Ferne Tabelle erstellen' wird gestartet.
4. Führen Sie die Schritte im Assistenten aus.

Erstellen neuer ferner Tabellen unter Verwendung von transparenter DDL - Beispiele

In den folgenden Beispielen wird gezeigt, welche Angaben zum Erstellen ferner Tabellen unter Verwendung von transparenter DDL erforderlich sind. Ferner wird die Verwendung von Datentypzuordnungen veranschaulicht.

Beim Erstellen ferner Tabellen unter Verwendung von transparenter DDL gilt Folgendes:

- Die ferne Datenquelle muss die Spaltendatentypen und die Primärschlüsseloption in der Anweisung CREATE TABLE unterstützen.

Beispiel: Die ferne Datenquelle unterstützt keine Primärschlüssel. Je nachdem, wie die Datenquelle auf von ihr nicht unterstützte Anforderungen antwortet, wird möglicherweise ein Fehler zurückgegeben oder die Anforderung wird ignoriert.

- Der ferne Server muss in der Klausel `OPTIONS` angegeben werden. Die Klausel `OPTIONS` kann verwendet werden, um den fernen Namen oder das ferne Schema der erstellten Tabelle zu überschreiben. Die Option `SQL_SUFFIX` ist am Ende der Anweisung `CREATE TABLE` zulässig. Sie können diese Option für eine beliebige relationale Datenquelle angeben, um der Anweisung `CREATE TABLE`, die in der Datenquelle abgesetzt wird, datenquellenspezifische Optionen hinzuzufügen.

Beispiel: Sie möchten die Tabelle `EMPLOY` auf einem Oracle-Server erstellen. Verwenden Sie in der Anweisung `CREATE TABLE` die DB2-Datentypen beim Angeben der einzelnen Spalten. Bei Verwendung des Befehlszeilenprozessors (CLP) lautet die Syntax zum Erstellen der Tabelle folgendermaßen:

```
CREATE TABLE EMPLOY
( EMP_NO      CHAR(6) NOT NULL,
  FIRSTNAME   VARCHAR(12) NOT NULL,
  MIDINT      CHAR(1) NOT NULL,
  LASTNAME    VARCHAR(15) NOT NULL,
  HIREDATE    DATE,
  JOB         CHAR(8),
  SALARY      DECIMAL(9,2),
  PRIMARY KEY (EMP_NO) )
OPTIONS (REMOTE_SERVER 'ORASERVER',
        REMOTE_SCHEMA 'J15USER1', REMOTE_TABNAME 'EMPLOY' )
```

EMPLOY

Der Name des Kurznamens, der der Tabelle zugeordnet ist.

REMOTE_SERVER 'ORASERVER'

Der Name, den Sie in der Anweisung `CREATE SERVER` für den Server angegeben haben. Bei diesem Wert muss die Groß-/Kleinschreibung beachtet werden.

REMOTE_SCHEMA 'J15USER1'

Der Name des fernen Schemas. Die Angabe dieses Parameters ist zwar optional, es wird jedoch empfohlen, einen Schemanamen anzugeben. Wenn dieser Parameter nicht angegeben ist, wird das Kurznamenschema für den Namen des fernen Schemas verwendet. Bei diesem Wert muss die Groß-/Kleinschreibung beachtet werden.

REMOTE_TABNAME 'EMPLOY'

Der Name der fernen Tabelle. Die Angabe dieses Parameters ist optional. Wenn dieser Parameter nicht angegeben ist, wird der Name der lokalen Tabelle für den Namen der fernen Tabelle verwendet. Dieser Wert muss ein gültiger Name in der fernen Datenquelle sein. Es darf sich dabei nicht um einen vorhandenen Tabellennamen handeln. Bei diesem Wert muss die Groß-/Kleinschreibung beachtet werden.

Im oben beschriebenen Beispiel verwendet die föderierte Datenbank regressive Datentypzuordnungen, um die DB2-Datentypen den Oracle-Datentypen zuzuordnen. Auf dem fernen Oracle-Server wird die Tabelle `EMPLOY` unter Verwendung von Oracle-Datentypen erstellt. In der folgenden Tabelle sind die Zuordnungen von den DB2-Datentypen zu den Oracle-Datentypen für die im Beispiel angegebenen Spalten dargestellt.

Tabelle 7. Ein Beispiel regressiver Datentypzuordnungen von der föderierten Datenbank zu Oracle

Spalte	In der Anweisung CREATE TABLE angegebener DB2-Datentyp	In der fernen Tabelle verwendeter Oracle-Datentyp
EMP_NO	CHAR(6) NOT NULL	CHAR(6) NOT NULL
FIRST_NAME	VARCHAR(12) NOT NULL	VARCHAR2(12) NOT NULL
MID_INT	CHAR(1) NOT NULL	CHAR(1) NOT NULL
LAST_NAME	VARCHAR(15) NOT NULL	VARCHAR2(15) NOT NULL
HIRE_DATE	DATE	DATE
JOB	CHAR(8)	CHAR(8)
SALARY	DECIMAL(9,2)	NUMBER(9,2)

Ändern ferner Tabellen unter Verwendung von transparenter DDL

Tabellen ferner Datenquellen, die über die föderierte Datenbank erstellt wurden, können unter Verwendung von transparenter DDL geändert werden. Tabellen, die direkt in der fernen Datenquelle erstellt wurden, können nicht geändert werden.

Vorbereitung

Zu den Zugriffsrechten der Berechtigungs-ID der transparenten DDL-Anweisungen muss mindestens eine der folgenden Berechtigungen gehören:

- Berechtigung SYSADM oder DBADM
- Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen:
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder der explizite Schemaname der Tabelle nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist

Zum Absetzen transparenter DDL-Anweisungen muss Ihre Berechtigungs-ID über die notwendigen Zugriffsrechte für den Kurznamen verfügen (damit der Server mit föderierten Datenbanken die Anforderung akzeptiert) sowie über die vergleichbaren Zugriffsrechte für den Server der fernen Datenquelle (damit die Datenquelle die Anforderung akzeptiert).

Einschränkungen

Beim Ändern einer fernen Tabelle unter Verwendung von transparenter DDL gelten die folgenden Einschränkungen:

- Tabellen, die ursprünglich in der fernen Datenquelle erstellt wurden, können nicht geändert werden.
- Ein vorhandener Primärschlüssel kann nicht in einer fernen Tabelle geändert oder gelöscht werden.
- Durch das Ändern einer fernen Tabelle werden alle Pakete inaktiviert, die von dem Kurznamen abhängig sind, der der fernen Tabelle zugeordnet ist.

- Die ferne Datenquelle muss die Änderungen in der Anweisung ALTER TABLE unterstützen. Nehmen Sie beispielsweise an, die ferne Datenquelle unterstützt keine Primärschlüssel. Je nachdem, wie die Datenquelle auf von ihr nicht unterstützte Anforderungen antwortet, wird möglicherweise ein Fehler zurückgegeben oder die Anforderung wird ignoriert.
- Sie können keine Kommentare für eine Spalte angeben.
- Sie können keine Spalteninhalte generieren.
- Sie können einen Primärschlüssel, aber keinen Fremdschlüssel oder Prüfungen auf Integritätsbedingungen angeben. Die für einen Primärschlüssel verwendeten Spalten müssen NOT NULL sein und dürfen keine Spalten mit LOBs enthalten.
- Die Parameter vorhandener Spalten, z. B. der Datentyp oder die Länge, können nicht geändert werden.
- Die Klausel DEFAULT in der Anweisung ALTER TABLE wird nicht unterstützt.

Informationen zu dieser Task

Sie können entweder die DB2-Steuerzentrale oder die Anweisung ALTER TABLE verwenden, um Tabellen zu ändern, die über IBM InfoSphere Federation Server und unter Verwendung von transparenter DDL erstellt wurden. Verwenden Sie die DB2-Steuerzentrale, um das Angeben nicht unterstützter Parameter oder Optionen zu vermeiden. Bei Verwendung der Anweisung ALTER TABLE haben Sie folgende Möglichkeiten:

- Hinzufügen neuer Spalten
- Hinzufügen des Primärschlüssels für die Tabelle

Verwenden Sie die Anweisung ALTER TABLE nicht, um Spaltenoptionen hinzuzufügen oder zu ändern. Verwenden Sie stattdessen die Anweisung ALTER NICKNAME.

Vorgehensweise

Setzen Sie zum Ändern einer fernen Tabelle unter Verwendung von transparenter DDL die Anweisung ALTER TABLE ab:

Beispiel: Sie möchten der fernen Tabelle EMPLOYEE, die Sie unter Verwendung von transparenter DDL erstellt haben, einen Primärschlüssel hinzufügen. Verwenden Sie die folgende Anweisung ALTER TABLE, um die Tabelle zu ändern:

```
ALTER TABLE EMPLOYEE
  ADD PRIMARY KEY (EMP_NO, WORK_DEPT)
```

Die für einen Primärschlüssel verwendeten Spalten müssen NOT NULL sein, und es darf sich nicht um Spalten handeln, die LOBs (große Objekte) enthalten.

Beispiel: Sie möchten die Spalten ORDER_DATE und SHIP_DATE der fernen Tabelle SPALTEN hinzufügen, die unter Verwendung von transparenter DDL erstellt wurde. Verwenden Sie die folgende Anweisung ALTER TABLE, um die Tabelle zu erstellen:

```
ALTER TABLE SPALTEN
  ADD COLUMN ORDER_DATE DATE
  ADD COLUMN SHIP_DATE DATE
```

Löschen ferner Tabellen unter Verwendung von transparenter DDL

Tabellen ferner Datenquellen, die über die föderierte Datenbank erstellt wurden, können unter Verwendung von transparenter DDL gelöscht werden. Tabellen, die direkt in der fernen Datenquelle erstellt wurden, können nicht gelöscht werden.

Vorbereitung

Zu den Zugriffsrechten der Berechtigungs-ID der transparenten DDL-Anweisungen muss mindestens eine der folgenden Berechtigungen gehören:

- Berechtigung SYSADM oder DBADM
- Berechtigung CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich sowie eine der folgenden Berechtigungen:
 - Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder der explizite Schemaname der Tabelle nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Tabelle auf ein vorhandenes Schema verweist

Zum Absetzen transparenter DDL-Anweisungen muss Ihre Berechtigungs-ID über die notwendigen Zugriffsrechte für den Kurznamen verfügen (damit der Server mit föderierten Datenbanken die Anforderung akzeptiert) sowie über die vergleichbaren Zugriffsrechte für den Server der fernen Datenquelle (damit die Datenquelle die Anforderung akzeptiert).

Einschränkungen

Tabellen, die ursprünglich in der fernen Datenquelle erstellt wurden, können nicht gelöscht werden.

Informationen zu dieser Task

Zum Löschen einer fernen Tabelle, die über die föderierte Datenbank unter Verwendung von transparenter DDL erstellt wurde, können Sie entweder die DB2-Steuerzentrale oder die Anweisung DROP verwenden.

Durch das Löschen eines Kurznamens für eine ferne Tabelle, die unter Verwendung von transparenter DDL erstellt wurde, wird nur der lokale Kurzname für diese Tabelle gelöscht. Durch die Anweisung DROP NICKNAME wird nicht die ferne Tabelle gelöscht. Sie müssen die Anweisung DROP TABLE verwenden, um die ferne Tabelle zu löschen.

Beim Löschen einer fernen Tabelle wird die Tabelle zuerst in der Datenquelle gelöscht, und anschließend wird der zugehörige Kurzname für die ferne Tabelle in der föderierten Datenbank gelöscht. Durch das Löschen des Kurznamens werden alle auf diesem Kurznamen basierenden Pakete inaktiviert.

Vorgehensweise

Setzen Sie zum Löschen einer fernen Tabelle die Anweisung DROP TABLE ab.

Beispiel: Setzen Sie zum Löschen einer Tabelle mit dem Namen SPALTEN die folgende DROP-Anweisung ab:

```
DROP TABLE SPALTEN
```

Dabei ist *SPALTEN* der lokale Name für die ferne Tabelle.

Kapitel 8. Verwalten von Transaktionen in einem System föderierter Datenbanken

Die Transaktionsverarbeitung in einem System föderierter Datenbanken ermöglicht Ihnen das Lesen und Aktualisieren von Datenbanken in einer einzelnen Transaktion, wobei die Konsistenz der Daten erhalten bleibt. Systeme föderierter Datenbanken unterstützen Protokolle für einphasiges und zweiphasiges Commit. Sie müssen bei der Verwaltung Ihres Systems föderierter Datenbanken das entsprechende Protokoll definieren.

Transaktionsunterstützung in föderierten Systemen

Kenntnisse der Konzepte zur Transaktionsverarbeitung in verteilten DB2 Database für Linux-, UNIX- und Windows-Umgebungen erleichtern Ihnen das Verständnis von Transaktionen in föderierten Systemen.

Sie sollten mit folgenden Konzepten für verteilte Transaktionsverarbeitung vertraut sein, um die Transaktionsverarbeitung in föderierten Systemen zu verstehen:

- Arbeitseinheit (UOW = Unit of Work)
- Ferne Arbeitseinheit (RUOW = Remote Unit of Work)
- Verteilte Arbeitseinheit (DUOW = Distributed Unit of Work)
- Aktualisierung auf mehreren Systemen
- Transaktionsmanager (TM)
- Ressourcenmanager (RM)
- Typ-1-Verbindung
- Typ-2-Verbindung
- Einphasiges Commit
- Zweiphasiges Commit

Die Funktionsweise dieser Konzepte ist in Systemen mit und ohne föderierte Datenbanken identisch. Der Geltungsbereich der jeweiligen Konzepte ändert sich jedoch in einem föderierten System.

Eine Arbeitseinheit beispielsweise beginnt implizit, wenn Daten in einer Datenbank geschrieben oder gelesen werden. Bei einer Arbeitseinheit in einem föderiertem System kann die Datenbank eine föderierte oder eine Datenquellendatenbank sein. Bei einer verteilten Arbeitseinheit in einem föderiertem System können Sie sowohl auf eine föderierte als auch auf eine Datenquellendatenbank zugreifen.

Eine Anwendung muss eine Arbeitseinheit durch Absetzen einer COMMIT- oder ROLLBACK-Anweisung beenden, unabhängig von der Anzahl der Datenbanken, auf die zugegriffen wird. Mit der Anweisung COMMIT werden alle Änderungen in einer Arbeitseinheit dauerhaft gemacht. Die Anweisung ROLLBACK entfernt diese Änderungen aus einer Datenbank. Die von einer Arbeitseinheit vorgenommenen Änderungen werden für andere Anwendungen nach einem erfolgreichen Commit sichtbar.

Empfehlung: Führen Sie für Arbeitseinheiten in Ihren Anwendungen stets eine explizite COMMIT- oder ROLLBACK-Operation aus.

Die Daten in einer verteilten Arbeitseinheit, zu der Aktualisierungen mehrerer Datenbanken auf verschiedenen Systemen gehören, müssen konsistent sein. Mit dem Protokoll für die Aktualisierung auf mehreren Systemen oder für zweiphasiges Commit wird die Datenkonsistenz in mehreren Datenbanken innerhalb einer verteilten Arbeitseinheit sichergestellt.

Föderierte Transaktionen unterstützen das Protokoll für einphasiges und für zweiphasiges Commit. Mit der Serveroption `DB2_TWO_PHASE_COMMIT` wird Unterstützung für zweiphasiges Commit in folgenden Datenquellen aktiviert:

- Datenquellen der DB2-Produktfamilie
- Informix
- Oracle
- Sybase
- MS SQL Server

Wenn eine Datenquelle als föderierte Datenquelle für zweiphasiges Commit deklariert wird, d. h., wenn die Serveroption `DB2_TWO_PHASE_COMMIT` auf „Y“ eingestellt wird, erfolgt ein Commit für diese Datenquelle mithilfe des Protokolls für zweiphasiges Commit, auch wenn es sich bei der Transaktion um eine Aktualisierung auf einem oder mehreren Systemen handelt.

Wenn eine Datenquelle als föderierte Datenquelle für einphasiges Commit deklariert wird (Standardeinstellung) und es sich bei der Transaktion um eine Aktualisierung auf einem einzigen System handelt, erfolgt ein Commit für diese Datenquelle mithilfe des Protokolls für einphasiges Commit.

Im folgenden Beispiel für ein einphasiges Commit wird Oracle als Datenquelle für einphasiges Commit definiert:

```
SELECT * FROM oracle_nickname  
UPDATE oracle_nickname  
COMMIT
```

Im folgenden Beispiel für ein zweiphasiges Commit werden Oracle und DRDA als Datenquellen für zweiphasiges Commit definiert:

```
SELECT * FROM oracle_nickname  
UPDATE oracle_nickname  
  
SELECT * FROM drda_nickname  
UPDATE drda_nickname  
COMMIT
```

Was ist eine Aktualisierung in einem föderierten System?

Eine Aktualisierung in einem föderierten System ist mehr als nur eine Transaktion, die eine `INSERT`-, `UPDATE`- oder `DELETE`-Anweisung umfasst. Es gibt bestimmte Operationen, die in einem föderierten System als Aktualisierungen betrachtet werden, und bestimmte Aktualisierungstypen, die in einem föderierten System zulässig sind.

In einem föderierten System können Aktualisierungen lokal oder per Fernzugriff ausgeführt werden.

- Unter Aktualisierungen lokaler Systeme versteht man Aktualisierungen von DB2-Tabellen oder -Sichten, die keine Verweise auf Kurznamen enthalten.
- Aktualisierungen ferner Systeme sind Aktualisierungen von Objekten an einer fernen Datenquelle. Zu fernen Datenquellen zählen:

- Eine andere DB2 Database für Linux-, UNIX- und Windows-Datenbank oder -Instanz auf dem Server mit föderierten Datenbanken
- Eine andere DB2 Database für Linux-, UNIX- und Windows-Datenbank oder -Instanz auf einem anderen Server
- Andere Datenquellen als DB2 Database für Linux, UNIX und Windows, z. B. DB2 für System i, Informix, Oracle und Teradata

Vier Aktionstypen werden vom Server mit föderierten Datenbanken als Aktualisierungstransaktionen betrachtet. Die folgende Tabelle zeigt die Aktualisierungen, die Sie in einem föderierten System ausführen können.

Tabelle 8. Aktualisierungstypen und das System, auf dem die Aktualisierungen ausgeführt werden

Aktionstyp	Lokales System	Fernes System	Erläuterung
Lokale Aktualisierung (DDL und DML)	J	N	Aktualisierung eines Objekts in der föderierten Datenbank.
Ferne Aktualisierung (Kurzname)	N	J	Aktualisierung eines fernen Datenquellenobjekts, für das Sie einen Kurznamen erstellt haben.
Dynamisches SQL in Durchgriffssitzungen	N	J	Aktualisierung eines fernen Datenquellenobjekts. Sie können mit einer Durchgriffssitzung keine lokalen Objekte aktualisieren. Selbst in Durchgriffssitzungen gesendete SELECT-Abfragen werden als Aktualisierungsaktion betrachtet.
Transparente DDL	J	J	Ein Transaktionspaar zum Erstellen, Ändern oder Löschen ferner Tabellen und ihrer zugehörigen Kurznamen in einer föderierten Datenbank. Beispiel: ein Transaktionspaar, mit dem eine ferne Tabelle in einer Datenquelle und ein Kurzname auf einem Server mit föderierten Datenbanken erstellt werden.

Was ist eine Aktualisierungstransaktion in einer Durchgriffssitzung?

Ein Server mit föderierten Datenbanken betrachtet alle dynamischen SQL-Anweisungen, die über Durchgriffssitzungen gesendet werden, als Aktualisierungen. Dieses Verhalten gewährleistet die Datenintegrität.

Wenn eine dynamische SQL-Anweisung, die über eine Durchgriffssitzung gesendet wird, erfolgreich ist, wird die Transaktion als Aktualisierung aufgezeichnet. Dabei kann es sich um beliebige SQL-Anweisungen handeln, einschließlich SELECT-Anweisungen.

Datenquellen, die DDL-Anweisungen automatisch festschreiben

Einige Datenquellen wie Oracle schreiben die aktuelle Transaktion im Rahmen einer DDL-Anweisungsdurchführung automatisch auf den Systemen ihrer Datenquellen fest.

Wenn Sie eine ferne Tabelle mit einer transparenten DDL oder in einer Durchgriffs-sitzung erstellen, können diese Datenquellen nach der Tabellenerstellung kein Roll-back der fernen Tabelle durchführen. Sie müssen die ferne Tabelle manuell löschen.

Benutzerdefinierte Funktionen, die zur Verarbeitung in eine Datenquelle verschoben werden

Wenn eine benutzerdefinierte Funktion eine Aktualisierung in einer Datenquelle ausführt, erhält der Server mit föderierten Datenbanken keine Informationen über diese Aktualisierung.

Da der Server mit föderierten Datenbanken diese benutzerdefinierten Funktionen nicht als Aktualisierungsanweisungen betrachtet, ist der gesamte Schutz, den das föderierte System auf die Aktualisierungsoperationen anwendet, nicht anwendbar. Folglich kann die Datenintegrität in bestimmten Situationen beeinträchtigt werden.

Wichtig: Die Datenintegrität ist nicht gewährleistet, wenn eine benutzerdefinierte Funktion, die in eine Datenquelle verschoben wird, eine Aktualisierung ausführt.

Kapitel 9. Ausführen von Transaktionen mit zweiphasigem Commit

Sie können die Funktionalität für zweiphasiges Commit eines Systems föderierter Datenbanken nutzen, um Daten für eine oder mehrere Datenquellen in einer einzelnen Transaktion zu aktualisieren.

Zweiphasiges Commit für föderierte Transaktionen

Ein föderiertes System kann zweiphasiges Commit für Transaktionen verwenden, die auf eine oder mehrere Datenquellen zugreifen. Bei zweiphasigem Commit wird die Verarbeitung von DUOW-Transaktionen mithilfe des Industriestandardprotokolls X/Open XA koordiniert.

Bei zweiphasigem Commit erfolgt die Commitverarbeitung in zwei Phasen: der Vorbereitungsphase und der Commitphase. In der Vorbereitungsphase auf einem föderierten System fragt ein Server mit föderierten Datenbanken alle föderierten Datenquellen für zweiphasiges Commit ab, die in eine Transaktion einbezogen sind. Mit dieser Abfrage wird geprüft, ob die einzelnen Datenquellen bereit sind, die Daten festzuschreiben oder rückgängig zu machen. In der Commitphase weist der Server mit föderierten Datenbanken die einzelnen Datenquellen für zweiphasiges Commit an, die Daten festzuschreiben oder die Transaktion rückgängig zu machen.

In einer Umgebung für einphasiges Commit wird bei der Aktualisierung mehrerer Datenquellen jeweils nur eine Datenquelle mit einer separaten Commitoperation aktualisiert. Dies kann zu Problemen bei der Datensynchronisation führen, wenn einige Datenquellen erfolgreich aktualisiert werden, andere dagegen nicht.

Wenn beispielsweise bei einer Transaktion mit einphasigem Commit ein Geldbetrag von einem Konto abgehoben und auf einem anderen Konto eingezahlt wird, kann es passieren, dass das System die Abhebung erfolgreich festschreibt, die Einzahlung dagegen nicht. Die Einzahlung kann rückgängig gemacht werden, die Abhebung jedoch nicht, weil sie bereits erfolgreich festgeschrieben wurde. Im Ergebnis ist der Betrag quasi "verloren".

In einer Umgebung für zweiphasiges Commit werden die Abhebungs- und die Einzahlungstransaktion gemeinsam vorbereitet und auch gemeinsam festgeschrieben bzw. rückgängig gemacht. Im Ergebnis bleibt die Integrität des Betrags erhalten.

Planung von föderiertem zweiphasigen Commit

Föderiertes zweiphasiges Commit bietet nicht in allen Geschäftsumgebungen Vorteile. Zudem müssen Sie bestimmte Faktoren bei der Entscheidung über den Einsatz von föderiertem zweiphasigen Commit berücksichtigen.

Sie müssen folgende Punkte beim Einsatz von zweiphasigem Commit für föderierte Transaktionen berücksichtigen:

- Unterstützen Ihr Betriebssystem und Ihre Datenquellenumgebung zweiphasiges Commit für föderierte Transaktionen?

- Ist in Ihrer Geschäftsumgebung zweiphasiges Commit für föderierte Transaktionen erforderlich?

Für die Entscheidung darüber, ob zweiphasiges Commit für Ihre Geschäftsumgebung geeignet ist, müssen Sie verstehen, wie zweiphasiges Commit für föderierte Transaktionen funktioniert und welche Probleme damit gelöst werden.

- Wie werden der Server mit föderierten Datenbanken und die kompatiblen Datenquellen für die Verwendung von zweiphasigem Commit in föderierten Transaktionen konfiguriert?

Sie müssen beim Einsatz von zweiphasigem Commit Basisanforderungen an den Server mit föderierten Datenbanken und die Datenquellen für die Verwendung von zweiphasigem Commit für föderierte Transaktionen sowie leistungsbezogene Überlegungen berücksichtigen.

- Sie müssen die interne Funktionsweise von zweiphasigem Commit für föderierte Transaktionen verstehen, um unbestätigte Transaktionen manuell auflösen zu können.

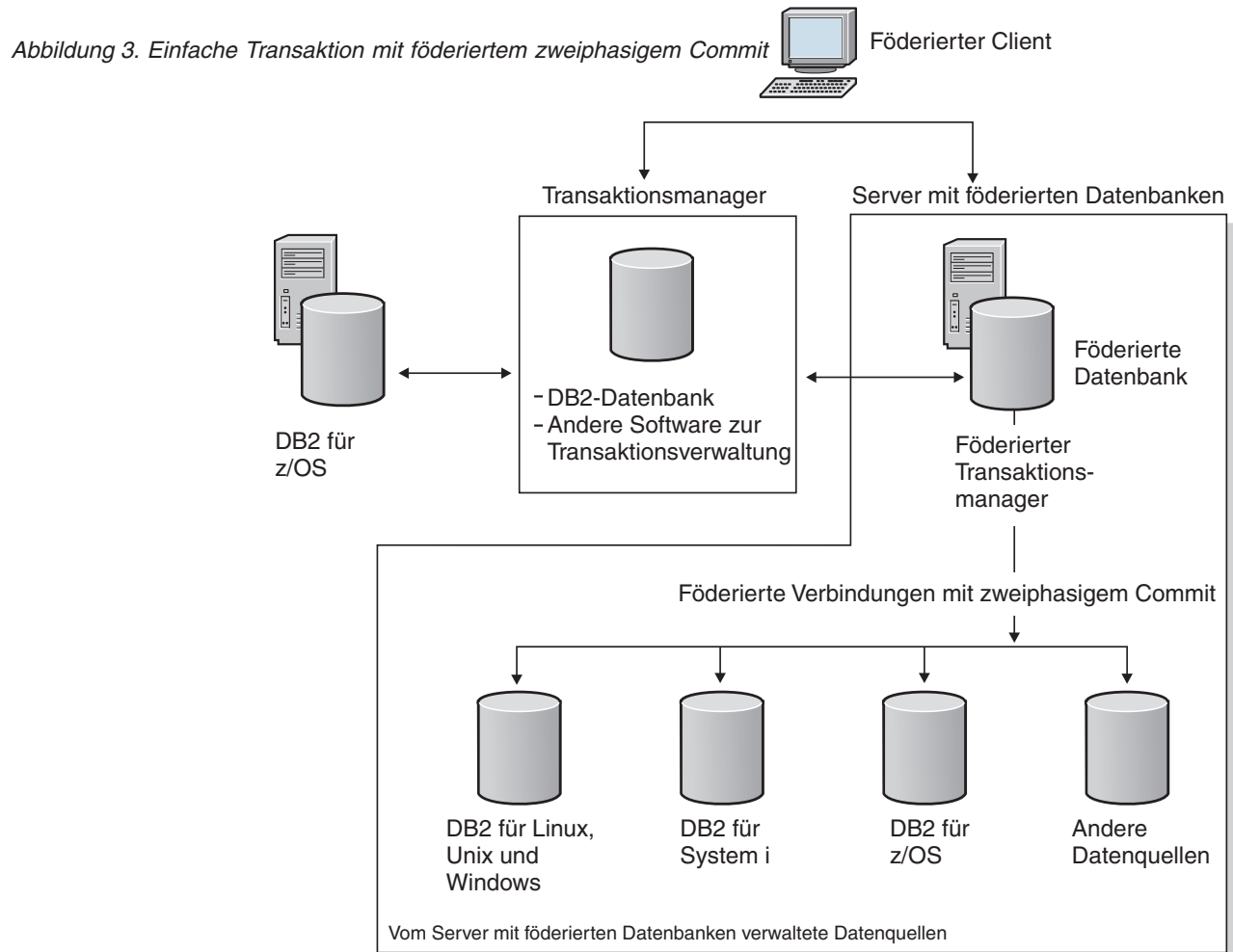
Mit zweiphasigem Commit für föderierte Transaktionen lassen sich Probleme ohne Eingriff von außen beheben. Bei größeren Netzausfallzeiten, Hardwarefehlern oder dem dringenden Bedarf zur Freigabe von Systemressourcen können Sie Probleme jedoch auch manuell durch *heuristische Verarbeitung* beheben.

Föderierte Architektur für zweiphasiges Commit

Föderiertes zweiphasiges Commit basiert auf der in DB2 verfügbaren Funktion für zweiphasiges Commit. Bei zweiphasigem Commit enthält das X/Open-Modell für die verteilte Transaktionsverarbeitung (DTP) mehrere Komponenten: Transaktions-IDs, Transaktionsmanager und Ressourcenmanager. In föderierten Systemen, die föderiertes zweiphasiges Commit verwenden, wird eine weitere Komponente hinzugefügt: der föderierte Transaktionsmanager.

Ein Server mit föderierten Datenbanken wird zu einem föderierten Transaktionsmanager, wenn er die Aktivität von einer oder mehreren fernen Datenquellen koordiniert, die das Protokoll für zweiphasiges Commit verwenden. Ein föderierter Transaktionsmanager führt bestimmte Transaktionsverwaltungsfunktionen im Auftrag des Transaktionsmanagers aus. Der Client oder die Anwendung, der bzw. die eine DOUW-Transaktion einleitet, und der Transaktionsmanager haben keine Informationen über die Aktivität, die der föderierte Transaktionsmanager in den fernen Datenquellen koordiniert. Der föderierte Transaktionsmanager kommuniziert mit den DB2-Datenbanktransaktionsmanagern mithilfe einer XA-Schnittstelle. Neben den X/Open-Anforderungen für zweiphasiges Commit muss außerdem der Zugriff auf die Transaktionsmanagerdatenbank von der föderierten Instanz möglich sein. Die Ressourcenmanager halten sich beim Commit oder Rollback einer Transaktion an die von einem Transaktionsmanager bereitgestellten Anweisungen.

Die folgende Abbildung zeigt ein Beispiel für eine einfache Transaktion mit zweiphasigem Commit in einem typischen föderierten System, von der Client-Initiierung bis zu den Datenquellenaktualisierungen.



In der obigen Abbildung ist die Verbindung vom Client zum Transaktionsmanager eine Typ-2-Verbindung. Jede Datenbankverbindung hat außerdem eine eigene Einstellung für den Synchronisationspunkt. Ein Synchronisationspunkt ist ein Zeitpunkt, an dem alle wiederherstellbaren Daten, auf die ein Programm zugreift, konsistent sind. Zweiphasige Verbindungen mit Synchronisationspunkt unterstützen DUOW-Transaktionen mit Aktualisierungen mehrerer Datenquellen.

Wenn der Client eine Verbindung zu der DB2-Datenbank herstellt, erhält der Transaktionsmanager Informationen zu der Transaktion, jedoch ist keine zusätzliche Koordination seitens des Servers mit föderierten Datenbanken erforderlich. Wenn der Server mit föderierten Datenbanken über das Protokoll für zweiphasiges Commit eine Verbindung zu den Datenquellen herstellt, wird er zum föderierten Transaktionsmanager. Der Server mit föderierten Datenbanken überwacht und koordiniert die zweiphasigen Commits. Zu diesem Zeitpunkt weiß der Transaktionsmanager nichts von den mit zweiphasigem Commit ausgeführten Transaktionen mit den Datenquellen. Der Transaktionsmanager weiß nur, dass auf dem Server mit föderierten Datenbanken eine einzelne Transaktion verarbeitet wird.

Datenquellen können bei einem Fehler in einem föderierten System keine Resynchronisation einleiten. Die Resynchronisationsverarbeitung wird vom Server mit föderierten Datenbanken eingeleitet.

Der Versuch, über mehrere Pfade in derselben Transaktion mit föderiertem zweiphasigem Commit auf eine Datenquelle zuzugreifen, kann unvorhersehbare Folgen haben. Wenn der Server mit föderierten Datenbanken beispielsweise ein Ressourcenmanager für einen externen Transaktionsmanager ist, erfolgt der Zugriff auf die Datenquelle möglicherweise indirekt über den Server und direkt als Ressourcenmanager des Transaktionsmanagers. In diesem Fall erkennt die Datenquelle unter Umständen nicht, ob diese beiden Pfade von derselben globalen Transaktion stammen. Möglicherweise erstellt die Datenquelle zwei Transaktionseinträge für dieselbe globale Transaktion und behandelt die Transaktionen als separate Vorgänge, was unvorhersehbare Ergebnisse zur Folge haben kann. Die Datenquelle erkennt jedoch möglicherweise, dass die beiden Pfade von derselben globalen Transaktion stammen, und weist den zweiten Pfad zurück.

Zweiphasiges Commit für föderierte Transaktionen - Beispiele

Ein föderiertes System, das zweiphasiges Commit verwendet, kann auf unterschiedliche Arten konfiguriert werden. Welche Konfiguration Sie wählen, richtet sich nach der erforderlichen Lösung.

Die Konfigurationen können Typ-1- oder Typ-2-Verbindungen verwenden.

Unter Typ-1-Verbindungen versteht man Verbindungen, bei denen ein Anwendungsprozess entsprechend den Regeln für RUOWs mit einem Anwendungsserver verbunden ist.

Unter Typ-2-Verbindungen versteht man Verbindungen, bei denen ein Anwendungsprozess mit einem Anwendungsserver verbunden ist und die Regeln für anwendungsorientierte RUOWs festlegt. Der Anwendungsserver ist dann der aktuelle Server für den Prozess.

Die folgende Abbildung zeigt eine DB2-Typ-1-Verbindung, bei der ein Server mit föderierten Datenbanken als Transaktionsmanager fungiert.

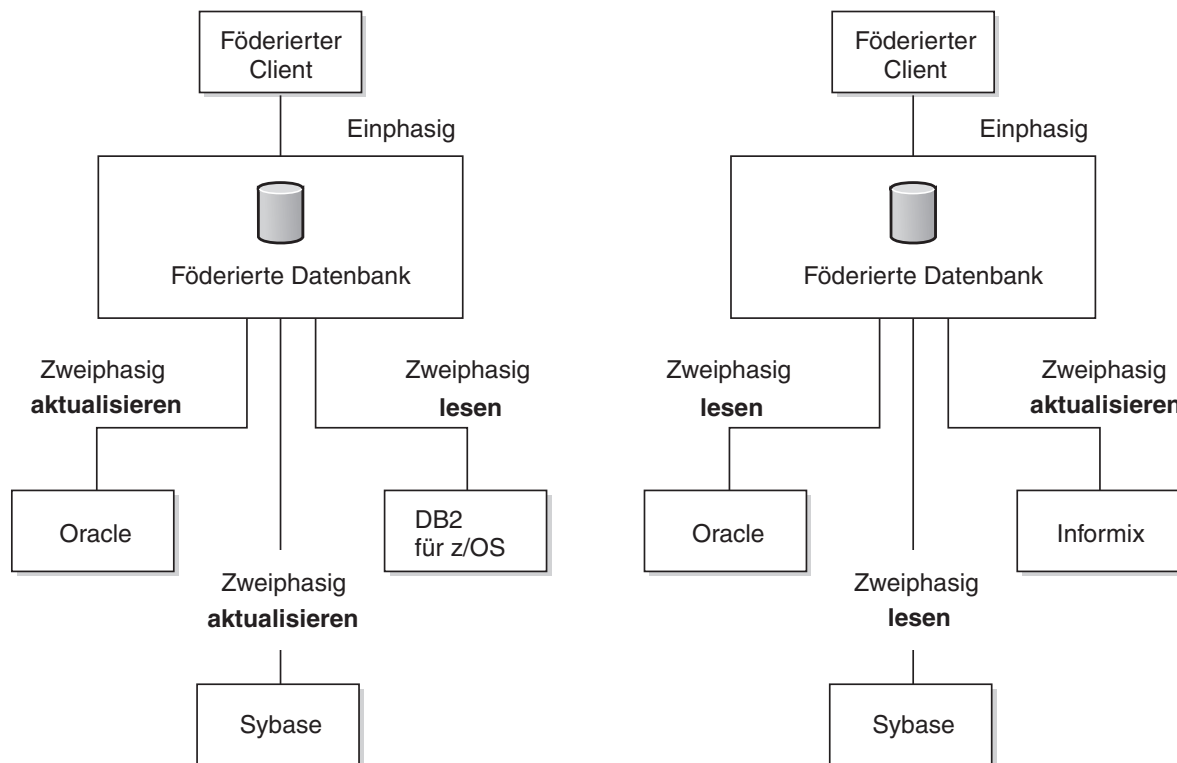


Abbildung 4. DB2-Typ-1-Verbindung mit einem Server mit föderierten Datenbanken als Transaktionsmanager

Die folgende Abbildung zeigt eine DB2-Typ-2-Verbindung, bei der ein Server mit föderierten Datenbanken als Ressourcenmanager fungiert. Bei dieser Konfiguration müssen alle föderierten Datenquellen Unterstützung für föderiertes zweiphasiges Commit bieten und für zweiphasiges Commit aktiviert sein.

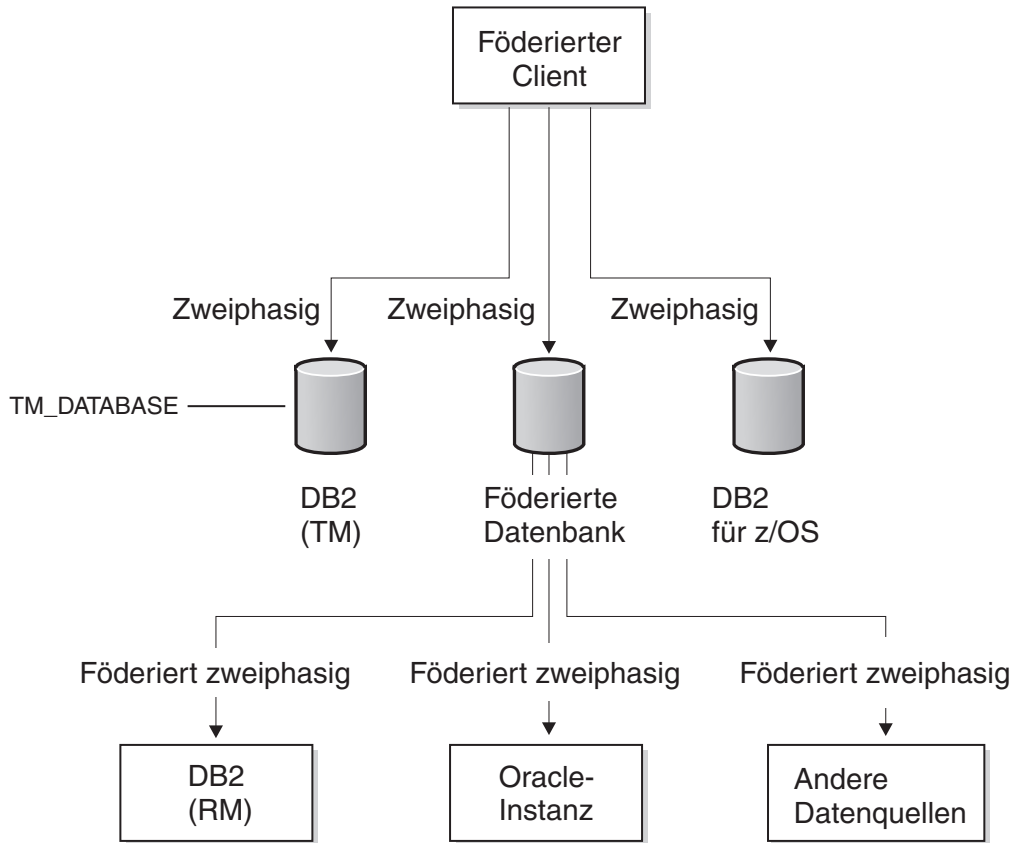


Abbildung 5. DB2-Typ-2-Verbindung mit einem Server mit föderierten Datenbanken als Ressourcenmanager

Die folgende Abbildung zeigt eine DB2-Typ-2-Verbindung, bei der ein Server mit föderierten Datenbanken als Transaktionsmanager fungiert.

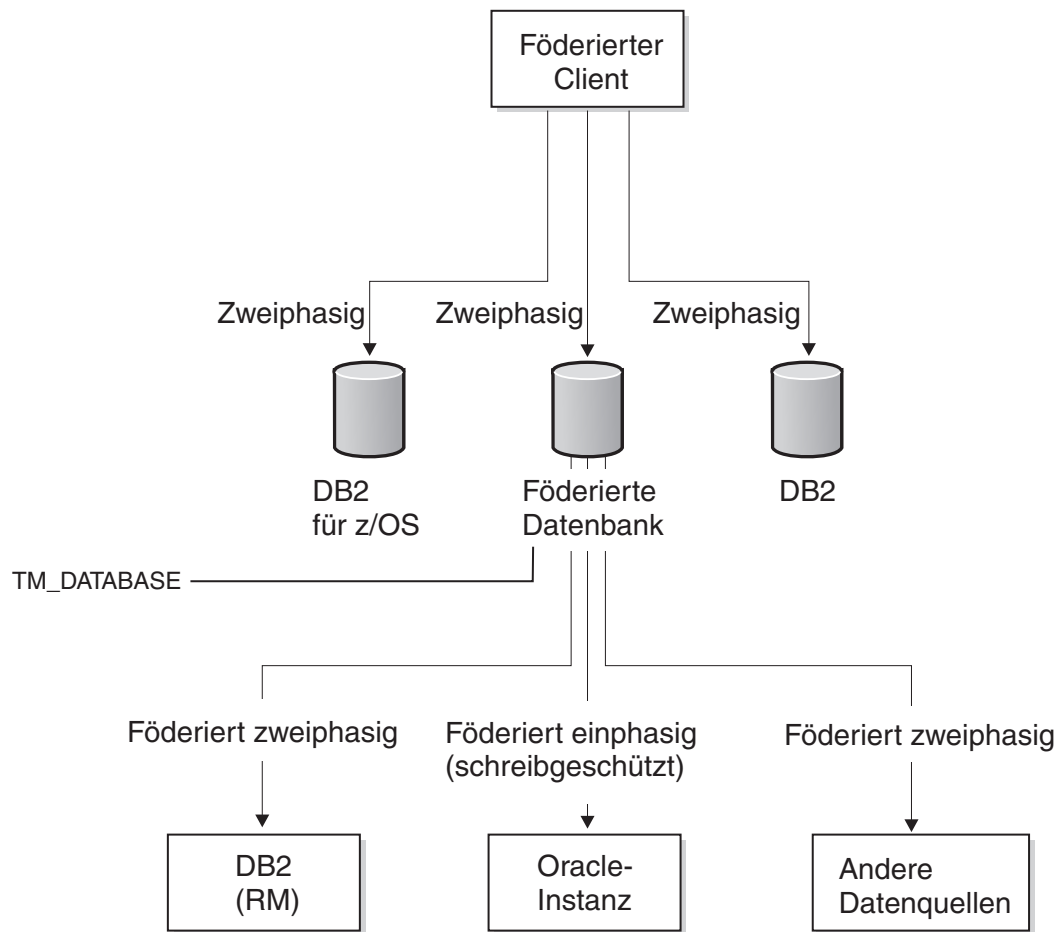


Abbildung 6. DB2-Typ-2-Verbindung mit einem Server mit föderierten Datenbanken als Transaktionsmanager

Die folgende Abbildung zeigt eine XA-Verbindung, bei der ein Server mit föderierten Datenbanken als Ressourcenmanager fungiert.

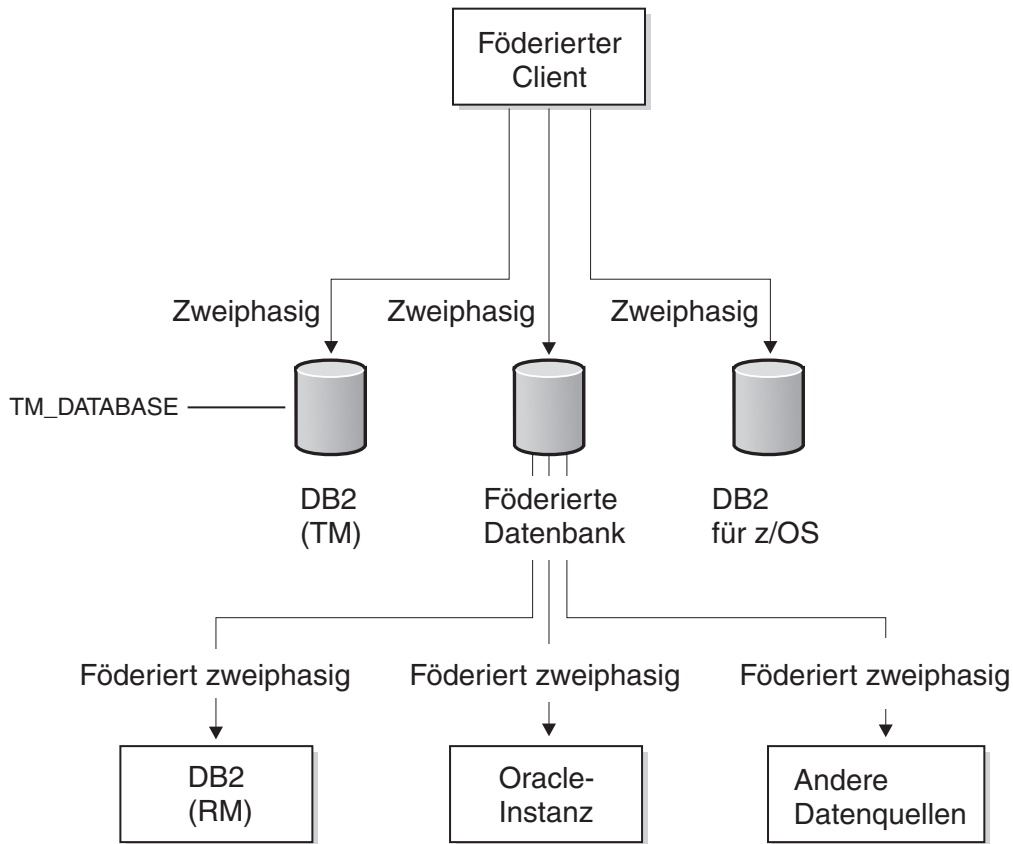


Abbildung 7. XA-Verbindung mit einem Server mit föderierten Datenbanken als Ressourcenmanager

Verfahren zur Verarbeitung von Transaktionen mit föderiertem zweiphasigem Commit

Der Server mit föderierten Datenbanken pflegt die Datenkonsistenz und Atomizität der von ihm verwalteten Datenquellen. Der Bereich der möglichen Transaktionen hängt davon ab, welcher Verbindungstyp verwendet wird und ob der Server mit föderierten Datenbanken der Transaktionsmanager oder der Ressourcenmanager für die Verbindung ist.

Atomizität ist ein Datenbankprinzip, bei dem Gruppen von Operationen innerhalb von unteilbaren Gruppen definiert werden. Mit diesem Prinzip wird sichergestellt, dass die Datenbank jederzeit konsistent ist, da beim Fehlschlagen einer einzelnen Operation innerhalb der unteilbaren Transaktion die gesamte Transaktion fehlschlägt. Damit wird vermieden, dass die Datenintegrität aufgrund einer Teiländerung beeinträchtigt wird.

Beispiel: Eine Transaktion, bei der ein Geldbetrag von einem Konto auf ein anderes übertragen werden soll, muss der Betrag von dem ersten Konto abgeboben und auf das zweite eingezahlt werden. Wenn nur die Abhebung erfolgreich ist, ist der Betrag auf dem ersten Konto quasi nicht mehr vorhanden.

Der Server mit föderierten Datenbanken verarbeitet föderierte Aktualisierungsanforderungen nach strengen Regeln. Unter einer föderierten Aktualisierung versteht man eine der folgenden Aktionen:

- Eine föderierte Einfüge-, Aktualisierungs- oder Löschoption, bei der die entsprechende Datenquelle Einfüge-, Aktualisierungs- oder Löschoptionen unterstützt. Bestimmte Datenquellen unterstützen beispielsweise keine Aktualisierungsoperationen. Andere Datenquellen sind schreibgeschützt, was zur Folge hat, dass Einfüge-, Aktualisierungs- oder Löschoptionen nicht zulässig sind.
- Eine erfolgreiche Durchgriffsoperation innerhalb einer Durchgriffssitzung.
- Eine transparente DDL-Operation, die gleichermaßen als lokale und föderierte Aktualisierung betrachtet wird, da bei ihr sowohl eine lokale als auch eine Datenbankaktualisierung über Remotezugriff ausgeführt wird.
- Eine föderierte gespeicherte Prozedur mit `MODIFY SQL ACCESS`.

Anmerkung: Greifen Sie auf eine Datenquelle innerhalb einer einzelnen Transaktion nicht über mehrere Pfade zu. Eine solche Transaktion kann in einem Deadlock enden. Das bedeutet, dass die Transaktion blockiert ist. Verweisen Sie beispielsweise in einer Transaktion nicht mit mehreren Servern mit föderierten Datenbanken auf dieselben Datenquellen.

In der folgenden Tabelle sind die Vorgänge bei einer einzelnen DUOW-Transaktion aufgeführt: der Verbindungstyp, der Committyp, die Rolle des Servers mit föderierten Datenbanken bei der Transaktion, die zulässigen Operationen und die Verwendungsweise einer transparenten DDL.

Tabelle 9. Vorgänge bei einer einzelnen DUOW-Transaktion

Commit-typ	Verbindungstyp	Rolle des Servers mit föderierten Datenbanken	Operationen	Transparente DDL
Einphasig	Lokale DB2-Typ-1- oder XA-Transaktion	Subtransaktionsmanager. Tritt auch als Transaktionskoordinator auf, ermittelt das Ergebnis der Transaktion und übermittelt es an die mitwirkenden Ressourcenmanager.	Leseoperationen mit ein- und zweiphasigem Commit sind zulässig. Genau eine einphasige Datenquelle kann aktualisiert werden, solange das die einzige Aktualisierung bei der Transaktion ist.	Zulässig und verwaltet entsprechend den Regeln für Datenquellen mit einphasigem Commit. Jede Anweisung, die ausgegeben wird, muss die einzige Aktualisierung bei einer Transaktion mit einphasigem Commit sein. Die Koexistenz mit anderen Aktualisierungen von föderierten Datenquellen für zweiphasiges Commit in derselben Transaktion ist nicht möglich. Es wird dringend empfohlen, COMMIT- oder ROLLBACK-Anweisungen abzusetzen, bevor und nachdem transparente DDL-Transaktionen erfolgen.
Zweiphasig	Lokale DB2-Typ-1- oder XA-Transaktion	Transaktionsmanager. Tritt auch als Transaktionskoordinator auf, ermittelt das Ergebnis der Transaktion und übermittelt es an die mitwirkenden Ressourcenmanager.	Leseoperationen mit ein- und zweiphasigem Commit sind zulässig. Es können mehrere zweiphasige Datenquellen aktualisiert werden.	Zulässig und verwaltet entsprechend den Regeln für Datenquellen mit zweiphasigem Commit. Die Koexistenz mit anderen Aktualisierungen von föderierten Datenquellen für zweiphasiges Commit in derselben Transaktion ist möglich.

Tabelle 9. Vorgänge bei einer einzelnen DUOW-Transaktion (Forts.)

Commit-typ	Verbindungstyp	Rolle des Servers mit föderierten Datenbanken	Operationen	Transparente DDL
Einphasig	Globale DB2-Typ-2- oder XA-Transaktion	Kann der Transaktionsmanager sein. Falls nicht der Transaktionsmanager, wird nur das Ergebnis vom externen Transaktionskoordinator an die beteiligten Ressourcenmanager weitergeleitet.	Leseoperationen mit ein- und zweiphasigem Commit sind zulässig. Einphasige Aktualisierungen sind nicht zulässig, mit Ausnahme von mit DB2 koordinierten Transaktionen, die über eine zweiphasige eingehende DRDA-Verbindung (Distributed Relational Database Architecture) föderierte einphasige Aktualisierungen ausführen können.	Zulässig und verwaltet entsprechend den Regeln für Datenquellen mit einphasigem Commit. Jede Anweisung, die ausgegeben wird, muss die einzige Aktualisierung bei einer Transaktion mit einphasigem Commit sein. Die Koexistenz mit anderen Aktualisierungen von föderierten Datenquellen für zweiphasiges Commit in derselben Transaktion ist nicht möglich. Es wird dringend empfohlen, COMMIT- oder ROLLBACK-Anweisungen abzusetzen, bevor und nachdem transparente DDL-Transaktionen erfolgen.
Zweiphasig	Globale DB2-Typ-2- oder XA-Transaktion	Kann der Transaktionsmanager sein. Falls nicht der Transaktionsmanager, wird nur das Ergebnis vom externen Transaktionskoordinator an die beteiligten Ressourcenmanager weitergeleitet.	Leseoperationen mit ein- und zweiphasigem Commit sind zulässig. Es können mehrere zweiphasige Datenquellen aktualisiert werden.	Zulässig und verwaltet entsprechend den Regeln für Datenquellen mit zweiphasigem Commit. Die Koexistenz mit anderen Aktualisierungen von föderierten Datenquellen für zweiphasiges Commit in derselben Transaktion ist möglich.

Verfahren zur Pflege der Datenkonsistenz und Atomizität

Server mit föderierten Datenbanken versuchen, Datenkonsistenz zu gewährleisten und die Transaktionsatomizität von Datenquellen sicherzustellen.

Konflikte zwischen der Synchronisationspunkteinstellung einer Anwendung und der Aktualisierungsfunktion einer Zieldatenquelle haben einen Fehler (SQL30090, Ursachencode 18) zur Folge.

Lokale Aktualisierungen inklusive DDL, die an der föderierten Datenbank vorgenommen werden, können innerhalb derselben Transaktion nicht als Aktualisierung einer föderierten einphasigen Datenquelle gemischt werden. Transparente DDL

Verwendung von DDL und transparenter DDL

Lokale Aktualisierungen inklusive DDL, die an der föderierten Datenbank vorgenommen werden, können innerhalb derselben Transaktion nicht als Aktualisierung einer föderierten einphasigen Datenquelle gemischt werden. Eine transparente DDL stellt eine Ausnahme dar. Bei einer transparenten DDL sind sowohl lokale Aktualisierungen als auch Datenquellenaktualisierungen unabhängig davon zulässig, welcher Verbindungstyp verwendet wird und ob die Datenquelle für ein- oder zweiphasiges Commit konfiguriert ist.

Eine transparente DDL erstellt in einer fernen Datenquelle eine Tabelle und in der lokalen föderierten Datenbank einen Kurznamen für die ferne Tabelle. Ein Server mit föderierten Datenbanken betrachtet transparente DDL-Transaktionen als Aktualisierungen.

Eine transparente DDL bietet die Möglichkeit, ferne Tabellen über das DB2-Datenbanksystem zu erstellen und zu modifizieren, ohne Durchgriffssitzungen verwenden zu müssen. Die SQL-Anweisungen für transparente DDL lauten: CREATE TABLE, ALTER TABLE und DROP TABLE. Beispiel: Mit einer CREATE TABLE-Anweisung einer transparenten DDL werden eine ferne Tabelle in einer Datenquelle und ein Kurzname für diese Tabelle auf einem Server mit föderierten Datenbanken erstellt. Die Anweisung enthält eine lokale und eine ferne Aktualisierungsoperation.

Einige Datenquellen, z. B. Oracle, lassen keine transparente DDL in einer föderierten zweiphasigen Commitverbindung zu.

Aktivieren von zweiphasigem Commit für föderierte Transaktionen

Für den Einsatz von zweiphasigem Commit für bestimmte Datenquellen müssen Sie die zugehörigen Server mit föderierten Datenbanken aktivieren. Der Aktivierungsprozess umfasst die Vorbereitung des Servers mit föderierten Datenbanken und die Änderung der Serverdefinition der Datenquelle.

Vorbereitung

- Wenn Sie föderiertes zweiphasiges Commit für eine Datenquelle aktivieren, erhöhen Sie die Anzahl der Datensätze, die in das Datenbankprotokoll des Servers mit föderierten Datenbanken und der Datenquelle geschrieben werden. Überlegen Sie, welche Auswirkungen dies auf die Verwaltung und Pflege dieser Protokolldateien hat, um sicherzustellen, dass sie Ihre lokalen Richtlinien einhalten.
- Sie müssen als Datenquelle eine unterstützte föderierte Datenquelle für zweiphasiges Commit verwenden.
- Föderiertes zweiphasiges Commit wird in einer MPP-Umgebung (MPP = Massively Parallel Processing, exklusive Parallelverarbeitung) oder einer abgeschirmten Umgebung (mit Ausnahme von Sybase) nicht unterstützt. Nur für Sybase unter UNIX wird föderiertes zweiphasiges Commit in der abgeschirmten Umgebung unterstützt.
- Für Datenquellen von DB2 für System i Version 5.3 und frühere Versionen sowie für Datenquellen von DB2 für z/OS müssen Sie sicherstellen, dass für den Konfigurationsparameter SPM_NAME der Standardwert - der Hostname des Servers - definiert ist. SPM_NAME nimmt standardmäßig eine Variante der ersten sieben Zeichen des TCP/IP-Hostnamens an. Bei DB2 für System i ab Version 5.4 müssen Sie SPM_NAME nicht definieren.

Informationen zu dieser Task

Mit der Serveroption `DB2_TWO_PHASE_COMMIT` wird zweiphasiges Commit für Datenquellen aktiviert. Die Registrierung einer Serverdefinition für eine Datenquelle erfolgt mit der Anweisung `CREATE SERVER`. Der Wert, den Sie für `DB2_TWO_PHASE_COMMIT` festlegen, bleibt für alle Verbindungen bestehen, die mit dieser Serverdefinition hergestellt werden. Sie können den Wert jederzeit mit der Anweisung `ALTER SERVER` ändern. Nachdem die Anweisung `CREATE SERVER` oder `ALTER SERVER` erfolgreich festgeschrieben wurde, steht die neue Einstellung dem Benutzer bei künftigen Anforderungen von abgehenden Verbindungen zur Verfügung.

Clients und Anwendungsprogramme können mit der Anweisung `SET SERVER OPTION` den aktuellen Wert der Serveroption `DB2_TWO_PHASE_COMMIT` zeitweilig überschreiben. Die Anweisung `SET SERVER OPTION` muss sofort ausgeführt werden, nachdem die Verbindung zur Datenbank des Servers mit föderierten Datenbanken hergestellt wurde und bevor eine Verbindung zu den fernen Datenquellen hergestellt wird. Der Befehl ist nur so lange wirksam, wie die Verbindung mit der föderierten Datenbank besteht. Sobald der Server mit föderierten Datenbanken eine Verbindung mit der fernen Datenquelle hergestellt hat, können Sie die Serveroption `DB2_TWO_PHASE_COMMIT` nicht mehr ändern.

Wenn Sie die Option `XA_OPEN_STRING_OPTIONS` in eine `CREATE SERVER`-Anweisung einschließen, können Sie spezielle Informationen in die `XA_OPEN`-Standardzeichenfolge einbetten. Folgende Informationen können eingebettet werden:

- Eindeutige IDs für Transaktionen zusätzlich zu den von IBM InfoSphere Federation Server bereitgestellten IDs
- Benutzerdefinierte Parameter zur Art der Verarbeitung von Transaktionen
- Eine benutzerdefinierte Zeichenfolge zum Anhängen an die `XA_OPEN`-Anforderung

Bei einem `XA_OPEN`-Aufruf hängt der Wrapper normalerweise am Anfang der ersten Transaktion zu einer fernen Datenquelle, bei der zweiphasiges Commit verwendet wird, den Wert der benutzerdefinierten Zeichenfolge an die `XA_OPEN`-Standardzeichenfolge für den `XA_OPEN`-Aufruf an.

Sie können sowohl `DB2_TWO_PHASE_COMMIT` als auch `XA_OPEN_STRING_OPTIONS` in eine `CREATE SERVER`-, `SET SERVER`- oder `ALTER SERVER`-Anweisung einschließen.

Vorgehensweise

Im Allgemeinen müssen Sie wie folgt vorgehen, um zweiphasiges Commit für eine föderierte Datenquelle zu aktivieren:

1. Führen Sie die Anweisung `CREATE SERVER`, `ALTER SERVER` oder `SET SERVER` mit der Einstellung `Y` für die Option `DB2_TWO_PHASE_COMMIT` aus.
2. Optional: Führen Sie die Anweisung `CREATE SERVER`, `ALTER SERVER` oder `SET SERVER` mit der Option `XA_OPEN_STRING_OPTIONS` aus.

Beispiele für Serveroptionen

Dieses Beispiel zeigt, wie zweiphasiges Commit mit der Anweisung `CREATE SERVER` festgelegt wird:

```
CREATE SERVER Net8_Server TYPE ORACLE VERSION 8.1.7 WRAPPER NET8
OPTIONS (DB2_TWO_PHASE_COMMIT 'Y');
```

Dieses Beispiel zeigt, wie zweiphasiges Commit mit der Anweisung `ALTER SERVER` inaktiviert wird:

```
ALTER SERVER Net8_Server OPTIONS (SET DB2_TWO_PHASE_COMMIT 'N');
```

Dieses Beispiel zeigt, wie eine XA-Tracedatei für den Sybase-Wrapper mit der Anweisung `ALTER SERVER` und der Serveroption `XA_OPEN_STRING_OPTIONS` auf `D:\Temp\sybase_xa.log` festgelegt wird:

```
ALTER SERVER Ctlib_Server OPTIONS (ADD XA_OPEN_STRING_OPTIONS
'-LD:\Temp\sybase_xa.log');
```

Dieses Beispiel zeigt, wie zweiphasiges Commit mit der Anweisung `SET SERVER OPTION` zeitweilig inaktiviert wird:

```
SET SERVER OPTION DB2_TWO_PHASE_COMMIT TO 'N' FOR SERVER Net8_Server;
```

Anforderungen an Datenquellen und deren Konfiguration für Transaktionen mit föderiertem zweiphasigem Commit

Bevor Sie zweiphasiges Commit für eine Datenquelle aktivieren, müssen Sie sicherstellen, dass die Datenquelle unterstützt wird.

Föderierte Systeme unterstützen zweiphasiges Commit bei folgenden Datenquellen:

- Datenquellen der DB2-Produktfamilie über das Protokoll DRDA (Distributed Relational Database Architecture):
 - DB2 Universal Database für Linux, UNIX und Windows Version 8.1 oder höher
 - DB2 Universal Database für z/OS Version 7.1 oder höher
 - DB2 Universal Database für System i ab Version 5.3
- Informix IDS Version 7.31 oder höher, Version 9.40 oder höher, Version 10.0 oder höher
- Informix XPS Version 8.40 oder höher
- Microsoft SQL Server 2000 und Microsoft SQL Server 2005 für einen Server mit föderierten Datenbanken, nur unter Windows
- Oracle Version 8.1.7 oder höher, mit der XA-Bibliothek
- Sybase Adaptive Server Enterprise Version 12 oder höher, mit der XA-Bibliothek für einen Server mit föderierten Datenbanken, nur unter Windows

Wenn Sie versuchen, föderiertes zweiphasiges Commit für eine nicht unterstützte Datenquelle zu aktivieren, wird der Fehler SQL1881N zurückgegeben.

Konfigurieren von DRDA-Datenquellen

Der Server mit föderierten Datenbanken stellt mithilfe des offenen Protokolls DRDA Konnektivität für DB2-Datenquellen bereit. Dies entspricht der vom DB2 Connect-Server bereitgestellten Unterstützung.

Darüber hinaus interagiert der Server mit föderierten Datenbanken bei zweiphasigem Commit über das standardisierte XA-Modell mit den einzelnen Datenquellen.

Vorbereitung

Einschränkungen:

- Nicht alle DB2-Datenquellen bieten native Unterstützung für XA über DRDA. Bei den Datenquellen, die diese Unterstützung nicht bieten, wie z. B. DB2 für z/OS und DB2 für System i, verwendet der Server mit föderierten Datenbanken den Synchronisationspunktmanager (SPM). Der Synchronisationspunktmanager führt eine Zuordnung zwischen den mit XA und den nicht mit XA gesteuerten zweiphasigen Commitabläufen durch, die alle DB2-Server unterstützen. Bei der Bereitstellung des Zugriffs auf föderiertes zweiphasiges Commit über den Synchronisationspunktmanager wird aufgrund von Inkompatibilitäten zwischen der föderierten Unterstützung und dem Synchronisationspunktmanager nicht die gesamte XA-Semantik unterstützt. Beispielsweise ist es nicht möglich, Transaktionen zu verschachteln. Alle Transaktionen müssen festgeschrieben oder rückgängig gemacht werden, bevor eine neue Transaktion gestartet werden kann.
- Föderiertes zweiphasiges Commit unterstützt DB2 für z/OS; DB2 für z/OS lässt jedoch das Absetzen einer Anweisung SAVEPOINT in einer Transaktion mit föderiertem zweiphasigem Commit nicht zu.

- In einer mit DB2 koordinierten Transaktion kann ein DB2 für z/OS-Client eine föderierte einphasige Aktualisierung über eine zweiphasige eingehende DRDA-Verbindung durchführen. Allerdings kann eine solche Aktualisierung nicht über eine zweiphasige eingehende XA DRDA-Verbindung ausgeführt werden. Eben- sowenig ist eine Mischung von ein- und zweiphasigen Aktualisierungen über eine zweiphasige eingehende DRDA-Verbindung möglich.
- Die Serveroption XA_OPEN_STRING_OPTIONS wird bei DRDA-Datenquellen nicht unterstützt. Wenn Sie die Option verwenden, wird der Fehler SQL1881 zurückgegeben.

Anforderungen:

- Bei den DB2-Datenquellen, die XA nicht nativ, sondern über den Synchroni- sationspunktmanager unterstützen, müssen Sie sicherstellen, dass die Parameter SPM_NAME und SVCENAME in der Konfiguration des Datenbankmanagers ordnungsgemäß auf ihre Standardwerte eingestellt sind.

Vorgehensweise

Gehen Sie wie folgt vor, um eine DRDA-Datenquelle zu konfigurieren:

Führen Sie die Anweisung CREATE SERVER, ALTER SERVER oder SET SERVER mit der Einstellung Y für die Option DB2_TWO_PHASE_COMMIT aus. Der DRDA-Wrapper generiert automatisch die folgende XA OPEN-Zeichenfolge für DRDA-Datenquellen:

```
DB=dbname,UID=benutzer-id,PWD=kennwort,TPM=FDB2,HOLD_CURSOR=T
```

Konfigurieren von Oracle-Datenquellen

Es gibt mehrere Anforderungen und Einschränkungen bei der Verwendung von Oracle-Datenquellen für föderiertes zweiphasiges Commit.

Vorbereitung

Einschränkungen:

- Sowohl Durchgriffs-DDL als auch transparente DDL, die an Oracle gerichtet sind, schlagen mit dem Fehler SQL30090, Ursachencode 21 (ORA-2089) fehl. Dagegen funktioniert normales, in Durchgriffssitzungen übergebenes SQL.

Anforderungen:

- Der auf dem Server mit föderierten Datenbanken verwendete Oracle-Client sollte vollständig installiert sein, damit gewährleistet ist, dass alle für XA relevanten Bibliotheken vorhanden sind. Stellen Sie sicher, dass djsxlinkOracle erfolgreich ausgeführt wurde, damit alle DB2- und Oracle-Bibliotheken verfügbar und ord- nungsgemäß verlinkt sind. Beachten Sie, dass die djsxlink-Scripts automatisch ausgeführt werden, wenn Sie den Oracle-Client vor dem Server mit föderierten Datenbanken installieren.
- Sie müssen allen Benutzern, die Transaktionen mit zweiphasigem Commit über den Server mit föderierten Datenbanken ausführen, folgende Zugriffsrechte ertei- len:
 - grant select on dba_pending_transactions to USERID;
 - grant select on dba_2pc_pending to USERID;
 - grant force transaction to USERID;

- Außerdem haben Sie die Möglichkeit, den Benutzern, die Transaktionen mit zweiphasigem Commit über den Server mit föderierten Datenbanken ausführen, das folgende Zugriffsrecht zu erteilen:
 - grant force any transaction to USERID;
- Wenn Sie planen, mehr als 10 Transaktionen mit zweiphasigem Commit gleichzeitig auszuführen, empfiehlt es sich, den Parameter `distributed_transactions` Ihres Oracle-Servers zu erhöhen, der sich in der Datei `init.ora` befindet.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Oracle-Datenquelle zu konfigurieren:

1. Führen Sie die Anweisung `CREATE SERVER`, `ALTER SERVER` oder `SET SERVER` mit der Einstellung `Y` für die Option `DB2_TWO_PHASE_COMMIT` aus.

Der Oracle-Wrapper erstellt automatisch die folgende XA OPEN-Zeichenfolge für Oracle-Datenquellen:

```
Oracle_XA=Acc=Pbenutzer-id/kennwort+Seqtm=0+DB=dbname+SqlNet=dblink+Threads=true
```

Beispiel:

```
XA_OPEN_STRING_OPTIONS '+LogDir=/home/user/directory+DbgFl=0x7'
```

2. Geben Sie mit der Serveroption `XA_OPEN_STRING_OPTIONS` weitere XA-Optionen an.

Konfigurieren von Informix-Datenquellen

Es gibt mehrere Anforderungen und Einschränkungen bei der Verwendung von Informix-Datenquellen für föderiertes zweiphasiges Commit.

Vorbereitung

Einschränkungen:

- Sie können mit einer Mischung aus Servern mit ein- und zweiphasigem Commit in einer einzigen Verbindung mit einem Server mit föderierten Datenbanken nicht auf Informix-Kurznamen zugreifen.
- Die Cursoroption `WITH HOLD` wird nicht unterstützt.
- Die Serveroption `XA_OPEN_STRING_OPTIONS` wird bei Informix-Datenquellen nicht unterstützt.

Anforderungen:

- In der Informix-Datenbank muss die Protokollfunktion aktiviert sein.
- Die XA-Bibliothek von Informix lässt nur eine einzige Verbindung pro Thread zu. Dies hat zur Folge, dass der Server mit föderierten Datenbanken in einer einzigen Verbindung nicht mit mehreren Servern, die für föderiertes zweiphasiges Commit aktiviert sind, auf Informix-Datenquellen zugreifen kann. Führen Sie die optionalen Schritte in der folgenden Prozedur aus, wenn eine Anwendung mehrere Server verwenden muss, die für föderiertes zweiphasiges Commit aktiviert sind.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Informix-Datenquelle zu konfigurieren:

1. Führen Sie die Anweisung `CREATE SERVER`, `ALTER SERVER` oder `SET SERVER` mit der Einstellung `Y` für die Option `DB2_TWO_PHASE_COMMIT` aus.

Der Informix-Wrapper generiert automatisch die folgende XA OPEN-Zeichenfolge für Informix-Datenquellen:

```
DB=dbname;RM=rmname;CON=con;USER=benutzer;PASSWD=kennwort
```

2. Führen Sie folgende Schritte aus, wenn eine Anwendung mehrere Server verwenden muss, die für föderiertes zweiphasiges Commit aktiviert sind:
 - a. Kopieren Sie die folgenden Informix-Wrapperbibliotheken:
libdb2informix.a, libdb2informixF.a und libdb2informixU.a.
 - b. Definieren Sie mehrere Instanzen des Informix-Wrappers, indem Sie in der Klausel LIBRARY innerhalb der Anweisung CREATE SERVER eine andere Kopie der Informix-Wrapperbibliotheken angeben.
 - c. Definieren Sie jeden Server mit föderiertem zweiphasigem Commit für die verschiedenen Wrapperinstanzen.

Beispiel:

```
CREATE WRAPPER wrapper1 library 'libdb2informix.a'  
CREATE SERVER server1 type informix version 9.4 wrapper wrapper1 options  
  (node 'inf1', dbname 'firstdb', db2_two_phase_commit 'Y');  
CREATE WRAPPER wrapper2 library 'libdb2informix2.a'  
CREATE SERVER server2 type informix version 9.4 wrapper wrapper2 options  
  (node 'inf2', dbname 'seconddb', db2_two_phase_commit 'Y');
```

Konfigurieren von Microsoft SQL Server-Datenquellen

Es gibt mehrere Anforderungen und Einschränkungen bei der Verwendung von Microsoft SQL Server-Datenquellen für föderiertes zweiphasiges Commit.

Vorbereitung

Einschränkungen:

- Bei Microsoft SQL Server-Datenquellen wird föderiertes zweiphasiges Commit nur unterstützt, wenn IBM InfoSphere Federation Server unter Windows installiert ist.
- Die DB2-Isolationsstufe wird nicht an den Microsoft SQL-Server weitergegeben.

Anforderungen:

- Damit föderiertes zweiphasiges Commit gemeinsam mit Microsoft SQL genutzt werden kann, muss dem Server die zusätzliche Serveroption XA_OPEN_STRING_OPTIONS hinzugefügt werden:

```
alter server S1 options(add xa_open_string_options  
'RMRecoveryGuid=c200e360-38c5-11ce-ae62-08002b2b79ef');
```

Dabei gilt: RMRecoveryGuid = Ressourcenmanager-ID.

Die Ressourcenmanager-ID ist an der folgenden Position in der Microsoft SQL Server-Registrierdatenbank verfügbar:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer]  
"ResourceMgrID" = "{resource manager ID}"
```

Vorgehensweise

Gehen Sie wie folgt vor, um eine Microsoft SQL Server-Datenquelle zu konfigurieren:

1. Führen Sie die Anweisung CREATE SERVER, ALTER SERVER oder SET SERVER mit der Einstellung Y für die Option DB2_TWO_PHASE_COMMIT aus.

Der Microsoft SQL Server-Wrapper generiert automatisch die folgende XA OPEN-Zeichenfolge für Microsoft SQL Server-Datenquellen:

```
TM=tmname
```

2. Geben Sie mit der Serveroption XA_OPEN_STRING_OPTIONS neben dem erforderlichen Wert für RMRecoveryGuid weitere XA-Optionen an.

Konfigurieren von Sybase-Datenquellen

Es gibt mehrere Anforderungen und Einschränkungen bei der Verwendung von Sybase-Datenquellen für föderiertes zweiphasiges Commit.

Vorbereitung

Einschränkungen:

- Bei Sybase-Datenquellen wird föderiertes zweiphasiges Commit nur unterstützt, wenn IBM InfoSphere Federation Server unter Windows installiert ist.
- Sowohl Durchgriffs-DDL als auch transparente DDL, die an Sybase gerichtet sind, schlagen mit dem Fehler SQL910N fehl. Dagegen funktioniert normales, in Durchgriffssitzungen übergebenes SQL.

Anforderungen:

- Der Administrator der Sybase-Datenbank muss eine Lizenz zum verteilten Transaktionsmanagement von Sybase Adaptive Server Enterprise (ASE) besitzen und die Funktion mit folgendem Befehl im isql-Tool aktivieren:

```
sp_configure 'enable dtm', 1
```

Sybase ASE muss erneut gestartet werden, damit dieser Parameter wirksam wird.
- Der in einer offenen Zeichenfolge angegebene Benutzername muss auf der entsprechenden Sybase ASE-Instanz die Rolle `dtm_tm_role` haben. Der Benutzer mit Verwaltungsaufgaben kann diese Rolle im isql-Tool mit folgendem Befehl zuweisen:

```
sp_role "grant", dtm_tm_role, user_name
```
- Damit eine Datenquelle von Sybase Adaptive Server Enterprise (ASE) als Ressourcenmanager für eine föderierte Datenbank agiert, muss in der Datei `xa_config` im Verzeichnis `$$SYBASE/$SYBASE_OCS/config` (Version 12 oder höher) ein Eintrag für einen logischen Ressourcenmanager (LRM) vorhanden sein, mit dem der Ressourcenmanagername dem Sybase ASE-Namen zugeordnet wird. Weitere Informationen finden Sie in der Dokumentation zu Sybase ASE XA. Der LRM-Name wird vom Server mit föderierten Datenbanken in der Zeichenfolge `XA OPEN` verwendet. Der Server mit föderierten Datenbanken verwendet den Sybase ASE-Knotennamen als LRM-Namen.
- Stellen Sie sicher, dass sich der in der XA-Konfigurationsdatei `xa_config` angegebene Servername in der Initialisierungsdatei `sql.ini` im Verzeichnis `$$SYBASE/ini` befindet.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Sybase-Datenquelle zu konfigurieren:

1. Installieren Sie die Sybase XA-Bibliotheksdatei `libxadtm.dll` auf dem Server mit föderierten Datenbanken.
2. Erstellen Sie in der Datei `$$SYBASE/$SYBASE_OCS/config/xa_config` folgende Einträge, bevor Sie die Funktionen für föderiertes zweiphasiges Commit verwenden. Wenn Sie keine Schreibberechtigungen für die Datei `xa_config` haben, müssen Sie in einem anderen Verzeichnis eine Datei namens `xa_config` erstellen und deren absoluten Pfad in der Umgebungsvariablen `XACONFIGFILE` festlegen, die sich in der Datei `db2dj.ini` befindet:

```
;one comment line is required  
lrm=lrm_name  
server=servername
```

Dabei steht *servername* für den Namen eines Eintrags in der Datei \$SYBASE/ini/sql.ini.

3. Der Sybase-Wrapper erstellt automatisch die folgende XA_OPEN-Standardzeichenfolge für Sybase-Datenquellen:

-Nrmmame -Ubenutzer-ID -Pkennwort

Wenn Sie andere Optionen für die Zeichenfolge XA_OPEN angeben müssen, verwenden Sie dazu die Serveroption XA_OPEN_STRING_OPTIONS.

Recovery nach Fehlern durch föderiertes zweiphasiges Commit

Ein föderiertes System kann nach Fehlern, die bei zweiphasigem Commit auftreten, durch eine automatische Resynchronisation oder durch eine manuelle Recovery der unbestätigten Transaktionen wiederhergestellt werden.

Resynchronisation in föderierten Systemen

Föderiertes zweiphasiges Commit beinhaltet einen automatischen Prozess, der versucht, Fehler zu behandeln, die bei Committransaktionen auftreten.

Neben Fehlern auf dem Server mit föderierten Datenbanken steigt in einer föderierten Umgebung die Wahrscheinlichkeit von Fehlern infolge von Netz-, Kommunikations- oder Datenquellenstörungen.

Zur Gewährleistung der Datenintegrität behandelt der Server mit föderierten Datenbanken diese Fehler während des föderierten zweiphasigen Commitprozesses.

Fehler in der ersten Phase

Wenn eine Datenbank mitteilt, dass bei der Vorbereitung einer UOW für das Commit ein Fehler aufgetreten ist, macht der Server mit föderierten Datenbanken die UOW in der zweiten Phase des Commitprozesses rückgängig. In der zweiten Phase sendet der Server mit föderierten Datenbanken eine Rollbacknachricht an alle beteiligten Datenquellen, die auf das Ergebnis der Transaktion warten.

Fehler in der zweiten Phase

Die Fehlerbehandlung in dieser Phase hängt davon ab, ob die Transaktion in der zweiten Phase festgeschrieben oder rückgängig gemacht wird. In der zweiten Phase wird die Transaktion nur rückgängig gemacht, wenn die Transaktion in der ersten Phase einen Fehler erkannt hat.

Wenn das Commit oder Rollback der UOW in einer der beteiligten Datenquellen fehlschlägt, möglicherweise aufgrund eines Kommunikationsfehlers, versucht der Server mit föderierten Datenbanken in einem als Resynchronisation bezeichneten Prozess, das Commit oder Rollback zu wiederholen. Die Resynchronisation wird vom Server mit föderierten Datenbanken automatisch eingeleitet und gesteuert. Wenn die aufrufende Anwendung über zweiphasiges Commit eine Verbindung zum Server mit föderierten Datenbanken herstellt, wird sie vom SQL-Kommunikationsbereich (SQLCA) über das erfolgreiche Commit informiert. Wenn die aufrufende Anwendung über einphasiges Commit eine Verbindung zum Server mit föderierten Datenbanken herstellt, wird ihre Verbindung zu diesem Server getrennt.

Die meisten Datenquellen können bei einem Fehler in einem föderierten System keine Resynchronisation einleiten. Die Resynchronisationsverarbeitung wird vom Server mit föderierten Datenbanken eingeleitet.

Unter bestimmten Umständen kann ein Server mit föderierten Datenbanken während der Transaktionsverarbeitung fehlschlagen, beispielsweise bei einem Stromausfall. Dank der Resynchronisation lassen sich alle DUOW-Transaktionen ohne Eingriff von außen lösen.

Die Resynchronisation versucht, alle unbestätigten Transaktionen abzuschließen. Im Rahmen einer normalen Resynchronisation stellt der Resynchronisationsagent eine Verbindung zur Ressourcenmanagerdatenbank einer Transaktion her und gibt eine Commit- oder Rollbackentscheidung aus. Diese Entscheidung wird anschließend vom föderierten Transaktionsmanager an die Datenquellen weitergeleitet, die an der DUOW-Transaktion beteiligt waren.

Manuelles Recovery von unbestätigten Transaktionen

Wenn Sie nicht warten können, bis die unbestätigten Transaktionen während der Resynchronisation automatisch aufgelöst werden, können Sie sie manuell auflösen. Dieser Prozess wird mitunter als heuristische Verarbeitung bezeichnet.

Beispiel: Die Kommunikationsverbindung zwischen dem externen Transaktionsmanager und einem föderierten Transaktionsmanager wird mitten in einer Transaktion unterbrochen. Wenn Sie genügend Informationen zu der Transaktion haben, können Sie Ressourcen auf dem Server mit föderierten Datenbanken und auf den fernen Datenquellen freigeben, indem Sie ein Rollback der Transaktion vom Server mit föderierten Datenbanken durchführen.

Verwenden Sie die heuristische Verarbeitung nur dann, wenn Sie die Ursache für den Transaktionsfehler kennen und gesperrte Ressourcen sofort freigeben müssen. In den meisten Situationen empfiehlt es sich, Transaktionen mit der automatisierten Resynchronisation wiederherzustellen. In einem föderierten System gibt es viele Ebenen von Transaktionsverwaltung. Transaktionen auf heuristischem Wege wiederherzustellen, ist ein komplexer und riskanter Prozess.

Es gibt drei Hauptverfahren zur Ausführung von heuristischer Verarbeitung:

- Befehl LIST INDOUBT TRANSACTIONS
Sie können diesen Befehlszeilenbefehl für heuristische Verarbeitung einsetzen.
- Fenster 'Manager für unbestätigte Transaktionen'
Sie können dieses GUI-Tool für heuristische Verarbeitung einsetzen.
- Heuristische APIs
Sie können diese APIs in Ihren Anwendungen für heuristische Verarbeitung einsetzen.

Je nach den Umständen des Fehlers unterscheiden sich die konkreten Operationen und Tasks, die Sie zur Ausführung von heuristischer Verarbeitung einsetzen.

Wenn in föderierten Systemen eine Anforderung für heuristische Verarbeitung an einen föderierten Transaktionsmanager gesendet wird, muss die resultierende Entscheidung über ein Commit oder Rollback mit dem tatsächlichen Status der unbestätigten Transaktion auf dem Server mit föderierten Datenbanken kompatibel sein. Andernfalls wird eine Fehlernachricht zurückgegeben.

Der Status von föderiertem zweiphasigen Commit bei unbestätigten Transaktionen unterscheidet sich leicht vom zweiphasigen DB2-Basiscommit bei unbestätigten Transaktionen:

- Der Status (d) bedeutet, dass in der Transaktion die Commitbestätigung von einer oder mehreren föderierten Datenquellen fehlt.

- Der Status (b) bedeutet, dass in der Transaktion die Rollbackbestätigung von einer oder mehreren föderierten Datenquellen fehlt.

Wenn sich für eine Transaktion mit dem Status (d) oder (b) kein erfolgreiches Commit oder Rollback durchführen lässt, können Sie mit der Option (f) festlegen, dass die Transaktion vergessen wird. Bei Verwendung der Option (f) werden jedoch alle Einträge der Transaktion vom Server mit föderierten Datenbanken gelöscht, und Sie müssen alle verbleibenden Synchronisationsprobleme mit den beteiligten Datenquellen manuell bereinigen. Verwenden Sie die Option (f) nur, wenn es absolut notwendig ist, beispielsweise wenn ein ferner Server ausfällt oder Verbindungen zu fernem Servern unterbrochen werden und unbedingt Ressourcen freigegeben werden müssen, und auch dann nur mit größter Vorsicht.

Anmerkung: Da (d) und (b) neue Statustypen für WebSphere Federation Server v9.1 sind, werden sie von föderierten Clients mit älteren Versionen nicht unterstützt. Wenn Sie einen Vorversionsclient zum manuellen Recovery von unbestätigten Transaktionen verwenden, werden die Statustypen (d) und (b) stattdessen dem Status (m) zugeordnet, was kein präziser Wert ist. Stellen Sie sicher, dass ein föderierter Client der Version 9.1 verwendet wird, um ungenaue Informationen beim manuellen Recovery von unbestätigten Transaktionen zu vermeiden. Standardmäßig ist auf dem Computer, auf dem WebSphere Federated Server v9.1 ausgeführt wird, der föderierte v9.1-Client enthalten.

Verfolgen der Statustypen von DUOW-Transaktionen in Datenquellen

Wenn Sie beschließen, unbestätigte Transaktionen manuell aufzulösen, anstatt ihre Auflösung der automatischen Synchronisation zu überlassen, ist es unerlässlich, die Transaktionen in dem föderierten System zu verfolgen. Bei der Verfolgung einer unbestätigten DUOW-Transaktion können Sie die fehlgeschlagenen Transaktionen nur ermitteln, indem Sie deren Transaktions-ID erfassen.

Sie müssen diese Transaktions-ID in den Datenbankmanagern aller Datenquellen suchen, auf die ein Server mit föderierten Datenbanken bei einer DUOW-Transaktion zugegriffen haben könnte.

Setzen Sie in Ihrer Anwendungsdatenbank, in der föderierten Datenbank und in allen Datenquellen in der DUOW-Transaktion den Befehl LIST INDOUBT TRANSACTIONS ab, um die Kennung und den Status der einzelnen Transaktionen zu ermitteln, die an der DUOW beteiligt sind, die Sie verfolgen wollen.

Anmerkung: Für jede Datenquelle, die das Protokoll für zweiphasiges Commit unterstützt, könnte ein eigener Befehl erforderlich sein. Suchen Sie in der Dokumentation zu dem Befehl nach der Zeichenfolge XID (Transaktions-ID) für Ihre konkrete Datenquelle.

Der föderierte Transaktionsmanager generiert während der Transaktionsverarbeitung eine Transaktions-ID im Hexadezimalformat. Diese Transaktions-ID beginnt mit der Format-ID F2PC, was dem Wert 46325243 im Hexadezimalformat entspricht. Der föderierte Transaktionsmanager sendet die Transaktions-ID an die Datenquellen. Bevor ein Server mit föderierten Datenbanken jedoch eine Transaktions-ID an eine Datenquelle sendet, ändert er die Transaktions-ID so, dass sie mit dem Format der Transaktions-ID der entsprechenden Datenquelle übereinstimmt. Im Rahmen dieser Änderungen wird der Abschnitt für die Länge des Verzweigungsqualifikationsmerkmals in der Transaktions-ID aktualisiert und der Abschnitt für das Verzweigungsqualifikationsmerkmal der Transaktions-ID hinzugefügt.

Möglicherweise müssen Sie Transaktions-IDs über mehrere Datenquellen hinweg vergleichen. Daher müssen Sie beim Verfolgen einer Transaktions-ID wissen, welchen Teil der Transaktions-ID Sie in Ihrer Umgebung präzise vergleichen können.

Beispiel: Der Transaktionsmanager ist eine DB2-Datenbank. Wenn Sie in der Datenbank den Befehl LIST INDOUBT TRANSACTIONS absetzen, wird für die Transaktions-ID der Transaktion eine Zeichenfolge in hexadezimaler Darstellung zurückgegeben, die der folgenden ähnelt:

```
463250430000019 000000004739314533463135 2E47453934000000 000000000000E80000
```

Diese Zeichenfolge besteht aus mehreren Teilen:

Format-ID	Länge der Transaktions-ID	Länge des Verzweigungsqualifikationsmerkmals	Transaktions-ID
46325043	00000019	00000000	4739314533463135 2E47453934000000 000000000000E800 00

Bei den in der Zeichenfolge aufgeführten Werten handelt es sich um Hexadezimalwerte. Die Länge der Transaktions-ID (Hexadezimalwert 19) beispielsweise steht für den Dezimalwert 25.

Wird dieselbe Transaktions-ID von einem Server mit föderierten Datenbanken, der als föderierter Transaktionsmanager agiert, an eine Datenquelle übergeben, die als Ressourcenmanager agiert, wird die Zeichenfolge der Transaktions-ID angehängt. Beispiel: Die Transaktions-ID, die an einen Ressourcenmanager für ein DB2-Datenbanksystem für eine Windows-Datenquelle übergeben wird, ändert sich in das folgende Format:

```
463252430000019 000000014739314533463135 2E47453934000000 000000000000E8000001
```

Nachstehend finden Sie die Zeichenfolge dieser geänderten Transaktions-ID, unterteilt in die Standardteile:

Format-ID	Länge der Transaktions-ID	Länge des Verzweigungsqualifikationsmerkmals	Transaktions-ID	Verzweigungsqualifikationsmerkmal
46325043	00000019	00000001	4739314533463135 2E47453934000000 000000000000E800 00	01

Die Länge des Verzweigungsqualifikationsmerkmals enthält nun eine 1, und es wurde ein Abschnitt mit einem Verzweigungsqualifikationsmerkmal hinzugefügt. Der Wert der Transaktions-ID wurde jedoch nicht geändert. Sie können die Transaktions-ID weiterhin über mehrere Datenquellen hinweg verfolgen, indem Sie Ihre Suchzeichenfolge auf den Abschnitt mit der Transaktions-ID beschränken.

Fehlerbehebung bei Problemen mit föderiertem zweiphasigem Commit

Die Fehlerbehebung bei Problemen mit föderiertem zweiphasigem Commit erfolgt oft speziell für die Datenquelle, die das Problem verursacht.

Anwendungen müssen Fehlercodes behandeln, die anzeigen, dass Transaktionen das zulässige Zeitlimit überschritten haben. Dies betrifft neben der Prüfung der Fehlercodes -911 insbesondere die Fehlercodes -913 und -918.

Fehlerbehebung bei Oracle-Datenquellen

Versuchen Sie folgende Verfahren zur Fehlerbehebung bei Oracle-Datenquellen.

- Zur Protokollieren von Informationen:

```
db2 "alter server ora1 options (add XA_OPEN_STRING_OPTIONS
'+LogDir=C:\temp+DbgFl=0x7')
```

Dabei steht `C:\temp` für den vollständigen Pfad zu der Speicherposition, an der Sie die Protokolldatei erstellen wollen. Da die Protokollierung von Informationen die Leistung deutlich beeinträchtigen kann, sollten Sie Informationen nur bei der Behebung von Fehlern protokollieren.

- Fügen Sie zum Anzeigen von Traceinformationen folgende Zeilen zur Datei `sqlnet.ora` des Oracle-Clients hinzu:

```
TRACE_LEVEL_CLIENT=16
TRACE_DIRECTORY_CLIENT=C:\temp
```

Sie können für `TRACE_LEVEL_CLIENT` auch eine kleinere Zahl festlegen, beispielsweise 4 oder 8. Da das Anzeigen von Traceinformationen die Leistung deutlich beeinträchtigen kann, sollten Sie Informationen auf Trace-Ebene nur bei der Behebung von Fehlern aktivieren.

- Zum Auflisten der in der Oracle-Datenquelle anstehenden Transaktionen:

```
select * from dba_pending_transactions where formatid=1177702467;
select * from dba_2pc_pending;
```

Zum Auflisten des Status und der ID einer Transaktion:

```
select A.STATE, A.LOCAL_TRAN_ID, A.FAIL_TIME, A.GLOBAL_TRAN_ID,
B.FORMATID || '.' || B.GLOBALID || '.' || b
ANCHID as fmt_xid
from dba_2pc_pending A, dba_pending_transactions B
where A.GLOBAL_TRAN_ID = B.FORMATID || '.' ||
B.GLOBALID and STATE='prepared' and B.FORMATID=1177702467;
```

- Zum manuellen Auflösen von Transaktionen:

```
rollback force '4.31.157818';
commit force '10.24.154537'
```

Dabei ist `'4.31.157818'` die Oracle-Feld-ID `A.LOCAL_TRAN_ID`, die der Transaktions-ID in `GLOBAL_TRAN_ID` entspricht.

Fehlerbehebung bei Sybase-Datenquellen

Versuchen Sie folgende Verfahren zur Fehlerbehebung bei Sybase-Datenquellen.

- Prüfen Sie die Sybase XA-Protokolldatei `syb_xa_log`, die sich im Verzeichnis `$SYBASE` befindet.
- Prüfen Sie die Datei `db2diag.log` mit dem Tool `db2diag`.
- Zum Prüfen einer Transaktion auf dem Sybase-Server:

```
$ isql -Ubenutzername -Pkennwort -Sservername
1> sp_transactions
2> go
```

Wenn Sie eine ungültige oder überflüssige Transaktion finden, bitten Sie den Sybase-Administrator, die Transaktion zu löschen.

Leistungsverhalten bei föderiertem zweiphasigen Commit

Datenquellen, die für Transaktionen mit zweiphasigem Commit konfiguriert werden, haben Leistungsnachteile gegenüber Datenquellen, die für Transaktionen mit einphasigem Commit konfiguriert werden.

Wenn eine Datenquelle zweiphasiges Commit verwendet, agiert der Server mit föderierten Datenbanken als Koordinator, der sicherstellt, dass alle Teilnehmer ordnungsgemäß synchronisiert werden. Diese Koordination wird durch zusätzliche Protokollierung auf dem Server mit föderierten Datenbanken sowie durch zusätzliche Kommunikation zwischen den Datenquellen erreicht. Daher ist bei einer föderierten Transaktion, die auf eine Datenquelle für zweiphasiges Commit zugreift, ein höherer Verarbeitungsaufwand erforderlich als bei einer Transaktion, die auf eine Datenquelle für einphasiges Commit zugreift. Folglich sollten Sie eine Datenquelle nur dann für zweiphasiges Commit aktivieren, wenn dies für die föderierten Transaktionen erforderlich ist.

Eine föderierte Transaktion, für die nur einphasiges Commit erforderlich ist, beispielsweise die Aktualisierung eines einzelnen Systems, die jedoch in einer Datenquelle für zweiphasiges Commit ausgeführt wird, hat in der Regel Leistungsnachteile gegenüber derselben Transaktion, wenn diese in einer Datenquelle ohne zweiphasiges Commit ausgeführt wird.

Bei einer mit zweiphasigem Commit gesteuerten Transaktion fällt der folgende Zusatzaufwand bei der Verarbeitung an, unabhängig davon, ob für die Transaktion zweiphasiges Commit erforderlich ist oder nicht:

1. Bei allen Transaktionen sind zusätzliche Einträge in die Datenbankprotokolldatei des Servers mit föderierten Datenbanken erforderlich.
Mithilfe dieser zusätzlichen Protokolleinträge kann der Server mit föderierten Datenbanken die Datenquellen in der Transaktion verfolgen und dadurch spätere Commit- und Rollbackoperationen koordinieren.
2. Bei allen Transaktionen ist zusätzliche Datenbankkommunikation zwischen dem Server mit föderierten Datenbanken und der Datenquelle erforderlich.
3. Bei Einfüge-, Aktualisierungs- und Löschoptionen sind eine oder mehrere zusätzliche Datenbankprotokolleinträge in der fernen Datenquelle erforderlich.

Der meiste Zusatzaufwand bei der Verarbeitung fällt an den Transaktionsgrenzen an und wird nicht vom Inhalt der Transaktion beeinflusst. Folglich ist die prozentuale Erhöhung der Verarbeitungsdauer bei einer kurzen Transaktion größer als bei einer langen Transaktion.

Bei gleichzeitig ablaufenden Transaktionen schreibt der Server mit föderierten Datenbanken mit einer einzigen Schreiboperation mehrere Protokolleinträge aus dem Puffer in die Protokolldatei. Daher fällt bei Anwendungen, die Transaktionen mit föderiertem zweiphasigem Commit gleichzeitig ausführen, weniger Aufwand an als bei denselben Transaktionen, wenn diese nacheinander ausgeführt werden.

Leistungssteigerung bei föderiertem zweiphasigem Commit

Sie können Maßnahmen treffen, um die Leistung bei der Ausführung von Transaktionen in einer Konfiguration mit föderiertem zweiphasigem Commit zu verbessern.

Stellen Sie sich folgende Konfigurationen vor:

- Positionieren Sie zur Reduzierung des Zeitraums, der zum Schreiben der zusätzlichen Protokolleinträge auf dem Server mit föderierten Datenbanken erforderlich ist, die Protokolldateien der föderierten Datenbank auf einer Einheit, die in der Lage ist, Fast Write-Transaktionen auszuführen, vorzugsweise auf einer Einheit, die über einen Schreibcache verfügt. In der Regel verursachen die Transaktionen mit den Protokollschreibvorgängen auf dem Server mit föderierten Datenbanken die meiste zusätzliche Verarbeitungszeit. Daher lässt sich die Leistung des zweiphasigen Commits mit hoher Wahrscheinlichkeit durch die korrekte Positionierung der Protokolldateien verbessern. Eine solche Verbesserung gilt insbesondere für schreibgeschützte Transaktionen. Die Positionierung der Protokolldateien des Servers mit föderierten Datenbanken wird vom Datenbankkonfigurationsparameter NEWLOGPATH gesteuert.
- Positionieren Sie bei Einfüge-, Aktualisierungs- und Löschttransaktionen die Protokolldateien der fernen Datenquelle auf einem Datenträger, der Fast Write-Operationen ausführen kann.
- Gehen Sie wie folgt vor, um den Aufwand aufgrund von XA-Nachrichten zu verringern, die zwischen dem Server mit föderierten Datenbanken und den Datenquellen versendet werden:
 - Positionieren Sie den Server mit föderierten Datenbanken auf demselben Computer wie eine der Datenquellen für zweiphasiges Commit.
 - Falls Sie den Server mit föderierten Datenbanken nicht auf demselben Computer wie eine Datenquelle positionieren können, erhöhen Sie die Netzgeschwindigkeit, und verringern Sie die Latenzzeit zwischen dem Server und den Datenquellen, um eine Leistungssteigerung zu erreichen.

Aktivieren Sie bei Anwendungen zweiphasiges Commit für eine Datenquelle nur dann, wenn für mindestens eine Transaktion in der Anwendung zweiphasiges Commit erforderlich ist. Setzen Sie auf jedem Server den Standardwert für DB2_TWO_PHASE_COMMIT auf N, und aktivieren Sie mit der Anweisung SET SERVER OPTION zweiphasiges Commit speziell in denjenigen Anwendungen, für die zweiphasiges Commit erforderlich ist.

Kapitel 10. Einfügen, Aktualisieren und Löschen von Daten in einem föderierten System

Während der Entwicklung Ihrer föderierten Umgebung müssen Sie Daten unabhängig davon, ob Sie ferne Informationen ändern oder Daten zwischen Datenquellen versetzen, in Ihren Datenquellen einfügen, aktualisieren und löschen.

Bevor Sie Daten in einer fernen Datenquelle modifizieren, müssen Sie sicherstellen, dass Sie über die entsprechenden Zugriffsrechte verfügen, um die Anweisungen INSERT, UPDATE und DELETE für den Kurznamen abzusetzen. Sie sollten über Kenntnisse der referenziellen Integrität, der Beibehaltung der Anweisungsatomizität und der Semantik von Zuordnungen verfügen.

Zugriffsrechte für INSERT-, UPDATE- und DELETE-Anweisungen

Zum Absetzen von INSERT-, UPDATE- und DELETE-Anweisungen für Kurznamen sind ähnliche Zugriffsrechte erforderlich wie zum Absetzen dieser Anweisungen für Tabellen. Darüber hinaus müssen Sie die entsprechenden Zugriffsrechte für Datenquellen besitzen, um in den zugrunde liegenden Objekten Auswahl-, Einfüge-, Aktualisierungs- und Löschoperationen auszuführen.

Sie können die Zugriffsrechte SELECT, INSERT, UPDATE und DELETE für einen Kurznamen erteilen oder widerrufen.

Beim Erteilen oder Widerrufen von Zugriffsrechten für einen Kurznamen werden jedoch keine Zugriffsrechte an der Datenquelle erteilt oder widerrufen. An der Datenquelle müssen die Zugriffsrechte für den Wert REMOTE_AUTHID erteilt oder widerrufen werden, der in der Benutzerzuordnung auf dem Server mit föderierten Datenbanken angegeben wurde.

Die Berechtigungs-ID der Anweisung muss über die erforderlichen Zugriffsrechte für den Kurznamen verfügen (damit die föderierte Datenbank die Anforderung akzeptiert). Die Benutzer-ID an der Datenquelle, die der Berechtigungs-ID (über eine Benutzerzuordnung) zugeordnet ist, muss die erforderlichen Zugriffsrechte für das zugrunde liegende Tabellenobjekt besitzen (damit die Datenquelle die Anforderung akzeptiert).

Bei Übergabe einer Abfrage an die föderierte Datenbank werden die Zugriffsrechte für den Kurznamen in dieser Abfrage geprüft. Die Berechtigungsanforderungen des Datenquellenobjekts, auf das der Kurzname verweist, werden nur angewendet, wenn die Abfrage tatsächlich verarbeitet wird. Wenn Sie nicht über das Zugriffsrecht SELECT für den Kurznamen verfügen, können Sie keine Auswahl aus dem Datenquellenobjekt treffen, auf das der Kurzname verweist.

Ebenso bedeutet das Zugriffsrecht UPDATE für den Kurznamen nicht, dass Sie automatisch zur Aktualisierung des Datenquellenobjekts berechtigt sind, für das der Kurzname steht. Das Bestehen der Zugriffsrechtsprüfung auf dem Server mit föderierten Datenbanken heißt nicht, dass Sie die Zugriffsrechtsprüfung an der fernen Datenquelle bestehen. Die Berechtigungs-ID eines Servers mit föderierten Datenbanken wird der Benutzer-ID der Datenquelle über Benutzerzuordnungen zugeordnet. Die Zugriffsrechtsprüfung wird an der Datenquelle umgesetzt.

Einschränkungen für INSERT-, UPDATE- und DELETE-Anweisungen in föderierten Systemen

In einem föderierten System gelten bestimmte Einschränkungen für INSERT-, UPDATE- und DELETE-Anweisungen.

Für Aktualisierungen von Kurznamen gelten folgende Einschränkungen:

- Schreibgeschützte Datenquellenobjekte wie JOIN-Sichten können nicht aktualisiert werden.
- Sie können in föderierten Sichten, die mit UNION ALL-Anweisungen erstellt wurden, keine Einfüge-, Aktualisierungs- oder Löschoperationen ausführen. Mit UNION ALL-Anweisungen erstellte föderierte Sichten sind schreibgeschützt.
- Eine föderierte Einfüge-, Aktualisierungs- oder Löschoperation oder ein Aufruf einer föderierten Prozedur mit der SQL-Datenzugriffsangabe MODIFIES SQL DATA ist in einer Funktion, einem Verweis auf die Datenänderungstabelle (Option 'data-change-table-reference'), einer Compound-Anweisung, in der ATOMIC angegeben ist (mit Ausnahme der DB2 für Linux-, UNIX- und Windows-Datenquelle), einem Trigger und der Ausführungsumgebung einer Anwendung ungültig, wenn folgende Bedingungen erfüllt sind:
 - Ein Sicherungspunkt (SAVEPOINT) ist aktiviert (mit Ausnahme der DB2 für Linux-, UNIX- und Windows-Datenquelle).
 - Ein verschiebbarer Cursor wird verwendet.
 - Die Zielsicht enthält mehrere Tabellen oder Kurznamen.

Nicht unterstützte Datenquellen

Das föderierte System bietet keine Unterstützung für Einfüge-, Aktualisierungs- und Löschoperationen für Kurznamen in nicht relationalen Datenquellen.

Die von Federation nicht unterstützten Datenquellen lauten:

- BioRS
- Excel
- Dateien mit Tabellenstruktur
- Web-Services
- XML

Referenzielle Integrität in einem föderierten System

In einem föderierten System implementiert die föderierte Datenbank keine referenzielle Integrität zwischen den Datenquellen.

Referenzielle Integritätsbedingungen an einer Datenquelle können jedoch Auswirkungen auf Aktualisierungen des Kurznamens haben. Es kann beispielsweise vorkommen, dass Sie Daten, die sich auf dem Server mit föderierten Datenbanken befinden, in einen Kurznamen einfügen müssen. Wenn der Server mit föderierten Datenbanken die Einfügung an die Datenquelle sendet, verstößt er gegen eine referenzielle Integritätsbedingung der Datenquelle. Der Server mit föderierten Datenbanken ordnet den Fehler einem föderierten Fehler zu.

Für die referenzielle Integrität zwischen den Datenquellen sind Anwendungen verantwortlich.

Anweisungen INSERT, UPDATE und DELETE und große Objekte (LOBs)

Sie können in föderierten Systemen Leseoperationen in fernen LOBs ausführen. In bestimmten Datenquellen werden Schreiboperationen in LOBs unterstützt.

Bei Federation können Sie Leseoperationen in LOBs ausführen, die sich in einer beliebigen relationalen Datenquelle befinden. Sie können Schreiboperationen in LOBs ausführen, die sich in folgenden Datenquellen befinden:

- Oracle mit dem NET8-Wrapper
- DB2 für z/OS, DB2 für System i und DB2 Database für Linux, UNIX und Windows mit dem DRDA-Wrapper

Unter bestimmten Bedingungen können Sie Schreiboperationen in LOBs ausführen, die sich in anderen Datenquellen befinden, indem Sie den Spaltentyp für den Kurznamen in VARCHAR ändern.

Beibehalten der Atomizität von Anweisungen in einem föderierten System

Bei Aktualisierungsoperationen versuchen föderierte Systeme stets, die Daten beim Abschluss einer DML-Anweisung in einem atomaren Zustand zu halten. Wenn die Daten in einem atomaren Zustand vorliegen, ist gewährleistet, dass sie erfolgreich verarbeitet werden oder unverändert erhalten bleiben.

Wenn ein Client oder eine Anwendung eine INSERT-, UPDATE- oder DELETE-Anweisung für einen Kurznamen absetzt, verarbeitet ein Server mit föderierten Datenbanken diese Anweisung intern entweder als einzelne DML-Anweisung oder als Serie mehrerer DML-Anweisungen. Wenn ein Server mit föderierten Datenbanken mehrere DML-Anweisungen zur Verarbeitung an eine Zieldatenquelle senden muss, wird die Atomizität der Daten möglicherweise beeinträchtigt. Um zu verhindern, dass die Datenatomizität beeinträchtigt wird, überwachen föderierte Systeme mithilfe von Datenquellensicherungspunkt-APIs die Serie der DML-Anweisungen.

Bestimmte Datenquellen stellen extern für das föderierte System keine Sicherungspunkt-APIs bereit. In diesem Fall werden föderierte INSERT-, UPDATE- oder DELETE-Anweisungen ohne den Schutz der Sicherungspunkt-APIs ausgeführt.

Tritt bei einer föderierten Einfüge-, Aktualisierungs- oder Löschttransaktion ein Fehler auf, können an den Datenquellen Ergebnisse einer partiellen Aktualisierung auftreten. Zur Behebung von Problemen aufgrund von Inkonsistenz führt ein föderiertes System automatisch ein internes Rollback der Transaktion aus, bevor es einen SQLCODE-Fehler an die Anwendungen zurückgibt.

Folgende Datenquellen stellen extern für den Server mit föderierten Datenbanken keine Sicherungspunkt-APIs bereit:

- DB2 für System i
- DB2 für VM und VSE
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Teradata

Wenn eine vollständige Einfüge-, Aktualisierungs- oder Löschttransaktion zur Verarbeitung in die Datenquelle verschoben wird, geht der Server mit föderierten Datenbanken davon aus, dass die Datenquelle bei einem Fehler die Atomizität der Anweisung beibehält. Wird nur ein Teil einer Einfüge-, Aktualisierungs- oder Löschttransaktion zur Verarbeitung in die Datenquelle verschoben, wird bei einem Fehler die gesamte Transaktion rückgängig gemacht.

Modifizieren von Daten in einem föderierten System

Sie können Daten in einem System föderierter Datenbanken modifizieren, indem Sie Einfüge-, Aktualisierungs- und Löschopeationen für Daten in Datenquellenobjekten durchführen.

Einfügen von Daten in Datenquellenobjekte

Verwenden Sie zum Einfügen von Daten in Datenquellen in der Anweisung INSERT die Kurznamen der Datenquellenobjekte.

Damit Daten mit einem Kurznamen eingefügt werden können, müssen alle nachstehenden Zugriffsrechte erteilt sein:

- Die Berechtigungs-ID der Anweisung muss über das Zugriffsrecht INSERT für den Kurznamen verfügen (damit die föderierte Datenbank die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss das Zugriffsrecht INSERT für das zugrunde liegende Tabellenobjekt besitzen (damit die Datenquelle die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss der Berechtigungs-ID auf dem Server mit föderierten Datenbanken über eine Benutzerzuordnung zugeordnet werden.

Einschränkungen

Federation unterstützt keine INSERT-Operationen mit nicht relationalen Datenquellen.

Vorgehensweise

Setzen Sie zum Einfügen von Daten in Datenquellenobjekte die Anweisung INSERT ab.

Beispiel: Eine Informix-Tabelle besteht aus zwei Spalten. Die erste Spalte enthält Daten vom Typ INTEGER, die zweite Spalte enthält Daten vom Typ VARCHAR (bis 20 Zeichen). Der Kurzname *infx_table_nn* wird beim Server mit föderierten Datenbanken für die Informix-Tabelle registriert.

Sie können über den Kurznamen *infx_table_nn* INSERT-, UPDATE- und DELETE-Anweisungen in der Informix-Tabelle absetzen. Mit der folgenden Anweisung wird eine neue Zeile mit Informationen in die Informix-Tabelle eingefügt:

```
INSERT INTO db2user1.infx_table_nn VALUES(1,'Walter')
```

Aktualisieren von Daten in Datenquellenobjekten

Verwenden Sie zum Aktualisieren von Daten in Datenquellen in der Anweisung UPDATE die Kurznamen der Datenquellenobjekte.

Vorbereitung

Damit Daten mit einem Kurznamen aktualisiert werden können, müssen alle nachstehenden Zugriffsrechte erteilt sein:

- Die Berechtigungs-ID der Anweisung muss über das Zugriffsrecht UPDATE für den Kurznamen verfügen (damit die föderierte Datenbank die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss das Zugriffsrecht UPDATE für das zugrunde liegende Tabellenobjekt besitzen (damit die Datenquelle die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss der Berechtigungs-ID auf dem Server mit föderierten Datenbanken über eine Benutzerzuordnung zugeordnet werden.

Einschränkungen

Federation unterstützt bei bestimmten Datenquellen keine UPDATE-Operationen (siehe „Einschränkungen für INSERT-, UPDATE- und DELETE-Anweisungen in föderierten Systemen“ auf Seite 144).

Setzen Sie zum Aktualisieren von Daten in Datenquellenobjekten die Anweisung UPDATE ab.

Beispiel: Eine Informix-Tabelle besteht aus zwei Spalten. Die erste Spalte enthält Daten vom Typ INTEGER, die zweite Spalte enthält Daten vom Typ VARCHAR (bis 20 Zeichen). Der Kurzname *infx_table_nn* wird beim Server mit föderierten Datenbanken für die Informix-Tabelle registriert.

Sie können über den Kurznamen *infx_table_nn* INSERT-, UPDATE- und DELETE-Anweisungen in der Informix-Tabelle absetzen. Mit der folgenden Anweisung wird eine Zeile mit Informationen in der Informix-Tabelle aktualisiert:

```
UPDATE db2user1.infx_table_nn SET c2='Bill' WHERE c1=2
```

Löschen von Daten aus Datenquellenobjekten

Verwenden Sie zum Löschen von Daten aus Datenquellen in der Anweisung DELETE die Kurznamen der Datenquellenobjekte.

Vorbereitung

Damit Daten mit einem Kurznamen gelöscht werden können, müssen alle nachstehenden Zugriffsrechte erteilt sein:

- Die Berechtigungs-ID der Anweisung muss über das Zugriffsrecht DELETE für den Kurznamen verfügen (damit die föderierte Datenbank die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss das Zugriffsrecht DELETE für das zugrunde liegende Tabellenobjekt besitzen (damit die Datenquelle die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss der Berechtigungs-ID auf dem Server mit föderierten Datenbanken über eine Benutzerzuordnung zugeordnet werden.

Einschränkungen

Federation unterstützt bei bestimmten Datenquellen keine DELETE-Operationen (siehe „Einschränkungen für INSERT-, UPDATE- und DELETE-Anweisungen in föderierten Systemen“ auf Seite 144).

Vorgehensweise

Setzen Sie zum Löschen von Daten aus Datenquellenobjekten die Anweisung DELETE ab.

Beispiel: Eine Informix-Tabelle besteht aus zwei Spalten. Die erste Spalte enthält Daten vom Typ INTEGER, die zweite Spalte enthält Daten vom Typ VARCHAR (bis 20 Zeichen). Der Kurzname *infx_table_nn* wird beim Server mit föderierten Datenbanken für die Informix-Tabelle registriert.

Sie können über den Kurznamen *infx_table_nn* INSERT-, UPDATE- und DELETE-Anweisungen in der Informix-Tabelle absetzen. Mit der folgenden Anweisung wird eine Zeile mit Informationen in der Informix-Tabelle gelöscht:

```
DELETE FROM infx_table_nn WHERE c1=3
```

Zuordnungssemantik in einem föderierten System

Beim Zuordnen von Daten zu einer Kurznamenspalte ändert sich je nach den vom föderierten System verwendeten Zuordnungsregeln möglicherweise der Datentyp. Damit Sie die gewünschten Ergebnisse erhalten, müssen Sie die Zuordnungsregeln verstehen.

Für die Ermittlung des Zieldatentyps einer Zuordnung zu einer Kurznamenspalte gelten folgende Regeln:

- Ermittlung des lokalen Quellentyps: Der lokale Quellentyp wird durch den lokalen Spaltentyp oder den lokalen Ergebnistyp der Ausdrücke ermittelt. Wenn die Quelle konstant ist, stimmt der lokale Quellentyp mit dem Typ der Konstanten überein.
- Ermittlung des Zieltyps:
 - Wenn die Zuordnungsquelle keinen Typ aufweist, beispielsweise Parametermarken und Nullwerte (NULLs), lautet der Zieltyp MIN(local_target_type, remote_target_type). Dabei ist local_target_type der lokale Datentyp der aktualisierten Spalte und remote_target_type der Datentyp der Datenquelle der aktualisierten Spalte. remote_target_type bezieht sich auf den Standardtyp für die progressive Typenzuordnung des Datentyps der fernen Zielspalte.
 - Wenn die Zuordnungsquelle keine Nullwerte oder Parametermarken aufweist, lautet der Zieltyp MIN(local_target_type, remote_target_type, local_source_type).

Definition von MIN(type1, type2)

- Type1 und type2 stimmen nicht überein.
- MIN(type1,type2) = MIN(type2, type1)
- MIN(type1, type2) = remote_target_type(local_target_type), wenn MIN(type1, type2) = DECIMAL(0,0)
- BLOB ist nur mit BLOB kompatibel, sodass MIN(BLOB(x), BLOB(y))=BLOB(z), wobei gilt: z=min(x,y)
- Die Datentypen TIME und DATE sind nicht kompatibel.
- Datums-/Zeittypen und Zeichenfolgen sind kompatibel.
- In Unicode-Datenbanken sind Zeichenfolgen und Grafikzeichenfolgen kompatibel.

In den folgenden Tabellen werden mindestens zwei Datentypen für Numerisch, Zeichenfolge, Grafikzeichenfolge und Datum/Zeit aufgeführt.

Tabelle 10. Datentypen für Numerisch

type1	type2	MIN(type1, type2)
SMALLINT	SMALLINT oder INTEGER oder BIGINT oder REAL oder DOUBLE	SMALLINT
INTEGER	BIGINT oder REAL oder DOUBLE	INTEGER
BIGINT	REAL oder DOUBLE	BIGINT
REAL	DOUBLE	REAL
DECIMAL(w,x)	SMALLINT	DECIMAL(p,0), wobei gilt: p=w-x, wenn p<5; SMALLINT, andernfalls
DECIMAL(w,x)	INTEGER	DECIMAL(p,0), wobei gilt: p=w-x, wenn p<11; INTEGER, andernfalls
DECIMAL(w,x)	BIGINT	DECIMAL(p,0), wobei gilt: p=w-x, wenn p<19; BIGINT, andernfalls
DECIMAL(w,x)	DECIMAL(y,z)	DECIMAL(p,s), wobei gilt: p=min(w,y)+min(w-x,y-z), s=min(x,z)
DECIMAL(w,x)	DOUBLE oder REAL	DECIMAL(w,x)

In der folgenden Tabelle werden mindestens zwei Datentypen für Zeichenfolge aufgeführt.

Tabelle 11. Datentypen für Zeichenfolge

type1	type2	MIN(type1, type2)
CHAR(x)	CHAR(y) oder VARCHAR(y) oder LONG VARCHAR oder CLOB(y)	CHAR(z), wobei gilt: z=min(x,y)
VARCHAR(x)	VARCHAR(y) oder LONG VARCHAR oder CLOB(y)	VARCHAR(z), wobei gilt: z=min(x,y)
LONG VARCHAR	CLOB(y)	LONG VARCHAR, wobei gilt: x>32700, CLOB(x), wobei gilt: x<=32700
CLOB(x)	CLOB(y)	CLOB(z), wobei gilt: z=min(x,y)

In der folgenden Tabelle werden mindestens zwei Datentypen für Grafikzeichenfolge aufgeführt.

Tabelle 12. Datentypen für Grafikzeichenfolge

type1	type2	MIN(type1, type2)
GRAPHIC(x)	GRAPHIC(y) oder VARGRAPHIC(y) oder LONG VARGRAPHIC oder DBCLOB(y)	GRAPHIC(z), wobei gilt: z=min(x,y)
VARGRAPHIC(x)	VARGRAPHIC(y) oder LONG VARGRAPHIC oder DBCLOB(y)	VARGRAPHIC(z), wobei gilt: z=min(x,y)

Tabelle 12. Datentypen für Grafikzeichenfolge (Forts.)

type1	type2	MIN(type1, type2)
LONG VARGRAPHIC	DBCLOB(y)	LONG VARGRAPHIC, wobei gilt: $x > 32700$, DBCLOB(x), wobei gilt: $x \leq 32700$
DBCLOB(x)	DBCLOB(y)	DBCLOB(z) wobei gilt: $z = \min(x, y)$

In der folgenden Tabelle werden mindestens zwei Datentypen für Datum/Zeit aufgeführt.

Tabelle 13. Datentypen für Datum/Zeit

type1	type2	MIN(type1, type2)
DATE	TIMESTAMP	DATE
TIME	TIMESTAMP	TIME

Wenn die Daten von Datentyp CHAR, die Sie einfügen, kürzer sind als die Zielgröße, füllt die Datenquelle den Rest der Spalte auf.

Wenn Sie Daten vom Datentyp DATE oder TIME in eine ferne Spalte mit dem Datentyp TIMESTAMP einfügen, füllt die Datenquelle den Rest der Spalte auf.

Zuordnungssemantik in einem föderierten System - Beispiele

Die folgende Tabelle zeigt mehrere Beispiele für die Anwendung von föderierter Zuordnungssemantik in Abfragen mit einem lokalen und fernen Typ.

Tabelle 14. Beispiele für Zuordnungssemantik

Lokaler Typ	Ferner Typ	Ihre Abfrage	Generierte Fernabfrage
FLOAT	INTEGER	SET c1=123.23	SET c1=INTEGER(123.23)
INTEGER	FLOAT	SET c1=123.23	SET c1=INTEGER(123.23)
FLOAT	INTEGER	SET c1=123	SET c1=123
CHAR(10)	CHAR(20)	SET c1='123'	SET c1='123' ('123' hat einen Typ VARCHAR(3) und ist der kürzeste Typ)
CHAR(10)	CHAR(20)	SET c1=char23col	SET c1=CHAR(char23col, 10)
CHAR(10)	CHAR(20)	SET c1=expr1	<ul style="list-style-type: none"> • SET c1=expr1 -- wenn expr1 char(n) $n \leq 10$ zurückgibt • SET c1=CHAR(expr1, 10) -- wenn expr1 char(n) $n > 10$ zurückgibt
TIMESTAMP	DATE	SET c1=CURRENT_TIMESTAMP	SET c1=DATE(CURRENT_TIMESTAMP)

Datenquellenspezifische Einschränkungen für Datentypwerte

Für einige Datentypen unterstützt der Server mit föderierten Datenbanken einen größeren Wertebereich als die Datenquelle.

Wenn Sie diese Datentypen verwenden, achten Sie darauf, die Werte zu verwenden, die von der Datenquelle unterstützt werden. Wenn Sie einen Wert einfügen, der sich außerhalb des unterstützten Wertebereichs befindet, setzt die Datenquelle den Wert entweder um oder gibt einen Fehler zurück.

Wenn Sie ein Vergleichselement mit einem Wert verwenden, der sich außerhalb des unterstützten Wertebereichs befindet, und das Vergleichselement in der Datenquelle ausgewertet wird, kann dies zu folgenden Resultaten führen:

- Das Vergleichselement gibt möglicherweise ein anderes Ergebnis zurück als dasselbe Vergleichselement auf dem Server mit föderierten Datenbanken.
- Das Vergleichselement kann zu einem Fehler führen.

Zeitangaben und Zeitmarken mit 24 im Stundenfeld

Wenn Sie ein Vergleichselement verwenden, das eine Zeitangabe oder Zeitmarke mit 24 im Stundenfeld enthält, tritt möglicherweise ein Fehler auf. Zur Vermeidung von Fehlern können Sie diese Zeitangaben und Zeitmarken konvertieren. Sie können eine Anweisung ähnlich der folgenden Anweisung UPDATE verwenden:

```
UPDATE my_table SET timestamp_col = timestamp_col + 0 SECONDS;
```

Durch diese Aktualisierung wird das Zeitfeld in 00:00:00 geändert und das Datum um einen Tag erhöht.

Der Server mit föderierten Datenbanken lässt Zeitangaben und Zeitmarken zu, die 24 im Stundenfeld enthalten. Der Unterschied zwischen einer Zeitmarke mit 24 Stunden und einer Zeitmarke mit 00 Stunden besteht darin, dass sie bei Vergleichsoperationen ungleich, in der Arithmetik jedoch gleich sind.

Beispiel

- Das folgende Vergleichselement gibt 'falsch' zurück:
 $2008-05-14-24:00:00.000000 = 2008-05-15-00:00:00.000000$
- Das folgende Vergleichselement gibt 'wahr' zurück:
 $(2008-05-14-24:00:00.000000 + 0 \text{ SECONDS}) = (2008-05-15-00:00:00.000000 + 0 \text{ SECONDS})$

Einige Datenquellen, wie beispielsweise Oracle und Sybase, lassen keine Zeitangaben und Zeitmarken mit 24 im Stundenfeld zu.

Leere Zeichenfolgen bei Oracle

Sie können Fehler aufgrund von leeren Zeichenfolgen vermeiden, indem Sie in Anwendungen, die Oracle-Kurznamen verwenden, keine leeren Zeichenfolgen verwenden. Sie können stattdessen den NULL-Wert oder eine aus einem einzelnen Leerzeichen bestehende Zeichenfolge (' ') verwenden.

In VARCHAR-Spalten auf einem Server mit föderierten Datenbanken, der nicht mit VARCHAR2 kompatibel ist, wird zwischen der leeren Zeichenfolge und NULL-Werten unterschieden. Dies führt zu einem unterschiedlichen Verhalten bei den lokalen Tabellen einerseits und den Kurznamen andererseits. Bei VARCHAR-Spalten in einer fernen Datenquelle, die mit VARCHAR2 kompatibel ist, werden die leere Zeichenfolge und NULL-Werte hingegen als äquivalent betrachtet. Falls eine leere Zeichenfolge ('') in eine VARCHAR-Spalte eingefügt wird, wird sie in einen NULL-Wert konvertiert.

In den folgenden Beispielen ist MY_TABLE eine lokale Tabelle in der föderierten Datenbank und MY_NICK ein Kurzname in einer fernen Tabelle einer Oracle-Datenbank:

Beispiel 1

In diesem Beispiel wird eine leere Zeichenfolge in MY_TABLE und MY_NICK eingefügt, die jeweils eine Spalte namens NOT_NULL_COL enthalten, in der eine Eingabe von Daten erforderlich ist (NOT NULL). Die Anweisung INSERT für MY_TABLE wird erfolgreich ausgeführt, aber die Anweisung INSERT für MY_NICK wird mit einem Fehler beendet, weil die leere Zeichenfolge in einen NULL-Wert konvertiert wird, der einen Konflikt mit der Einschränkung verursacht, dass die Eingabe von Daten erforderlich ist.

```
INSERT INTO my_table(not_null_col) VALUES('');  
INSERT INTO my_nick(not_null_col) VALUES('');
```

Beispiel 2

Im folgenden Beispiel ist sowohl die föderierte als auch die ferne Tabelle anfänglich leer. Die erste Anweisung SELECT gibt keine Zeilen zurück, weil der Server mit föderierten Datenbanken zwischen der leeren Zeichenfolge und NULL-Werten unterscheidet. Die zweite Anweisung SELECT gibt jedoch eine Zeile zurück, weil die Oracle-Datenbank während der Einfügeoperation die leere Zeichenfolge in einen NULL-Wert konvertiert.

```
INSERT INTO my_table(my_col) VALUES('');  
SELECT * FROM my_table WHERE my_col IS NULL;  
INSERT INTO my_nick(my_col) VALUES('');  
SELECT * FROM my_nick WHERE my_col IS NULL;
```

Föderierte Aktualisierungen mit Anwendungssicherungspunkten

Mit Anwendungssicherungspunkten in einem föderierten System können Sie die Operationen steuern, die eine Untergruppe von SQL-Anweisungen in einer Transaktion ausführt.

Sie können in einer einzelnen Transaktion mehrere Sicherungspunkte verwenden. Eine Untergruppe von SQL-Anweisungen, die unter einem Sicherungspunkt gruppiert ist, kann als eine Einheit behandelt werden. Sie können eine Transaktion in eine einzelne logische Ebene oder in verschachtelte logische Ebenen von Sicherungspunkteinheiten unterteilen.

Nachdem Sie einen Sicherungspunkt in Ihrer Anwendung gesetzt haben, können Sie diesen Sicherungspunkt zu einem späteren Zeitpunkt freigeben oder für die Arbeitsgänge, die nach dem gesetzten Sicherungspunkt ausgeführt wurden, ein Rollback durchführen. Wenn beispielsweise eine einzelne SQL-Anweisung in einem Sicherungspunkt fehlschlägt, bleibt die Einheit als Ganzes intakt. Sie können einen der folgenden Schritte ausführen:

- Sie können alle SQL-Anweisungen, die im Sicherungspunkt ausgeführt wurden, rückgängig machen, indem Sie ein Rollback bis zum Sicherungspunkt durchführen.
- Sie können den Sicherungspunkt freigeben, wenn er für das System nicht mehr benötigt wird.

Sie können föderierte Aktualisierungen mit Operationen für Anwendungssicherungspunkte für DB2 Database für Linux, UNIX und Windows-Datenbanken ausführen.

Föderierte Aktualisierungen mit Anwendungssicherungspunkten - Beispiele

In diesem Beispiel für Anwendungssicherungspunkte wird der Effekt von Rollbackoperationen in verschachtelten Sicherungspunkten veranschaulicht.

Beispiel: Erstellen Sie den Kurznamen NN_DEPT für die Tabelle DEPARTMENT im Schema DEPTDATA auf dem Server DB2_SERVER. Der Kurzname NN_DEPT enthält die Spalten DEPT_NO, DEPT_NAME und MANAGER_NO.

```
CREATE NICKNAME NN_DEPT FOR DB2_SERVER.DEPTDATA.DEPARTMENT;
```

Fügen Sie eine Zeile ein, bevor Sie den Sicherungspunkt SP1 erstellen. Fügen Sie eine weitere Zeile ein, bevor Sie den Sicherungspunkt SP2 erstellen. Fügen Sie zwei weitere Zeilen ein, bevor Sie den Sicherungspunkt SP3 erstellen. Fügen Sie eine fünfte Zeile ein.

```
INSERT INTO NN_DEPT VALUES ('A10', 'SALES', 'ADAM');
SAVEPOINT SP1 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('B10', 'MARKETING', 'BRIAN');
SAVEPOINT SP2 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('C10', 'ENGINEERING', 'CINDY');
INSERT INTO NN_DEPT VALUES ('C20', 'TESTING', 'DOUG');
SAVEPOINT SP3 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('D10', 'SUPPORT', 'EMILY');
```

Der Kurzname NN_DEPT enthält jetzt fünf Zeilen mit den Abteilungen A10, B10, C10, C20 und D10. Sie wollen für einige dieser Zeilen ein Rollback durchführen. In den folgenden Beispielen werden die Ergebnisse der Rollbackoperation für jeden Sicherungspunkt beschrieben:

- ROLLBACK TO SAVEPOINT SP3;

Die letzte Zeile mit der Abteilung D10 befindet sich nicht mehr im Kurznamen NN_DEPT. Die Zeilen, die Sie vor dem Erstellen des Sicherungspunkts SP3 eingesetzt haben (A10, B10, C10, C20), sind weiterhin im Kurznamen NN_DEPT vorhanden.

- ROLLBACK TO SAVEPOINT SP2;

Die Zeilen mit den Abteilungen C10, C20 und D10 befinden sich nicht mehr im Kurznamen NN_DEPT. Die Zeilen, die Sie vor dem Erstellen des Sicherungspunkts SP2 eingesetzt haben (A10, B10), sind weiterhin im Kurznamen NN_DEPT vorhanden.

- ROLLBACK TO SAVEPOINT SP1;

Die Zeilen mit den Abteilungen B10, C10, C20 und D10 befinden sich nicht mehr im Kurznamen NN_DEPT. Die Zeile, die Sie vor dem Erstellen des Sicherungspunkts SP1 eingesetzt haben (A10), ist weiterhin im Kurznamen NN_DEPT vorhanden.

Einschränkungen bei föderierten Aktualisierungen mit Anwendungssicherungspunkten

In föderierten Systemen bestehen Einschränkungen für Sicherungspunktoperationen.

Die Unterstützung von Sicherungspunkten für Schreiboperationen in fernen Daten in föderierten Anwendungen ist auf DB2 Database für Linux, UNIX und Windows beschränkt. Die folgenden Einschränkungen gelten für das Datenquellenobjekt, für das der Kurzname definiert ist (als Ziel einer Schreiboperation):

- In Version 9.5 kann das Datenquellenobjekt in einer Sicherungspunktoperation selbst ein Kurzname sein.
- In früheren Versionen ist es nicht möglich, dass das Datenquellenobjekt in einer Sicherungspunktoperation ein Kurzname ist.

Sie können Sicherungspunkte im seriellen Modus, in der MPP-Umgebung (Massively Parallel Processing - exklusive Parallelverarbeitung) und in Umgebungen mit föderiertem ein- oder zweiphasigem Commit aktivieren. Alle in diesen Umgebungen vorhandenen Einschränkungen gelten auch für Sicherungspunktoperationen.

Kapitel 11. Importieren und Exportieren von Daten für Kurznamen

Mit dem Befehl `IMPORT` können Sie Daten in einen Kurznamen importieren. Mit dem Befehl `EXPORT` können Sie Daten aus einer Abfrage exportieren, die auf einen Kurznamen verweist.

Bei folgenden Datenquellen können Sie nur die `IMPORT`-Befehle verwenden:

- DB2-Produktfamilie
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

Einschränkungen für den Import von Daten in Kurznamen

Es gibt Einschränkungen bei der Verwendung des Befehls `IMPORT` zum Importieren von Daten in einen Kurznamen.

Bei der Verwendung des Befehls `IMPORT` zum Importieren von Daten in einen Kurznamen gelten folgende Einschränkungen:

- Das ferne Objekt, für das der Kurzname definiert ist, muss eine Tabelle sein. Sie können keine Daten in einen Kurznamen importieren, der für eine Sicht oder ein Synonym definiert wurde.
- Die unterstützten Dateitypen sind `IXF`, `ASC` und `DEL`.
- Die Klausel `ALLOW WRITE ACCESS` muss angegeben werden. Diese Klausel ruft den Onlineimportmodus auf. Die Klausel `ALLOW WRITE ACCESS` ermöglicht gleichzeitig ablaufenden Anwendungen den Lese- und Schreibzugriff auf die Zieltabelle des Imports.
- Sie können den Modus `COMMITCOUNT AUTOMATIC` nicht bei Kurznamen verwenden.
- Der Wert `COMMITCOUNT n` muss so angegeben werden, dass n eine gültige Zahl ungleich Null ist.
- Bei Kurznamen werden nur die Operationen `INSERT` und `INSERT_UPDATE` unterstützt.
- Die bei Kurznamen nicht unterstützten Spaltentypen sind `LOBs` und generierte Spalten. Zum Import von `LOB`-Daten in eine ferne Tabelle muss die entsprechende Kurznamenspalte den Datentyp `VARCHAR` aufweisen.
- Folgende Dateitypänderungswerte (`filetype`) werden bei Kurznamen nicht unterstützt:
 - `dldelfiletype`
 - `generatedignore`
 - `generatedmissing`
 - `identityignore`
 - `identitymissing`
 - `indexixf`
 - `indexschema`

```
lobsinfile
nodefaults
no_type_idfiletype
usedefaults
```

- Die Hierarchie (typisierte Tabelle) wird bei Kurznamen nicht unterstützt.

Bei der Übergabe eines IMPORT-Befehls, der diese Einschränkungen nicht beachtet, wird der SQL-Fehlercode -27999N zurückgegeben. Beispiel:

```
SQL27999N Die angeforderte IMPORT-Operation für ein fernes Ziel (Kurzname) kann nicht
ausgeführt werden. Ursachencode = "ursachencode"
```

IMPORT-Befehl bei Kurznamen - Beispiele

Die Beispiele zeigen, wie Daten aus verschiedenen Dateitypen in Kurznamen importiert werden.

Dateityp DEL

In diesem Beispiel werden Daten mit der Option INSERT_UPDATE aus dem Dateityp DEL importiert:

```
IMPORT FROM import_file_1.del OF DEL
ALLOW WRITE ACCESS
COMMITCOUNT 50
INSERT_UPDATE INTO NICKNAME_1;
```

Dateityp IXF

In diesem Beispiel werden Daten mit der Option INSERT aus dem Dateityp IXF importiert:

```
IMPORT FROM import_file_1.ixf OF IXF
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

Dateityp ASC

In diesem Beispiel werden Daten mit der Option INSERT aus dem Dateityp ASC importiert. In dem Beispiel ist der Dateänderungswert STRIPTBLANKS enthalten, um gegebenenfalls abschließende Leerzeichen in den Daten abzuschneiden. Der Parameter METHOD L gibt den Anfang und das Ende der Spaltennummern an.

```
IMPORT FROM import_file_1.asc OF ASC MODIFIED BY STRIPTBLANKS
METHOD L(1 6, 8 32, 34 44, 46 48)
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

Einschränkungen für den Export von Daten mit Kurznamen

Es gibt Einschränkungen bei der Verwendung des Befehls EXPORT zum Exportieren von Daten aus einer Abfrage, die auf einen Kurznamen verweist.

Bei der Verwendung des Befehls EXPORT zum Exportieren von Daten mit einem Kurznamen gelten folgende Einschränkungen:

- Die Beschreibung der Zieltabelle, die für die Ausführung der CREATE-Operation beim Import erforderlich ist, wird nicht im Dateiformat IXF gespeichert. Verwenden Sie das Dienstprogramm db2look für die Erfassung der Informationen, die Sie für die Neuerstellung der Tabelle benötigen.
- Sie können Daten in die Dateitypen IXF und DEL exportieren. Der Dateityp ASC wird beim Datenexport aus Kurznamen nicht unterstützt.

Kapitel 12. Arbeiten mit Kurznamen

Bei einem Kurznamen handelt es sich um eine Kennung, die von einer Anwendung verwendet wird, um auf ein Datenquellenobjekt, z. B. eine Tabelle oder eine Sicht, zu verweisen. In einem föderierten System können Sie Kurznamen dazu verwenden, auf Datenquellenobjekte zuzugreifen und die Leistung von Abfragen für ferne Datenquellen zu verbessern.

Kurznamen in einem föderierten System

Wenn Sie Daten von Datenquellen auswählen oder modifizieren wollen, fragen Sie mithilfe von SELECT-, INSERT-, UPDATE- und DELETE-Anweisungen die Kurznamen ab. Sie übergeben Abfragen in DB2 SQL an die föderierte Datenbank.

Sie können Daten aus lokalen Tabellen und fernen Datenquellen mit einer einzigen SQL-Anweisung so miteinander verknüpfen, als wären alle Daten lokal gespeichert. Beispielsweise können Sie Daten verknüpfen, die sich in folgenden Objekten befinden:

- Lokale DB2-Tabelle in der föderierten Datenbank, Oracle-Tabelle und Sybase-Sicht
- DB2 UDB für z/OS-Tabelle auf einem Server, DB2 UDB für z/OS-Tabelle auf einem anderen Server und Excel-Arbeitsblatt

Indem das föderierte System SQL-Anweisungen so verarbeitet, als wären die Datenquellen gewöhnliche relationale Tabellen oder Sichten innerhalb der föderierten Datenbank, kann es relationale Daten mit Daten verknüpfen, die in nicht relationalen Formaten vorliegen.

Tabellen und Sichten in der föderierten Datenbank sind *lokale Objekte*. Erstellen Sie für diese Objekte keine Kurznamen. Verwenden Sie stattdessen in Ihrer SQL-Anweisung den tatsächlichen Objektnamen.

Ferne Objekte sind Objekte, die sich nicht in der föderierten Datenbank befinden. Für diese Objekte müssen Sie Kurznamen erstellen. Beispiel:

- Tabellen und Sichten in einer anderen Datenbank oder Instanz des föderierten Systems
- Tabellen und Sichten in einer anderen Datenbank oder Instanz auf einem anderen System
- Tabellen und Sichten in Datenquellen wie Oracle, Sybase und ODBC

Mit der Option WITH HOLD deklarierte Cursor

Sie können die Option WITH HOLD für einen Cursor verwenden, der für einen Kurznamen definiert wird.

Für den DRDA-Wrapper und eine DB2 Database für Linux, UNIX und Windows-Datenquelle bleiben Cursor, die Sie mit der Option WITH HOLD deklarieren, über mehrere UOWs geöffnet.

Wenn Sie einen Cursor mit der Option WITH HOLD deklarieren, die Datenquelle diese Option jedoch nicht unterstützt, wird das Attribut ignoriert.

Trigger

Ein Kurzname kann kein Aktualisierungsziel in einem Trigger sein.

Sie können in den Triggerhauptteil SELECT-Anweisungen für Kurznamen einbeziehen. Sie können in den Triggerhauptteil keine INSERT-, UPDATE- oder DELETE-Anweisungen für Kurznamen einbeziehen.

Zugriff auf Daten mit Kurznamen

In einem föderierten System können Sie ohne großen Aufwand auf Daten zugreifen, unabhängig von deren Speicherposition. Zum Zugriff auf Daten erstellen Sie Kurznamen für Ihre Datenquellenobjekte wie Tabellen und Sichten.

Wenn beispielsweise der Kurzname DEPT für die ferne Tabelle EUROPE.PERSON-DEPT steht, können Sie mit der Anweisung `SELECT * FROM DEPT` die Informationen in der fernen Tabelle abfragen. Beim Abfragen eines Kurznamens müssen Sie sich die Verbindungsdetails zu der Datenquelle nicht merken. Sie müssen sich also nicht um Folgendes kümmern:

- Name des Objekts an der Datenquelle
- Server, auf dem sich das Datenquellenobjekt befindet
- Typ der Datenquelle, in der sich das Objekt befindet, beispielsweise Informix und Oracle
- Von der Datenquelle verwendete Abfragesprache oder SQL-Version
- Datentypzuordnungen zwischen der Datenquelle und dem Server mit föderierten Datenbanken
- Funktionszuordnungen zwischen der Datenquelle und dem Server mit föderierten Datenbanken

Der Server mit föderierten Datenbanken findet die zur Verarbeitung Ihrer Abfragen erforderlichen Informationen in den zugrunde liegenden Metadaten im Katalog der föderierten Datenbank. Diese Metadaten werden aus den Datenquellen zusammengestellt, wenn der Server und die Datenbank eingerichtet und für den Zugriff auf Datenquellen konfiguriert werden.

Nachdem das föderierte System eingerichtet wurde, können Sie mithilfe der Kurznamen die Datenquellen abfragen oder die Konfiguration des föderierten Systems weiter verbessern.

Mit Kurznamen verwendbare SQL-Anweisungen

SQL-Anweisungen unterstützen die Verwendung von Kurznamen.

Tabelle 15. Gebräuchliche SQL-Anweisungen zur Verwendung mit Kurznamen

SQL-Anweisung	Beschreibung	Erforderliche Berechtigung
ALTER NICKNAME	Modifiziert einen vorhandenen Kurznamen durch Änderung des lokalen Spaltennamens, des lokalen Datentyps, der föderierten Spaltenoptionen oder der informativen Integrationsbedingungen. Die Tabelle oder Sicht an der Datenquelle ist nicht betroffen.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht ALTER oder CONTROL für den Kurznamen • Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist • Definierender Benutzer des Kurznamens, wie in der Spalte DEFINER der Katalogsicht für den Kurznamen eingetragen
ALTER TABLE	Ändert eine ferne Tabelle, die über eine föderierte Datenbank mit transparenter DDL erstellt wurde. Sie können keine Tabellen ändern, die nativ in der Datenquelle erstellt wurden. Kann mithilfe informativer Integrationsbedingungen eine referenzielle Integritätsbedingung zu einem Kurznamen hinzufügen.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht ALTER oder CONTROL für den Kurznamen • Zugriffsrecht ALTERIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist
COMMENT ON	Fügt Kommentare in die Katalogbeschreibungen verschiedener Objekte wie Funktionen, Funktionszuordnungen, Indizes, Kurznamen, Server, Serveroptionen, Typenzuordnungen und Wrapper ein oder ersetzt diese Kommentare.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht ALTER oder CONTROL für das Objekt • Zugriffsrecht ALTERIN für das Schema • Definierender Benutzer des Objekts, wie in der Spalte DEFINER der Katalogsicht für das Objekt eingetragen
CREATE ALIAS	Definiert einen Aliasnamen für einen Kurznamen.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Berechtigung IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder der explizite Schemaname des Aliasnamens nicht vorhanden ist • Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname des Aliasnamens auf ein vorhandenes Schema verweist
CREATE INDEX mit Klausel SPECIFICATION ONLY	Erstellt eine Indexspezifikation (Metadaten), die dem Abfrageoptimierungsprogramm anzeigt, dass ein Datenquellenobjekt einen Index besitzt. Es wird kein Index erstellt, sondern nur die Spezifikation.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht CONTROL oder INDEX für das zugrunde liegende Datenquellenobjekt — sowie <i>entweder</i> Berechtigung IMPLICIT_SCHEMA für die Datenbank oder Zugriffsrecht CREATEIN für das Schema

Tabelle 15. Gebräuchliche SQL-Anweisungen zur Verwendung mit Kurznamen (Forts.)

SQL-Anweisung	Beschreibung	Erforderliche Berechtigung
CREATE TABLE mit Klausel OPTIONS	Erstellt eine ferne Tabelle über die föderierte Datenbank mit transparenter DDL.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich — <i>sowie entweder</i> Berechtigung IMPLICIT_SCHEMA für die Datenbank oder Zugriffsrecht CREATEIN für das Schema
CREATE TABLE mit Fullselect AS und DATA INITIALLY DEFERRED REFRESH-Klauseln	Erstellt eine MQT (Materialized Query Table) mit einem Fullselect, der auf einen Kurznamen verweist.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht CREATETAB für die Datenbank und Zugriffsrecht USE für den Tabellenbereich — <i>sowie entweder</i> Berechtigung IMPLICIT_SCHEMA für die Datenbank oder Zugriffsrecht CREATEIN für das Schema • Zugriffsrecht CONTROL für die Tabelle oder Sicht • Zugriffsrecht SELECT für die Tabelle oder Sicht und Zugriffsrecht ALTER, wenn REFRESH DEFERRED angegeben wird
CREATE VIEW	Erstellt eine Sicht, die auf einen oder mehrere Kurznamen verweist.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht CONTROL oder SELECT für den Kurznamen — <i>sowie entweder</i> Berechtigung IMPLICIT_SCHEMA für die Datenbank oder Zugriffsrecht CREATEIN für das Schema
DELETE	Löscht Zeilen aus dem Datenquellenobjekt, beispielsweise aus einer Tabelle oder Sicht, die einen Kurznamen besitzt.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht DELETE für den Kurznamen und Zugriffsrecht DELETE für das zugrunde liegende Datenquellenobjekt • Zugriffsrecht CONTROL für das zugrunde liegende Datenquellenobjekt
DROP	<p>Löscht ein Objekt, beispielsweise einen Kurznamen, eine föderierte Sicht oder eine Indexspezifikation. Die Tabelle, die Sicht oder der Index an der Datenquelle ist nicht betroffen.</p> <p>Beim Löschen von Tabellen, die mit transparenter DDL erstellt wurden, wird der entsprechende Kurzname für diese Tabelle ebenfalls gelöscht.</p>	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht DROPIN für das Schema des Objekts • Zugriffsrecht CONTROL für das Objekt

Tabelle 15. Gebräuchliche SQL-Anweisungen zur Verwendung mit Kurznamen (Forts.)

SQL-Anweisung	Beschreibung	Erforderliche Berechtigung
GRANT	Erteilt Zugriffsrechte für Kurznamen und föderierte Sichten, beispielsweise ALTER, DELETE, INDEX, INSERT, SELECT oder UPDATE. Zugriffsrechte an der Datenquelle müssen separat erteilt werden.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Klausel WITH GRANT OPTION für jedes angegebene Zugriffsrecht • Zugriffsrecht CONTROL für das Objekt
INSERT	Fügt Zeilen in das Datenquellenobjekt ein, beispielsweise in eine Tabelle oder Sicht, die einen Kurznamen besitzt.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht INSERT für den Kurznamen und Zugriffsrecht INSERT für das zugrunde liegende Datenquellenobjekt • Zugriffsrecht CONTROL für das zugrunde liegende Datenquellenobjekt
LOCK TABLE	Sperrt das ferne Objekt an der Datenquelle. Verhindert, dass gleichzeitig ablaufende Anwendungsprozesse eine Datenquellentabelle ändern, die einen Kurznamen besitzt.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht SELECT für die zugrunde liegende Tabelle • Zugriffsrecht CONTROL für die zugrunde liegende Tabelle
REVOKE	Widerruft Zugriffsrechte für Kurznamen und föderierte Sichten, beispielsweise ALTER, DELETE, INDEX, INSERT, SELECT oder UPDATE. Zugriffsrechte an der Datenquelle müssen separat widerrufen werden.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht CONTROL für das Objekt
SELECT	Wählt Zeilen aus dem Datenquellenobjekt aus, beispielsweise aus einer Tabelle oder Sicht, die einen Kurznamen besitzt.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht SELECT für den Kurznamen und Zugriffsrecht SELECT für das zugrunde liegende Datenquellenobjekt • Zugriffsrecht CONTROL für das zugrunde liegende Datenquellenobjekt
UPDATE	Aktualisiert die Werte in den angegebenen Spalten und Zeilen des Datenquellenobjekts, beispielsweise in einer Tabelle oder Sicht, die einen Kurznamen besitzt.	<ul style="list-style-type: none"> • SYSADM oder DBADM • Zugriffsrecht UPDATE für den Kurznamen und Zugriffsrecht UPDATE für das zugrunde liegende Datenquellenobjekt • Zugriffsrecht CONTROL für das zugrunde liegende Datenquellenobjekt

Wenn Sie eine Abfrage an die föderierte Datenbank übergeben, werden die Zugriffsrechte für den Kurznamen in dieser Abfrage geprüft. Die Berechtigungsanforderungen des Datenquellenobjekts, auf das der Kurzname verweist, werden nur angewendet, wenn die Abfrage an der Datenquelle verarbeitet wird.

Damit Daten, die einen Kurznamen besitzen, ausgewählt, eingefügt, aktualisiert oder gelöscht werden können, muss die Berechtigungs-ID der Anweisung folgende Zugriffsrechte enthalten:

- Das geeignete Zugriffsrecht für den Kurznamen, damit die föderierte Datenbank die Anforderung akzeptiert
- Das geeignete Zugriffsrecht für das zugrunde liegende Datenquellenobjekt, damit die Datenquelle die Anforderung akzeptiert

Zum Aktualisieren einer Datenquelle mit einem Kurznamen benötigen Sie beispielsweise das Zugriffsrecht UPDATE für den Kurznamen und das Zugriffsrecht UPDATE für das zugrunde liegende Datenquellenobjekt.

Zugriff auf neue Datenquellenobjekte

Für den Zugriff auf neue Datenquellenobjekte müssen Sie Kurznamen für diese Objekte erstellen. Verwenden Sie für Datenquellen, die keine Kurznamen haben, die Anweisung CREATE NICKNAME.

Vorbereitung

Das föderierte System muss für den Zugriff auf die Datenquelle konfiguriert sein.

In der föderierten Datenbank muss eine Serverdefinition für den Datenquellen-server vorhanden sein, auf dem sich das Objekt befindet. Sie erstellen eine Serverdefinition mit der Anweisung CREATE SERVER.

Damit Daten mit einem Kurznamen eingefügt, aktualisiert oder gelöscht werden können, müssen alle nachstehenden Zugriffsrechte erteilt sein:

- Die Berechtigungs-ID der Anweisung muss über die erforderlichen Zugriffsrechte SELECT, INSERT, UPDATE und DELETE für den Kurznamen verfügen, damit die föderierte Datenbank die Anforderung akzeptiert.
- Die Benutzer-ID an der Datenquelle muss die erforderlichen Zugriffsrechte SELECT, INSERT, UPDATE und DELETE für das zugrunde liegende Tabellenobjekt besitzen, damit die Datenquelle die Anforderung akzeptiert.
- Die Benutzer-ID an der Datenquelle muss der Berechtigungs-ID auf dem Server mit föderierten Datenbanken über eine Benutzerzuordnung zugeordnet werden.

Sie müssen die erforderlichen Berechtigungen für die Anweisung CREATE NICKNAME haben.

- SYSADM oder DBADM
- Berechtigung IMPLICIT_SCHEMA für die föderierte Datenbank, wenn der implizite oder der explizite Schemaname des Kurznamens nicht vorhanden ist
- Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname des Kurznamens vorhanden ist

Informationen zu dieser Task

Mitunter müssen Sie auf Datenquellenobjekte zugreifen, die keine Kurznamen haben. Dies können neue Objekte sein, die zu einer Datenquelle hinzugefügt wurden, beispielsweise eine neu erstellte Sicht. Dies können vorhandene Objekte sein, die nicht beim Server mit föderierten Datenbanken registriert wurden, als dieser das erste Mal konfiguriert wurde. In jedem Fall müssen Sie einen Kurznamen für das Objekt erstellen.

Vorgehensweise

Setzen Sie für den Zugriff auf neue Datenquellenobjekte die Anweisung CREATE NICKNAME ab.

Erstellen von Kurznamen für relationale und nicht relationale Datenquellen

Die Anweisung CREATE NICKNAME weist kleine Unterschiede bei relationalen und nicht relationalen Datenquellen auf.

Bei relationalen Datenquellen lautet die Syntax der Anweisung CREATE NICKNAME wie folgt:

```
CREATE NICKNAME kurzname FOR servername."fernes_schema"."objektname"
```

kurzname

Ein eindeutiger Kurzname für das Datenquellenobjekt. Kurznamen können bis zu 128 Zeichen lang sein.

Der Kurzname besteht aus zwei Teilen - dem Schema und dem Kurznamen. Wenn Sie beim Erstellen des Kurznamens das Schema auslassen, wird die Authentifizierung-ID des Benutzers, der den Kurznamen erstellt, zum Schema des Kurznamens. Die Standardwerte für den Schemanamen werden auf Basis der folgenden Hierarchie gewählt:

1. Sonderregister CURRENT SCHEMA
2. Sonderregister SESSION_USER
3. Sonderregister SYSTEM USER
4. Mit der Datenbank verbundene Berechtigungs-ID

FOR *servername."fernes_schema"."objektname"*

Eine dreiteilige Kennung für das ferne Datenquellenobjekt. Wenn Ihre Datenquelle keine Schemata unterstützt, übergehen Sie das Schema in der Anweisung CREATE NICKNAME.

- *servername* ist der Name, den Sie dem Datenquellenserver in der Anweisung CREATE SERVER zugeordnet haben.
- *fernes_schema* ist der Name des fernen Schemas, zu dem das Objekt gehört.
- *objektname* ist der Name des fernen Objekts, auf das Sie zugreifen wollen.

OPTIONS (*optionsliste*)

Informationen zu dem Kurznamen, mit deren Hilfe der SQL-Abfragecompiler und der Wrapper effiziente Abfragen ausführen können.

Bei bestimmten nicht relationalen Datenquellen lautet die Syntax der Anweisung CREATE NICKNAME wie folgt:

```
CREATE NICKNAME kurzname spaltendefinitionsliste  
FOR SERVER servername  
OPTIONS (optionsliste)
```

kurzname

Ein eindeutiger Kurzname für das Datenquellenobjekt, wie oben für relationale Datenquellen beschrieben.

spaltendefinitionsliste

Eine Liste der Kurznamenspalten und Datentypen.

FOR SERVER *servername*

Der lokale Name, den Sie in den Serverdefinitionsangaben in der Anweisung CREATE SERVER für den fernen Server erstellt haben.

OPTIONS (*optionsliste*)

Informationen zu dem Kurznamen, mit deren Hilfe der SQL-Abfragecompiler und der Wrapper effiziente Abfragen ausführen können.

Namen für Kurznamenspalten und Indizes

Wenn Sie einen Kurznamen erstellen, verwendet der Server mit föderierten Datenbanken einen Algorithmus, um Namen für Kurznamenspalten und Indizes zu generieren.

Dieser Algorithmus wurde in Version 9.5 verbessert, sodass die Namen für die Kurznamenspalten und die Indizes den ursprünglichen Namen möglichst genau entsprechen.

Der Algorithmus ist nur für relationale Datenquellen gültig. Für nicht relationale Datenquellen bleibt der in der Anweisung CREATE NICKNAME angegebene Name unverändert.

Die folgende Tabelle enthält Beispiele für Namen von Kurznamenspalten, die aus Namen von fernen Spalten generiert wurden. In diesen Beispielen befinden sich die fernen Spalten in separaten fernen Tabellen. Die Namen für Kurznamenspalten, die von den Namen der fernen Spalten abweichen, sind in Fettdruck dargestellt.

Tabelle 16. Namen für Kurznamenspalten, die aus Namen von fernen Spalten generiert wurden.

Name der fernen Spalten	Name, der mit dem aktuellen Algorithmus für Datenquellen generiert wurde, die ...			Name, der vor Version 9.5 generiert wurde (der Algorithmus setzt nicht-alphanumerische Zeichen um und konvertiert sie)
	Kennungen in Großbuchstaben umsetzen	Kennungen in Kleinbuchstaben umsetzen	Kennungen nicht ändern	
spalte1	spalte1	SPALTE1	SPALTE1	SPALTE1
Spalte1	Spalte1	Spalte1	SPALTE1	SPALTE1
SPALTE1	SPALTE1	SPALTE1	SPALTE1	SPALTE1
spalte-1	spalte-1	spalte-1	spalte-1	SPALTE_1
Spalte-1	Spalte-1	Spalte-1	Spalte-1	SPALTE_1
SPALTE-1	SPALTE-1	SPALTE-1	SPALTE-1	SPALTE_1

Wenn sich die fernen Spalten in derselben Tabelle befinden, verwendet der Server mit föderierten Datenbanken den vorhandenen Algorithmus, der eindeutige Namen generiert. Beispiel: Bei den Namen für Kurznamenspalten für die Datenquellen, bei denen die Kennungen nicht geändert werden, werden die Namen der fernen Spalten wie folgt generiert:

- spalte1 wird zu SPALTE1
- Spalte1 wird zu SPALTE10
- SPALTE1 wird zu SPALTE11

Zugreifen auf Datenquellen unter Verwendung von Durchgriffssitzungen

Sie können SQL-Anweisungen direkt an Datenquellen übergeben, indem Sie einen speziellen Modus verwenden, der Durchgriff genannt wird. In einer Durchgriffssitzung können Sie SQL-Anweisungen in der SQL-Programmversion übergeben, die für diese Datenquelle verwendet wird.

Verwenden Sie eine Durchgriffssitzung, wenn Sie eine Operation durchführen wollen, die mit SQL nicht möglich ist. Verwenden Sie beispielsweise eine Durchgriffssitzung, um unter Verwendung der nativen Programmversion der Datenquelle eine Prozedur oder einen Index zu erstellen oder Abfragen auszuführen.

Die Datenquellen, die die Durchgriffsfunktion unterstützen, akzeptieren nur SQL-Anweisungen in einer Durchgriffssitzung.

Sie können eine Durchgriffssitzung ebenso verwenden, um von SQL nicht unterstützte Aktionen auszuführen, wie beispielsweise bestimmte Verwaltungstasks. Welche Verwaltungstasks Sie ausführen können, hängt von der Datenquelle ab. Bei DB2 Database für Linux, UNIX und Windows können Sie beispielsweise das Statistikdienstprogramm für die Datenquelle ausführen. Sie können jedoch die ferne Datenbank nicht starten oder stoppen.

Sie können in einer Durchgriffssitzung nur jeweils eine Datenquelle abfragen. Verwenden Sie zum Öffnen einer Sitzung den Befehl SET PASSTHRU. Wenn Sie den Befehl SET PASSTHRU RESET verwenden, wird die Durchgriffssitzung geschlossen. Wenn Sie anstelle des Befehls SET PASSTHRU RESET den Befehl SET PASSTHRU verwenden, wird die aktuelle Durchgriffssitzung geschlossen und eine neue geöffnet.

Sie können einen Cursor in einer Durchgriffssitzung mit der Option WITH HOLD deklarieren. Wenn Sie diese Option beim DRDA-Wrapper und einer DB2 Database für Linux, UNIX und Windows-Datenquelle verwenden, bleiben die deklarierten Cursor für mehrere UOWs geöffnet. Wenn Sie einen Cursor mit der Option WITH HOLD deklarieren, die Datenquelle diese Option jedoch nicht unterstützt, wird das Attribut ignoriert.

Bei nicht relationalen Datenquellen können Sie Durchgriffssitzungen nicht verwenden.

Zugreifen auf heterogene Daten über föderierte Sichten

Eine *föderierte Sicht* ist eine Sicht in der föderierten Datenbank, deren Basistabellen sich in fernen Datenquellen befinden. Die föderierte Sicht verweist mit Kurznamen auf die Basistabellen, nicht mit den Tabellennamen der Datenquelle.

Vorbereitung

Sie müssen eine der folgenden Berechtigungen haben, um die Anweisung CREATE VIEW abzusetzen:

- SYSADM oder DBADM
- Für jeden Kurznamen in einem Fullselect die beiden nachstehenden Zugriffsrechte bzw. Berechtigungen:

- Zugriffsrecht CONTROL oder SELECT für die zugrunde liegende Tabelle oder Sicht
- Eine(s) der folgenden Berechtigungen oder Zugriffsrechte:
 - Berechtigung IMPLICIT_SCHEMA für die föderierte Datenbank, wenn der implizite oder der explizite Schemaname der Sicht nicht vorhanden ist
 - Zugriffsrecht CREATEIN für das Schema, wenn der Schemaname der Sicht auf ein vorhandenes Schema verweist

Beim Definieren einer Sicht für einen Kurznamen einer föderierten Datenbank werden die Zugriffsrechte für die zugrunde liegenden Objekte nicht berücksichtigt.

Einschränkungen

Mit UNION ALL-Anweisungen erstellte föderierte Sichten sind schreibgeschützt.

Föderierte Sichten, die in der Klausel FROM mehrere Kurznamen enthalten, sind schreibgeschützt.

Föderierte Sichten, die in der Klausel FROM nur einen einzigen Kurznamen enthalten, sind möglicherweise schreibgeschützt.

- Wenn sich der Kurzname in der Klausel FROM auf eine nicht relationale Datenquelle bezieht, ist die föderierte Sicht schreibgeschützt.
- Wenn Sie beim Erstellen der Sicht andere Kurznamen als Vergleichselemente oder Unterabfragen einschließen, kann die föderierte Sicht aktualisiert werden.

Informationen zu dieser Task

Wenn Sie eine Abfrage über eine föderierte Sicht ausführen, werden die Daten von der fernen Datenquelle abgerufen. Das Erstellen einer föderierten Datenbanksicht von Daten einer Datenquelle wird manchmal als „Erstellen einer Sicht für einen Kurznamen“ bezeichnet. Dies liegt daran, dass Sie beim Erstellen der Sicht auf die Kurznamen anstatt auf die Datenquellen verweisen.

Diese Sichten bieten einer global integrierten Datenbank ein hohes Maß an Datenunabhängigkeit, ähnlich wie Sichten, die in mehreren lokalen Tabellen definiert werden, dies für Manager zentralisierter relationaler Datenbanken tun.

Vorgehensweise

Setzen Sie zum Erstellen einer föderierten Sicht die Anweisung CREATE VIEW ab.

Die Berechtigungsanforderungen der Datenquelle für die Tabelle oder Sicht, auf die der Kurzname verweist, werden angewendet, wenn die Abfrage verarbeitet wird. Die Berechtigungs-ID der Anweisung kann mit einer Benutzerzuordnung einer anderen fernen Berechtigungs-ID zugeordnet werden.

Erstellen von föderierten Sichten - Beispiele

Die folgenden Beispiele zeigen, wie föderierte Sichten erstellt werden, um von mehreren Datenquellen auf Daten zuzugreifen. Die Beispiele zeigen die Syntax der Anweisung CREATE VIEW für Federation.

Beispiel: Erstellen einer föderierten Sicht, die ähnliche Daten von verschiedenen Datenquellenobjekten mischt

Sie arbeiten mit Kundendaten auf verschiedenen Servern: einem in Europa, einem in Asien und einem in Südamerika. Die europäischen Kundendaten befinden sich in einer Oracle-Tabelle. Der Kurzname für diese Tabelle lautet `ORA_EU_CUST`. Die asiatischen Kundendaten befinden sich in einer Sybase-Tabelle. Der Kurzname für diese Tabelle lautet `SYB_AS_CUST`. Die südamerikanischen Kundendaten befinden sich in einer Informix-Tabelle. Der Kurzname für diese Tabelle lautet `INFMX_SA_CUST`. Jede Tabelle verfügt jeweils über eine Spalte für die Kundennummer (`CUST_NO`), für den Namen des Kunden (`CUST_NAME`), für die Produktnummer (`PROD_NO`) und für die Bestellmenge (`QUANTITY`). Setzen Sie die folgende Anweisung ab, um aus diesen Kurznamen eine Sicht zu erstellen, in der diese Kundendaten gemischt werden:

```
CREATE VIEW FV1
AS SELECT * FROM ORA_EU_CUST
UNION
SELECT * FROM SYB_AS_CUST
UNION
SELECT * FROM INFMX_SA_CUST
```

Beispiel: Verknüpfen von Daten zum Erstellen einer föderierten Sicht

Sie arbeiten mit den Kundendaten auf einem Server und mit den Vertriebsdaten auf einem anderen Server. Die Kundendaten befinden sich in einer Oracle-Tabelle. Der Kurzname für diese Tabelle lautet `ORA_EU_CUST`. Die Vertriebsdaten befinden sich in einer Sybase-Tabelle. Der Kurzname für diese Tabelle lautet `SYB_SALES`. Sie wollen die Kundeninformationen mit den von diesen Kunden getätigten Käufen abgleichen. Jede Tabelle enthält eine Spalte mit der Kundennummer (`CUST_NO`). Setzen Sie die folgende Anweisung ab, um aus diesen Kurznamen eine föderierte Sicht zu erstellen, in der diese Daten verknüpft werden:

```
CREATE VIEW FV4
AS SELECT A.CUST_NO, A.CUST_NAME, B.PROD_NO, B.QUANTITY
FROM ORA_EU_CUST A, SYB_SALES B
WHERE A.CUST_NO=B.CUST_NO
```

Erstellen eines Kurznamens für einen Kurznamen

Sie können einen Kurznamen für einen Kurznamen erstellen.

Vorgehensweise

Führen Sie die in diesem Beispiel beschriebenen Schritte aus, um einen Kurznamen für einen Kurznamen zu erstellen.

Beispiel: Zugriff auf ein Microsoft Excel-Arbeitsblatt über einen AIX-Server mit föderierten Datenbanken

Sie haben einen Server mit föderierten Datenbanken unter AIX und einen Server mit föderierten Datenbanken unter Windows. Sie wollen von beiden Servern auf das Excel-Arbeitsblatt zugreifen. Der Excel-Wrapper wird jedoch nur auf Servern mit föderierten Datenbanken unterstützt, die unter Windows ausgeführt werden. Gehen Sie wie folgt vor, um über den AIX-Server mit föderierten Datenbanken auf das Excel-Arbeitsblatt zuzugreifen:

1. Installieren Sie IBM InfoSphere Federation Server auf dem Windows-Server mit föderierten Datenbanken.
2. Konfigurieren Sie den Windows-Server für den Zugriff auf Excel-Datenquellen.
3. Erstellen Sie auf dem Windows-Server einen Kurznamen für das Excel-Arbeitsblatt.
4. Installieren Sie IBM InfoSphere Federation Server auf dem AIX-Server mit föderierten Datenbanken.
5. Konfigurieren Sie den AIX-Server für den Zugriff auf Datenquellen der DB2-Produktfamilie.
6. Erstellen Sie auf dem AIX-Server einen Kurznamen für den Excel-Kurznamen auf dem Windows-Server.

Datenauswahl in einem föderierten System

Die Art Ihrer Datenauswahl in einem föderierten System richtet sich nach dem Datenquellentyp, aus dem Sie Daten auswählen.

Einige Typen der verteilten Anforderungen, die mit einem föderierten System verwendet werden, fragen Folgendes ab:

- Eine einzelne ferne Datenquelle
- Eine lokale und eine ferne Datenquelle
- Mehrere ferne Datenquellen
- Eine Kombination von fernem und lokalen Datenquellen

Verwenden Sie zur Auswahl von Daten aus Datenquellen in der Anweisung SELECT die Kurznamen der Datenquellenobjekte.

Die föderierte Datenbank ist eine lokale Datenquelle. Tabellen und Sichten in der föderierten Datenbank sind lokale Objekte. Sie erstellen für diese Objekte keine Kurznamen, sondern verwenden in den SELECT-Anweisungen den tatsächlichen Objektnamen.

Zu fernem Datenquellen gehören folgende: Eine andere DB2 Database für Linux, UNIX und Windows-Datenbankinstanz auf dem Server mit föderierten Datenbanken, eine andere DB2 Database für Linux, UNIX und Windows-Datenbankinstanz auf einem anderen Server sowie weitere Datenquellen. Objekte, die sich in fernem Datenquellen befinden, sind ferne Objekte.

Datenauswahl in einem föderierten System - Beispiele

Dieser Abschnitt veranschaulicht ein Szenario, in dem ein Server mit föderierten Datenbanken auf mehrere Datenquellen zugreift, und stellt Beispiele für SELECT-Anweisungen bereit.

Beispiel: Ein Server mit föderierten Datenbanken ist für den Zugriff auf eine DB2 für z/OS-Datenquelle, eine DB2 für System i-Datenquelle und eine Oracle-Datenquelle konfiguriert. In jeder Datenquelle ist eine Tabelle gespeichert, die Vertriebsinformationen enthält. Diese Konfiguration wird in der folgenden Abbildung dargestellt.

DB2-Clients (Endbenutzer und Anwendung)

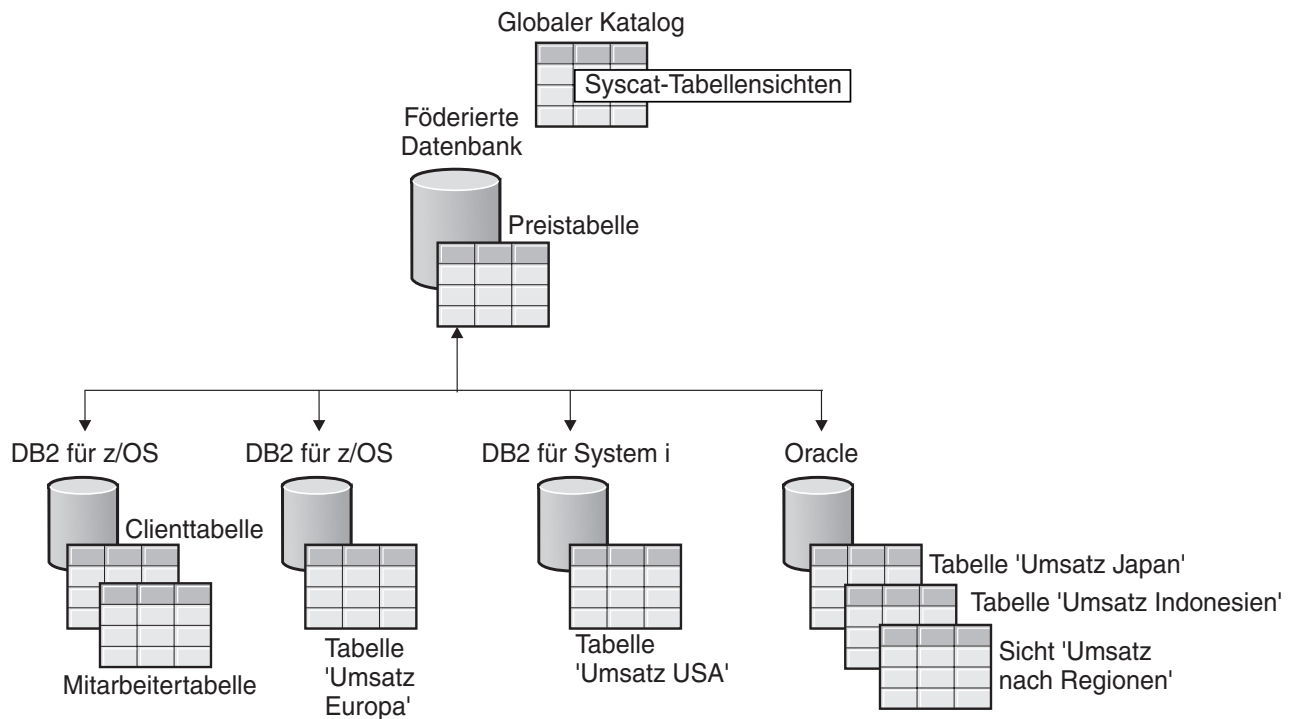


Abbildung 8. Beispiel für ein föderiertes System mit DB2- und Oracle-Datenquellen

In den Vertriebstabellen gibt es Spalten, in denen die Kundennummer (CUST_NO), die Bestellmenge (QUANTITY) und die Nummer des bestellten Produkts (PROD_NO) verzeichnet sind. Außerdem ist in der föderierten Datenbank eine lokale Tabelle mit Preisinformationen vorhanden. Die Preistabelle enthält Spalten, in denen die Produktnummer (PROD_NO) und der aktuelle Preis (PRICE) verzeichnet sind.

Die Kurznamen für die fernen Datenquellenobjekte werden in den SYSCAT.TABLES-Tabellen gespeichert, wie in den folgenden Tabellen aufgeführt. Die Spalte TYPE gibt den Objekttyp an, beispielsweise Kurzname (N), lokale Tabelle (T) oder Sicht (V).

Tabelle 17. Informationen zur Datenquelle

Objektname der Datenquelle	Typ des Objekts	Position
PRICES	Lokale Tabelle	Föderierte Datenbank
EUROPE_SALES	Ferne Tabelle	DB2 für z/OS-Datenbank
US_SALES	Ferne Tabelle	DB2 für System i-Datenbank
JAPAN_SALES	Ferne Tabelle	Oracle-Datenbank
SALES_BY_REGION	Ferne Sicht	Oracle-Datenbank

Tabelle 18. SYSCAT-Tabellen

TABNAME	TYPE
PRICES	T
FED_PRICES	N
Z_EU_SALES	N

Tabelle 18. SYSCAT-Tabellen (Forts.)

TABNAME	TYPE
SI_US_SALES	N
ORA_JAPANSALES	N
ORA_REGIONSALES	N
...	

Damit Daten mit einem Kurznamen ausgewählt werden können, müssen alle nachstehenden Zugriffsrechte erteilt sein:

- Die Berechtigungs-ID der Anweisung muss über das Zugriffsrecht SELECT für den Kurznamen verfügen (damit die föderierte Datenbank die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss das Zugriffsrecht SELECT für das zugrunde liegende Tabellenobjekt besitzen (damit die Datenquelle die Anforderung akzeptiert).
- Die Benutzer-ID an der Datenquelle muss der Berechtigungs-ID auf dem Server mit föderierten Datenbanken über eine Benutzerzuordnung zugeordnet werden.

Bei den folgenden Beispielen für die Anweisung SELECT wird das oben beschriebene föderierte Beispielsystem verwendet.

Beispiel: Abfragen einer einzelnen Datenquelle:

Z_EU_SALES enthält die Produkte, die von Ihren europäischen Kunden bestellt wurden. Außerdem enthält diese Tabelle die bei jedem Verkauf bestellte Menge. Diese Abfrage verwendet eine Anweisung SELECT mit einer Klausel ORDER BY, um die Verkäufe in Europa aufzulisten, und sortiert die Liste nach der Kundennummer:

```
SELECT CUST_NO, PROD_NO, QUANTITY
   FROM Z_EU_SALES
  ORDER BY CUST_NO
```

Beispiel: Verknüpfen einer lokalen und einer fernen Datenquelle:

Die Tabelle PRICES befindet sich in der föderierten Datenbank. Sie enthält die Preisliste der Produkte, die Sie vertreiben. Sie wollen die Preise aus dieser lokalen Tabelle auswählen, die den in Z_EU_SALES aufgeführten Produkten entsprechen. Außerdem wollen Sie die Ergebnismenge nach der Kundennummer sortieren.

```
SELECT sales.CUST_NO, sales.PROD_NO, sales.QUANTITY
   FROM Z_EU_SALES sales, PRICES
  WHERE sales.PROD_NO=PRICES.PROD_NO
  ORDER BY sales.CUST_NO
```

Beispiel: Abfragen mehrerer ferner Datenquellen:

Sie wollen alle Vertriebsinformationen aus den einzelnen Regionen zusammenstellen und die Ergebnismenge nach der Produktnummer ordnen.

```
WITH GLOBAL_SALES (Customer, Product, Quantity) AS
  (SELECT CUST_NO, PROD_NO, QUANTITY FROM Z_EU_SALES
   UNION ALL
   SELECT CUST.NO, PROD.NO, QUANTITY FROM IS_US_SALES
   UNION ALL
```



```

SELECT CUST.NO,PROD.NO, QUANTITY FROM ORA_JAPANSALES)
SELECT Customer, Product, Quantity
FROM GLOBAL_SALES
ORDER BY Product

```

Eine Sicht in der Oracle-Datenquelle führt die Verkaufszahlen für Japan und Indonesien auf. Diese Sicht hat den Kurznamen ORA_SALESREGION. Sie wollen diese Angaben mit den Verkaufszahlen aus den USA kombinieren und die Produktpreise neben den einzelnen Verkäufen anzeigen.

```

SELECT us_jpn_ind.CUST_NO, us_jpn_ind.PROD_NO,
       us_jpn_ind.QUANTITY, us_jpn_ind.QUANTITY*PRICES.PRICE
AS SALEPRICE FROM
(SELECT CUST_NO, PROD_NO, QUANTITY
FROM ORA_SALESREGION
UNION ALL
SELECT CUST_NO, PROD_NO, QUANTITY
FROM iS_US_SALES us ) us_jpn_ind,PRICES
WHERE us_jpn_ind.PROD_NO = PRICES.PROD_NO
ORDER BY SALEPRICE DESC

```

Informative Integrationsbedingungen für Kurznamen

Sie können mithilfe von informativen Integrationsbedingungen für Kurznamen die Abfrageleistung verbessern. Informative Integrationsbedingungen sind Regeln, mit denen das Optimierungsprogramm die Leistung verbessert, die jedoch vom Datenbankmanager nicht umgesetzt werden.

Sie können für Kurznamen Folgendes angeben:

- Referenzielle Integritätsbedingungen
- Prüfungen auf Integritätsbedingung
- Integritätsbedingungen der funktionalen Abhängigkeiten
- Integritätsbedingungen über Primärschlüssel
- Eindeutige Integritätsbedingungen

Angeben von informativen Integrationsbedingungen für Kurznamen (DB2-Steuerzentrale)

Sie können informative Integrationsbedingungen für Kurznamen angeben, um die Leistung von Abfragen in fernen Datenquellen zu verbessern. Allerdings implementiert oder prüft der Server mit föderierten Datenbanken die Integritätsbedingungen nicht. Sie können informative Integrationsbedingungen für einen Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile angeben.

Einschränkungen

Nachdem Sie informative Integrationsbedingungen für einen Kurznamen definiert haben, können Sie die Spaltennamen für diesen Kurznamen nur ändern, wenn Sie die informativen Integrationsbedingungen wieder entfernen.

Informationen zu dieser Task

Bei relationalen Datenquellen können Sie informative Integrationsbedingungen angeben, wenn Sie einen Kurznamen ändern.

Bei nicht relationalen Datenquellen können Sie informative Integrationsbedingungen angeben, wenn Sie einen Kurznamen erstellen oder ändern.

Vorgehensweise

Gehen Sie wie folgt vor, um informative Integrationsbedingungen für einen Kurznamen mithilfe der DB2-Steuerzentrale anzugeben:

1. Wählen Sie den Ordner **Kurznamen** aus:
 - Klicken Sie, wenn Sie einen Kurznamen erstellen, im Teilfenster der DB2-Steuerzentrale mit den Objektdetails in der Liste der Aktionen den Eintrag **Kurznamen erstellen** an. Der Assistent für Kurznamen wird geöffnet.
 - Öffnen Sie über das Fenster 'Kurznamen erstellen' das Notizbuch 'Kurznamen hinzufügen' oder 'Merkmale' für einen Kurznamen:
 - a. Wenn Sie einen einzelnen Kurznamen erstellen wollen, klicken Sie **Hinzufügen** an. Das Notizbuch 'Kurznamen hinzufügen' wird geöffnet.
 - b. Wenn mit der Aufspürfunktion eine Liste von Kurznamen generiert wurde, wählen Sie den Kurznamen aus, dem Sie informative Integrationsbedingungen hinzufügen wollen. Klicken Sie anschließend **Merkmale** an. Das Notizbuch 'Merkmale' wird geöffnet.
 - Wenn Sie einen Kurznamen ändern, klicken Sie den Kurznamen an, den Sie ändern wollen. Klicken Sie im Teilfenster der DB2-Steuerzentrale mit den Objektdetails in der Liste der Aktionen den Eintrag **Ändern** an. Das Notizbuch 'Kurznamen ändern' wird geöffnet.
2. Legen Sie auf der Seite 'Schlüssel' die referenziellen Integritätsbedingungen für den Kurznamen fest. Sie können eine Integritätsbedingung über Primärschlüssel, über eindeutige Schlüssel oder über Fremdschlüssel definieren.
3. Legen Sie auf der Seite 'Prüfung auf Integritätsbedingung' die Prüfungen auf Integritätsbedingung oder die Integritätsbedingungen der funktionalen Abhängigkeiten für den Kurznamen fest.
4. Klicken Sie **OK** an, um die informativen Integrationsbedingungen festzulegen und das Notizbuch zu schließen.

Angeben von informativen Integrationsbedingungen für Kurznamen (DB2-Befehlszeile)

Sie können informative Integrationsbedingungen zu Kurznamen hinzufügen, um die Leistung von Abfragen in fernen Datenquellen zu verbessern. Allerdings implementiert oder prüft der Server mit föderierten Datenbanken die Integritätsbedingungen nicht. Sie können informative Integrationsbedingungen für einen Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile angeben.

Einschränkungen

Nachdem Sie informative Integrationsbedingungen für einen Kurznamen definiert haben, können Sie die Spaltennamen für diesen Kurznamen nur ändern, wenn Sie die informativen Integrationsbedingungen wieder entfernen.

Informationen zu dieser Task

Bei relationalen Datenquellen können Sie informative Integrationsbedingungen angeben, wenn Sie einen Kurznamen ändern.

Bei nicht relationalen Datenquellen können Sie informative Integrationsbedingungen angeben, wenn Sie einen Kurznamen erstellen oder ändern.

Vorgehensweise

Zum Angeben von informativen Integrationsbedingungen über die DB2-Befehlszeile müssen Sie die Anweisung CREATE NICKNAME oder ALTER NICKNAME mit den entsprechenden Attributen für die Integritätsbedingungen absetzen.

Angeben von informativen Integrationsbedingungen für Kurznamen - Beispiele

Die folgenden Beispiele zeigen die Verwendung informativer Integrationsbedingungen für Kurznamen. Sie verwenden CREATE- oder ALTER NICKNAME-Anweisungen für Prüfungen auf Integritätsbedingung, referenzielle Integritätsbedingungen und andere Datenstrukturen.

Beispiel: Informative Prüfung auf Integritätsbedingung

In der folgenden fernen Tabelle sind die Daten in der Gehaltsspalte stets größer als 10000.

```
CREATE TABLE account.salary (  
    empno INTEGER NOT NULL PRIMARY KEY,  
    salary INTEGER NOT NULL  
);
```

Erstellen Sie einen Kurznamen für diese Tabelle:

```
CREATE NICKNAME account.salary FOR myserv.account.salary;
```

Fügen Sie anschließend informative Prüfungen auf Integritätsbedingung für den Kurznamen ein, indem Sie die folgende Anweisung absetzen:

```
ALTER NICKNAME account.salary ADD CONSTRAINT cons1 CHECK( salary > 10000 )  
NOT ENFORCED  
ENABLE QUERY OPTIMIZATION;
```

Beispiel: Informative referenzielle Integritätsbedingung: Kurzname zu Kurzname

In diesem Beispiel gibt es zwei Kurznamen: N1 und N2. Die Spalte F1 des Kurznamens N2 enthält den Schlüsselwert in der Spalte P1 des Kurznamens N1. Sie können die referenzielle Integritätsbedingung für den Kurznamen N2 durch Absetzen der folgenden Anweisung definieren:

```
ALTER NICKNAME SCHEMA1.N2 ADD CONSTRAINT ref1  
    FOREIGN KEY (F1) REFERENCES SCHEMA1.N1 (P1)  
    NOT ENFORCED;
```

Beispiel: Informative referenzielle Integritätsbedingung: Kurzname zu Tabelle

In diesem Beispiel enthält der Kurzname N3 mit der Spalte F1 den Schlüsselwert in der Spalte P1 der Tabelle T1. Sie können die referenzielle Integritätsbedingung für den Kurznamen N3 durch Absetzen der folgenden Anweisung definieren:

```
ALTER NICKNAME SCHEMA1.N3 ADD CONSTRAINT ref1  
    FOREIGN KEY (F1) REFERENCES SCHEMA1.T1 (P1)  
    NOT ENFORCED;
```

Beispiel: Informative referenzielle Integritätsbedingung: Tabelle zu Kurzname

In diesem Beispiel enthält die Tabelle T2 mit der Spalte F1 den Schlüsselwert in der Spalte P1 des Kurznamens N4. Sie können die referenzielle Integritätsbedingung für die Tabelle T2 durch Absetzen der folgenden Anweisung definieren:

```
ALTER TABLE SCHEMA1.T2 ADD CONSTRAINT ref1
    FOREIGN KEY (F1) REFERENCES SCHEMA1.N4 (P1)
    NOT ENFORCED;
```

Beispiel: Funktionale Abhängigkeit

In diesem Beispiel ermittelt das Spaltenpaar C1 und C2 eindeutig den Wert in der Spalte P1. Sie können die funktionale Abhängigkeit durch Absetzen der folgenden Anweisung definieren:

```
ALTER NICKNAME SCHEMA1.NICK1 ADD CONSTRAINT FD1 CHECK( P1 DETERMINED BY (C1,C2) )
    NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Beispiel: Datei mit Tabellenstruktur

Diese Anweisung definiert einen Primärschlüssel für eine Datei mit Tabellenstruktur:

```
CREATE NICKNAME MY_FILE (
    X INTEGER NOT NULL,
    Y INTEGER,
    PRIMARY KEY (X) NOT ENFORCED
) FOR SERVER MY_SERVER OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT');
```

Sternschema

Die Anweisung erstellt vier Dimensionstabellen und eine Faktentabelle:

```
CREATE TABLE SCHEMA.FACT (
    LOCATION_CODE INTEGER NOT NULL,
    PRODUCT_CODE  INTEGER NOT NULL,
    CUSTOMER_CODE INTEGER NOT NULL,
    SDATE         DATE NOT NULL,
    SALES         INTEGER NOT NULL
);

CREATE TABLE SCHEMA.LOCATION (
    LOCATION_CODE INTEGER NOT NULL PRIMARY KEY,
    STATE         CHAR(2) NOT NULL,
    SHOP_ID      INTEGER NOT NULL,
    ...
);

CREATE TABLE SCHEMA.PRODUCT (
    PRODUCT_CODE  INTEGER NOT NULL PRIMARY KEY,
    PRODUCT_CAT  INTEGER NOT NULL,
    PRODUCT_NAME  VARCHAR(20) NOT NULL,
    ...
);

CREATE TABLE SCHEMA.CUSTOMER (
    CUSTOMER_CODE INTEGER NOT NULL PRIMARY KEY,
    NAME          VARCHAR(20) NOT NULL,
    TEL           VARCHAR(10) NOT NULL,
    ...
);

CREATE TABLE SCHEMA.TIMEDIM (
    SDATE         DATE NOT NULL UNIQUE,
    YEAR          INTEGER NOT NULL,
    QUARTER       INTEGER NOT NULL,
    ...
);
```

Der Server mit föderierten Datenbanken erstellt folgende Kurznamen für die Fakt-tabelle und die Dimensionstabellen:

```
CREATE NICKNAME SCHEMA.FACT FOR SERVER.SCHEMA.FACT;  
CREATE NICKNAME SCHEMA.LOCATION FOR SERVER.SCHEMA.LOCATION;  
CREATE NICKNAME SCHEMA.PRODUCT FOR SERVER.SCHEMA.PRODUCT;  
CREATE NICKNAME SCHEMA.CUSTOMER FOR SERVER.SCHEMA.CUSTOMER;  
CREATE NICKNAME SCHEMA.TIMEDIM FOR SERVER.SCHEMA.TIMEDIM;
```

Sie können die folgende Fremdschlüsselbeziehung durch Absetzen der folgenden Anweisungen definieren:

```
ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT L1 FOREIGN KEY (LOCATION_CODE)  
REFERENCES SCHEMA.LOCATION(LOCATION_CODE)  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;  
  
ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT P1 FOREIGN KEY (PRODUCT_CODE)  
REFERENCES SCHEMA.PRODUCT(PRODUCT_CODE)  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;  
  
ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT C1 FOREIGN KEY (CUSTOMER_CODE)  
REFERENCES SCHEMA.CUSTOMER(CUSTOMER_CODE)  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;  
  
ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT S1 FOREIGN KEY (SDATE)  
REFERENCES SCHEMA.TIMEDIM(SDATE)  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Wenn der Wert der Spalte TEL im Kurznamen CUSTOMER eindeutig ist, können Sie mit der nachstehenden Anweisung die folgende eindeutige informative Integrationsbedingung hinzufügen:

```
ALTER NICKNAME SCHEMA.CUSTOMER ADD CONSTRAINT U1 UNIQUE( TEL )  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Wenn der Wert der Spalte SHOP_ID im Kurznamen LOCATION den Wert der Spalte LOCATION_ID eindeutig ermittelt, können Sie mit der nachstehenden Anweisung die folgende funktionale Abhängigkeit definieren:

```
ALTER NICKNAME SCHEMA.LOCATION  
ADD CONSTRAINT F1 CHECK( LOCATION_ID DETERMINED BY SHOP_ID )  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Da der Wert der Spalte QUARTER im Kurznamen TIMEDIM zwischen 1 und 4 liegt, können Sie mit der nachstehenden Anweisung die folgende informative Prüfung auf Integritätsbedingung definieren:

```
ALTER NICKNAME SCHEMA.TIMEDIM  
ADD CONSTRAINT Q1 CHECK( QUARTER BETWEEN 1 AND 4 )  
NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Die Anweisungen in diesem Beispiel erstellen Kurznamen für ferne Tabellen. Die Kurznamen besitzen Primärschlüssel, wenn die fernen Tabellen Primärschlüssel besitzen. Wenn Sie Kurznamen für Sichten erstellen, besitzen die Kurznamen keine Primärschlüssel. In diesem Fall können Sie den Kurznamen ändern, um eine informative Integrationsbedingung über Primärschlüssel hinzuzufügen. Beispiel:

```
CREATE NICKNAME SCHEMA.LOCATION FOR SERVER.SH.V_LOCATION;  
ALTER NICKNAME SCHEMA.LOCATION  
ADD CONSTRAINT P1 PRIMARY KEY ( LOCATION_CODE ) NOT ENFORCED;
```

Aktualisieren von Statistikdaten für Kurznamen

Durch das Abrufen der Statistikdaten für Kurznamen wird sichergestellt, dass das Abfrageoptimierungsprogramm die neuesten Informationen zum Kurznamen verwendet, wenn es die Abfragezugriffspläne erstellt.

Aktualisierungsfunktion für Kurznamenstatistiken - Übersicht

Das Abrufen von Statistikdaten für einen Kurznamen dient dazu, dem Abfrageoptimierungsprogramm präzise und aktuelle Angaben zu dem Kurznamen bereit zu stellen. Das Optimierungsprogramm ermittelt mithilfe dieser Informationen einen optimalen Zugriffsplan für eine Abfrage.

Wenn Sie einen Kurznamen für ein Datenquellenobjekt registrieren, fügt der Server mit föderierten Datenbanken dem Systemkatalog in der föderierten Datenbank Informationen zu diesem Datenquellenobjekt hinzu. Das Abfrageoptimierungsprogramm plant mithilfe dieser Informationen die Vorgehensweise beim Abruf von Daten aus dem Datenquellenobjekt. Die föderierte Datenbank erkennt Änderungen an den Datenquellenobjekten nicht automatisch. Daher können die Angaben im Systemkatalog veraltet sein.

Sie können die gerade verfügbaren Statistikdaten zu Datenbankkurznamen, -spalten und -indizes aus der fernen Datenquelle abrufen. Sie können Statistikdaten für Kurznamen abrufen und aktualisieren, indem Sie die DB2-Steuerzentrale oder den DB2-Befehlszeilenprozessor verwenden.

Normalerweise verwenden Sie die DB2-Steuerzentrale oder den DB2-Befehlszeilenprozessor, um Statistikdaten für Kurznamen manuell zu aktualisieren, wenn neue Statistikdaten für eine ferne Tabelle erfasst wurden. Darüber hinaus sorgt die Funktion zur automatischen Statistikerfassung standardmäßig dafür, dass die Statistikdaten für Kurznamen auf dem neuesten Stand sind.

Normalerweise verwenden Sie die DB2-Steuerzentrale oder den DB2-Befehlszeilenprozessor, um Statistikdaten für Kurznamen manuell zu aktualisieren, wenn neue Statistikdaten für eine ferne Tabelle erfasst wurden. Darüber hinaus sorgt die Funktion zur automatischen Statistikerfassung standardmäßig dafür, dass die Statistikdaten für Kurznamen auf dem neuesten Stand sind.

Sie können eine Kurznamenstatistik für folgende relationale Datenquellen abrufen:

- DB2-Produktfamilie (DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Sie können eine Kurznamenstatistik für folgende nicht relationale Datenquellen abrufen:

- BioRS
- Excel
- Dateien mit Tabellenstruktur

- XML im Rootkurznamen

Die erfassten Statistikdaten variieren je nach Datenquelle.

Sie können folgende Statistikdaten mit der katalogbasierten Methode zur Statistikerfassung aktualisieren:

- CARD
- FPAGES
- NPAGES
- OVERFLOW
- COLCARD
- HIGH2KEY
- LOW2KEY
- NLEAF
- NLEVELS
- CLUSTERFACTOR
- CLUSTERRATIO
- FULLKEYCARD
- FIRSTKEYCARD

Sie können folgende Statistikdaten mit der datenbasierten Methode zur Statistikerfassung aktualisieren:

- CARD
- COLCARD
- HIGH2KEY
- LOW2KEY
- FULLKEYCARD
- FIRSTKEYCARD

Sie können die Statistikdaten zu einem oder allen Kurznamen in einem Schema in einer bestimmten Serverdefinition abrufen. Sie können auch Statistikdaten zu Kurznamen in Datenquellenobjekten mit Oracle Label Security abrufen. Bei einer Datenbank mit eingeschränktem Zugriff sind die Statistikdaten für HIGH2KEY und LOW2KEY nur für die Benutzer mit dem entsprechenden Zugriffsrecht sichtbar. Bei Datenbanken mit uneingeschränktem Zugriff ist die Kurznamenstatistik in Objekten mit Oracle Label Security ungeschützt und stellt möglicherweise ein Sicherheitsrisiko dar. In diesen Fällen können Sie den Zugriff auf die Systemkataloge beschränken, die schutzwürdige Informationen enthalten. Wenn das Abrufen fehlschlägt, macht die Datenbank die Änderungen rückgängig.

Sie können Statistikdaten zu Kurznamen in jedem Betriebssystem abrufen, das Server mit föderierten Datenbanken unterstützt.

Methoden für den Abruf von Kurznamenstatistik

Sie können die Methode für die Statistikerfassung wählen und Ihre Auswahl auf bestimmte Spalten und Indizes beschränken. Die katalogbasierte Methode bietet bessere Leistungswerte. Die datenbasierte Methode dagegen stellt Ihnen aktuellere Statistikdaten bereit, benötigt aber mehr Zeit zur Ausführung.

Sie können eine der folgenden Methoden für die Statistikerfassung auswählen:

- **Katalogbasierte Methode**

Bei der katalogbasierten Methode werden die Statistikdaten aus den Katalogtabellen der Datenquelle in die föderierte Katalogtabelle kopiert. Es werden nur diejenigen Statistikdaten kopiert, die semantisch zu föderierten Statistikdaten zugeordnet werden können. Allerdings sind die Statistikdaten zu den Kurznamen nur so präzise und aktuell wie die Informationen, die gerade im Katalog der fernen Quelle enthalten sind. Sind die statistischen Daten nicht auf dem neuesten Stand, sind auch die erfassten Statistikdaten zu den Kurznamen nicht auf dem neuesten Stand. Stellen Sie bei Verwendung der katalogbasierten Methode sicher, dass in der fernen Quelle aktuelle Statistikdaten vorhanden sind.

Da die Statistikdaten aus dem Katalog der fernen Quelle in den Katalog des Servers mit föderierten Datenbanken kopiert werden, ist die katalogbasierte Methode der Statistikerfassung in der Regel sehr schnell.

- **Datenbasierte Methode**

Die datenbasierte Methode ist nicht von den Statistikdaten an der fernen Quelle abhängig. Diese Methode generiert auf empirischem Weg ihre eigenen Statistikdaten, und zwar mithilfe der Ergebnisse der Abfragen, die sie für die Kurznamen absetzt. Bei dieser Methode sind die erfassten Statistikdaten eine präzise Darstellung der fernen Daten.

Die datenbasierte Methode kann viel Zeit in Anspruch nehmen, wenn die betroffenen Kurznamen eine hohe Zeilenzahl aufweisen. In der Regel umfassen die Abfragen umfangreiche Sortiervorgänge und Zusammenfassungen. Wählen Sie die datenbasierte Statistikerfassung daher nur, wenn mit der katalogbasierten Methode keine befriedigenden Statistikdaten generiert werden können.

Wenn Sie die Leistung der datenbasierten Methode zu Lasten der Qualität der erfassten Statistikdaten steigern wollen, können Sie die Statistikerfassung auf die Spalten- und Indextypen beschränken, die den größten Nutzen daraus ziehen. Zu diesen Spaltentypen zählen Spalten, die in Vergleichselemente, Joinschlüssel oder Gruppierungsoperationen einbezogen sind oder die zu einem oder mehreren Indizes gehören.

Bei der katalogbasierten Methode müssen Sie die Statistikerfassung normalerweise nicht auf bestimmte Spalten oder Indizes beschränken, da bei dieser Erfassungsmethode nur ein geringer Systemaufwand anfällt.

Abrufen von Kurznamenstatistik

Sie können Statistikdaten für einen Kurznamen abrufen, um sicherzustellen, dass das Abfrageoptimierungsprogramm die Angaben zu dem Kurznamen verwendet, die an der Datenquelle verfügbar sind. Sie können die Kurznamenstatistik für einen, mehrere oder alle Kurznamen aktualisieren.

Informationen zu dieser Task

Das Abfrageoptimierungsprogramm stellt statistische Daten wie die Kurznamenstatistik HIGH2KEY und LOW2KEY für Datenquellenobjekte zusammen, die eine kennsatzbasierte Zugriffssteuerung (LBAC) oder Oracle Label Security verwenden. In Datenbanken, die diese Form der Sicherheit nutzen, können nur die Benutzer mit der entsprechenden Zugriffsebene die Statistikdaten anzeigen. In nicht eingeschränkten Datenbanken können Sie den Zugriff einschränken oder die Kurznamenstatistik HIGH2KEY und LOW2KEY auf einen leeren oder bedeutungslosen Wert einstellen.

Vorbereitung

Wenn Sie die Statistikdaten über die Befehlszeile aktualisieren, gelten folgende Voraussetzungen:

- Der Server mit föderierten Datenbanken erstellt die Protokolldatei auf dem Server. Die Verzeichnisse, die Sie in dem Pfad aufführen, müssen vorhanden sein.
- Die Zugriffsrechte für die abgeschirmte Benutzer-ID der föderierten Instanz müssen das Zugriffsrecht zum Erstellen der Protokolldatei an der angegebenen Speicherposition einschließen.

Einschränkungen

Die Benutzer-ID, mit der Sie die Verbindung zur föderierten Datenbank herstellen, muss der Tabelle für die ferne Datenquelle zugeordnet werden.

Wenn der lokale Spaltenname oder -typ nicht die standardmäßige Typenzuordnung von dem fernen Spaltennamen oder -typ darstellt, ruft das Dienstprogramm für die Aktualisierung der Kurznamenstatistik keine Statistikdaten zu den Spalten ab.

Informationen zu dieser Task

Standardmäßig umfasst der Versuch zur Statistikerfassung alle Spalten und Indizes eines Kurznamens. Sie können die Statistik auf bestimmte Spalten und Indizes beschränken und eine Protokolldatei angeben.

Vorgehensweise

Rufen Sie zur Aktualisierung der Kurznamenstatistik über die DB2-Befehlszeile oder innerhalb einer Anwendung die gespeicherte Prozedur SYSPROC.NNSTAT auf.

Gehen Sie wie folgt vor, um die Kurznamenstatistik über die DB2-Steuerzentrale zu aktualisieren:

1. Wählen Sie die Kurznamen aus, für die Sie aktuelle Statistikdaten abrufen wollen.
2. Wenn kein DB2-Toolskatalog vorhanden ist, wird ein Fenster angezeigt, über das Sie den DB2-Toolskatalog erstellen können. Erstellen Sie den Toolskatalog, wenn Sie die Aktualisierung der Kurznamenstatistik terminieren wollen.
3. Geben Sie die Einstellungen für die Aktualisierung an.

Abrufen von Statistikdaten für mehrere Kurznamen (DB2-Steuerzentrale)

Sie können Statistikdaten für mehrere Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile aktualisieren.

Vorgehensweise

Gehen Sie wie folgt vor, um die Statistikdaten für mehrere Kurznamen über die DB2-Steuerzentrale zu aktualisieren:

1. Wählen Sie die erforderlichen Kurznamen aus:
 - Kurznamen mit Serverdefinition:
 - a. Erweitern Sie den Ordner **Objekte föderierter Datenbanken**. Wählen Sie den Wrapperordner aus, mit dem Sie arbeiten wollen.
 - b. Erweitern Sie den Ordner **Serverdefinitionen**. Wählen Sie den Serverordner aus, der die Kurznamen enthält, mit denen Sie arbeiten wollen.

- c. Klicken Sie den Ordner **Kurznamen** doppelt an.
 - d. Klicken Sie die Namen der Kurznamen mit der rechten Maustaste an, und wählen Sie **Statistikdaten** aus.
 - e. Wählen Sie **Aktualisieren** aus. Das Notizbuch 'Statistik aktualisieren' für mehrere Kurznamen wird geöffnet.
 - Gehen Sie wie folgt vor, um die Statistikdaten für mehrere Kurznamen zu aktualisieren, die einer bestimmten Datenbankdefinition zugeordnet sind:
 - a. Erweitern Sie den Ordner **Datenbanken**. Wählen Sie den Ordner **Datenbank** aus, der die Kurznamendefinitionen enthält, mit denen Sie arbeiten wollen.
 - b. Klicken Sie den Ordner **Kurznamen** doppelt an.
 - c. Klicken Sie die Namen der Kurznamen, die Sie aktualisieren wollen, mit der rechten Maustaste an, und wählen Sie **Statistikdaten** aus.
 - d. Wählen Sie **Aktualisieren** aus. Das Notizbuch 'Statistik aktualisieren' für mehrere Kurznamen wird geöffnet.
2. Geben Sie die Einstellungen für die Aktualisierung an:
- Seite 'Statistik aktualisieren':
 - a. Zeigen Sie die Kurznamen im Fenster 'Statistik aktualisieren' an, und stellen Sie sicher, dass dies die Kurznamen sind, deren Statistikdaten Sie aktualisieren wollen. Diejenigen Kurznamen, die Sie nicht aktualisieren wollen, können Sie inaktivieren.
 - b. Wählen Sie **Details** aus, um die zu aktualisierenden Statistikdaten auf Spalten- und Indexebene zu wählen. Das Notizbuch 'Details' wird geöffnet.
 - Seite 'Details':
 - a. Wählen Sie alle oder gezielte Spalten des Kurznamens für die Aktualisierung aus.
 - b. Wählen Sie alle oder gezielte Indizes des Kurznamens für die Aktualisierung aus.
 - c. Wählen Sie eine vorhandene Protokolldatei aus, oder geben Sie den vollständig qualifizierten Pfad einer neuen Protokolldatei ein.
 - Seite 'Methode': Wählen Sie eine der folgenden Methoden für die Statistikerfassung aus:
 - a. Katalogbasierte Methode; gültig für relationale Kurznamen
 - b. Datenbasierte Methode; gültig für relationale und nicht relationale Kurznamen
 - c. Beide Methoden; die Standardeinstellung
 - **Optional:** Geben Sie auf der Seite 'Zeitplan' an, wann Sie die Aktualisierung der Kurznamenstatistik ausführen wollen.

Abrufen von Statistikdaten für einen einzelnen Kurznamen (DB2-Steuerzentrale)

Sie können Statistikdaten für einen einzelnen Kurznamen abrufen, der einer Serverdefinition zugeordnet ist. Sie können Statistikdaten für Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile aktualisieren.

Vorgehensweise

Gehen Sie wie folgt vor, um die Statistikdaten für einen einzelnen Kurznamen über die DB2-Steuerzentrale zu aktualisieren:

1. Wählen Sie die erforderlichen Kurznamen aus.
 - Kurznamen mit Serverdefinitionen:

- a. Erweitern Sie den Ordner **Objekte föderierter Datenbanken**. Wählen Sie den Wrapperordner aus, mit dem Sie arbeiten wollen.
 - b. Erweitern Sie den Ordner **Serverdefinitionen**. Wählen Sie den Serverordner aus, der den Kurznamen enthält, mit dem Sie arbeiten wollen.
 - c. Klicken Sie den Ordner **Kurznamen** doppelt an.
 - d. Klicken Sie den Namen des Kurznamens mit der rechten Maustaste an, und wählen Sie **Statistikdaten** aus.
 - e. Wählen Sie **Aktualisieren** aus. Das Notizbuch 'Statistik aktualisieren' für einen einzelnen Kurznamen wird geöffnet.
 - Kurznamen mit Datenbankdefinition:
 - a. Erweitern Sie den Ordner **Datenbanken**. Wählen Sie den Ordner **Datenbank** aus, der die Kurznamendefinitionen enthält, mit denen Sie arbeiten wollen.
 - b. Klicken Sie den Ordner **Kurznamen** doppelt an.
 - c. Klicken Sie die Namen des Kurznamens, den Sie aktualisieren wollen, mit der rechten Maustaste an, und wählen Sie **Statistikdaten** aus.
 - d. Wählen Sie **Aktualisieren** aus. Das Notizbuch 'Statistik aktualisieren' für einen einzelnen Kurznamen wird geöffnet.
2. Geben Sie die Einstellungen für die Aktualisierung an:
- Seite 'Statistik aktualisieren':
 - a. Wählen Sie alle oder gezielte Spalten des Kurznamens für die Aktualisierung aus.
 - b. Wählen Sie alle oder gezielte Indizes des Kurznamens für die Aktualisierung aus.
 - c. Wählen Sie eine vorhandene Protokolldatei aus, oder geben Sie den vollständig qualifizierten Pfad einer neuen Protokolldatei ein.
 - Seite 'Methode': Wählen Sie eine der folgenden Methoden für die Statistikerfassung aus:
 - a. Katalogbasierte Methode; gültig für relationale Kurznamen
 - b. Datenbasierte Methode; gültig für relationale und nicht relationale Kurznamen
 - c. Beide Methoden; die Standardeinstellung
 - **Optional:** Geben Sie auf der Seite 'Zeitplan' an, wann Sie die Aktualisierung der Kurznamenstatistik ausführen wollen.

Abrufen von Kurznamenstatistik über die Befehlszeile - Beispiele

Die folgenden Beispiele zeigen, wie mit der gespeicherten Prozedur SYSPROC.NNSTAT eine Kurznamenstatistik über die Befehlszeile abgerufen wird.

Beispiel: Abrufen aller Statistikdaten

Der Server mit föderierten Datenbanken ruft die Statistikdaten für die Kurznamen auf dem Server DB2SERV ab und erstellt kein Protokoll.

```
CALL SYSPROC.NNSTAT('DB2SERV',NULL,NULL,NULL,NULL,0,NULL,?)
```

Abrufen aller Statistikdaten für ein Schema und Zurückgeben eines Protokolls

Beispiel: Abrufen von Statistikdaten mit der katalogbasierten Methode

Der Server mit föderierten Datenbanken ruft die Statistikdaten für den Kurznamen STAFF im Schema ADMIN ab. Die Statistikdaten werden für die Spalten 1 bis 5

und für die Indizes 1 und 2 zusammengestellt. Die Datenerfassung erfolgt mit der katalogbasierten Methode. Der Server mit föderierten Datenbanken schreibt das Protokoll in die Datei /home/iiuser/reportlogs/log1.txt.

```
CALL SYSPROC.NNSTAT(  
  NULL, 'ADMIN','STAFF','COL1, COL2, COL3, COL4, COL5','IND1, IND2',1,  
  '/home/iiuser/reportlogs/log1.txt',?)
```

In diesem Beispiel ruft der Server mit föderierten Datenbanken die Statistikdaten für alle Kurznamen auf dem Server DB2Serv im Schema admin ab. Der Server mit föderierten Datenbanken schreibt das Protokoll in die Datei /home/iiuser/stats/recent.log.

```
CALL SYSPROC.NNSTAT(  
  'DB2Serv', 'admin', NULL, NULL, NULL, 0, '/home/iiuser/stats/recent.log', ?)
```

Einschränkungen bei HIGH2KEY- und LOW2KEY-Statistik

Wenn Sie die HIGH2KEY- und LOW2KEY-Statistik für einen DB2-Kurznamen erfassen, werden die Informationen bei einigen Spalten nicht erfasst.

Einschränkungen

Wenn der DRDA-Wrapper einen Kurznamen für eine ferne DB2-Tabelle oder für einen Kurznamen erstellt, erfasst er die HIGH2KEY- und LOW2KEY-Statistik nur für numerische Spalten und ausschließlich dann, wenn die Spaltenkardinalität größer als 3 ist.

Vorgehensweise

Verwenden Sie zur Erfassung der HIGH2KEY- und LOW2KEY-Statistik eines der folgenden Verfahren:

- Verwenden Sie SYSPROC.NNSTAT, wobei der Parameter METHOD auf 2 gesetzt ist. Diese Einstellung gibt die datenbasierte Statistikerfassung an. Bei der datenbasierten Methode werden Daten aus der fernen Tabelle abgefragt, um die Werte für die lokalen Statistikdaten zu berechnen. Diese Methode kann jedoch auf dem fernen Server und auf dem Server mit föderierten Datenbanken eine spürbare Ressourcennutzung verursachen.
- Setzen Sie die SQL-Anweisung UPDATE ab, um die Spalten HIGH2KEY und LOW2KEY in der Sicht SYSSTAT.COLUMNS zu aktualisieren. In diesem Fall müssen Sie die korrekten Werte für HIGH2KEY und LOW2KEY manuell ermitteln.

Erstellen eines DB2-Toolskatalogs

Wenn Sie die Statistikdaten für einen Kurznamen aktualisieren, können Sie mithilfe eines DB2-Toolskatalogs die nachfolgenden Aktualisierungen der Kurznamenstatistik terminieren. Wenn Sie keinen DB2-Toolskatalog besitzen, werden Sie aufgefordert, einen Katalog zu erstellen. Die Erstellung eines DB2-Toolskatalogs ist über die DB2-Steuerzentrale oder die Befehlszeile möglich, die Aktualisierung dagegen können Sie nur über die DB2-Steuerzentrale terminieren.

Vorbereitung

Der DB2-Verwaltungsserver muss installiert sein.

Vorgehensweise

Gehen Sie wie folgt vor, um einen DB2-Toolskatalog über die DB2-Steuerzentrale zu erstellen:

1. Wenn Sie die Kurznamenstatistik aktualisieren, wird das Fenster 'Statistik aktualisieren' geöffnet.
2. Wählen Sie das System aus, auf dem Sie eine Datenbank für den DB2-Toolskatalog erstellen wollen.

Die Datenbank muss auf einem katalogisierten System erstellt werden, das keinen Metadatenpeicher besitzt. Wenn das gewünschte System nicht katalogisiert ist, müssen Sie es katalogisieren, bevor Sie die Datenbank für den DB2-Toolskatalog erstellen.

Statusanzeige bei Aktualisierungen von Kurznamenstatistik (DB2-Steuerzentrale)

Nachdem Sie eine Aktualisierung der Statistikdaten zu einem Kurznamen angefordert haben, können Sie den Status der Aktualisierung anzeigen. Sie können den Status der Aktualisierung von Statistikdaten für Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile anzeigen.

Vorgehensweise

Gehen Sie wie folgt vor, um den Status von Aktualisierungen der Kurznamenstatistik über die DB2-Steuerzentrale anzuzeigen:

1. Wählen Sie **Statistik aktualisieren** aus.
2. Wählen Sie **Ergebnisse anzeigen** aus, und prüfen Sie die Statusinformationen.

Statusanzeige bei Aktualisierungen von Kurznamenstatistik (DB2-Befehlszeile)

Nachdem Sie eine Aktualisierung der Statistikdaten zu einem Kurznamen angefordert haben, können Sie den Status der Aktualisierung anzeigen. Sie können den Status der Aktualisierung von Statistikdaten für Kurznamen über die DB2-Steuerzentrale oder die DB2-Befehlszeile anzeigen.

Vorgehensweise

In der Tabelle SYSPROC.FED_STATS können Sie den Status der Aktualisierung von Statistikdaten für Kurznamen über die DB2-Befehlszeile anzeigen.

Das folgende Beispiel zeigt eine Beschreibung der Tabelle SYSPROC.FED_STATS (die tatsächliche Spaltenlänge wurde zur Vereinfachung des Beispiels verringert):

```
db2 describe table sysproc.fed_stats
```

Spaltenname	Typenschema	Typenname	Länge	Skalierung	Nullwerte
SERVER	SYSIBM	VARCHAR	128	0	Yes
SCHEMA	SYSIBM	VARCHAR	128	0	Yes
NICKNAME	SYSIBM	VARCHAR	128	0	Yes
STATS_UPDATE_TIME	SYSIBM	TIMESTAMP	10	0	No
LOG_FILE_PATH	SYSIBM	VARCHAR	1000	0	Yes
SQLCODE	SYSIBM	INTEGER	4	0	Yes
SQLSTATE	SYSIBM	CHARACTER	5	0	Yes
STATUS	SYSIBM	VARCHAR	1000	0	Yes

8 Satz/Sätze ausgewählt.

```
db2 "select * from sysproc.fed_stats"
```

SERVER	SCHEMA	NICKNAME	STATS_UPDATE_TIME	LOG_FILE_PATH	SQLCODE
ORA8	HAROLDL	NICK1	2006-05-02-12.03.24.927112	-	1791 42704

SQLSTATE	STATUS
----------	--------

SQL1791N Die angegebene Serverdefinition, das angegebene Serverschema oder der angegebene Serverkurzname ist nicht vorhanden.

1 Satz/Sätze ausgewählt.

Gespeicherte Prozedur SYSPROC.NNSTAT

Abrufen der gerade verfügbaren Statistikdaten zu einem oder mehreren Kurznamen. Die Statistikdaten werden im Systemkatalog auf der föderierten Datenbank gespeichert.

Berechtigung

SYSPROC.NNSTAT ist eine abgeschirmte Prozedur. Die Zugriffsrechte für die abgeschirmte Benutzer-ID der föderierten Instanz müssen das Zugriffsrecht zum Erstellen der Protokolldatei an der angegebenen Speicherposition einschließen.

Syntax

```
CALL SYSPROC.NNSTAT(  
    SERVER          VARCHAR(128)  
    SCHEMA          VARCHAR(128)  
    NICKNAME        VARCHAR(128)  
    COLNAMES        CLOB(2M)  
    INDEXNAMES      CLOB(2M)  
    METHOD           SMALLINT  
    LOG_FILE_PATH   VARCHAR(1000)  
    OUT_SQLCODE     INTEGER  
    OUT_TRACE       VARCHAR(2000)  
)
```

Parameter

Server Der Server, auf dem der Server mit föderierten Datenbanken die Kurznamenstatistik zusammenstellt. Diesen Server registriert der Benutzer, um eine Datenquelle in der föderierten Datenbank zu definieren. Wenn Sie einen (1) Kurznamen angeben, können Sie für diesen Parameter den Wert NULL angeben.

Schema

Wenn NULL angegeben wird, ruft der Server mit föderierten Datenbanken alle Kurznamen unter dem angegebenen Server ab. Wenn der Serverparameter den Wert NULL hat, ruft der Server mit föderierten Datenbanken die Statistikdaten für den Kurznamen unter dem angegebenen Schema ab. Wenn der Schema- und der Kurznamenparameter den Wert NULL haben und Sie einen Server angeben, ruft der Server mit föderierten Datenbanken die Statistikdaten auf dem angegebenen Server ab.

Nickname

Der Name des Kurznamens. Wenn Sie einen Kurznamen angeben, müssen Sie auch ein Schema angeben.

Colnames

Die Namen der Spalten, die als Spaltennamen-IDs angegeben werden.

Sie können diesen Parameter nur bei einem einzelnen Kurznamen angeben. Wenn Sie Spaltennamen angeben, müssen Sie auch ein Schema und einen Kurznamen angeben.

Wird NULL angegeben, werden die Statistikdaten für alle Spalten erfasst. NULL ist der Standardwert.

Nur die angegebenen Spalten werden verarbeitet. Wird eine leere Zeichenfolge (") angegeben, werden keine Spalten verarbeitet.

Indexnames

Die Namen der Indizes, die als Indexnamen-IDs angegeben werden.

Sie können diesen Parameter nur bei einem einzelnen Kurznamen angeben. Wenn Sie Indexnamen angeben, müssen Sie auch ein Schema und einen Kurznamen angeben. Nur die angegebenen Indizes werden verarbeitet.

Wird NULL angegeben, werden die Statistikdaten für alle Indizes erfasst. NULL ist der Standardwert.

Wird eine leere Zeichenfolge (") angegeben, werden keine Indizes verarbeitet.

Methode

Die Methode zum Erfassen von Statistikdaten aus der Datenquelle.

0 oder NULL

Als erstes wird die katalogbasierte Methode verwendet. Wenn diese Methode fehlschlägt, wird die datenbasierte Methode eingesetzt. Dies ist die Standardeinstellung.

- 1 Katalogbasierte Statistikerfassung. Bei der katalogbasierten Methode werden der lokalen Kurznamenstatistik Informationen aus fernen Katalogen zugeordnet. Diese Methode ist nur für relationale Kurznamen gültig.
- 2 Datenbasierte Statistikerfassung. Bei der datenbasierten Methode werden Daten aus der fernen Tabelle abgefragt, um die Werte für die lokalen Statistikdaten zu berechnen. Diese Methode ist für relationale und nicht relationale Kurznamen gültig.

Diese Methode ist die Standardeinstellung für relationale Kurznamen, wenn die katalogbasierte Methode bei einem bestimmten Kurznamen fehlschlägt. In der Regel können die Statistikdaten nicht erfasst werden, weil Kurznamen für ferne Sichten definiert sind. In diesem Fall sind an der fernen Quelle keine Statistikdaten verfügbar.

Log_File_Path

Der Pfadname und der Dateiname für die Protokolldatei. Der Server mit föderierten Datenbanken erstellt die Protokolldatei auf dem Server. Die Verzeichnisse, die Sie in dem Pfad aufführen, müssen vorhanden sein. Verwenden Sie unter Windows zwei umgekehrte Schrägstriche, um den Protokollpfad anzugeben. Beispiel: c:\\temp\\nnstat.log. Wenn Sie NULL angeben, erstellt der Server mit föderierten Datenbanken kein Protokoll.

Ausgabeparameter

out_SQLCode

Der SQL-Fehler, der sich aus der Statistik ergibt.

out_Trace

Der Trace.

Beispiel 1: In diesem Beispiel ruft der Server mit föderierten Datenbanken die Statistikdaten für den Kurznamen STAFF im Schema ADMIN ab. Die Statistikdaten werden für die Spalten 1, 3, 4, 6, 7 und 10 und für die Indizes 1 bis 3 zusammengestellt. Die Datenerfassung erfolgt mit der datenbasierten Methode. Der Server mit föderierten Datenbanken schreibt das Protokoll in die Datei /home/iuser/reportlogs/log1.txt.

```
CALL SYSPROC.NNSTAT(  
    NULL, 'ADMIN','STAFF','COL1, COL3, COL4, COL6, COL7, COL10',  
    'IND1, IND2, IND3',2,'/home/iuser/reportlogs/log1.txt',?)
```

Beispiel 2: In diesem Beispiel ruft der Server mit föderierten Datenbanken die Statistikdaten für alle Kurznamen auf dem Server DB2SERV im Schema ADMIN ab. Der Server mit föderierten Datenbanken schreibt das Protokoll in die Datei c:\reports\log1.txt.

```
CALL SYSPROC.NNSTAT(  
    'DB2SERV','ADMIN',NULL,NULL,NULL,0,'c:\reports\log1.txt',?)
```

Beispiel 3: In diesem Beispiel ruft der Server mit föderierten Datenbanken die Statistikdaten für alle Kurznamen auf dem Server DB2SERV ab und erstellt kein Protokoll.

```
CALL SYSPROC.NNSTAT(  
    'DB2SERV',NULL,NULL,NULL,NULL,0,NULL,?)
```

Automatisches Aktualisieren von Kurznamenstatistikdaten

Die automatische Erfassung statistischer Daten ist eine Funktion, die aktuelle Statistikdaten zu Tabellen und Kurzname erfasst. Diese Funktion ist standardmäßig aktiviert.

Vorbereitung

Für die Ausführung der automatischen Erfassung statistischer Daten für eine Datenquelle muss sichergestellt werden, dass eine Benutzerzuordnung für den Instanzeigner zur Herstellung einer Verbindung zur Datenquelle definiert ist.

Informationen zu dieser Task

Die automatische Erfassung statistischer Daten ist Teil der Funktion zur automatisierten Tabellenverwaltung, die bestimmt, wann Datenbankstatistikdaten aktualisiert werden müssen. Bei Kurznamen verwendet die automatische Erfassung statistischer Daten die katalogbasierte Methode der gespeicherten Prozedur für Kurznamenstatistikdaten (NNSTAT). Die katalogbasierte Methode ruft Statistikdaten für Kurznamen aus Kataloginformationen der fernen Site ab.

Sie können anpassen, welche Tabellen und Kurznamen von der automatischen Erfassung statistischer Daten berücksichtigt werden. So können Sie beispielsweise anpassen, welche Tabellen von der Funktion zur automatischen Erfassung statistischer Daten berücksichtigt werden, und speziell alle Kurznamen oder Kurznamen auf bestimmten Servern einschließen oder ausschließen.

Die automatische Erfassung statistischer Daten wird über eine Verwaltungsrichtlinie gesteuert. Dieselbe Richtlinie wird auch für die automatische Statistikerstellung (AUTO_RUNSTATS) sowie für Statistikdaten zu Kurznamen verwendet. Musterdateien und Beispiele stehen zur Verfügung, die Sie anpassen können, um eine Richtlinie zu erstellen oder zu aktualisieren.

Wenn Sie nicht möchten, dass Statistikdaten für Kurznamen automatisch erfasst werden, können Sie die Funktion für die automatische Erfassung statistischer Daten inaktivieren oder eine Richtlinie angeben, die die Standardrichtlinie ersetzt.

Vorgehensweise

Gehen Sie wie folgt vor, um die Standardfunktionsweise der automatischen Erfassung statistischer Daten für Tabellen und Kurznamen zu ändern:

- Erstellen Sie eine Richtlinie für automatische RUNSTATS-Operationen für Tabellen. Verwenden Sie die gespeicherten Systemprozeduren `SYSPROC.AUTOMAINT_SET_POLICY` und `SYSPROC.AUTOMAINT_SET_POLICYFILE`.
- Erfassen Sie Informationen zu automatisierten Verwaltungsrichtlinien für automatische RUNSTATS-Operationen für Tabellen. Verwenden Sie die gespeicherten Systemprozeduren `SYSPROC.AUTOMAINT_GET_POLICY` und `SYSPROC.AUTOMAINT_GET_POLICYFILE`.
- Definieren Sie den Wert OFF für die Konfigurationsparameter `AUTO_MAINT`, `AUTO_TBL_MAINT` und `AUTO_RUNSTATS`, um die automatische Erfassung statistischer Daten zu inaktivieren.

Kapitel 13. Arbeiten mit fernen XML-Daten

Federation unterstützt den fernen XML-Datentyp, der es Ihnen ermöglicht, auf XML-Daten in einer Datenbank von DB2 Database für Linux und UNIX und Windows zuzugreifen und die XML-Daten zu bearbeiten.

Das föderierte System hält sich an die gleiche XML-Semantik wie das DB2-Datenbanksystem. Der native XML-Datenspeicher ermöglicht das Speichern von und den Zugriff auf XML-Dokumente, die in hierarchischer Form als Spalte einer relationalen Tabelle gespeichert sind. Sie definieren die XML-Spalte mit dem XML-Datentyp.

Sie können einen relationalen Kurznamen für eine ferne Tabelle oder Sicht erstellen, die den XML-Datentyp enthält. Sie können außerdem den XML-Wrapper verwenden, um einen nicht relationalen Kurznamen für ein XML-Dokument zu erstellen.

Sie können den Kurznamen dann in den Sprachen XQuery und SQL verwenden. Die Sprache XQuery ist der primäre Mechanismus zum Abfragen von XML-Dokumenten. Mit SQL können Sie Basisoperationen ausführen, wie z. B. das Auswählen von XML-Spalten und das Einfügen, Aktualisieren oder Löschen von XML-Daten. Sie können SQL und XQuery auch integrieren, um mithilfe von SQL/XML-Funktionen und -Vergleichselementen sowie XQuery-Funktionen sowohl für vorhandene relationale Daten als auch für XML-Daten Abfragen zu erstellen.

Erstellen eines Kurznamens für ferne XML-Daten - Beispiele

Zum Arbeiten mit fernen XML-Daten müssen Sie einen Kurznamen für eine ferne Tabelle mit XML-Daten oder für ein XML-Dokument erstellen.

Beispiel: Erstellen Sie einen relationalen Kurznamen, der einer fernen DB2-Tabelle mit dem Namen XMLTABLE1 entspricht. Die Tabelle XMLTABLE1 enthält die Spalte XMLCOL, die mit dem XML-Datentyp definiert ist:

```
CREATE NICKNAME NNXML1 FOR SERVER1.SCHEMA1.XMLTABLE1;
```

Beispiel: Erstellen Sie mithilfe des XML-Wrappers einen nicht relationalen Kurznamen für ein XML-Dokument:

```
CREATE NICKNAME NNXML2
  (file_path VARCHAR(64) OPTIONS(DOCUMENT 'FILE'),
   doc XML)
FOR SERVER XML_SERVER;
```

Bearbeiten von XML-Daten - Beispiele

Nach dem Erstellen eines Kurznamens können Sie XML-Daten in der fernen Tabelle oder im XML-Dokument auf dem föderierten System abfragen und bearbeiten. Das föderierte System unterstützt alle Datenbearbeitungsoperationen für XML-Daten, die das DB2-Datenbanksystem unterstützt.

Die folgenden Beispiele veranschaulichen die verschiedenen Methoden, die Sie unter Verwendung des Kurznamens NNXML1 zum Arbeiten mit XML-Daten verwenden können.

Bearbeiten von XML-Daten mit SQL

Mit SQL können Sie grundlegende Auswahl-, Einfüge-, Aktualisierungs- und Löschoperationen ausführen.

Beispiel: Verwenden Sie die Anweisungen SELECT und INSERT, um auf XML-Daten vom Kurznamen NNXML1 zuzugreifen und diese einzufügen.

Mit der folgenden Anweisung SELECT wird das XML-Dokument auf dem föderierten Server ausgewählt:

```
SELECT xmlcol FROM NNXML1;
```

Mit der folgenden Anweisung INSERT wird eine Zeichenfolge in eine XML-Spalte des Kurznamens eingesetzt:

```
INSERT INTO NNXML1 (xmlcol) VALUES ('<a><b>My data</b></a>');
```

Bearbeiten von XML-Daten mit SQL/XML-Funktionen

Mit SQL/XML-Funktionen können Sie verschiedene Operationen ausführen, wie z. B. das Abfragen, Prüfen und Veröffentlichen von XML-Daten.

Beispiel: Verwenden Sie die Funktion XMLVALIDATE, um XML-Daten mithilfe des XML-Schemas zu prüfen, das aus der Schemaspezifikation im XML-Instanzdokument bezogen wird:

```
SELECT XMLVALIDATE(xmlcol) FROM NNXML1;
```

Bearbeiten von XML-Daten mit XQuery

Sie können XML-Daten mithilfe von XQuery-Funktionen wie z. B. der Funktion xmlcolumn bearbeiten.

Beispiel: Verwenden Sie XQuery, um Element b, das ein untergeordnetes Element des Stammelements a ist, aus der XML-Spalte XMLCOL abzurufen:

```
xquery db2-fn:xmlcolumn('NNXML1.XMLCOL')/a/b;
```

Prüfen von XML-Dokumenten anhand von XML-Schemata

Sie können prüfen, ob ein XML-Dokument gültig ist. Sie müssen das XML-Schema registrieren und die Funktion XMLVALIDATE explizit aufrufen, um das XML-Dokument zu prüfen. Die Prüfung ist optional, wird aber empfohlen.

Registrieren von XML-Schemata

Die Schemaregistrierung ist für die XML-Prüfung erforderlich. Außerdem müssen Sie das Schema registrieren, bevor Sie ein mit Annotationen versehenes Schema zerlegen (Dekomposition).

Informationen zu dieser Task

Sie registrieren ein XML-Schema im XML-Schema-Repository (XSR). Der Registrierungsprozess erstellt ein XSR-Objekt.

Die XML-Schemaregistrierung wird auf dem föderierten Server vorgenommen.

Vorgehensweise

Gehen Sie wie folgt vor, um XSR-Objekte zu registrieren:

1. Registrieren Sie das XML-Schemadokument im XML-Schema-Repository mithilfe gespeicherter Prozeduren oder mithilfe von Befehlen über den Befehlszeilenprozessor.
2. Geben Sie zusätzliche XML-Schemadokumente an, die in das XSR-Objekt eingeschlossen werden sollen, wenn das XML-Schema aus mehreren Schemadokumenten besteht.
3. Beenden Sie den Registrierungsprozess mit dem XML-Schema-Repository.

Prüfen von XML-Dokumenten

Sie müssen die Funktion XMLVALIDATE explizit aufrufen, um ein bestimmtes XML-Dokument zu prüfen.

Informationen zu dieser Task

In der Regel prüfen Sie XML-Daten während einer Einsetzungs- oder Aktualisierungsoperation mithilfe der Funktion XMLVALIDATE. Die Funktion XMLVALIDATE prüft einen XML-Wert anhand eines XML-Schemas oder anhand des XML-Schemas aus der Schemaspezifikation im Instanzdokument.

Wenn Sie eine XML-Spalte in einen Kurznamen einsetzen oder in einem Kurznamen aktualisieren und die Funktion XMLVALIDATE verwenden, wird auf dem föderierten Server eine Prüfung ausgeführt. Anschließend serialisiert der föderierte Server die Daten und sendet sie an die Datenquelle. Während der Serialisierung werden die Datentypenannotationen, die von XMLVALIDATE hinzugefügt wurden, nicht beibehalten.

Vorgehensweise

Gehen Sie wie folgt vor, um ein XML-Dokument zu prüfen:

Rufen Sie die Funktion XMLVALIDATE als Teil einer SQL-Anweisung auf.

- Wenn XMLVALIDATE ohne Schemaspezifikation aufgerufen wird, wird das Schema auf der Basis des Attributs *schemaLocation* im Instanzdokument ermittelt. Wenn keine Schemainformationen vorhanden sind, wird eine Fehlermeldung abgesetzt.
- Wenn XMLVALIDATE mit einem registrierten Schema oder mit einer URI (Uniform Resource Identifier) aufgerufen wird, wird die Prüfung anhand dieses bestimmten Schemas ausgeführt. Wird ein externes Schema angegeben, überschreibt das externe Schema die interne Schemaspezifikation.

Beispiel: Prüfen Sie ein XML-Dokument mithilfe des XML-Schemas MYSCHEMA.MYDOCUMENTS:

```
INSERT INTO NNXML1(XMLCOL)
VALUES (
XMLVALIDATE(
? ACCORDING TO XMLSCHEMA ID MYSCHEMA.MYDOCUMENTS
)
)
```

Wenn das XML-Schema, das der SQL-ID MYSCHEMA.MYDOCUMENTS zugeordnet ist, im XML-Repository vorhanden ist, wird der XML-Eingabewert geprüft.

Zerlegen von XML-Dokumenten, die über mit Annotationen versehene XML-Schemata verfügen, in Kurznamen

Mit der Dekomposition (Zerlegung) auf der Basis von mit Annotationen versehenen XML-Schemata können Sie Dokumente zerlegen, die auf Annotationen basieren, die in einem XML-Schema angegeben sind.

Vorbereitung

Die mit Annotationen versehenen Schemadokumente müssen im XML-Schema-Repository (XSR) gespeichert und für dieses Repository registriert werden.

Informationen zu dieser Task

Wenn es erforderlich ist, dass Sie auf XML-Daten nicht als hierarchische, sondern als relationale Daten zugreifen, können Sie die XML-Daten zerlegen bzw. zerkleinern. Sie können ein XML-Dokument in einen Kurznamen zerlegen, der auf eine ferne Tabelle verweist.

Vorgehensweise

Gehen Sie wie folgt vor, um XML-Dokumente, die über mit Annotationen versehene XML-Schemata verfügen, in einen Kurznamen zu zerlegen:

1. Versehen Sie die Schemadokumente für die XML-Schemadekomposition mit Annotationen.
2. Erstellen Sie den Kurznamen, den Sie für die Dekomposition verwenden wollen. Der Name des Kurznamens muss mit den Werten in dem mit Annotationen versehenen Schema übereinstimmen.
3. Registrieren Sie das Schema mithilfe des Befehls REGISTER XMLSCHEMA.
4. Erteilen Sie das Zugriffsrecht USAGE für das XSR-Objekt, um bestimmten Benutzern die Verwendung eines bestimmten XML-Schemas zu erlauben.
5. Aktivieren Sie das Schema für die Dekomposition mithilfe der Anweisung ALTER XSROBJECT.
6. Verwenden Sie den Befehl DECOMPOSE XML DOCUMENT, um das XML-Instanzdokument zu zerlegen.

Föderierte Verarbeitung ferner XML-Daten

Das Senden und Empfangen von XML-Daten zwischen dem föderierten System und dem fernen Datenquellenclient geschieht als serialisierte XML-Zeichenfolge im Zeichen- oder Binärformat.

Wie der föderierte Server XML-Daten sendet und empfängt, wird durch Serialisierungs-, Parsing- und Codepagekonventionen beeinflusst.

Föderiertes Parsing ferner XML-Daten

Das föderierte System verwendet den DB2-XML-Parser, um ferne XML-Daten zu verarbeiten.

Der Parser erfordert korrekt formatierte XML-Daten, die Datenverschlüsselungsregeln entsprechen. Der Parser führt keine Schemaprüfungen durch.

- Während des Datenabrufs setzt der Parser eine Fehlernachricht ab, wenn die XML-Daten nicht korrekt formatiert sind.

- Abhängig von dem Ort, an dem das Parsing vorgenommen wird, setzt der föderierte Server oder die Zieldatenquelle während des Einsetzens von Daten eine Fehlernachricht ab, wenn die XML-Daten nicht korrekt formatiert sind.

Handhabung von Leerzeichen für Federation

Beim XML-Parsing können Sie Begrenzungsleerzeichen in XML-Dokumenten beibehalten oder entfernen. Die Leerzeichen, die zwischen Elementen auftreten, sind *Begrenzungsleerzeichen (Boundary Whitespaces)*.

Der föderierte Server setzt die gleichen Leerzeichenkonventionen um wie das DB2-Datenbanksystem.

Für Anwendungshostvariablen und Parametermarken ermittelt das Sonderregister `CURRENT IMPLICIT XMLPARSE OPTION`, ob Leerzeichen während der föderierten Bindeoperation entfernt werden. Wenn die Datenquelle andere Regeln für die Handhabung von Leerzeichen verwendet, versucht das föderierte System, die Unterschiede in der Semantik zu kompensieren.

Codepageprobleme bei fernen XML-Daten

Bestimmte Codepageprobleme können sich auf föderierte Anweisungen auswirken.

Der föderierte Server hält sich an die Verschlüsselungsregeln des DB2-XML-Parsers.

Bei föderierten Anweisungen können die folgenden Probleme auftreten, wenn serialisierte ferne XML-Daten bearbeitet werden:

- Bei fernen XML-Daten, die im Binärformat serialisiert werden:
 - Wenn Daten vom fernen Client an den föderierten Wrapper gesendet werden, kommt es nicht zum Datenverlust.
 - Wenn eine interne Verschlüsselung vorhanden ist, bei der es sich nicht um ein gültiges IBM Schema für Codeumsetzung handelt, ersetzt der Wrapper das Verschlüsselungsattribut durch ein gültiges IBM Schema für Codeumsetzung, oder er entfernt das Verschlüsselungsattribut aus der XML-Deklaration, und der DB2-XML-Parser entschlüsselt den Wert.
- Bei fernen XML-Daten, die im Zeichenformat serialisiert werden:
 - Die Codepage der XML-Daten befindet sich in der Codepage der föderierten Datenbank. Wenn der Datenquellenclient Daten an den föderierten Wrapper sendet, kann es zu Datenverlust kommen. Wenn die Konvertierung zu einer Zeichenersetzung führt, wird möglicherweise eine Warnung generiert. Dies hängt vom Verhalten der Datenquellen und von der Wrapperimplementierung ab.
 - Wenn eine interne Verschlüsselung vorhanden ist, wird sie vom Wrapper entfernt, weil sich die serialisierte Zeichenfolge in der Codepage der föderierten Datenbank befindet.

Bei nicht relationalen Wrappern entschlüsselt der DB2-XML-Parser XML-Daten mithilfe der internen Verschlüsselung des Dokuments.

Einschränkungen beim fernen XML-Datentyp

Föderierte Systeme führen beim fernen XML-Datentyp zu Einschränkungen.

Sie können den fernen XML-Datentyp nur mit der Datenquelle aus DB2 für Linux, UNIX und Windows und mit dem XML-Wrapper verwenden.

Die folgenden SQL-Operationen können Sie nicht ausführen:

- Ändern des XML-Datentyps für einen Kurznamen. Dies führt zum Fehler SQL0270N.
- Erstellen eines föderierten Indexes SPECIFICATION ONLY mit der Klausel 'xml-indexspezifikation'. Dies führt zum Fehler SQL0104N.
- Definieren einer föderierten gespeicherten Prozedur mit dem XML-Datentyp. Dies führt zum Fehler SQL1254N.
- Erstellen einer Zuordnung zwischen dem XML-Datentyp und einem anderen Datentyp. Dies führt zum Fehler SQL0604N.

Die folgenden Einschränkungen gelten beim XML-Wrapper für XML-Spalten:

- Sie können für eine XML-Spalte keine Spaltenoptionen angeben.
- Nur der Stammelementkurzname kann eine einzelne XML-Spalte enthalten, die auf den gesamten Inhalt eines XML-Dokuments verweist.
- Jeder Stammelementkurzname muss mit einem einzelnen XML-Dokument übereinstimmen.
- Untergeordnete Kurznamen, XPath-Optionen für Kurznamen und Namensbereichsoptionen für Kurznamen spielen keine Rolle.

Kapitel 14. Fehlertoleranz in verschachtelten Tabellenausdrücken

Die Fehlertoleranz in verschachtelten Tabellenausdrücken ist ein Mechanismus, mit dessen Hilfe die Ausführung einer Abfrage fortgesetzt werden kann, während bestimmte SQL-Fehler in verschachtelten Tabellenausdrücken toleriert werden. Dank der Fehlertoleranz können Sie maximale Abfrageergebnisse aus verfügbaren Daten erzielen, anstatt dass eine Fehlermeldung für einen Teil der Abfrage empfangen und die gesamte Abfrage beendet wird.

Wird auf dem Server mit föderierten Datenbanken ein zulässiger Fehler gefunden, lässt der Server den Fehler zu und setzt die Verarbeitung des übrigen Teils der Abfrage fort, anstatt einen Fehler für die gesamte Abfrage zurückzugeben. Die Ergebnismenge, die der Server mit föderierten Datenbanken zurückgibt, kann ein Teilergebnis oder ein leeres Ergebnis sein.

Wenn der Server mit föderierten Datenbanken Fehler toleriert, gibt er selbst dann Abfrageergebnisse zurück, wenn die Datenquellen, auf die die Abfrage zugreift, nicht verfügbar sind. Dieser Mechanismus ist nützlich, wenn Sie trotz unvollständiger Abfrageergebnisse alle verfügbaren Informationen zurückgeben müssen. Ein Arzt könnte beispielsweise Informationen zu einem bestimmten Krankheitszustand benötigen. Er führt eine Abfrage aus, um Informationen von fernen Datenquellen aus verschiedenen Krankenhäusern zu erfassen. Auch wenn von einem oder mehreren Krankenhäusern keine Datenbanken zur Verfügung stehen, sind die Ergebnisse der verfügbaren Datenbanken für den Arzt dennoch von großem Wert.

Abfragen mit UNION ALL-Zweigen können von der Fehlertoleranz profitieren. Ohne diesen Mechanismus stoppt der Server mit föderierten Datenbanken die Abfrage, wenn bei der Verarbeitung eines Zweigs der Abfrage ein Fehler auftritt. Wenn Sie bei diesem Mechanismus den zu tolerierenden Fehler auf demselben Zweig der Abfrage angeben, toleriert der Server mit föderierten Datenbanken den Fehler und setzt die Navigation zu den übrigen verfügbaren Zweigen fort. Die UNION ALL-Operation gibt die Ergebnisse von allen verfügbaren Datenquellen zurück.

Beispiel: Mit der folgenden Abfrage werden Daten von drei Kurznamen auf drei verschiedenen Servern ausgewählt.

```
SELECT c1 from nickname1_on_server1
UNION ALL
SELECT c1 from nickname2_on_server2
UNION ALL
SELECT c1 from nickname3_on_server3
```

Wenn nickname2_on_server2 nicht verfügbar ist oder wenn der ferne Server server2 während der Abfrageverarbeitung nicht verfügbar ist, erhalten Sie die Ergebnismenge von nickname1_on_server1 und nickname3_on_server3, indem die Fehler bei nickname2 und server2 ignoriert werden. Eine Ergebnismenge von zwei der drei Zweige entspricht der Ausführung der folgenden Abfrage:

```
SELECT c1 from nickname1_on_server1
UNION ALL
SELECT c1 from nickname3_on_server3
```

Sie können die SQL-Fehler angeben, die Sie bei der Abfrageverarbeitung in einem verschachtelten Tabellenausdruck zulassen wollen. Die Fehlertypen, die der Server mit föderierten Datenbanken toleriert, sind Fehler bei Remoteverbindungen, Berechtigungen und Authentifizierung.

Angeben der Fehlertoleranz in verschachtelten Tabellenausdrücken

Sie geben die zu tolerierenden Fehler in einem verschachtelten Tabellenausdruck mit der Klausel RETURN DATA UNTIL an.

Informationen zu dieser Task

Wenn Sie die Klausel RETURN DATA UNTIL verwenden, müssen Sie die Fehlercodes angeben, die toleriert werden sollen. In der folgenden Tabelle werden die Fehler aufgeführt, die in der Klausel *specific-condition-value* zulässig sind. Sie müssen einen SQLSTATE-Wert oder einen SQLSTATE- und SQLCODE-Wert angeben, der mit einem gültigen Fehlercode übereinstimmt. Die in der Tabelle aufgeführten SQLCODE-Werte sind erforderlich.

Tabelle 19. In der Klausel 'specific-condition-value' zulässige Fehler

SQLSTATE	Fehlercode	SQLCODE
08001	SQL30080N	-30080
08001	SQL30081N	-30081
08001	SQL30082N	-30082
08001	SQL1336N	-1336
08004	Beliebig	Beliebig
28000	Beliebig	Beliebig
42501	Beliebig	Beliebig
42512	Beliebig	Beliebig
42704	SQL0204N	-204
42720	Beliebig	Beliebig

Vorgehensweise

Erstellen Sie zur Angabe von verschachtelten Tabellenausdrücken für Fehlertoleranz eine SQL-Anweisung, die die Klausel RETURN DATA UNTIL enthält.
RETURN DATA UNTIL *specific-condition-value*

RETURN DATA UNTIL

Alle vor dem Auftreten der angegebenen Bedingung vom Fullselect zurückgegebenen Zeilen werden in der Ergebnismenge zurückgegeben.

specific-condition-value

Gibt die Bedingung und die Werte für die Fehlertoleranz ein.

FEDERATED

Erforderliches Schlüsselwort. Die bestimmte Bedingung, die Sie angeben, darf nur Fehler einschließen, die an einer föderierten Datenquelle auftreten.

SQLSTATE VALUE *zeichenfolgekonstante*

Sie können eine bestimmte Bedingung als SQLSTATE-Wert angeben. Wenn VALUE angegeben wird, muss die Länge der Zeichenfolgekonstanten 5 betragen. Ein SQLSTATE-Wert kann auf einen bestimmten SQLCODE-Wert eingegrenzt werden. Sie können mehrere SQLCODE-Werte angeben, die denselben SQLSTATE-Wert in einer Klausel *specific-condition-value* gemeinsam nutzen.

Fehlertoleranz in verschachtelten Tabellenausdrücken - Beispiel

Die folgenden Beispiele zeigen, wie mithilfe der Klausel RETURN DATA UNTIL Abfrageergebnisse zurückgegeben werden, wenn eine oder mehrere Datenquellen nicht verfügbar sind.

Beispiel: Mit der folgenden SQL-Anweisung werden Daten von drei verschiedenen Servern ausgewählt: SQLServer, Oracle und Sybase.

```
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN(SELECT c1 FROM n1_from_SQLServer) etq1
UNION ALL
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN (SELECT c1 FROM n2_from_Oracle) etq2
UNION ALL
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN(SELECT c1 FROM n3_from_Sybase) etq3;
```

Szenario 1: Ein Server ist nicht verfügbar.

In diesem Szenario ist der Oracle-Server nicht verfügbar. Der SQLServer-Server und der Sybase-Server sind verfügbar. In dieser Situation wird bei der Abfrage im zweiten Zweig der UNION ALL-Operation eine leere Ergebnismenge mit dem Fehler 30080 zurückgegeben, der als tolerierbar definiert ist. Die Abfrage gibt die Ergebnisse von n1_from_SQLServer und n3_from_Sybase zurück. Die Warnung sqlwarn5='E' wird ausgegeben.

Die Ergebnismenge entspricht der Ausführung der folgenden Abfrage:

```
SELECT c1 FROM n1_from_SQLServer
UNION ALL
SELECT c1 FROM n3_from_Sybase;
```

Szenario 2: Alle Server sind nicht verfügbar.

In diesem Szenario sind sämtliche Server (SQLServer, Oracle und Sybase) nicht verfügbar. In dieser Situation gibt die UNION ALL-Operation eine leere Ergebnismenge zurück. Die Warnung sqlwarn5='E' wird ausgegeben.

Szenario 3: Alle Server sind verfügbar.

Wenn alle Server verfügbar sind, entspricht die Ergebnismenge der Abfrage der Ausführung derselben Abfrage ohne Angabe der Klausel RETURN DATA UNTIL.

Datenquellenunterstützung für Fehlertoleranz in verschachtelten Tabellenausdrücken

Fehlertoleranz wird für mehrere relationale Datenquellen und für nicht relationale Kurznamen unterstützt.

Fehlertoleranz in verschachtelten Tabellenausdrücken wird für folgende relationale Datenquellen unterstützt:

- DB2-Produktfamilie (DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Sie können nicht relationale Kurznamen in verschachtelten Tabellenausdrücken für Fehlertoleranz verwenden. Der Server mit föderierten Datenbanken kann die zulässigen Verbindungs-, Authentifizierungs- oder Berechtigungsfehler tolerieren, wenn die nicht relationalen Wrapper einen gültigen Fehlercode zurückgeben.

Einschränkungen bei der Fehlertoleranz in verschachtelten Tabellenausdrücken

Bei der Definition fehlertoleranter verschachtelter Tabellenausdrücke gibt es bestimmte Einschränkungen.

Wenn Sie eine Abfrage oder Sicht mit einem Ausdruck definieren, der die Klausel RETURN DATA UNTIL enthält, ist die Abfrage oder Sicht schreibgeschützt. Cursor, die in Ausdrücken mit der Klausel RETURN DATA UNTIL deklariert werden, sind schreibgeschützt. In jeder dieser Situationen werden Fehler zurückgegeben.

Kapitel 15. Überwachen von Servern mit föderierten Datenbanken und Kurznamen

Mithilfe der Diagnosemonitor- und Systemmonitorelemente können Sie ein System föderierter Datenbanken überwachen.

Diagnoseanzeiger für Kurznamen und föderierte Server

Sie können Diagnoseanzeiger in der DB2-Diagnosezentrale verwenden, um den Status Ihrer Kurznamen und Server für föderierte System zu überwachen.

Der Diagnoseanzeiger für Kurznamen lautet `db.fed_nicknames_op_status`. Der Diagnoseanzeiger für Serverdefinitionen lautet `db.fed_servers_op_status`. Die Diagnoseanzeiger für föderierte Systeme werden bei der Installation des Diagnosemonitors installiert.

Standardmäßig aktiviert die Diagnosezentrale die Diagnoseanzeiger für föderierte Systeme nicht. Der Benutzer muss die Anzeiger aktivieren.

Wenn der Status eines Kurznamens oder Servers abnormal ist, geben die Diagnoseanzeiger eine Warnung aus. Sie können die Überwachungsergebnisse mit der Diagnosezentrale oder über die Befehlszeile anzeigen.

Diagnoseanzeiger werden von Servern mit föderierten Datenbanken unterstützt, die die Betriebssysteme AIX, HP-UX, Linux, Microsoft Windows und Solaris verwenden.

In Tabelle 20 werden die Diagnoseanzeiger für Kurznamen und Server für föderierte Systeme beschrieben.

Tabelle 20. Diagnoseanzeiger für Kurznamen und Server

Diagnoseanzeiger	Beschreibung
<code>db.fed_nicknames_op_status</code>	<p>Gibt den Ergebnisstatus aller relationalen Kurznamen an, die in einer Datenbank auf einem Server mit föderierten Datenbanken definiert wurden.</p> <p>Gibt eine Warnung aus, wenn ein Kurzname ungültig ist. Enthält Details zu den ungültigen Kurznamen sowie Informationen zu Schritten, die Sie zur Behebung dieses Problems ausführen können.</p>
<code>db.fed_servers_op_status</code>	<p>Gibt den Ergebnisstatus aller Server mit föderierten Datenbanken an, die in einer Datenbank auf einem Server mit föderierten Datenbanken definiert wurden.</p> <p>Gibt eine Warnung aus, wenn ein Server nicht verfügbar ist. Enthält Details zu den nicht verfügbaren Servern sowie Informationen zu Schritten, die Sie zur Behebung dieses Problems ausführen können.</p>

Die Diagnoseanzeiger können die folgenden Datenquellen auswerten:

- DB2-Produktfamilie (DRDA)
- Excel

- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Dateien mit Tabellenstruktur
- Teradata
- XML (nur Rootkurznamen)

Aktivieren der Diagnoseanzeiger von föderierten Systemen

Sie müssen die Diagnoseanzeiger für föderierte Systeme aktivieren, um den Status von Kurznamen und Servern zu überwachen. Der Diagnoseanzeiger für Kurznamen lautet `db.fed_nicknames_op_status`. Der Diagnoseanzeiger für Serverdefinitionen lautet `db.fed_servers_op_status`.

Vorgehensweise

Öffnen Sie die DB2-Diagnosezentrale und konfigurieren Sie die Diagnoseanzeiger, um die Diagnoseanzeiger für föderierte Systeme zu aktivieren. Sie können diese Aufgabe auch über den Befehlszeilenprozessor ausführen.

Überwachen des Status von Kurznamen und Servern für föderierte Datenbanken

Die Überwachung des Status von Kurznamen und Servern kann bei der Bestimmung und Lösung von Problemen mit Ihrem föderierten System nützlich sein. Mit Diagnoseanzeigern in der Diagnosezentrale können Sie den Status von Kurznamen und Servern für föderierte Systeme überwachen.

Vorbereitung

- Stellen Sie sicher, dass die SELECT-Zugriffsrechte für die Kurznamen im Server mit föderierten Datenbanken definiert wurden.
- Setzen Sie den Konfigurationsparameter `FEDERATED` des Datenbankmanagers auf `YES (JA)`.
- Wenn für die Datenquelle eine Authentifizierung erforderlich ist, muss die Datenquelle über Benutzerzuordnungen der ID des Diagnosemonitors verfügen. Der Diagnosemonitor verwendet diese Zuordnung zur Herstellung einer Verbindung zur Datenquelle.

Einschränkungen

„Diagnoseanzeiger für Kurznamen und föderierte Server“ auf Seite 201 listet die Datenquellen auf, die der Diagnoseanzeiger auswerten kann.

Informationen zu dieser Task

Sie können die Überwachungsergebnisse mit der Diagnosezentrale oder über die Befehlszeile anzeigen. Verwenden Sie die DB2-Steuerzentrale, oder den DB2-Befehlszeilenprozessor, um die von den Diagnoseanzeigern festgestellten Probleme zu beheben.

Vorgehensweise

Geben Sie den Befehl GET HEALTH SNAPSHOT ein, um diese Task von der Befehlszeile aus auszuführen.

Gehen Sie wie folgt vor, um diese Task über die DB2-Steuerzentrale auszuführen:

1. Öffnen Sie die Diagnosezentrale.
2. Öffnen Sie den Empfehlungsadvisor, um Empfehlungen zur Lösung der ungültigen Kurznamen oder der nicht verfügbaren Server anzuzeigen.
3. Geben Sie den Befehl GET HEALTH SNAPSHOT ein, um diese Task von der Befehlszeile aus auszuführen.

Überwachen des Status von Kurznamen und Servern für föderierte Datenbanken - Beispiel

Dieses Thema enthält ein Beispiel für eine Diagnosemomentaufnahme einer Datenbank.

Die Namen der Diagnoseanzeiger für föderierte Systeme lauten db.fed_nicknames_op_status und db.fed_servers_op_status. Sie müssen diese Diagnoseanzeiger entweder über die Diagnosezentrale oder durch Verwendung der folgenden Befehle im CLP aktivieren:

```
db2 update alert cfg for databases using db.fed_nicknames_op_status set
    THRESHOLDSCHECKED YES
db2 update alert cfg for databases using db.fed_servers_op_status set
    THRESHOLDSCHECKED YES
```

Der folgende Befehl ruft eine Diagnosemomentaufnahme der Datenbank sowie die föderierten Diagnoseanzeiger ab, sofern diese aktiviert wurden:

```
db2 get health snapshot for database on <datenbankname>
```

In diesem Beispiel lautet der Datenbankname 'fedhi'. Die Ausgabe dieses Befehls gibt an, dass sich beide Diagnoseanzeiger in einem normalen Status befinden. 'Normal' bedeutet, dass die Kurznamen und Server gültig sind.

Diagnosemomentaufnahme der Datenbank

```
Zeitmarke für Momentaufnahme = 02/10/2006 12:10:55.063004
Datenbankname                 = FEDHI
Datenbankpfad                  = C:\DB2\NODE0000\SQL00006\
Aliasname der Eingabedatenbank = FEDHI
Betriebssystem auf Datenbank-Server = NT
Position der Datenbank         = Lokal
Höchster Alertstatus für Datenbank = Achtung
```

Diagnoseanzeiger:

```
    Name des Diagnoseanzeigers      = db.fed_servers_op_status
    Wert                             = 0
```

```
Bewertungszeitmarke = 02/10/2006 12:09:10.961000
    Alertstatus       = Normal
```

```
    Name des Diagnoseanzeigers      = db.fed_nicknames_op_status
    Wert                             = 0
```

```
Bewertungszeitmarke = 02/10/2006 12:09:10.961000
    Alertstatus       = Normal
```

```
    Name des Diagnoseanzeigers      = db.db_op_status
    Wert                             = 0
```

```
Bewertungszeitmarke = 02/10/2006 12:08:10.774000
    Alertstatus       = Normal
```

```

Name des Diagnoseanzeigers          = db.sort_shrmem_util
Wert                                  = 0
Einheit                              = %
Bewertungszeitmarke = 02/10/2006 12:08:10.774000
Alertstatus                          = Normal

Name des Diagnoseanzeigers          = db.spilled_sorts
Wert                                  = 0
Einheit                              = %
Bewertungszeitmarke = 02/10/2006 12:09:10.961000
Alertstatus                          = Normal

```

Ausführen des Snapshot Monitor für föderierte Systeme - Übersicht

Sie können den Snapshot Monitor (Überwachungsprogramm für Momentaufnahmen) verwenden, um die zu einem bestimmten Zeitpunkt verfügbaren Informationen zu Datenquellen für föderierte Systeme und den zugehörigen Anwendungen zu erfassen.

Momentaufnahmen dienen zur Ermittlung des Status eines föderierten Systems. Wenn Momentaufnahmen in regelmäßigen Abständen erstellt werden, können sie auch zur Beobachtung von Trends und zur Prognose potenzieller Fehler eingesetzt werden.

Die Ausgabe des Snapshot Monitor ist in den folgenden Formaten verfügbar:

- In Textform, über die Schnittstelle des Befehlszeilenprozessors des Snapshot Monitor.
- Als Ausgabe von Tabellenfunktionen. Diese Ausgabe ist sinnvoll zum Schreiben von Abfragen, bei denen die Ausgabe begrenzt ist.

Folgende Momentaufnahmen sind für föderierte Workloads besonders nützlich:

Momentaufnahme von dynamischen SQL-Anweisungen

Stellt eine Momentaufnahme aller dynamischen SQL-Anweisungen bereit, die sich momentan im Anweisungscache befinden. Der Cache beinhaltet föderierte und nicht föderierte Anweisungen.

Momentaufnahme von Anwendungen

Stellt Informationen über eine bestimmte Anwendung sowie den Text der momentan ausgeführten SQL-Anweisung bereit.

Momentaufnahme von fernen Datenbanken

Stellt Informationen über eine bestimmte Datenbank in einem föderierten System bereit.

Momentaufnahme von allen fernen Datenbanken

Stellt Informationen über jede aktive Datenbank in einem föderierten System bereit.

Momentaufnahme von fernen Anwendungen

Stellt Informationen auf Anwendungsebene für jede aktive Anwendung in einem föderierten System bereit.

Überwachen von föderierten Abfragen

Durch die Überwachung von Abfragen können Sie die Leistung Ihres föderierten Systems ermitteln. Wenn Sie Informationen zur Verarbeitung einer Abfrage durch ein föderiertes System benötigen, können Sie eine Momentaufnahme der fernen Abfrage abrufen.

Informationen zu dieser Task

Der Snapshot Monitor überwacht zwei Aspekte jeder Abfrage, die vom Server mit föderierten Datenbanken verarbeitet wurde:

- Die gesamte föderierte Abfrage, die von der Anwendung übergeben wurde und auf Kurznamen, lokale Tabellen oder beides verweist.
- Bei Abfragen mit Kurznamen ein oder mehrere *ferne Fragmente*. Bei fernen Fragmenten handelt es sich um die Anweisungen, die automatisch generiert und in ihren nativen Programmversionen für die föderierte Abfrage an ferne Datenquellen übergeben werden.

Bei der Überwachung von föderierten Abfragen muss sowohl die lokal im föderierten Server ausgeführte Verarbeitung als auch die Verarbeitung in fernen Servern als Antwort auf ferne Abfragefragmente berücksichtigt werden. Die Momentaufnahme der dynamischen SQL-Anweisung und die Tabellenfunktion `SNAPSHOT_DYN_SQL` enthalten Informationen über einzelne föderierte Abfragen zum Zeitpunkt der Übergabe an den föderierten Server und über ferne Abfragefragmente, die der föderierte Server an andere Datenquellen sendet.

Vorbereitung

Sie müssen den Monitorschalter `STATEMENT` auf `ON` setzen, damit die föderierte Datenbank Momentaufnahmeinformationen für ferne Abfragen sammelt.

Vorgehensweise

Verwenden Sie bei aktiver Verbindung zur föderierten Datenbank eine der folgenden Methoden, um Abfragen auf dem föderierten Server zu überwachen:

- Textausgabe:

```
GET SNAPSHOT FOR DYNAMIC SQL on datenbankname
```

Dabei ist *datenbankname* der Name der föderierten Datenbank des Servers.

- Tabellenfunktion:

```
CREATE TABLE table_snap AS (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL  
('datenbankname', -1)) as snaptab) definition only;  
INSERT INTO snap (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL ('dbname', -1))  
as snaptab);
```

Anschließend können Sie eine Abfrage mithilfe der Momentaufnahmetabelle schreiben, die jeweils pro (föderierter oder nicht föderierter) Abfrage und pro Abfragefragment eine Zeile im Anweisungscache des Servers enthält.

Der Name von fernen Abfragefragmenten ist der Server, an den sie gesendet wurden. Er wird dem fernen Abfragetext im Feld `stmt_text` der Tabellenfunktion in eckigen Klammern vorangestellt. Sie können beispielsweise die folgende Abfrage verwenden, um nach fernen Fragmenten mit langer Laufzeit zu suchen:

```
SELECT total_exec_time, rows_read, total_usr_cpu_time, num_executions,  
       substr(stmt_text,1,30)  
FROM TABLE(SNAPSHOT_DYN_SQL ('dbname', -1))AS snaptab  
-- remote fragments only  
WHERE stmt_text LIKE '[%]%'  
ORDER BY total_exec_time;
```

Beim Vergleich der Ausführungszeit einer gesamten föderierten Anweisung mit den Ausführungszeiten von fernen Fragmenten, die für die Anweisung an andere Datenquellen gesendet wurden, wird deutlich, wo die meiste Verarbeitungszeit aufgewendet wird.

Informationen dazu, welche Abfragefragmente für eine föderierte Abfrage an ferne Quellen gesendet werden, finden Sie in einem EXPLAIN-Ausführungsplan für die Abfrage.

Ausführen des Snapshot Monitor für föderierte Abfragen - Beispiel

Dieses Thema enthält ein Beispiel einer Ausgabe für die textbasierte dynamische SQL-Momentaufnahme einer föderierten Abfrage, die sich auf eine ferne Oracle-Datenquelle bezieht.

Die folgende Anweisung ruft eine Momentaufnahme aller Anweisungen ab, die sich momentan im Anweisungscache befinden, einschließlich föderierter Anweisungen und ferner Fragmente, die an andere Datenquellen gesendet wurden:

```
GET SNAPSHOT FOR DYNAMIC SQL ON <datenbankname>
```

Der Datenbankname ist der Name der lokalen föderierten Datenbank.

Die Ausgabe im folgenden Beispiel ist das Ergebnis der Anweisung:

```
GET SNAPSHOT FOR DYNAMIC SQL ON FEDDB
```

Das Beispiel zeigt eine föderierte Anweisung und ein fernes Fragment, das im Pushdown-Modus von dieser föderierten Anweisung weitergeleitet wird. Sie können ferne Fragmente identifizieren, indem Sie nach dem Namen des fernen Servers suchen, der dem Text der fernen Anweisung in eckigen Klammern vorangestellt wird. In diesem Beispiel lautet der Name des fernen Oracle-Servers 'ORA9'. Der erste Eintrag zeigt die föderierte SQL-Anweisung, die auf Kurznamen verweist, sowie deren gesamte abgelaufene Zeit. Der zweite Eintrag zeigt die ferne Anweisung, die an die Quelle [ORA9] gesendet wurde, die ihrerseits auf die fernen Oracle-Tabellennamen verweist.

Dynamisches SQL - Momentaufnahmeergebnis

Datenbankname = FEDDB

```
Anzahl Ausführungen = 1
Anzahl Kompilierungen = 1
Schlechteste Vorbereitungszeit (ms) = 475
Beste Vorbereitungszeit (ms) = 475
Intern gelöschte Zeilen = 0
Intern eingefügte Zeilen = 0
Gelesene Zeilen = 5
Intern aktualisierte Zeilen = 0
Geschriebene Zeilen = 0
Sortiervorgänge für Anweisungen = 0
Überläufe für Sortiervorgänge für Anweisungen = 0
Gesamtzeit für Sortiervorgänge = 0
Logische Lesevorgänge im Pufferpool = Nicht gesammelt
Physische Lesevorgänge im Pufferpool = Nicht gesammelt
Logische Lesevorgänge in temporären Daten des Pufferpools = Nicht gesammelt
Physische Lesevorgänge in temporären Daten des Pufferpools = Nicht gesammelt
Logische Lesevorgänge im Pufferpoolindex = Nicht gesammelt
Physische Lesevorgänge im Pufferpoolindex = Nicht gesammelt
Logische Lesevorgänge im temporären Pufferpoolindex = Nicht gesammelt
Physische Lesevorgänge im temporären Pufferpoolindex = Nicht gesammelt
Logische XDA-Lesevorgänge im Pufferpool = Nicht gesammelt
Physische XDA-Lesevorgänge im Pufferpool = Nicht gesammelt
Logische Lesevorgänge in temporären XDA-Daten des Pufferpools = Nicht gesammelt
Physische Lesevorgänge in temporären XDA-Daten des Pufferpools = Nicht gesammelt
Gesamte Ausführungszeit (sek.ms) = 1.816884
Gesamte Benutzer-CPU-Zeit (sek.ms) = 0.000000
```

```

Gesamte System-CPU-Zeit (sek.ms) = 0.020000
Anweisungstext = select count(*) from orat.supplier,
orat.nation where s_nationkey =
n_nationkey and n_name <> 'FRANCE'

Anzahl Ausführungen = 1
Anzahl Kompilierungen = 1
Schlechteste Vorbereitungszeit (ms) = 0
Beste Vorbereitungszeit (ms) = 0
Intern gelöschte Zeilen = 0
Intern eingefügte Zeilen = 0
Gelesene Zeilen = 1
Intern aktualisierte Zeilen = 0
Geschriebene Zeilen = 0
Sortiervorgänge für Anweisungen = 0
Überläufe für Sortiervorgänge für Anweisungen = 0
Gesamtzeit für Sortiervorgänge = 0
Logische Lesevorgänge im Pufferpool = Nicht gesammelt
Physische Lesevorgänge im Pufferpool = Nicht gesammelt
Logische Lesevorgänge in temporären Daten des Pufferpools = Nicht gesammelt
Physische Lesevorgänge in temporären Daten des Pufferpools = Nicht gesammelt
Logische Lesevorgänge im Pufferpoolindex = Nicht gesammelt
Physische Lesevorgänge im Pufferpoolindex = Nicht gesammelt
Logische Lesevorgänge im temporären Pufferpoolindex = Nicht gesammelt
Physische Lesevorgänge im temporären Pufferpoolindex = Nicht gesammelt
Logische XDA-Lesevorgänge im Pufferpool = Nicht gesammelt
Physische XDA-Lesevorgänge im Pufferpool = Nicht gesammelt
Logische Lesevorgänge in temporären XDA-Daten des Pufferpools = Nicht gesammelt
Physische Lesevorgänge in temporären XDA-Daten des Pufferpools = Nicht gesammelt
Gesamte Ausführungszeit (sek.ms) = 1.337672
Gesamte Benutzer-CPU-Zeit (sek.ms) = 0.000000
Gesamte System-CPU-Zeit (sek.ms) = 0.000000
Statement text = [ORA9] SELECT COUNT(*) FROM "TPCH"."NATION"
A0, "TPCH"."SUPPLIER" A1 WHERE
(A0."N_NAME" <> 'FRANCE ') AND
(A1."S_NATIONKEY" = A0."N_NATIONKEY")

```

Bei der Momentaufnahme wurden keine Pufferpoolinformationen gesammelt, da Pufferpoolinformationen für ferne Abfragen nicht gültig sind.

Systemmonitorelemente der föderierten Datenbanken

Dieses Thema beschreibt die Monitorelemente, die Informationen über föderierte Systeme liefern.

Ein Zugriff eines föderierten Systems auf verschiedene Datenquellen, die sich auf verschiedenen, relationalen oder nicht relationalen Plattformen von IBM oder anderen Softwareanbietern befinden können. Das föderierte System integriert den Zugriff auf verteilte Daten und stellt für Benutzer ein einzelnes Datenbankimage einer heterogenen Umgebung dar.

Die folgenden Elemente listen Informationen dazu auf, wie und wie oft Anwendungen, die in einem föderierten System ausgeführt werden, auf eine Datenquelle zugreifen. Sie listen ebenfalls Informationen dazu auf, wie eine gegebene Anwendung, die auf einer Instanz eines Servers mit föderierten Datenbanken ausgeführt wird, auf eine Datenquelle zugreift. Zu diesen Elementen gehören Folgende:

- `datasource_name` - Monitorelement für den Datenquellennamen
- `disconnects` - Monitorelement für Verbindungstrennungen
- `insert_sql_stmts` - Monitorelement für Einfügungen
- `update_sql_stmts` - Monitorelement für Aktualisierungen

- delete_sql_stmts - Monitorelement für Löschvorgänge
- dynamic_sql_stmts - Monitorelement für versuchte dynamische SQL-Anweisungen
- create_nickname - Monitorelement für das Erstellen von Kurznamen
- passthru - Monitorelement für Durchgriffe
- stored_procs - Monitorelement für gespeicherte Prozeduren
- remote_locks - Monitorelement für ferne Sperren
- sp_rows_selected - Monitorelement für von gespeicherten Prozeduren zurückgegebene Zeilen
- select_time - für die Antwortzeit bei Abfragen
- insert_time - Monitorelement für die Antwortzeit bei Einfügungen
- update_time - Monitorelement für die Antwortzeit bei Aktualisierungen
- delete_time - Monitorelement für die Antwortzeit bei Löschvorgängen
- create_nickname_time - Monitorelement für die Antwortzeit beim Erstellen von Kurznamen
- passthru_time - Monitorelement für die Durchgriffszeit
- stored_proc_time - Monitorelement für die Zeit für gespeicherte Prozeduren
- remote_lock_time - Monitorelement für die Zeit für ferne Sperren

Im folgenden Beispiel ist die Momentaufnahme für dynamic_sql_statement dargestellt:

```
Anweisungstext = [ORACLE817]SELECT A0.C1,A0.C2 FROM ORA_T A0 WHERE A0.C3 = :H0
```

Der Anweisungstext beginnt bei allen fernen Anweisungen mit dem Namen der fernen Datenquelle in eckigen Klammern, gefolgt von dem eigentlichen Text, der an die ferne Datenquelle gesendet wird.

Kapitel 16. Interaktion zwischen Clientanwendungen und Datenquellen

Für Clientanwendungen werden die Datenquellen in einem föderierten System als eine einzige, kollektive Datenbank dargestellt. Um Daten von Datenquellen abzurufen, übergeben Anwendungen Abfragen in DB2-SQL an die föderierte Datenbank. Die föderierte Datenbank verteilt die Abfragen dann an die entsprechenden Datenquellen und gibt diese Daten entweder an die Anwendungen zurück oder führt die angeforderte Aktion aus.

Die föderierte Datenbank kann Daten aus lokalen Tabellen und fernen Datenquellen in derselben SQL-Anweisung verknüpfen. Sie können beispielsweise Daten aus einer lokalen DB2-Tabelle, einer Informix-Tabelle und einer Sybase-Sicht in einer einzigen SQL-Anweisung verknüpfen. Das föderierte System verarbeitet SQL-Anweisungen so, als ob die Datenquellen gewöhnliche relationale Tabellen oder Sichten innerhalb der föderierten Datenbank wären, und kann hierdurch relationale Daten und nicht relationale Daten verknüpfen.

In einem föderierten System können Sie über Kurznamen auf Datenquellen zugreifen. Bei einem *Kurznamen* handelt es sich um ein Objekt einer föderierten Datenbank, das von einer Anwendung verwendet wird, um auf ein Datenquellenobjekt, z. B. eine Tabelle oder eine Sicht, zu verweisen. Zum Schreiben in eine Datenquelle — beispielsweise zur Aktualisierung einer Datenquellentabelle — können Anwendungen DB2-SQL (mit Kurznamen) verwenden. Alternativ können Anwendungen die SQL-Programmversion der Datenquelle (ohne Kurznamen) in einer speziellen Sitzung, einer so genannten *Durchgriffssitzung* verwenden, um direkt auf die Datenquellen zuzugreifen.

Anwendungen, die DB2-SQL und Kurznamen verwenden, können auf sämtliche Datentypen zugreifen, die von der föderierten Datenbank erkannt werden.

Der Katalog der föderierten Datenbank enthält Informationen über Objekte in der föderierten Datenbank und über Objekte in den Datenquellen. Da der Katalog Informationen über die gesamte föderierte Datenbank enthält, wird er als *globaler Katalog* bezeichnet.

Kapitel 17. Verweis auf Datenquellenobjekte in SQL-Anweisungen durch Kurznamen

Bei einem föderierten System werden die für Datenquellenobjekte definierten Kurznamen verwendet, um die Objekte in den SQL-Anweisungen darzustellen. Das föderierte System erkennt vollständig qualifizierte Datenquellen-, Schema- und Objektnamen in SQL-Anweisungen nicht.

Datenquellenobjekte müssen in der föderierten Datenbank registrierte Kurznamen aufweisen, damit Sie sie in Ihre Abfragen aufnehmen können. Im Allgemeinen können Sie Kurznamen in einer SQL-Anweisung dort angeben, wo lokale Tabellen in einer SQL-Anweisung angegeben werden können.

Beispiel: Verwenden von Kurznamen in den Anweisungen SELECT, INSERT, UPDATE und DELETE

Sie definieren den Kurznamen NFXDEPT, um eine Tabelle in einer Informix-Tabelle mit dem Namen PERSON.DEPT darzustellen. Dabei gilt Folgendes:

- PERSON ist das Datenquellschema
- DEPT ist der Datenquellentabellenname

Die Anweisung `SELECT * FROM NFXDEPT` vom Server mit föderierten Datenbanken ist zulässig. Die Anweisung `SELECT * FROM PERSON.DEPT` ist jedoch nicht zulässig (außer in einer Durchgriffssitzung). Im Server mit föderierten Datenbanken ist `PERSON.DEPT` nicht als Kurzname registriert.

Beispiel: Verwenden von Kurznamen in der Anweisung CREATE TABLE

Sie möchten eine lokale Tabelle auf der Basis einer fernen Tabelle erstellen, für die Sie einen Kurznamen definiert haben. Ein Beispiel der Anweisung `CREATE TABLE` lautet:

```
CREATE TABLE tabellenname LIKE kurzname
```

Kurznamen in DDL-Anweisungen

Datenquellenobjekte müssen in der föderierten Datenbank registrierte Kurznamen aufweisen, damit Sie sie in Ihre DDL-Anweisungen aufnehmen können. Dieser Abschnitt enthält einige Beispiele von DDL-Anweisungen, die mit föderierten Systemen verwendet werden.

Verwenden von Kurznamen in der Anweisung COMMENT ON

Mit der Anweisung `COMMENT ON` werden im globalen Katalog der föderierten Datenbank Kommentare hinzugefügt oder ersetzt. Die Anweisung `COMMENT ON` ist gültig mit einem Kurznamen und Spalten, die in einem Kurznamen definiert sind. Diese Anweisung aktualisiert keine Datenquellenkataloge.

Verwenden von Kurznamen in den Anweisungen GRANT und REVOKE

Die Anweisungen `GRANT` und `REVOKE` sind gültig mit einem Kurznamen für bestimmte Berechtigungen und für alle Benutzer und Gruppen. Das System föde-

rierter Datenbanken setzt jedoch keine entsprechende GRANT- oder REVOKE-Anweisung für das Objekt in der Datenquelle ab, auf das der Kurzname verweist.

Nehmen Sie zum Beispiel an, dass Benutzer JON einen Kurznamen für eine Oracle-Tabelle erstellt, die über keinen Index verfügt. Der Kurzname lautet ORAREM1. Zu einem späteren Zeitpunkt definiert der Oracle-DBA einen Index für diese Tabelle. Benutzer EILEEN möchte nun, dass die föderierte Datenbank über das Vorhandensein dieses Index informiert wird, damit das Abfrageoptimierungsprogramm Strategien für einen effizienteren Zugriff auf die Tabelle entwickeln kann. EILEEN kann die föderierte Datenbank darüber informieren, dass ein neuer Index vorhanden ist, indem Sie eine Indexspezifikation für ORAREM1 erstellt.

Die Informationen über den Index werden in der Katalogsicht SYSSTAT.INDEXES gespeichert. Verwenden Sie die Anweisung GRANT, um EILEEN das Indexzugriffsrecht für diesen Kurznamen zu erteilen, damit sie die Indexspezifikation erstellen kann.

```
GRANT INDEX ON NICKNAME ORAREM1 TO USER EILEEN
```

Wenn Sie die Berechtigungen von Benutzer EILEEN zum Erstellen einer Indexspezifikation für Kurzname ORAREM1 widerrufen möchten, verwenden Sie hierzu die Anweisung REVOKE:

```
REVOKE INDEX ON ORAREM1 FROM USER EILEEN
```

Statistikanwendungen für Datenquellen

Wenn für ein Datenquellenobjekt ein Kurzname erstellt wird, wird der globale Katalog der föderierten Datenbank mit Informationen über dieses Objekt aktualisiert. Das Abfrageoptimierungsprogramm verwendet diese Informationen, um das Abrufen von Daten von dem Objekt zu planen.

Sie müssen sich unbedingt davon vergewissern, dass die Datenquelleninformationen aktuell sind. Die föderierte Datenbank erkennt Änderungen an Datenquellenobjekten nicht automatisch.

Im globalen Katalog gespeicherte Datenbankobjektstatistik

Die im globalen Katalog über ein Datenquellenobjekt gespeicherten Informationen sind vom Typ des Objekts abhängig. Für Datenbanktabellen und -sichten werden der Name des Objekts, die Spaltennamen und Attribute im globalen Katalog gespeichert.

Im Fall einer Tabelle oder eines Kurznamens enthalten die Informationen darüber hinaus Folgendes:

- Statistik. Beispielsweise die Anzahl der Zeilen und die Anzahl der Seiten, auf denen sich die Zeilen befinden. Um sicherzustellen, dass die föderierte Datenbank die neueste Statistik erhält, führen Sie das Äquivalent des Befehls RUNSTATS der Datenquelle bei der Tabelle aus, bevor Sie den Kurznamen erstellen.
- Indexbeschreibungen. Falls die Tabelle über keine Indizes verfügt, können Sie dem Katalog Metadaten liefern, die normalerweise in einer Indexdefinition enthalten sind. Nehmen Sie zum Beispiel an, dass ein Kurzname für eine ferne Tabelle erstellt wird, und dass anschließend ein Index für die Tabelle in der Datenquelle erstellt wird. Sie können eine Indexspezifikation auf dem Server mit föderierten Datenbanken erstellen, die diesen fernen Index darstellt. Eine Indexspezifikation wird durch Ausgeben der Anweisung CREATE INDEX und Verweisen auf den Kurznamen für die Tabelle erstellt. Verwenden Sie die Klausel

SPECIFICATION ONLY mit der Anweisung CREATE INDEX, um nur eine Indexspezifikation zu erstellen. Die Indexspezifikation informiert das föderierte Optimierungsprogramm darüber, dass ein ferner Index vorhanden ist. Es werden jedoch nur Metadaten generiert. Auf dem Server mit föderierten Datenbanken wird kein Index erstellt. Außerdem werden dem globalen Katalog keine statistischen Informationen bereitgestellt. Wenn Sie der Indexspezifikation exakt dieselbe Signatur (d. h. denselben Namen und dieselben Spalten in derselben Reihenfolge) wie dem fernen Index geben, können Sie SYSPROC.NNSTAT verwenden, um Statistikdaten zum Kurznamen und der Indexspezifikation zu aktualisieren.

Um festzustellen, welche Datenquelleninformationen im globalen Katalog gespeichert sind, führen Sie Abfragen bei den Katalogsichten SYSCAT.TABLES und SYSCAT.COLUMNS aus. Um zu erfahren, welche Datenquellenindexinformationen im Katalog gespeichert sind oder welche Daten eine bestimmte Indexspezifikation enthält, führen Sie eine Abfrage bei der Katalogsicht SYSCAT.INDEXES aus.

Aktualisieren von Statistik unter Verwendung der Sicht SYSSTAT anstatt der Sicht SYSCAT

Bei SYSCAT-Sichten handelt es sich um schreibgeschützte Katalogsichten im SYSCAT-Schema. SYSSTAT-Sichten hingegen sind aktualisierbare Katalogsichten, die statistische Informationen enthalten, die vom Optimierungsprogramm verwendet werden. SYSSTAT-Sichten befinden sich im SYSSTAT-Schema.

Wenn Sie eine UPDATE- oder INSERT-Operation für eine Sicht im Schema SYSCAT ausführen, schlägt diese Operation fehl. Verwenden Sie die aktualisierbaren Katalogsichten im SYSSTAT-Schema, um Statistik zu Kurznamen manuell zu ändern.

Definieren von Spaltenoptionen für Kurznamen

Spaltenoptionen sind Parameter in den Anweisungen CREATE NICKNAME und ALTER NICKNAME. Spaltenoptionen können beim ersten Erstellen eines Kurznamens oder durch Ändern eines bestehenden Kurznamens angegeben werden.

Die Informationen, die über die Spaltenoptionen bereitgestellt werden, werden im globalen Katalog gespeichert.

Nicht relationale Datenquellen

Spaltenoptionen sind für jeden nicht relationalen Wrapper eindeutig. Diese Optionen werden normalerweise beim Absetzen der Anweisung CREATE NICKNAME festgelegt.

Relationale Datenquellen

Es stehen zwei Spaltenoptionen zur Verfügung, die für relationale Datenquellen verwendet werden können: NUMERIC_STRING und VARCHAR_NO_TRAILING_BLANKS.

Angeben der Spaltenoption NUMERIC_STRING

Wenn eine Zeichenfolgespalte einer Datenquelle nur numerische Ziffern und keine anderen Zeichen oder Leerzeichen enthält, setzen Sie die Spaltenoption NUMERIC_STRING auf Y.

Durch das Setzen der Spaltenoption `NUMERIC_STRING` auf `Y` können Abfragen, die diese Spalte verwenden, in Bezug auf das Sortieren von Operationen und Vergleichsoperationen optimiert werden. Beispiel:

```
ALTER NICKNAME nickname
  ALTER COLUMN local_column_name
  OPTIONS (SET NUMERIC_STRING 'Y')
```

Angeben der Spaltenoption `VARCHAR_NO_TRAILING_BLANKS`

Wenn die Zeichenfolgespalte der Datenquelle keine abschließenden Leerzeichen enthält, setzen Sie die Spaltenoption `VARCHAR_NO_TRAILING_BLANKS` auf `Y`.

Einige Datenquellen, z. B. Oracle, verwenden nicht dieselbe Zeichenfolgevergleichslogik mit Leerzeichen als Füllzeichen wie die föderierte Datenbank. Dies gilt für Datentypen wie `VARCHAR` und `VARCHAR2`. Daher müssen Vergleichselemente, die diese Datentypen einbeziehen, vom Abfrageoptimierungsprogramm neu geschrieben werden, um konsistente Abfrageergebnisse zu gewährleisten.

Das erneute Schreiben von Abfrageanweisungen kann sich jedoch auf die Leistung auswirken. Durch das Angeben dieser Option für eine bestimmte Spalte werden dem Abfrageoptimierungsprogramm Informationen über diese Spalte bereitgestellt und die Generierung effizienterer SQL-Anweisungen ermöglicht.

Beispiel:

```
ALTER NICKNAME nickname
  ALTER COLUMN local_column_name
  OPTIONS (SET VARCHAR_NO_TRAILING_BLANKS 'Y')
```

Kapitel 18. Erstellen und Verwenden von föderierten Sichten

Eine Sicht, die eine Referenz auf einen Kurznamen im Fullselect enthält, ist eine *föderierte Sicht*. Auf die Basistabellen wird in der föderierten Sicht mithilfe von Kurznamen anstelle von Datenquellentabellennamen verwiesen.

Einschränkungen

Föderierte Sichten, die aus mehreren Datenquellenobjekten erstellt werden, sind schreibgeschützte Sichten und können nicht aktualisiert werden.

Föderierte Sichten, die nur aus einem Datenquellenobjekt erstellt werden, können schreibgeschützte Sichten oder Sichten ohne Schreibschutz sein.

- Eine föderierte Sicht, die aus einer einzelnen, nicht relationalen Datenquelle erstellt wird, ist schreibgeschützt.
- Eine föderierte Sicht, die aus einer einzelnen, relationalen Datenquelle erstellt wird, kann abhängig von den Angaben in der Anweisung CREATE VIEW Aktualisierungen zulassen.

Informationen zu dieser Task

Die Vorteile der Verwendung von föderierten Sichten ähneln den Vorteilen der Verwendung von Sichten, die für lokale Tabellen in einem zentralen Manager für relationale Datenbanken definiert sind:

- Sichten liefern eine integrierte Darstellung der Daten
- Sie können Tabellenspalten, die vertrauliche oder schutzwürdige Daten enthalten, aus einer Sicht ausschließen

Vorgehensweise

Sie erstellen eine föderierte Sicht aus Datenquellenobjekten, die über Kurznamen verfügen. Die Aktion des Erstellens einer Sicht einer föderierten Datenbank aus Datenquellendaten wird manchmal als „Erstellen einer Sicht zu einem Kurznamen“ bezeichnet. Diese Umschreibung spiegelt die Tatsache wider, dass zum Erstellen der föderierten Sicht die Anweisung CREATE VIEW (Fullselect) auf den Kurznamen jeder Datenquellentabelle und -sicht verweisen muss, die in der föderierten Sicht enthalten sein soll.

Erstellen von föderierten Sichten - Beispiele

Dieser Abschnitt enthält Beispiele für das Erstellen von föderierten Sichten.

Beispiel: Erstellen einer föderierten Sicht, die ähnliche Daten von verschiedenen Datenquellenobjekten zusammenführt

Sie arbeiten mit Kundendaten auf drei verschiedenen Servern, einem in Europa, einem in Asien und einem in Südamerika. Die Kundendaten von Europa befinden sich in einer Oracle-Tabelle. Der Kurzname für diese Tabelle lautet ORA_EU_CUST. Die Kundendaten von Asien befinden sich in einer Sybase-Tabelle. Der Kurzname für diese Tabelle lautet SYB_AS_CUST. Die Kundendaten von Südamerika befinden sich in einer Informix-Tabelle. Der Kurzname für diese Tabelle lautet INFMX_SA_CUST. Jede Tabelle verfügt jeweils über eine Spalte für die Kundennummer (CUS-

T_NO), für den Namen des Kunden (CUST_NAME), für die Produktnummer (PROD_NO) und für die Bestellmenge (QUANTITY). Die Syntax zum Erstellen einer Sicht aus diesen drei Kurznamen, in der diese Kundendaten zusammengeführt werden, lautet folgendermaßen:

```
CREATE VIEW FV1
  AS SELECT * FROM ORA_EU_CUST
  UNION
  SELECT * FROM SYB_AS_CUST
  UNION
  SELECT * FROM INFMX_SA_CUST
```

Beispiel: Verknüpfen von Daten zum Erstellen einer föderierten Sicht

Sie arbeiten mit den Kundendaten auf einem Server und mit den Vertriebsdaten auf einem anderen Server. Die Kundendaten befinden sich in einer Oracle-Tabelle. Der Kurzname für diese Tabelle lautet ORA_EU_CUST. Die Vertriebsdaten befinden sich in einer Sybase-Tabelle. Der Kurzname für diese Tabelle lautet SYB_SALES. Sie möchten die Kundeninformationen mit den von diesen Kunden getätigten Einkäufen abgleichen. Jede Tabelle enthält eine Spalte mit der Kundennummer (CUST_NO). Die Syntax zum Erstellen einer föderierten Sicht aus diesen beiden Kurznamen, in der diese Daten verknüpft werden, lautet folgendermaßen:

```
CREATE VIEW FV4
  AS SELECT A.CUST_NO, A.CUST_NAME, B.PROD_NO, B.QUANTITY
  FROM ORA_EU_CUST A, SYB_SALES B
  WHERE A.CUST_NO=B.CUST_NO
```

Kapitel 19. Pflegen von Datenintegrität mit Isolationsstufen

Die Isolationsstufe definiert den Grad der Isolation eines Anwendungsprozesses von anderen Anwendungsprozessen, die gleichzeitig ausgeführt werden.

Sie können die Datenintegrität für eine Datenquellentabelle mithilfe der Anforderung pflegen, dass die Tabellenzeilen bei einer bestimmten Isolationsstufe gesperrt werden.

Das Sperren erfolgt in der Basistabellenzeile in der Datenquelle. Der Datenbankmanager kann jedoch mehrere Zeilensperren durch eine einzige Tabellensperre ersetzen. Diese Maßnahme wird als *Sperreneskalation* bezeichnet. Einem Anwendungsprozess wird mindestens die angeforderte minimale Sperrstufe garantiert.

Die Isolationsstufen für die föderierte Datenbank lauten folgendermaßen:

- RR** Repeatable read (Wiederholtes Lesen)
- RS** Read stability (Lesestabilität)
- CS** Cursor stability (Cursorstabilität - Standard)
- UR** Uncommitted read (Nicht festgeschriebenes Lesen)

Die Isolationstypen sind dabei die Isolation auf Anweisungsebene und die Isolation auf Verbindungsebene.

Die Isolation kann beim Ausführen der folgenden Aktionen angegeben werden:

- Vorkompilieren oder Binden einer Anwendung. Sie können Isolationsstufen angeben, wenn Sie eine Anwendung vorbereiten oder binden. Die in den Befehlen BIND und PREP angegebene Isolationsstufe ist die Standardisolationsstufe, wenn der Server mit föderierten Datenbanken eine Verbindung zur fernen Datenquelle herstellt.
- Verwenden der Klausel WITH in einer SQL-Anweisung. Diese Aktion wird als Isolation auf Anweisungsebene bezeichnet. Sie können die Klausel WITH in den Anweisungen SELECT, UPDATE, INSERT und DELETE verwenden.

Wenn der Server mit föderierten Datenbanken für eine Anweisung keine Isolationsstufe findet, verwendet er die Isolationsstufe, die beim Herstellen der Verbindung zur Datenquelle eingerichtet wurde.

In der folgenden Tabelle sind die Datenquellen aufgelistet, die Isolation auf Verbindungsebene verwenden. Darüber hinaus sind die von den einzelnen Datenquellen verwendeten Isolationsstufen sowie die äquivalenten Isolationsstufen auf dem Server mit föderierten Datenbanken angegeben.

Tabelle 21. Datenquellen und Isolationsstufen

Datenquellen	Am stärksten restriktive Isolationsstufe	Stärker restriktive Isolationsstufe	Weniger restriktive Isolationsstufe	Am wenigsten restriktive Isolationsstufe
Föderierte Datenbank	Wiederholtes Lesen	Lesestabilität	Cursorstabilität	Nicht festgeschriebenes Lesen

Tabelle 21. Datenquellen und Isolationsstufen (Forts.)

Datenquellen	Am stärksten restriktive Isolationsstufe	Stärker restriktive Isolationsstufe	Weniger restriktive Isolationsstufe	Am wenigsten restriktive Isolationsstufe
DB2-Produktfamilie	Wiederholtes Lesen	Lesestabilität*	Cursorstabilität	Nicht festgeschriebenes Lesen
Informix	Wiederholtes Lesen	Wiederholtes Lesen	Cursorstabilität	Fehlerhaftes Lesen
JDBC	Serialisierbar	Wiederholtes Lesen	Lesen mit COMMIT	Lesen ohne COMMIT
Microsoft SQL Server	Serialisierbar	Wiederholtes Lesen	Lesen mit COMMIT	Lesen ohne COMMIT
ODBC	Serialisierbar	Wiederholtes Lesen	Lesen mit COMMIT	Lesen ohne COMMIT
Oracle	Serialisierbar	Serialisierbar	Lesen mit COMMIT	Lesen mit COMMIT
Sybase	Stufe 3	Stufe 3	Stufe 1	Stufe 0

*Für DB2 für VM- und VSE Server-Datenquellen ist die Isolationsstufe 'Wiederholtes Lesen'.

Das Sonderregister CURRENT ISOLATION wird vom Server mit föderierten Datenbanken nicht verwendet, wenn er eine Verbindung zu einer Datenquelle herstellt.

Die nicht relationalen Datenquellen weisen kein Konzept wie das der Isolationsstufen auf. OLE DB und Teradata verfügen zwar über das Konzept von Isolationsstufen, diese werden aber vom Server mit föderierten Datenbanken nicht unterstützt. Es besteht keine Isolationsstufenzuordnung zwischen den Isolationsstufen der föderierten Datenbank und den OLE DB-, Teradata- und nicht relationalen Datenquellen.

Isolation auf Anweisungsebene in einem föderierten System

Bei föderierten Datenquellen müssen Sie die Isolationsklausel WITH verwenden, um die Isolation einer Anweisung anzugeben.

Sie müssen die Isolationsklausel WITH in der Anweisung verwenden, wenn Sie die Isolation auf Anweisungsebene einsetzen möchten. Wenn Sie mit der Schnittstelle auf Aufrufebene (Call Level Interface, CLI) oder einer anderen API für die Isolation auf Anweisungsebene Attribute verwenden, hat dies keinen Einfluss auf die Anweisungsisolierung.

Zu den Datenquellen, die die Isolation auf Anweisungsebene in einem föderierten System unterstützen, gehören die Produktfamilie von DB2 und Microsoft SQL Server. Die Anweisungsisolierung für die DB2-Produktfamilie und SQL Server wird an ferne Datenquellen gesendet.

Verwenden Sie die Serveroption DB2_STATEMENT_ISOLATION, um die Isolation auf Anweisungsebene ein- bzw. auszuschalten. Sie können diese Option in der Anweisung CREATE SERVER oder ALTER SERVER angeben. Die Serveroption ist automatisch auf Y gesetzt.

Sie können die Isolationsklausel WITH in den folgenden Anweisungen verwenden:

SELECT
SELECT INTO
Searched DELETE
INSERT
Searched UPDATE
DECLARE CURSOR

Sperrenanforderungsklausel

Sie können die Sperrenanforderungsklausel in der Anweisung SELECT oder SELECT INTO verwenden. Zu den föderierten Datenquellen, die die Sperrenanforderungsklausel unterstützen, gehören DB2 für Linux, UNIX und Windows sowie die DB2 für z/OS.

Einschränkungen bei Verwendung der WITH-Klausel zum Angeben der Isolationsstufe

Für Isolationsstufen, die für Anweisungen angegeben werden, gelten die folgenden Bedingungen:

- Die WITH-Klausel darf nicht in Unterabfragen verwendet werden.
- Die Isolationsstufe UR findet nur Anwendung, wenn die Ergebnistabelle der Anweisung fullselect oder SELECT INTO schreibgeschützt ist. In anderen Fällen wird die Isolationsstufe UR für die Anweisung bei der DB2-Datenquellenfamilie von UR in CS geändert. Für die SQL Server-Datenquelle wird ein Upgrade der Isolationsstufe UR in Lesen mit COMMIT (Read committed) durchgeführt.
- Wenn Sie für die folgenden Datenquellen die Sperrenanforderungsklausel angeben, wird die Klausel vom Server mit föderierten Datenbanken ignoriert.
 - DB2 für System i
 - DB2 für VM
 - Microsoft SQL Server

Isolation auf Verbindungsebene in einem föderierten System

Der Server mit föderierten Datenbanken ordnet Ihre Isolationsstufe einer entsprechenden Isolationsstufe in der Datenquelle zu.

Der Wrapper legt für jede Verbindung zur Datenquelle die Isolationsstufe fest.

Wenn der Server mit föderierten Datenbanken eine Verbindung zur Datenquelle herstellt, wird die Isolationsstufe in der fernen Datenquelle auf eine Stufe gesetzt, die äquivalent zur Stufe auf dem Server mit föderierten Datenbanken ist. Wenn kein exaktes Äquivalent vorhanden ist, setzt der Server mit föderierten Datenbanken die Isolationsstufe auf die nächste restriktive Stufe. Nachdem eine Verbindung zu einer Datenquelle hergestellt ist, kann die Isolationsstufe für die Dauer der Verbindung nicht mehr geändert werden.

Alle Wrapper mit Ausnahme von Teradata protokollieren die Isolationsstufe der Verbindung. Beim Einrichten einer Verbindung setzen die Wrapper die Verbindungsisolationsstufe auf das Äquivalent der aktuellen Isolationsstufe. Die aktuelle Isolationsstufe ist die Isolationsstufe des aktuellen Abschnitts (erste föderierte Anweisung für eine Datenquelle). Der Teradata-Wrapper befindet sich stets in der Standardisolationsstufe READ, da er über keine Möglichkeit zum Ändern der Isolationsstufe verfügt.

Kapitel 20. Föderierte LOB-Unterstützung

Mit einem System föderierter Datenbanken können Sie auf große Objekte (Large Objects, LOBs) in fernen Datenquellen zugreifen und diese bearbeiten.

Ein föderiertes System unterstützt SELECT-Operationen bei LOBs in DRDA-, Informix-, Microsoft SQL Server-, Oracle- und Sybase-Datenquellen. Beispiel:

```
SELECT empname, picture FROM infmx_emp_table  
WHERE empno = '01192345'
```

Hierbei steht *picture* für eine LOB-Spalte und *infmx_emp_table* für einen Kurznamen, der auf eine Informix-Tabelle mit Mitarbeiterdaten verweist.

Ein föderiertes System unterstützt SELECT-, INSERT-, UPDATE- und DELETE-Operationen bei LOBs in den folgenden Datenquellen, wobei der DRDA-Wrapper verwendet wird:

- DB2 für z/OS (Version 7 oder höher)
- DB2 für System i (Version 5)
- DB2 Database für Linux, UNIX und Windows (Version 7 oder höher)

Die Lese- und Schreiboperationen, die von DB2 Database für Linux, UNIX und Windows unterstützt werden, sind in der folgenden Tabelle aufgeführt:

Tabelle 22. Lese- und Schreibunterstützung für LOBs

Datenquelle	Typ der Operation
DB2 für z/OS, DB2 für System i, DB2 Database für Linux, UNIX und Windows ¹	Lesen und Schreiben
BioRS	Nur Lesen
Informix	Nur Lesen
JDBC	Nur Lesen
Microsoft SQL Server	Nur Lesen
Oracle (NET8-Wrapper) ²	Lesen und Schreiben
ODBC	Nur Lesen
Sybase	Nur Lesen
Teradata	Nur Lesen
Web-Services	Nur Lesen und extern Binden (nur für CLOB)
XML	Nur Lesen

Anmerkung:

1. DB2 für System i (Version 5 oder höher) ist für die LOB-Unterstützung erforderlich. DB2 Information Integrator Version 8 kann nicht auf LOB-Daten von DB2 Database für Linux, UNIX und Windows Version 7 zugreifen.
2. Zum Ausführen von Einfügings-, Aktualisierungs- oder Löschoptionen bei Oracle-Spalten für große Objekte (LONG) müssen Sie die fernen Spalten von LONG-Spalten in LOBs migrieren und die Kurznamen erneut erzeugen.

Teradata-LOBs

Teradata-LOBs unterscheiden sich geringfügig von DB2-LOBs. Teradata

weist keine Datentypen auf, die so groß wie die unter DB2 für Linux, UNIX und Windows unterstützten LOBs sind. Manche Teradata-Datentypen können allerdings bis zu 64000 Byte lang sein. Diese Datentypen sind CHAR, VARCHAR, BYTE, VARBYTE, GRAPHIC und VARGRAPHIC. Diese Teradata-Datentypen werden DB2-LOB-Datentypen zugeordnet, wenn die Länge des jeweiligen Teradata-Datentyps den Grenzwert des entsprechenden DB2-Datentyps überschreitet.

LOB-Längen

Einige Datenquellen, z. B. Oracle und Informix, speichern die Längen von LOB-Spalten in ihren Systemkatalogen nicht. Wenn Sie einen Kurznamen für eine Tabelle erstellen, werden Informationen vom Systemkatalog der Datenquelle einschließlich der Spaltenlänge abgerufen. Da für die LOB-Spalten keine Länge vorhanden ist, geht die föderierte Datenbank von der maximalen Länge einer LOB-Spalte unter DB2 Database für Linux, UNIX und Windows aus. Die föderierte Datenbank speichert die maximale Länge im Katalog der föderierten Datenbank als Länge der Kurznamenspalte.

LOB-Querverweise

Für große Objekte (LOBs), die in fernen Datenquellen gespeichert sind, erfordern einige Anwendungen LOB-Querverweise. Ein LOB-Querverweis ist ein 4-Byte-Wert, der in einer Hostvariablen gespeichert ist. Eine Anwendung kann den LOB-Querverweis dazu verwenden, auf einen im Datenbanksystem gespeicherten LOB-Wert (oder LOB-Ausdruck) zu verweisen.

Mithilfe eines LOB-Querverweises kann eine Anwendung den LOB-Wert so bearbeiten, als wäre er in einer normalen Hostvariablen gespeichert. Wenn Sie LOB-Querverweise verwenden, müssen Sie den LOB-Wert nicht vom Datenquellen-server zur Anwendung (und möglicherweise wieder zurück) transportieren.

Die föderierte Datenbank kann LOBs von fernen Datenquellen abrufen, sie auf dem Server mit föderierten Datenbanken speichern und anschließend einen LOB-Querverweis auf das gespeicherte LOB absetzen. LOB-Querverweise werden in folgenden Fällen abgesetzt:

- Anwendungen setzen FREE LOCATOR SQL-Anweisungen ab
- Anwendungen setzen COMMIT-Anweisungen ab
- Die föderierte Instanz wird erneut gestartet

Einschränkungen bei großen Objekten (LOBs)

Föderierte Systeme geben einige Einschränkungen für LOBs vor.

Für LOBs gelten die folgenden Einschränkungen:

- Die föderierte Datenbank kann ferne LOBs nicht an eine Dateireferenzvariable binden
- LOBs werden in Durchgriffssitzungen nicht unterstützt
- LOBs werden als Parameter von gespeicherten Prozeduren nicht unterstützt

Leistungsaspekte für die LOB-Verarbeitung

Bei der Entwicklung föderierter Anwendungen, die LOB-Daten (Large Object) abrufen und verarbeiten, müssen Anwendungsentwickler und Datenbank-administratoren wissen, welchen Einfluss die LOB-Verarbeitung auf die Leistung hat.

Wenn eine Anwendung Daten von einer föderierten Datenquelle abrufen muss, muss der Server mit föderierten Datenbanken die Daten in seine eigenen Anwendungspuffer stellen, bevor er sie an die Anwendung sendet. Da LOBs in einem Pufferpool nicht verarbeitet werden, müssen die LOB-Daten zuerst einen Tabellenbereich für temporäre Tabellen passieren, der für den Server mit föderierten Datenbanken definiert ist. Um die Leistung verbessern zu können und die Ressourcenauslastung zu reduzieren, sollten Anwendungsentwickler LOB-Daten daher nur speichern, wenn dies wirklich nötig ist.

Ähnliches gilt, wenn der Server mit föderierten Datenbanken ferne LOB-Daten aktualisiert. Die Daten müssen einen dem Server mit föderierten Datenbanken zugeordneten Tabellenbereich für temporäre Tabellen passieren, bevor sie an die Datenquelle weitergegeben werden.

Temporäre LOBs verwenden den Tabellenbereich für temporäre Tabellen, der dem Server mit föderierten Datenbanken zugeordnet ist. Aus diesem Grund müssen Datenbankadministratoren diesen Tabellenbereich für temporäre Tabellen möglicherweise vergrößern, um sicherzustellen, dass der Arbeitsbereich für die Verarbeitung der LOBs ausreicht.

Empfehlung: Definieren Sie den Tabellenbereich für temporäre Tabellen als systemverwalteten Bereich (System Managed, SMS) und stellen Sie sicher, dass der Tabellenbereich für temporäre Tabellen sich auf Datenträgern mit einer hohen E/A-Bandbreite befindet. Auf diese Weise können Sie die Leistung beim Arbeiten mit LOBs maximieren.

Verwenden der DB2 Call Level Interface für den Zugriff auf föderierte LOBs

Der Server mit föderierten Datenbanken unterstützt zwei DB2 CLI-APIs zum Auswählen von LOB-Daten:

- Die API SQLFetch ruft in einer einzigen Operation das LOB vom Server mit föderierten Datenbanken oder der Datenquelle ab und stellt es in die Anwendungspuffer.
- Die API SQLGetData ruft das LOB blockweise ab und erfordert möglicherweise wiederholte Aufrufe der API, um das gesamte LOB in die Anwendungspuffer zu stellen.

Empfehlung: Verwenden Sie beim Abrufen von LOBs über einen Server mit föderierten Datenbanken die API SQLGetData. Auf diese Weise erzielen Sie optimale Leistung.

Der Server mit föderierten Datenbanken unterstützt die APIs SQLExecute und SQLPutData zum Aktualisieren von LOB-Daten. Die API SQLExecute aktualisiert die LOB-Daten in einer einzigen Operation, während die API SQLPutData möglicherweise wiederholte Aufrufe erfordert, um die gesamten LOB-Daten von den Anwendungspuffern an den Server zu senden. Alle APIs werden in einer föderierten Umgebung auf derselben Ebene ausgeführt.

Gesicherte und abgeschirmte Wrapper

Kurznamen, die für Wrapper erstellt wurden, die als gesichert oder abgeschirmt definiert sind, weisen beim Abrufen oder Aktualisieren von LOBs ähnliche Leistungen auf.

Kapitel 21. Verteilte Anforderungen für das Abrufen von Datenquellen

Abfragen, die an die föderierte Datenbank übergeben werden, können zwar Ergebnisse aus einer einzelnen Datenquelle anfordern, normalerweise handelt es sich jedoch um Anforderungen, die mehrere Datenquellen umfassen. Da eine typische Abfrage an mehrere Datenquellen verteilt wird, wird sie als *verteilte Anforderung* bezeichnet.

Im Allgemeinen verwendet eine verteilte Anforderung mindestens eine von drei SQL-Konventionen um anzugeben, wo Daten von Unterabfragen, Gruppenoperatoren und Join-Subselects abzurufen sind.

Verteilte Anforderungen für das Abrufen von Datenquellen - Beispiele

Die Beispiele in diesem Thema veranschaulichen verteilte Anforderungen mit einer Unterabfrage, Gruppenoperatoren und einer Joinoperation.

In den folgenden Beispielen ist der Server mit föderierten Datenbanken für den Zugriff auf eine DB2 für z/OS-Datenquelle, eine DB2 für System i-Datenquelle und eine Oracle-Datenquelle konfiguriert. In jeder Datenquelle ist eine Tabelle gespeichert, die Mitarbeiterdaten enthält. Der Server mit föderierten Datenbanken weist auf diese Tabellen durch Kurznamen, die auf den Speicherort der Tabellen zeigen.

zOS_EMPLOYEES

Kurzname für eine Tabelle in einer DB2 für z/OS-Datenquelle, die Mitarbeiterdaten enthält.

SYSTEMi_EMPLOYEES

Kurzname für eine Tabelle in einer DB2 für System i-Datenquelle, die Mitarbeiterdaten enthält.

ORA_EMPLOYEES

Kurzname für eine Tabelle in einer Oracle-Datenquelle, die Mitarbeiterdaten enthält.

ORA_REGIONS

Kurzname für eine Tabelle in einer Oracle-Datenquelle, die Informationen über die Regionen enthält, in denen die Mitarbeiter wohnen.

Die folgenden Beispiele veranschaulichen die drei bei verteilten Anforderungen verwendeten SQL-Konventionen. Dabei werden die für die einzelnen Tabellen definierten Kurznamen verwendet.

Beispiel: Eine verteilte Anforderung mit einer Unterabfrage

SYSTEMi_EMPLOYEES enthält die Telefonnummern von Mitarbeitern, die in Asien wohnen. Darüber hinaus enthält sie die diesen Telefonnummern zugeordneten Regionscodes. Die Regionen, für die die einzelnen Codes stehen, sind allerdings nicht aufgelistet. ORA_REGIONS enthält sowohl die Codes als auch die Regionen. Bei der folgenden Abfrage wird eine Unterabfrage verwendet, um den Regionscode für China zu ermitteln. Anschließend wird der Regionscode verwendet, um eine Liste der Mitarbeiter in SYSTEMi_EMPLOYEES zurückzugeben, die über eine Telefonnummer in China verfügen.

```
SELECT name, telephone FROM db2admin.SYSTEMi_employees
WHERE region_code IN
(SELECT region_code FROM dbadmin.ora_regions
WHERE region_name = 'CHINA')
```

Beispiel: Eine verteilte Anforderung mit Gruppenoperatoren

Der Server mit föderierten Datenbanken unterstützt drei Gruppenoperatoren: UNION, EXCEPT und INTERSECT.

- Der Gruppenoperator UNION wird verwendet, um die Zeilen zu kombinieren, die mindestens eine von zwei oder mehr SELECT-Anweisungen erfüllen.
- Der Gruppenoperator EXCEPT wird verwendet, um die Zeilen abzurufen, die die erste SELECT-Anweisung, nicht aber die zweite erfüllen.
- Der Gruppenoperator INTERSECT wird verwendet, um die Zeilen abzurufen, die beide SELECT-Anweisungen erfüllen.

Bei allen drei Gruppenoperatoren kann der Operand ALL verwendet werden, um anzugeben, dass doppelte Zeilen nicht aus dem Ergebnis gelöscht werden sollen. Dies erübrigt einen zusätzlichen Sortiervorgang.

Bei der folgenden Abfrage werden alle Mitarbeiternamen und Regionscodes abgerufen, die sowohl in SYSTEMi_EMPLOYEES als auch in zOS_EMPLOYEES enthalten sind, obwohl sich jede Tabelle in einer anderen Datenquelle befindet.

```
SELECT name, region_code
FROM as400_employees
INTERSECT
SELECT name, region_code
FROM zOS_employees
```

Beispiel: Eine verteilte Anforderung einer Verknüpfung

Eine relationale Verknüpfung erzeugt eine Ergebnismenge mit einer Kombination von Spalten, die von zwei oder mehr Tabellen abgerufen wurden. Es empfiehlt sich, Bedingungen anzugeben, die die Größe der Zeilen in der Ergebnismenge begrenzen.

Bei der nachfolgenden Abfrage werden Mitarbeiternamen und ihre entsprechenden Regionsnamen kombiniert, indem die in den beiden Tabellen enthaltenen Regionscodes verglichen werden. Jede Tabelle befindet sich in einer anderen Datenquelle.

```
SELECT t1.name, t2.region_name
FROM dbadmin.SYSTEMi_employees t1, dbadmin.ora_regions t2
WHERE t1.region_code = t2.region_code
```

Optimieren von verteilten Anforderungen mit Serveroptionen

In einem föderierten System können Sie Parameter, so genannte *Serveroptionen* verwenden, um dem globalen Katalog Informationen bereitzustellen, die für eine Datenquelle als Ganzes gelten, oder um die Interaktion der föderierten Datenbank mit einer Datenquelle zu steuern.

Informationen zu dieser Task

Serveroptionen beschreiben die Funktionalität einer bestimmten Datenquelle und erweitern die Informationen, die der Server mit föderierten Datenbanken von dieser Datenquelle besitzt. Sie können beispielsweise die folgenden Serveroptionen verwenden:

- Die Serveroption `VARCHAR_NO_TRAILING_BLANKS` informiert das Optimierungsprogramm darüber, dass keine `VARCHAR`-Spalte auf dem Datenquellenserver abschließende Leerzeichen enthält. Verwenden Sie diese Option nur, wenn Sie sicher sind, dass alle `VARCHAR2`-Spalten für jedes Objekt, auf das durch einen Kurznamen auf dem Server verwiesen wird, keine abschließenden Leerzeichen enthalten. Verwenden Sie andernfalls eine Spaltenoption, um die Spalten für einzelne Objekte auf dem Server anzugeben, die keine abschließenden Leerzeichen enthalten. Die Spaltenoption lautet ebenfalls `VARCHAR_NO_TRAILING_BLANKS`.
- Die Serveroption `PLAN_HINTS` stellt Sybase-Datenquellen Anweisungsfragmente bereit, die als *Planhinweise* bezeichnet werden. Die Planhinweise unterstützen das Optimierungsprogramm der Datenquelle bei der Entscheidung, welcher Index für den Zugriff auf eine Tabelle verwendet werden soll und welche Tabellenjoin-sortierung beim Abrufen von Daten für eine Ergebnismenge zu verwenden ist.

Normalerweise legt der Datenbankadministrator Serveroptionen für ein föderiertes System fest. Programmierer können die Serveroptionen, die beim Optimieren von Abfragen unterstützen, jedoch ebenfalls sinnvoll nutzen. Beispiel: Für die Datenquellen SYB1 und SYB2 ist die Serveroption `PLAN_HINTS` auf den Standardwert N (nein, diese Datenquelle nicht mit Planhinweisen versehen) gesetzt. Sie schreiben eine verteilte Anforderung, die Daten von SYB1 und SYB2 auswählt. Sie erwarten, dass die Optimierungsprogramme in diesen Datenquellen die Planhinweise zum Verbessern ihrer Strategien für den Zugriff auf diese Daten nutzen können. Sie können den Standardwert mit der Einstellung Y (yes: ja, Planhinweise einfügen) überschreiben, während Ihre Anwendung mit der föderierten Datenbank verbunden ist. Wenn die Verbindung zu den Datenquellen beendet wird, wird die Einstellung automatisch auf N zurückgesetzt.

Vorgehensweise

Gehen Sie wie folgt vor, um Serveroptionen zu definieren:

Verwenden Sie die Anweisung `SET SERVER OPTION`, um Serveroptionen zu definieren oder zu ändern. Um sicherzustellen, dass die Einstellung wirksam wird, geben Sie die Anweisung `SET SERVER OPTION` unmittelbar nach der `CONNECT`-Anweisung an. Die Serveroption ist für die Dauer der Verbindung zur föderierten Datenbank festgelegt.

Ziehen Sie die Möglichkeit in Betracht, die Anweisung dynamisch vorzubereiten. Die Anweisung `SET SERVER OPTION` trifft nur auf dynamische SQL-Anweisungen zu.

Bei statischem SQL hat die Verwendung der Anweisung `SET SERVER OPTION` nur Auswirkungen auf die Ausführung der statischen SQL-Anweisung. Sie hat keine Auswirkungen auf die Pläne, die das Optimierungsprogramm generiert.

Abbrechen einer föderierten Abfrage

Sie können eine Anwendung unterbrechen und eine laufende Abfrage abbrechen. Diese Unterstützung ermöglicht es Ihnen, eine Abfrage der fernen Datenquellenanwendung vor deren Fertigstellung abzubrechen und die Abfrageunterbrechung sowie den Abfrageabbruch an die ferne Datenquelle weiterzugeben.

Einschränkungen

In Version 9.7 können Sie nur für Datenquellen von DB2 Database für Linux, UNIX und Windows eine Anwendung unterbrechen und eine laufende Abfrage abbrechen.

Sie können föderierte SELECT-Anweisungen, INSERT-, UPDATE- und DELETE-Anweisungen, föderierte gespeicherte Prozeduren und Anweisungen in Durchgriffssitzungen abbrechen.

Eine föderierte Anweisung kann mehrere Segmente enthalten, die auf mehrere Datenquellen zugreifen. Wenn ein Segment auf eine nicht unterstützte Datenquelle zugreift, z. B. bei mehreren Abfragesegmenten für eine Sybase-Datenquelle, kann die Anweisung nicht abgebrochen werden.

Eine föderierte Abfrage kann nicht abgebrochen werden, wenn ATQ (Asynchrony Table Queue, Asynchrony-Tabellenwarteschlange) aktiviert ist.

Informationen zu dieser Task

Wenn der Server mit föderierten Datenbanken eine föderierte SQL-Anweisung ausführt und blockiert ist, weil er auf die Ergebnisse und die Antwort von der fernen Datenquelle wartet, befindet sich die Anwendung auf dem Server mit föderierten Datenbanken im Status 'föderierte Anforderung anstehend'. Sie können den Befehl LIST APPLICATIONS absetzen, um die Anwendungen zu ermitteln, die sich im Status 'föderierte Anforderung anstehend' befinden.

Wenn sich eine Anwendung im Status 'föderierte Anforderung anstehend' befindet, können Sie mithilfe der Tastenkombination Strg+C die gewünschte Anwendung unterbrechen und abbrechen oder mithilfe des Befehls FORCE APPLICATION die gewünschte Anwendung auf dem fernen Server stoppen.

Vorgehensweise

Verwenden Sie zum Abbrechen einer föderierten Abfrage eine der folgenden Methoden:

- Drücken der Tastenkombination Strg+C zum Unterbrechen der Anwendung
Sie können mithilfe der Tastenkombination Strg+C die auf dem fernen Server ausgeführte Anwendung unterbrechen und die Abfrage abbrechen. Die aktuelle SQL-Anweisung auf dem Server mit föderierten Datenbanken wird unterbrochen und abgebrochen; die Konsistenz der lokalen Datenbank bleibt erhalten. Die ferne Anweisung wird ebenfalls abgebrochen. Sowohl die abgehende als auch die eingehende Verbindung bleibt aktiv.
- Absetzen des Befehls FORCE APPLICATION
Sie können den Befehl FORCE APPLICATION absetzen, um eine Anwendung zu stoppen. Die Anwendung wird auf dem Server mit föderierten Datenbanken gestoppt; die Transaktionskonsistenz bleibt sowohl lokal als auch auf dem fernen Server erhalten. Die eingehende und die abgehende Verbindung werden beendet, und der ferne Teil der Anwendungsausführung für die ferne Datenquelle wird abgebrochen.

Kapitel 22. Direktes Abfragen von Datenquellen im Durchgriffsmodus

Sie können Durchgriffssitzungen verwenden, um Operationen durchzuführen, die mit DB2 SQL/API nicht möglich sind.

Informationen zu dieser Task

Durchgriffssitzungen sind in den folgenden Fällen nützlich:

- Anwendungen müssen Objekte in der Datenquelle erstellen oder INSERT-, UPDATE- oder DELETE-Operationen ausführen
- Die föderierte Datenbank unterstützt keine eindeutige Datenquellenoperation

Vorgehensweise

Gehen Sie wie folgt vor, um Datenquellen direkt mit Durchgriff abzufragen:

- Verwenden Sie die Anweisung SET PASSTHRU, um eine Durchgriffssitzung zu starten und direkt auf einen Server zuzugreifen. Diese Anweisung kann dynamisch abgesetzt werden. Ein Beispiel für diese Anweisung lautet folgendermaßen: SET PASSTHRU ORACLE1 Mit der Anweisung SET PASSTHRU wird eine Durchgriffssitzung bei der Datenquelle unter Verwendung des Servernamens ORACLE1 geöffnet. ORACLE1 ist der Name, den Sie beim Erstellen der Serverdefinition für den Datenquellenserver registrierten.
- Wenn die Durchgriffssitzung geöffnet ist, müssen Sie sicherstellen, dass Sie beim Verweisen auf Objekte in einer solchen Sitzung den wahren Namen des Objekts (nicht den Kurznamen) verwenden. Sie müssen die SQL-Programmversion der Datenquelle verwenden, es sei denn, es handelt sich bei der referenzierten Datenquelle um die föderierte Datenbank.
- Wenn in einer Durchgriffssitzung eine statische Anweisung übergeben wird, wird sie zur Verarbeitung an den Server mit föderierten Datenbanken gesendet. Wenn Sie eine SQL-Anweisung zur Verarbeitung an eine Datenquelle senden möchten, müssen Sie sie in der Durchgriffssitzung dynamisch vorbereiten und sie bei geöffneter Sitzung ausführen lassen. Gehen Sie wie folgt vor, um Anweisungen in einer Durchgriffssitzung dynamisch vorzubereiten:
 - Verwenden Sie zum Übergeben einer SELECT-Anweisung auch die Anweisung PREPARE, und verwenden Sie anschließend die Anweisungen OPEN, FETCH und CLOSE, um auf die Ergebnisse Ihrer Abfrage zuzugreifen.
 - Bei einer anderen unterstützten Anweisung als SELECT stehen Ihnen zwei Möglichkeiten zur Verfügung: Sie können die Anweisung PREPARE verwenden, um die unterstützte Anweisung vorzubereiten, und anschließend die Anweisung EXECUTE zum Ausführen der vorbereiteten Anweisung verwenden. Alternativ können Sie die Anweisung EXECUTE IMMEDIATE verwenden, um die Anweisung vorzubereiten und auszuführen.

Wenn Sie während einer Durchgriffssitzung den Befehl COMMIT oder ROLLBACK absetzen, wird durch diesen Befehl die aktuelle Arbeitseinheit (Unit of Work, UOW) beendet. Die Durchgriffssitzung wird jedoch nicht beendet.

Überlegungen und Einschränkungen bei der Verwendung des Durchgriffsmodus in föderierten Systemen

In diesem Abschnitt werden die Überlegungen und Einschränkungen beschrieben, die Sie bei der Verwendung einer Durchgriffssitzung kennen müssen.

Die folgenden Überlegungen und Einschränkungen gelten für alle Datenquellen:

- Anweisungen, die in einer Durchgriffssitzung vorbereitet werden, müssen in derselben Sitzung ausgeführt werden. Anweisungen, die in einer Durchgriffssitzung vorbereitet, aber außerhalb derselben Sitzung ausgeführt werden, schlagen fehl und führen zum Fehler SQLSTATE 56098. Anweisungen, die außerhalb einer Durchgriffssitzung vorbereitet, aber in einer Durchgriffssitzung ausgeführt werden, werden als SET PASSTHRU-Anweisung behandelt.
- Eine Anwendung kann mehrere SET PASSTHRU-Anweisungen ausgeben, es ist jedoch nur die letzte Sitzung aktiv. Wenn eine neue Anweisung SET PASSTHRU aufgerufen wird, beendet sie die vorangegangene Anweisung SET PASSTHRU. Innerhalb einer Durchgriffssitzung können keine Durchgriffe auf mehrere Datenquellen durchgeführt werden.
- Wenn in einer Anwendung mehrere Durchgriffssitzungen verwendet werden, müssen Sie sicherstellen, dass Sie ein Commit durchführen, bevor Sie eine weitere Durchgriffssitzung öffnen. Dadurch wird die UOW für die aktuelle Sitzung abgeschlossen.
- Parametermarken werden in Durchgriffssitzungen nicht unterstützt. Verwenden Sie Hostvariablen anstelle von Parametermarken.
- Sie können die WITH HOLD-Semantik für einen Cursor verwenden, der in einer Durchgriffssitzung definiert wurde. Es wird jedoch möglicherweise ein Fehler ausgegeben, wenn Sie versuchen, die WITH HOLD-Semantik nach einem Commit zu verwenden, die Datenquelle die WITH HOLD-Semantik aber nicht unterstützt.
- Hostvariablen, die in SQL-Anweisungen in einer Durchgriffssitzung definiert werden, müssen die Form Hn haben, wobei H in Großbuchstaben vorliegen und n eine eindeutige ganze Zahl sein muss. Die Werte für n müssen von Null ausgehend fortlaufend nummeriert werden.
- Der Durchgriff unterstützt keine LOBs.
- Der Durchgriff unterstützt keine Aufrufe gespeicherter Prozeduren.
- Der Durchgriff unterstützt die Anweisung SELECT INTO nicht.
- Der Durchgriff unterstützt keine SQL oder externe benutzerdefinierte Funktionen.
- Während einer Durchgriffssitzung können keine dynamischen SQL COMMIT- oder ROLLBACK-Anweisungen ausgeführt werden.
- Wenn während einer Durchgriffssitzung Aktualisierungs- oder Löschoptionen ausgeführt werden, können Sie die Bedingung WHERE CURRENT OF CURSOR nicht verwenden.
- Im Durchgriffsmodus werden dynamische SQL-Anweisungen fern ausgeführt, während statische SQL-Anweisungen zur Verarbeitung an den Server mit föderierten Datenbanken gesendet werden. Wenn Sie eine SQL-Anweisung zur Verarbeitung an eine Datenquelle übergeben möchten, müssen Sie sie in der Durchgriffssitzung dynamisch vorbereiten, und sie muss ausgeführt werden, solange die Sitzung noch offen ist.
- Statische SET PASSTHRU-Anweisungen in gespeicherten Prozeduren mit SQL-Hauptteil werden blockiert, wenn gespeicherte Prozeduren erstellt werden.

Dabei wird Fehler SQL0104N ausgegeben. Um in den Durchgriffsmodus und aus diesem zu gelangen, verwenden Sie die Anweisung EXECUTE IMMEDIATE.

Beispiel:

```
create procedure stp()
dynamic result sets 1
language sql
modifies sql data
begin
declare stmt varchar(100);

DECLARE cur1 CURSOR WITH RETURN TO CALLER
FOR SELECT * FROM t1 ;

set stmt = 'set passthru mvs7';
execute immediate stmt;

set stmt = 'insert into t1 values (20, 'passthru_insert')';
execute immediate stmt;
commit;

insert into t1 values (20, 'stp_insert');
commit;

OPEN cur1;

end
DB20000I Der Befehl SQL wurde erfolgreich ausgeführt.

call stp()
```

```
Ergebnismenge 1
-----
10 local
20 stp_insert

2 Satz/Sätze ausgewählt.
Rückgabestatus = 0
```

```
select * from t1

C1 C2
-----
10 remote
20 passthru_insert

2 Satz/Sätze ausgewählt.
```

```
set passthru reset
DB20000I Der Befehl SQL wurde erfolgreich ausgeführt.
```

```
select * from t1

C1 C2
-----
10 local
20 stp_insert

2 Satz/Sätze ausgewählt.
```

- Die Rückkehr aus einer gespeicherten Prozedur oder Compound-SQL-Anweisung beendet nicht automatisch den Durchgriffsmodus. Um den Durchgriffsmodus zu beenden, muss die Anweisung SET PASSTHRU RESET explizit aufgerufen werden, wie im oben stehenden Beispiel dargestellt.

Durchgriffssitzungen bei Oracle-Datenquellen

In diesem Abschnitt werden einige SQL-Überlegungen beschrieben, die Ihnen bekannt sein sollten, bevor Sie SQL-Anweisungen in einer Durchgriffssitzung an Oracle-Datenbanken übergeben.

- Jede auf einem Oracle-Server abgesetzte DDL-Anweisung wird während der Syntaxanalyse ausgeführt und unterliegt keiner Transaktionssemantik. Die Operation wird nach Beendigung von Oracle automatisch festgeschrieben. Wenn ein Rollback durchgeführt wird, wird die DDL nicht rückgängig gemacht.
- Wenn Sie eine SELECT-Anweisung aus Rohdatentypen absetzen, verwenden Sie die Funktion RAWTOHEX, um die Hexadezimalwerte zu empfangen. Wenn Sie ein INSERT in Rohdatentypen ausführen, geben Sie die hexadezimale Darstellung an.

Kapitel 23. Optimieren der Abfrageverarbeitung

Sie können ein föderiertes System optimieren, um die Leistung der Abfrageverarbeitung zu verbessern.

Informationen zu dieser Task

Wenn Sie SQL-Abfragen an die föderierte Datenbank übergeben, verarbeitet der SQL-Compiler die Abfrage, und das Abfrageoptimierungsprogramm analysiert sie und erstellt einen Zugriffsplan. Das Abfrageoptimierungsprogramm speichert die Zugriffsplaninformationen in den EXPLAIN-Tabellen in der föderierten Datenbank. Sie können die Tools db2exfmt, db2expln und dynexpln verwenden, um den Zugriffsplan für eine bestimmte SQL-Anweisung zu interpretieren.

Vorgehensweise

Gehen Sie wie folgt vor, um föderierte Systeme für die Verbesserung der Abfrageverarbeitung zu optimieren:

1. Übergeben Sie die zu optimierenden SQL-Abfragen an die föderierte Datenbank.
2. Analysieren Sie, wo die Abfrage ausgewertet wird.
3. Ermitteln Sie die Gründe für die im Zugriffsplan getroffenen Entscheidungen, und modifizieren Sie Ihr System so, dass mehr Möglichkeiten zur Durchführung von Pushdown-Operationen zur Verfügung stehen.

Veröffentlichungen zur Leistung föderierter Datenbanken

Eine Vielzahl von IBM Dokumenten mit detaillierten Informationen zur Leistungsoptimierung steht Ihnen zur Verfügung.

- Asynchronous execution of federated queries in WebSphere Federation Server V9.1 unter der Webadresse <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0611norwood/>
- Maximize the performance of WebSphere Information Integrator with Materialized Query Tables unter der Webadresse <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0605lin/index.html>
- Performance enhancements in IBM WebSphere Federation Server V9.1, Part 1: Improve performance of federated queries with new WebSphere Federation Server capabilities unter der Webadresse <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert/>
- Performance enhancements in IBM WebSphere Federation Server V9.1, Part 2: Performance characteristics of new functionality in WebSphere Federation Server unter der Webadresse <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert2/>
- Using data federation technology in IBM WebSphere Information Integrator: Data federation usage examples and performance tuning unter der Webadresse <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0507lin/>
- Parallelism in WebSphere Information Integrator V8.2 unter der Webadresse <http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0502harris/>

- Data Federation with IBM DB2 Information Integrator V8.1 unter der Webadresse <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247052.html?Open>
- "Using the federated database technology of IBM DB2 Information Integrator" unter der Webadresse <ftp://ftp.software.ibm.com/software/data/pubs/papers/iifed.pdf>.

Abfrageanalyse

Ein wichtiger Teil der Abfrageverarbeitung besteht in der Analyse, die bestimmt, wie die Abfrage für die bestmögliche Leistung optimiert wird.

Zum Abrufen von Daten aus Datenquellen übergeben Clients (Benutzer und Anwendungen) Abfragen in SQL an die föderierte Datenbank. Zum Verarbeiten der Abfrage zieht der SQL-Compiler Informationen aus dem globalen Katalog und dem Wrapper der Datenquelle hinzu. Dabei handelt es sich um Informationen zur Verbindung zur Datenquelle, zu Serverattributen, Zuordnungen, Indexinformationen und Kurznamenstatistiken.

Als Teil der Verarbeitung des SQL-Compilers wird die Abfrage vom *Abfrageoptimierungsprogramm* analysiert. Der Compiler entwickelt für die Verarbeitung der Abfrage alternative Strategien, die als *Zugriffspläne* bezeichnet werden. Die Zugriffspläne legen für die Abfrage unter Umständen Folgendes fest:

- Die Verarbeitung der Abfrage soll durch die Datenquellen erfolgen.
- Die Verarbeitung der Abfrage soll durch den Server mit föderierten Datenbanken erfolgen.
- Die Verarbeitung der Abfrage soll teilweise durch die Datenquellen und teilweise durch den Server mit föderierten Datenbanken erfolgen.

Die föderierte Datenbank wertet die Zugriffspläne in erster Linie auf der Grundlage von Informationen über das Leistungsspektrum der Datenquelle und die Attribute der Daten aus. Diese Informationen sind im Wrapper und im globalen Katalog enthalten. Die föderierte Datenbank zerlegt die Abfrage in Segmente, die als *Abfragefragmente* bezeichnet werden. Normalerweise ist es effizienter, ein Abfragefragment an eine Datenquelle im Pushdown-Modus weiterzuleiten, wenn die Datenquelle das Fragment verarbeiten kann. Das Abfrageoptimierungsprogramm berücksichtigt jedoch andere Faktoren, zu denen folgende gehören:

- Die zu verarbeitende Datenmenge
- Die Verarbeitungsgeschwindigkeit der Datenquelle
- Die vom Fragment zurückgegebene Datenmenge
- Die Bandbreite der Übertragung

Die Pushdown-Analyse wird nur für relationale Datenquellen ausgeführt. Nicht relationale Datenquellen verwenden das RRC-Protokoll (RRC = Request-Reply-Compensate).

In der folgenden Abbildung werden die vom SQL-Compiler ausgeführten Schritte bei der Verarbeitung einer Abfrage dargestellt.

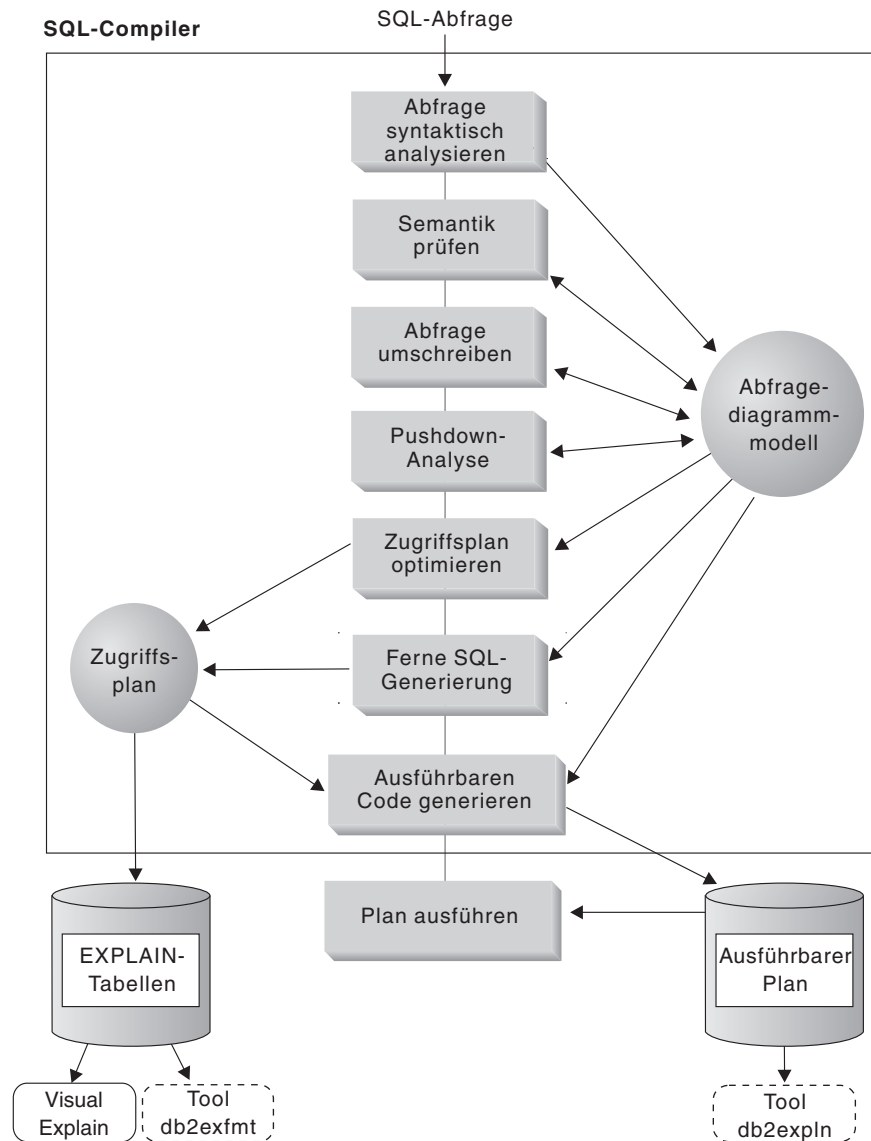


Abbildung 9. Flussdiagramm einer Abfrageanalyse des SQL-Compilers

Das Abfrageoptimierungsprogramm generiert auf der Grundlage des Ressourcenaufwands lokale und ferne Zugriffspläne für die Verarbeitung eines Abfragefragments. Die föderierte Datenbank wählt dann den Plan mit dem wahrscheinlich niedrigsten Ressourcenaufwand für die Verarbeitung der Abfrage aus.

Sollen Fragmente durch Datenquellen verarbeitet werden, übergibt der Server mit föderierten Datenbanken diese Fragmente an die Datenquellen. Nachdem die Datenquellen die Fragmente verarbeitet haben, werden die Ergebnisse abgerufen und an den Server mit föderierten Datenbanken zurückgegeben. Hat die föderierte Datenbank einen Teil der Verarbeitung durchgeführt, kombiniert sie ihre Ergebnisse mit den aus der Datenquelle abgerufenen Ergebnissen. Anschließend gibt der Server mit föderierten Datenbanken die Ergebnisse an den Client zurück.

Die Hauptaufgabe der Pushdown-Analyse ist es, festzustellen, welche Operationen fern ausgewertet werden können. Die Pushdown-Analyse ermittelt dies anhand der empfangenen SQL-Anweisung sowie anhand der ihr vorliegenden Informationen über die Funktionalität und die Semantik der fernen Datenquelle. Auf der

Basis dieser Analyse wertet das Abfrageoptimierungsprogramm die Alternativen aus und wählt unter Berücksichtigung des anfallenden Aufwands einen Zugriffsplan aus. Das Optimierungsprogramm führt unter Umständen eine Verarbeitung aus Gründen der Aufwandseffizienz nicht direkt in der fernen Datenquelle aus. Eine sekundäre Aufgabe ist der Versuch, die Abfrage neu zu schreiben, um so die Unterschiede hinsichtlich der Semantik und der SQL-Operationen zwischen dem Server mit föderierten Datenbanken und der Datenquelle zu kompensieren und die Abfrage zu optimieren.

Der vom Optimierungsprogramm ausgewählte endgültige Zugriffsplan kann Operationen enthalten, die in den fernen Datenquellen ausgewertet wurden. Für die über Remotezugriff ausgeführten Operationen erstellt der SQL-Compiler effizientes SQL, das während der Generierungsphase in der SQL-Variante der fernen Datenquelle formuliert wird. Der Prozess der Erstellung eines optimalen Abfrageplans, der alle Quellen berücksichtigt, wird als *globale Optimierung* bezeichnet.

Für nicht relationale Quellen verwenden die Wrapper das RRC-Protokoll.

Pushdown-Analyse

Sie gibt dem Abfrageoptimierungsprogramm an, ob eine ferne Datenquelle eine Operation ausführen kann. Eine Operation kann eine Funktion sein, wie z. B. relationale Bediener-, System- oder Benutzerfunktionen, oder ein SQL-Operator (GROUP BY, ORDER BY, usw.). Das Optimierungsprogramm entscheidet dann kostenbasiert, ob der Operator im Pushdown-Modus weitergeleitet werden soll. Selbst wenn die Pushdown-Analyse feststellt, dass eine bestimmte Operation in der fernen Quelle verarbeitet werden kann, entscheidet das Optimierungsprogramm möglicherweise dennoch, die Operation lokal auf dem föderierten Server auszuführen, wenn hierfür voraussichtlich weniger Ressourcen benötigt werden.

Die Pushdown-Analyse wird für relationale Datenquellen ausgeführt. Nicht relationale Quellen verwenden das RRC-Protokoll.

Funktionen, die nicht im Pushdown-Modus weitergeleitet werden können, wirken sich unter Umständen erheblich auf die Abfrageleistung aus. Beachten Sie die Auswirkungen, wenn ein selektives Vergleichselement erzwungen lokal ausgewertet wird anstatt in der fernen Datenquelle. Bei dieser Methode ist es möglicherweise erforderlich, dass der Server mit föderierten Datenbanken die gesamte Tabelle von der fernen Datenquelle abrufen und die Tabelle dann lokal mit dem Vergleichselement filtern muss. Bei einem begrenzten Netzwerk und einer großen Tabelle kann die Abfrageleistung beeinträchtigt werden.

Nicht im Pushdown-Modus weitergeleitete Operatoren können sich ebenfalls erheblich auf die Abfrageleistung auswirken. Das lokale Zusammenfassen von fernen Daten durch einen Operator GROUP BY beispielsweise könnte ebenso erfordern, dass der Server mit föderierten Datenbanken die gesamte Tabelle von der fernen Datenquelle abrufen muss.

Beispiel: Der Kurzname EMP verweist auf die Tabelle EMPLOYEE. Diese Tabelle hat 10.000 Zeilen. Eine Spalte enthält die Zipcodes (Postleitzahlen von Städten in den USA), eine andere die Gehälter der einzelnen Mitarbeiter. Die folgende Abfrage wird an den Server mit föderierten Datenbanken gesendet, um die Anzahl der Mitarbeiter pro Stadt zu berechnen, deren Gehalt höher als 50.000 ist und die in einem bestimmten Zipcode-Bereich wohnen:


```
SELECT CITY, COUNT(*) FROM EMP
WHERE ZIP BETWEEN 'CA1' AND 'CA5' AND SALARY > 50000
GROUP BY CITY;
```

Empfängt der SQL-Compiler diese Anweisung, überprüft er mehrere Möglichkeiten:

- Die Sortierfolgen der Datenquelle und des Servers mit föderierten Datenbanken sind identisch. Wahrscheinlich werden beide Vergleichselemente im Pushdown-Modus weitergeleitet, da sie voraussichtlich die Größe der von der Datenquelle zum Server mit föderierten Datenbanken gesendeten Zwischenergebnismenge reduzieren. Normalerweise ist es effizienter, Ergebnisse in der Datenquelle zu filtern und zu gruppieren anstatt die gesamte Tabelle auf den Server mit föderierten Datenbanken zu kopieren und die Operationen lokal auszuführen. Die Pushdown-Analyse ermittelt, ob Operationen in der Datenquelle ausgeführt werden können. Da die Sortierfolgen gleich sind, können die Operationen für die Vergleichselemente und für GROUP BY in der Datenquelle ausgeführt werden.
- Die Sortierfolgen sind gleich, und das Abfrageoptimierungsprogramm kalkuliert die hohe Verarbeitungsgeschwindigkeit des Servers mit föderierten Datenbanken mit ein. Möglicherweise ermittelt das Abfrageoptimierungsprogramm die lokale Ausführung der Operation GROUP BY als die beste Methode (d. h. als die Methode mit dem geringsten anfallenden Aufwand). Die Vergleichselemente werden zur Auswertung im Pushdown-Modus an die Datenquelle weitergeleitet. Hierbei handelt es sich um ein Beispiel für eine Kombination aus Pushdown-Analyse und globaler Optimierung.
- Die Sortierfolgen sind nicht gleich. Wahrscheinlich wird das Vergleichselement SALARY im Pushdown-Modus an die Datenquelle weitergeleitet, da numerische Spalten unabhängig von der jeweiligen Sortierfolge auf dieselbe Weise sortiert werden. Das Vergleichselement für ZIP wird jedoch nicht im Pushdown-Modus weitergeleitet, da seine Sortierung von einer Zeichendatenspalte abhängt. GROUP BY wird nur dann im Pushdown-Modus weitergeleitet, wenn sowohl das Vergleichselement für ZIP als auch für SALARY ebenfalls im Pushdown-Modus weitergeleitet werden.

Der SQL-Compiler überprüft die verfügbaren Zugriffspläne und wählt anschließend den effizientesten aus.

Grundsätzlich soll sichergestellt werden, dass das Abfrageoptimierungsprogramm das Weiterleiten von Funktionen und Operatoren zur Auswertung an die Datenquellen in Betracht zieht. Viele Faktoren bestimmen, ob eine Funktion oder ein SQL-Operator in einer fernen Datenquelle ausgewertet wird. Das Abfrageoptimierungsprogramm wird von folgenden Hauptfaktoren beeinflusst: Merkmale des Servers, Merkmale des Kurznamens und Merkmale der Abfrage.

Servermerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten

Zu den Servermerkmalen, die sich auf die Pushdown-Verarbeitung auswirken, gehören die SQL-Unterstützung, die Sortierfolge, die Optionen des Servers mit föderierten Datenbanken sowie die Typ- und Funktionszuordnungen.

Die Pushdown-Möglichkeiten werden bei nicht relationalen Datenquellen von anderen Faktoren beeinflusst als bei relationalen Datenquellen. Die SQL-Programmversion spielt für die meisten nicht relationalen Datenquellen beispielsweise keine Rolle, da sie kein SQL verwenden.

In den folgenden Themen werden datenquellenspezifische Faktoren beschrieben, die Pushdown-Möglichkeiten beeinflussen können.

SQL-Unterschiede

Zu den SQL-Merkmalen, die Auswirkungen auf die Weiterleitung im Pushdown-Modus haben, gehören das SQL-Leistungsspektrum, Einschränkungen und Begrenzungen sowie das für einen Server spezifische SQL.

- Leistungsspektrum von SQL. Jede Datenquelle unterstützt eine Variante der SQL-Programmversion und verschiedene Funktionalitätsebenen. Ein Beispiel ist die Liste GROUP BY. Die meisten Datenquellen unterstützen den Operator GROUP BY. Bei manchen Datenquellen bestehen jedoch Einschränkungen für die Anzahl der Elemente oder für die Zulassung eines Ausdrucks auf der Liste GROUP BY. Besteht in der fernen Datenquelle eine Einschränkung, führt der Server mit föderierten Datenbanken die Operation GROUP BY möglicherweise lokal aus.
- Einschränkungen für SQL. Datenquellen können unterschiedliche Einschränkungen für SQL haben. Einige Datenquellen erfordern z. B. Parametermarken für die Einbindung von Werten für ferne SQL-Anweisungen. Einschränkungen für Parametermarken müssen daher überprüft werden, um sicherzustellen, dass alle Datenquellen einen solchen Bindemechanismus unterstützen. Kann der Server mit föderierten Datenbanken keine passende Methode für die Einbindung eines Wertes in eine Funktion ermitteln, muss diese Funktion lokal ausgewertet werden.
- Begrenzungen für SQL. Der Server mit föderierten Datenbanken lässt möglicherweise die Verwendung von größeren ganzen Zahlen zu als die fernen Datenquellen. Werte, die die Begrenzung überschreiten, können jedoch nicht in Anweisungen eingebettet werden, die an die Datenquellen gesendet werden. Die Funktion oder der Operator, der mit dieser Konstante operiert, muss daher lokal ausgewertet werden.
- Serverspezifikationen. Zu dieser Kategorie gehören mehrere Faktoren. Ein Beispiel ist die Sortierung von NULL-Werten (je nach Sortierfolge höchster oder niedrigster Wert). Wird ein NULL-Wert beispielsweise in der Datenquelle nach anderen Kriterien sortiert als auf dem Server mit föderierten Datenbanken, können Operationen ORDER BY für einen Ausdruck mit Nullwerten nicht fern ausgewertet werden.

Datentyp VARCHAR2 in föderierten Systemen

Zur Optimierung der Abfrageverarbeitung muss Ihr Zugriffsplan das Auffüllen mit Leerzeichen bei Daten des Typs VARCHAR2 berücksichtigen.

Mit der Serveroption VARCHAR2_COMPAT aktivieren Sie die Unterstützung für Datenquellen, die mit VARCHAR2 kompatibel sind.

Die Semantik für die VARCHAR2-Kompatibilität bestimmt, wie Ihr Server mit föderierten Datenbanken und die Datenquelle den Datentyp VARCHAR2 verarbeiten. Dies ist davon abhängig, ob der Server mit föderierten Datenbanken und/oder die Datenquelle mit VARCHAR2 kompatibel ist. Für die Systeme sind die folgenden Konfigurationen möglich:

- Der Server mit föderierten Datenbanken ist mit VARCHAR2 kompatibel, stellt jedoch eine Verbindung zu einer Datenquelle her, die nicht mit VARCHAR2 kompatibel ist.
- Der Server mit föderierten Datenbanken ist nicht mit VARCHAR2 kompatibel, stellt jedoch eine Verbindung zu einer Datenquelle her, die mit VARCHAR2 kompatibel ist.

- Sowohl der Server mit föderierten Datenbanken als auch die Datenquelle ist mit VARCHAR2 kompatibel.

Wenn Ihr Server mit föderierten Datenbanken oder die DB2-Datenquelle mit VARCHAR2 kompatibel ist, wird der Datentyp VARCHAR2 als Synonym für den Datentyp VARCHAR behandelt.

Tipp: Um ein verbessertes Leistungsverhalten zu gewährleisten, sollte bei Systemen, die ausschließlich Verbindungen zu mit VARCHAR2 kompatiblen Datenquellen herstellen, der Server mit föderierten Datenbanken ebenfalls mit VARCHAR2 kompatibel sein.

In den folgenden Szenarios verarbeitet der Server mit föderierten Datenbanken leere Zeichenfolgenwerte und leitet die Operationen basierend auf der Semantik für die VARCHAR2-Kompatibilität im Pushdown-Modus an die Datenquelle weiter.

Szenario 1

Der Server mit föderierten Datenbanken ist nicht mit VARCHAR2 kompatibel, stellt jedoch eine Verbindung zu einer Datenquelle her, die mit VARCHAR2 kompatibel ist. Der Server mit föderierten Datenbanken lässt leere Zeichenfolgen zu, die Datenquelle hingegen nicht. Daher werden alle leeren Zeichenfolgen, die im Pushdown-Modus an die Datenquelle weitergeleitet werden, in NULL-Werte konvertiert.

Der Server mit föderierten Datenbanken verwendet standardmäßig die Semantik für das Auffüllen mit Leerzeichen beim Datentyp CHAR, bevor das Ergebnis im Pushdown-Modus an die Datenquelle weitergeleitet wird. Daher verwendet der Server mit föderierten Datenbanken das Auffüllen mit Leerzeichen nicht bei Operationen, die ausschließlich leere Zeichenfolgen angeben.

Beispiele

- Die folgenden Anweisungen INSERT und UPDATE enthalten nur leere Zeichenfolgen, die nicht mit Leerzeichen aufgefüllt sind. Die Anweisungen leiten leere Zeichenfolgenwerte im Pushdown-Modus weiter, die von der Datenquelle in NULL-Werte konvertiert werden.

Die folgenden Anweisungen werden vom Server mit föderierten Datenbanken beispielsweise nicht mit Leerzeichen aufgefüllt:

```
INSERT INTO n1 (c1) VALUES ('')
UPDATE n1 SET c1=''
INSERT INTO n1 (c1) VALUES (''), ('')
```

- Die folgende Anweisung INSERT sendet anstelle der leeren Zeichenfolge 10 Leerzeichen an die Datenquelle:

```
INSERT INTO n1 (col_char) VALUES (''), ('ibm')
```

Szenario 2

Der Server mit föderierten Datenbanken ist mit VARCHAR2 kompatibel, stellt jedoch eine Verbindung zu einer Datenquelle her, die nicht mit VARCHAR2 kompatibel ist. Die Semantik für die VARCHAR2-Kompatibilität des Servers mit föderierten Datenbanken lässt weder leere Zeichenfolgen zu, noch verwendet sie die Semantik für das Auffüllen mit Leerzeichen. Die Semantik der Datenquelle lässt hingegen leere Zeichenfolgen zu und verwendet die Semantik für das Auffüllen mit Leerzeichen. Daher leitet der Server mit föderierten Datenbanken Operationen,

die leere Zeichenfolgen einbeziehen, nur dann im Pushdown-Modus weiter, wenn die Semantik des Servers mit föderierten Datenbanken und die Datenkonsistenz erhalten bleiben.

Beispiele

- Die folgende Anweisung INSERT wird im Pushdown-Modus an die Datenquelle weitergeleitet, weil die leere Zeichenfolge in einen NULL-Wert konvertiert wird und lokal nicht verarbeitet werden kann:

```
INSERT INTO n1 (c1) VALUES ('')
```

- Die folgende Anweisung INSERT wird im Pushdown-Modus an die Datenquelle weitergeleitet, wenn Sie die Anweisung in zwei separate Anweisungen aufteilen:

```
INSERT INTO n1 SELECT * FROM n2
```

Sie müssen zwei Anweisungen verwenden, wenn die Ziel- oder die Quellentabelle aus einer Datenquelle stammt, die nicht mit VARCHAR2 kompatibel ist. Beispiel:

```
SELECT * FROM n2  
INSERT INTO n1 VALUES (:H0)
```

- In Ausdrücken verschachtelte Funktionen werden lokal im Server mit föderierten Datenbanken ausgeführt. Beispiel:

```
SELECT LENGTH(TRIM(c1)) FROM n1
```
- Alle Vergleichsoperationen werden im Server mit föderierten Datenbanken lokal ausgeführt, weil inkonsistente Ergebnisse möglich sind, wenn Vergleichsoperationen durch die Datenquelle ausgeführt werden.

Szenario 3

Sowohl der Server mit föderierten Datenbanken als auch die Datenquelle ist mit VARCHAR2 kompatibel. Weder der Server mit föderierten Datenbanken noch die Datenquelle lässt leere Zeichenfolgen zu oder verwendet die Semantik für das Auffüllen mit Leerzeichen. Daher werden Operationen, die leere Zeichenfolgen einbeziehen, die im Pushdown-Modus an die Datenquelle weitergeleitet, weil die Semantik erhalten bleibt.

Sortierfolge

Wenn Sie die Serveroption COLLATING_SEQUENCE auf 'Y' setzen, teilen Sie der föderierten Datenbank mit, dass die Sortierfolge der Datenquelle mit der Sortierfolge des Servers mit föderierten Datenbanken übereinstimmt. Durch diese Einstellung kann das Optimierungsprogramm in Betracht ziehen, sortierungsabhängige Verarbeitungsvorgänge im Pushdown-Modus an eine Datenquelle weiterzuleiten, wodurch eine bessere Leistung erzielt werden kann.

Wenn die Sortierfolge der Datenquelle von der Sortierfolge der föderierten Datenbank abweicht und Sie die Serveroption COLLATING_SEQUENCE auf 'Y' setzen, kann es zu fehlerhaften Ergebnissen kommen. Wenn der verwendete Plan beispielsweise Mischjoins enthält, leitet das Optimierungsprogramm Sortieroperationen möglicherweise im Pushdown-Modus an die Datenquellen weiter. Verfügt die Datenquelle nicht über dieselbe Sortierfolge, ist die Ergebnismenge der Joinergebnisse möglicherweise falsch. Setzen Sie die Serveroption COLLATING_SEQUENCE auf 'N', wenn Unklarheit darüber besteht, ob die Sortierfolge der Datenquelle mit der Sortierfolge der föderierten Datenbank übereinstimmt.

Sie können jedoch auch eine föderierte Datenbank so konfigurieren, dass sie die gleiche Sortierfolge verwendet wie die betreffende Datenquelle. Setzen Sie dann die Serveroption `COLLATING_SEQUENCE` auf 'Y'. Durch diese Einstellung kann das Optimierungsprogramm in Betracht ziehen, sortierungsabhängige Operationen für Zeichendaten spalten im Pushdown-Modus weiterzuleiten.

Um zu ermitteln, ob die Sortierfolge der Datenquelle und die Sortierfolge der föderierten Datenbank einander entsprechen, sind die folgenden Faktoren zu berücksichtigen:

- Unterstützung in der Landessprache
Die Sortierfolge steht in Zusammenhang mit der auf einem Server unterstützten Sprache. Vergleichen Sie die NLS-Informationen der föderierten Datenbank für Ihr Betriebssystem mit den entsprechenden NLS-Informationen, die für die Datenquelle gelten.
- Sprachensitive Sortierfolgen
Überprüfen Sie, ob die föderierte Datenbank oder die Datenquelle sprachensitive Sortierfolgen verwendet. Falls die Datenbank und die Datenquelle unterschiedliche Sortierfolgen verwenden, sollten Sie `COLLATING_SEQUENCE` nicht auf 'Y' setzen.
- Merkmale der Datenquelle
Einige Datenquellen werden mit Sortierfolgen erstellt, die von der Groß-/Kleinschreibung unabhängig sind. Dies kann bei Operationen, die von einer Reihenfolge abhängig sind, zu unterschiedlichen Ergebnissen aus der föderierten Datenbank führen.
- Anpassung
Einige Datenquellen stellen mehrere Optionen für Sortierfolgen zur Verfügung oder ermöglichen das Anpassen von Sortierfolgen.

Erfordert ein Abfragefragment von einem Server mit föderierten Datenbanken eine Sortierung, hängt der Ort für die Durchführung der Sortierung von mehreren Faktoren ab. Sind die Sortierfolgen in der föderierten Datenbank und in der Datenquelle gleich, kann die Sortierung in der Datenquelle ebenso ausgeführt werden wie auf dem Server mit föderierten Datenbanken. Das Abfrageoptimierungsprogramm kann ermitteln, ob die Abfrage am effizientesten durch eine lokale Sortieroperation oder eine ferne Sortieroperation verarbeitet wird.

Numerische Vergleiche können im Allgemeinen selbst bei unterschiedlicher Sortierfolge an beiden Orten durchgeführt werden. Es können jedoch fehlerhafte Ergebnisse auftreten, wenn die Gewichtung von Nullzeichen in der föderierten Datenbank anders ist als in der Datenquelle.

Bei Vergleichsanweisungen müssen Sie bei der Übergabe von Anweisungen an eine von der Groß-/Kleinschreibung unabhängige Datenquelle ebenfalls auf Fehlerquellen achten. Die den Zeichen 'I' und 'i' zugeordneten Wertigkeiten sind in einer Datenquelle, bei der die Groß-/Kleinschreibung nicht beachtet wird, gleich. Bei einer von der Groß-/Kleinschreibung unabhängigen Datenquelle mit einer englischen Codepage werden die Wörter **STEWART**, **StEWArT** und **stewart** als identisch interpretiert. Die föderierte Datenbank ist standardmäßig von der Groß-/Kleinschreibung abhängig und würde diesen Zeichen unterschiedliche Wertigkeiten zuordnen.

Sind die Sortierfolgen der föderierten Datenbank und der Datenquelle unterschiedlich, überträgt der Server die Daten auf die föderierte Datenbank, sodass die Sortierung lokal durchgeführt werden kann. Durch das lokale Sortieren der Daten

kommt der Server der Annahme der Benutzer entgegen, dass die Abfrageergebnisse nach der für den Server mit föderierten Datenbanken festgelegten Sortierfolge sortiert werden.

Enthält die Abfrage ein Gleichheitsvergleichselement für die Zeichenspalte, kann dieser Abschnitt der Abfrage im Pushdown-Modus weitergeleitet werden, auch wenn die Sortierfolgen nicht übereinstimmen (auf 'N' gesetzt). So würden beispielsweise mit dem Vergleichselement C1 = 'A' dieselben Werte abgerufen, unabhängig von der Sortierfolge; in diesem Fall wäre eine Weiterleitung im Pushdown-Modus an eine Datenquelle möglich, deren Sortierfolge von der des Servers mit föderierten Datenbank abweicht. Solche Vergleichselemente können jedoch nicht im Pushdown-Modus weitergeleitet werden, wenn die Sortierfolge in der Datenquelle die Groß-/Kleinschreibung nicht berücksichtigt (COLLATING_SEQUENCE='I'). Wenn eine Datenquelle von der Groß-/Kleinschreibung unabhängig ist, sind die Ergebnisse von C1= 'A' und C1 = 'a' identisch; dies ist jedoch nicht konsistent mit einer Umgebung, in der die Groß-/Kleinschreibung beachtet wird, wie beispielsweise DB2 Database für Linux, UNIX und Windows.

Administratoren können föderierte Datenbanken erstellen, deren Sortierfolge mit der Sortierfolge der Datenquelle übereinstimmt. Mit dieser Methode kann die Leistung beschleunigt werden, wenn alle Datenquellen dieselbe Sortierfolge verwenden oder die meisten bzw. alle Spaltenfunktionen für Datenquellen mit derselben Sortierfolge ausgeführt werden. Zum Beispiel werden in DB2 für z/OS Sortierungen, die durch ORDER BY-Klauseln definiert werden, mithilfe einer Sortierfolge implementiert, die auf einer EBCDIC-Codepage (Extended Binary-Coded Decimal Interchange Code) basiert. Soll der Server mit föderierten Datenbanken Daten von DB2 für z/OS nach ORDER BY-Klauseln sortiert abrufen, sollte die föderierte Datenbank für die Verwendung einer auf einer EBCDIC-Codepage basierenden vordefinierten Sortierfolge konfiguriert werden.

Sind die Sortierfolgen in der föderierten Datenbank und in der Datenquelle unterschiedlich und sollen die Daten in der Sortierfolge der Datenquelle sortiert werden, kann die Abfrage in einer Durchgriffssitzung übergeben oder die Abfrage in einer Sicht der Datenquelle angegeben werden.

Serveroptionen des Servers mit föderierten Datenbanken

Die Serveroptionen, die Sie festlegen, präzisieren die Informationen, die der Server mit föderierten Datenbanken über die ferne Datenquelle erhält.

Die oben aufgeführten, die Pushdown-Möglichkeiten beeinflussenden Faktoren sind Merkmale des Datenbankservers, die nicht verändert werden können. Durch eine sorgfältige Prüfung der folgenden Serveroptionen ist es möglich, die Abfrageleistung zu verbessern:

- COLLATING_SEQUENCE. Verwendet eine Datenquelle eine andere Sortierfolge als die föderierte Datenbank, können sortierungsabhängige Operationen für Zeichenwerte nicht fern in der Datenquelle ausgewertet werden. Ein Beispiel dafür ist das Ausführen von MAX-Spaltenfunktionen für eine Kurznamenzeichenspalte in einer Datenquelle mit einer anderen Sortierfolge. Da die Ergebnisse bei Auswertung der Funktion MAX in der fernen Datenquelle fehlerhaft sein können, führt die föderierte Datenbank die Ergebnisoperation und die Funktion MAX lokal aus.
- VARCHAR_NO_TRAILING_BLANKS. Diese Option ist für Zeichenfolgen mit variabler Länge ohne abschließende Leerzeichen bestimmt. Einige Datenquellen,

z. B. Oracle, verwenden keine Vergleichssemantik mit Leerzeichen als Füllzeichen, wie eine föderierte Datenbank. Dieser Unterschied beim Auffüllen kann zu unerwarteten Ergebnissen führen.

Beispiel:

```
'HALLO' = 'HALLO ' in DB2  
'HALLO' <> 'HALLO ' in Oracle!
```

Wenn abschließende Leerzeichen in VARCHAR-Spalten in einer Oracle-Datenquelle vorhanden sind, müssen Sie diese Option auf N setzen (Standardwert für Oracle). Diese Option beeinflusst die Leistung, da der Server mit föderierten Datenbanken die unterschiedliche Semantik ausgleichen muss, gewährleistet jedoch eine konsistente Ergebnismenge. Die Einstellung Y kann zu inkonsistenten Ergebnissen führen, wenn eine Oracle-Datenquelle abschließende Leerzeichen enthält.

Wenn sichergestellt ist, dass alle Spalten VARCHAR und VARCHAR2 in einer Datenquelle keine abschließenden Leerzeichen enthalten, kann das Definieren dieser Serveroption für die Datenquelle in Betracht gezogen werden. Dabei müssen jedoch alle Objekte einschließlich Sichten berücksichtigt werden, die potenziell Kurznamen haben können.

Empfehlung: Definieren Sie diese Option für jede Spalte separat und verwenden Sie die Spaltenoption VARCHAR_NO_TRAILING_BLANKS.

- DB2_MAXIMAL_PUSHDOWN. Diese Option gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Das Abfrageoptimierungsprogramm kann Zugriffspläne entweder auf der Basis des Aufwands auswählen oder auf der Basis der Benutzeranforderung, dass von den fernen Datenquellen so viel Abfrageverarbeitung wie möglich übernommen wird. Ist DB2_MAXIMAL_PUSHDOWN auf Y gesetzt, ist die Reduzierung des Datenaustauschs im Netz das vorrangige Kriterium für das Abfrageoptimierungsprogramm. Das Abfrageoptimierungsprogramm verwendet den Zugriffsplan, der die wenigsten Sendevorgänge an die Datenquelle ausführt. Wenn diese Serveroption auf Y gesetzt ist, muss der Server mit föderierten Datenbanken einen Zugriffsplan verwenden, der möglicherweise mehr Aufwand als nötig verursacht. Wird ein anderer Zugriffsplan verwendet als der Plan, der den geringsten Aufwand verursacht, kann dies die Leistung verringern. Ist die Serveroption DB2_MAXIMAL_PUSHDOWN auf Y gesetzt, wird eine Abfrage, die in einem kartesischen Produkt resultiert, nicht im Pushdown-Modus an ferne Datenquellen weitergeleitet. Abfragen, die in einem kartesischen Produkt resultieren, werden von der föderierten Datenbank verarbeitet. Die Serveroption DB2_MAXIMAL_PUSHDOWN muss nicht auf Y gesetzt werden, damit der Server mit föderierten Datenbanken die Verarbeitung von Abfragen im Pushdown-Modus an ferne Datenquellen sendet. Wenn diese Serveroption auf N gesetzt ist (dies ist die Standardeinstellung), leitet das Abfrageoptimierungsprogramm Abfragen im Pushdown-Modus zur Verarbeitung an die Datenquellen weiter. Ist diese Option auf N gesetzt, ist das primäre Kriterium für das Optimierungsprogramm jedoch der Aufwand und nicht der Datenaustausch im Netz.

„Servermerkmale mit Auswirkungen auf die globale Optimierung“ auf Seite 279 enthält eine Beschreibung zu den Serveroptionen COMM_RATE, CPU_RATIO und IO_RATIO, die ebenfalls Auswirkungen auf die Abfrageleistung haben können.

Faktoren der Typen- und Funktionszuordnung

Sowohl die Standarddatentyp- als auch die Standardfunktionszuordnungen sind in die Datenquellenwrapper integriert. Datentypzuordnungen beschreiben die Bezie-

hung zwischen dem Datentyp der Datenquelle und dem Datentyp des Servers mit föderierten Datenbanken. Sie können Standarddatentypzuordnungen anpassen. Funktionszuordnungen beschreiben die Beziehung zwischen einer Datenquellenfunktion und einer semantisch äquivalenten Funktion auf dem Server mit föderierten Datenbanken. In bestimmten Fällen kompensiert die föderierte Datenbank Funktionszuordnungen, die von einer Datenquelle nicht unterstützt werden.

Die Standarddatentypzuordnungen sind so konzipiert, dass jedem Datentyp der Datenquelle ausreichend Pufferspeicherbereich zur Verfügung steht, um einen Laufzeitpufferüberlauf und ein Abschneiden der Daten zu vermeiden. Sie können die Typzuordnung für eine bestimmte Datenquelle oder einen bestimmten Kurznamen so anpassen, dass sie den Anforderungen bestimmter Anwendungen entspricht und in einigen Fällen zu einer Leistungsverbesserung führt. Beispiel: DATE-Datentypen bei Oracle können sowohl einen Datums- als auch einen Zeitmarkenabschnitt enthalten und werden deshalb standardmäßig dem DB2-Datentyp TIMESTAMP zugeordnet. Wenn Sie auf eine Oracle-Datumsspalte zugreifen und wissen, dass sie nur Datumsabschnitte (keine Zeitmarkenabschnitte) enthält, können Sie die Anweisung ALTER NICKNAME verwenden, um den lokalen Datentyp des Kurznamens von TIMESTAMP in DATE zu ändern, wenn die Serveroption 'date_compat' inaktiviert ist. Bei der Auswertung von Vergleichselementen, die ausschließlich auf einem Datum basieren, wie beispielsweise SalesDate=DATE('2009-03-04'), wird durch diese Änderung die Verwendung einer Skalarfunktion (SCALAR) umgangen, mit deren Hilfe das Datum aus den kombinierten Datums-Zeitmarken-Informationen der Spalte extrahiert wird; hierdurch kann eine Leistungsverbesserung erreicht werden.

Die föderierte Datenbank kompensiert Funktionen, die von einer Datenquelle nicht unterstützt werden. Die Kompensation von Funktionen umfasst normalerweise das Abrufen der erforderlichen Daten von der Datenquelle und das lokale Anwenden der Funktion, was sich häufig auf die Leistung auswirkt. In den folgenden Fällen werden Funktionen kompensiert:

- Eine Funktion ist in der Datenquelle nicht vorhanden. Einige der SYSFUN-Funktionen sind beispielsweise nicht in Datenquellen von DB2 für z/OS vorhanden und erfordern so lokale Kompensation.
- Eine Funktion ist in der Datenquelle zwar vorhanden, die Merkmale des Operanden verletzen jedoch die Funktionseinschränkungen. Ein Beispiel hierfür ist der relationale Operator IS NULL. Die meisten Datenquellen unterstützen diesen Operator. In einigen Fällen gibt es jedoch Einschränkungen, sodass z. B. ein Spaltenname nur links neben dem Operator IS NULL zulässig ist.
- Eine fern ausgewertete Funktion kann ein anderes Ergebnis zurückgeben. Ein Beispiel hierfür ist der Operator '>' (größer als). Für Datenquellen mit abweichenden Sortierfolgen kann der Operator 'größer als' andere Ergebnisse liefern als bei einer lokalen Auswertung durch die föderierte Datenbank.

Kurznamenmerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten

Zu den Kurznamenmerkmalen mit Auswirkungen auf die Pushdown-Möglichkeiten gehören der lokale Datentyp einer Kurznamenspalte, die Optionen der föderierten Spalten sowie MQTs (Materialized Query Tables).

Es gibt verschiedene kurznamenspezifische Faktoren, die Pushdown-Möglichkeiten beeinflussen können. Der lokale Datentyp einer Kurznamenspalte kann die vom Optimierungsprogramm ausgewertete Anzahl der Möglichkeiten in einer verknüpfenden Sortierung beeinflussen. Kurznamen können durch eine Spaltenoption mar-

kiert werden, um anzuzeigen, dass die Spalten keine folgenden Leerzeichen enthalten. Dadurch hat der SQL-Compiler die Möglichkeit, ein effizienteres Format für das Vergleichselement der an die Datenquellen gesendeten SQL-Anweisung zu generieren.

Lokaler Datentyp einer Kurznamenspalte

Stellen Sie sicher, dass der lokale Datentyp einer Spalte nicht die Auswertung eines Vergleichselements in der Datenquelle verhindert.

Die standardmäßigen Datentypzuordnungen sollen einen möglichen Überlauf vermeiden. Ein Joinvergleichselement zwischen zwei Spalten mit unterschiedlichem Datentyp oder unterschiedlicher Länge verhindert jedoch, dass das Optimierungsprogramm das Hash-Join-Verfahren berücksichtigt. Damit das Optimierungsprogramm den Hash-Join berücksichtigt, müssen sowohl die Datentypen als auch die Längen der verknüpften Spalten exakt übereinstimmen. Beispiel: Oracle-Datenquellenspalten, die dafür konzipiert sind, ausschließlich ganzzahlige Werte zu enthalten, werden in der Oracle-Datenbank häufig mit der Angabe NUMBER erstellt; diese wird standardmäßig zu NUMBER (38). Eine Kurznamenspalte für diesen Oracle-Datentyp erhält den lokalen Datentyp FLOAT, da der Bereich einer Ganzzahl in DB2 nur grob dem Typ NUMBER (9) entspricht. In diesem Fall können Joins zwischen einer Spalte ganzer Zahlen in DB2 und einer mit NUMBER definierten Oracle-Spalte (die jedoch ausschließlich ganzzahlige Werte enthält) das Hash-Join-Verfahren nicht verwenden, da die Oracle-Spalte als Typ FLOAT zugeordnet wird. Kann die Domäne dieser mit NUMBER definierten Oracle-Spalte jedoch durch den DB2-Datentyp INTEGER erfasst werden, können Sie den entsprechenden lokalen Datentyp mithilfe der Anweisung ALTER NICKNAME ändern. Das Optimierungsprogramm kann dann das Hash-Join-Verfahren berücksichtigen, was zu einer Leistungsverbesserung führen kann.

Spaltenoptionen der föderierten Datenbank

Sie können Spaltenoptionen der föderierten Datenbank definieren, die das Abfrageoptimierungsprogramm zur Entwicklung von Zugriffsplänen verwendet.

Die Spaltenoptionen bewirken, dass der Wrapper die Daten in einer Spalte anders als sonst verarbeitet. Der SQL-Compiler und das Abfrageoptimierungsprogramm verwenden die Metadaten, um bessere Zugriffspläne für die Daten zu entwickeln. Die föderierte Datenbank behandelt das Objekt, auf das ein Kurzname verweist, wie eine Tabelle. Deshalb können für alle Datenquellenobjekte, für die ein Kurzname erstellt wird, Spaltenoptionen gesetzt werden.

Die Anweisung ALTER NICKNAME kann zum Hinzufügen oder Ändern von Spaltenoptionen für Kurznamen verwendet werden. Es gibt zwei Spaltenoptionen:

- NUMERIC_STRING. Diese Spaltenoption gilt für Zeichentypspalten (CHAR und VARCHAR). Angenommen, eine Datenquelle weist eine andere Sortierfolge als die föderierte Datenbank auf. Der Server mit föderierten Datenbanken würde in diesem Fall keine Spalten mit Zeichendaten in der Datenquelle sortieren. Stattdessen gibt er die Daten an die föderierte Datenbank zurück und führt die Sortierung lokal aus. Nehmen Sie jedoch an, die Spalte weist einen Zeichendatentyp auf und enthält ausschließlich numerische Zeichen ('0', '1', ..., '9'). Sie können dies durch die Zuordnung des Wertes 'Y' zu der Spaltenoption NUMERIC_STRING angeben. Auf diese Weise erhält das Abfrageoptimierungsprogramm die Option, den Sortiervorgang in der Datenquelle durchzuführen, da numerische Zeichen unabhängig von der Sortierfolge stets auf dieselbe Weise sortiert werden, auch wenn sie als Zeichenfolgen dargestellt werden. Wird die Sortierung

fern durchgeführt, kann der Systemaufwand für die Übertragung der Daten zum Server mit föderierten Datenbanken und die lokale Sortierung vermieden werden.

- `VARCHAR_NO_TRAILING_BLANKS`. Im Gegensatz zur gleichnamigen Serveroption kann diese Spaltenoption zur Angabe bestimmter Oracle-Spalten verwendet werden, die keine folgenden Leerzeichen enthalten. Der SQL-Compiler berücksichtigt diese Informationen dann bei der Pushdown-Analyse, wenn alle Operationen für Spalten mit dieser Einstellung überprüft werden. Der SQL-Compiler kann auf der Basis der Einstellung `VARCHAR_NO_TRAILING_BLANKS` ein anderes, aber semantisch äquivalentes Format für das Vergleichselement generieren, das in der an die Datenquelle gesendeten fernen SQL-Anweisung verwendet wird. Durch die Angabe des Wertes 'Y' wird in den meisten Fällen auch die Verwendung ferner Indizes (falls verfügbar) ermöglicht, wodurch die Abfrageleistung verbessert werden kann.

Abfragemerkmale mit Auswirkungen auf die Pushdown-Möglichkeiten

Ein SQL-Operator, der auf mehrere Datenquellen verweist, hat Auswirkungen auf die Pushdown-Verarbeitung.

Eine Abfrage kann auf einen SQL-Operator verweisen, der Kurznamen von mehreren Datenquellen einschließt. Kombiniert der Server mit föderierten Datenbanken die Ergebnisse von zwei Datenquellen unter Verwendung eines einzigen Operators, muss die Operation auf dem Server mit föderierten Datenbanken ausgeführt werden. Ein Beispiel hierfür ist ein Gruppenoperator, z. B. `UNION`. Der Operator kann nicht direkt in einer fernen Datenquelle ausgewertet werden.

Analysieren des Auswertungsortes einer Abfrage

Detaillierte Informationen des Abfrageoptimierungsprogramms befinden sich separat von dem eigentlichen Zugriffsplan in `EXPLAIN`-Tabellen. Mithilfe dieser Informationen ist eine intensive Analyse eines Zugriffsplans möglich. Durch die Überprüfung des Operators `SHIP` eines Zugriffsplans für den föderierten Zugriff kann festgestellt werden, welche SQL-Operationen im Pushdown-Modus an eine Datenquelle weitergeleitet und welche Operationen auf dem Server mit föderierten Datenbanken ausgeführt wurden.

Auf die `EXPLAIN`-Tabellen kann von allen unterstützten Betriebssystemen zugegriffen werden. Sie enthalten Informationen über statische und dynamische SQL-Anweisungen. Die folgenden Tools werden normalerweise verwendet, um Zugriffsplaninformationen aus den `EXPLAIN`-Tabellen abzurufen:

- Das Formattool der `EXPLAIN`-Tabelle. Verwenden Sie das Tool `db2exfmt`, um die Informationen der `EXPLAIN`-Tabellen in einem vordefinierten Format darzustellen.
- Die Tools `db2expln` und `dynexpln`. Sie können diese Tools verwenden, um den Zugriffsplan zu verstehen, der für eine bestimmte SQL-Anweisung ausgewählt wurde. Sie können auch die in die DB2-Steuerzentrale integrierte `EXPLAIN`-Funktion zusammen mit `Visual Explain` verwenden, um den für eine bestimmte SQL-Anweisung ausgewählten Zugriffsplan zu verstehen. Mit `EXPLAIN` können sowohl dynamische als auch statische SQL-Anweisungen bearbeitet werden. Der einzige Unterschied zwischen `EXPLAIN` und `Visual Explain` ist, dass `Visual Explain` die `EXPLAIN`-Informationen in einem Grafikformat darstellt. Die Detaillierungsebene der beiden Methoden ist im Übrigen identisch. Zur vollständigen Verwendung der Ausgabe von `db2expln` und `dynexpln` müssen Sie mit Folgendem vertraut sein:

- Den unterschiedlichen, unterstützten SQL-Anweisungen und der zu diesen Anweisungen gehörenden Terminologie (z. B. Vergleichselemente in einer Anweisung SELECT)
- Dem Zweck eines Pakets (Zugriffsplan)
- Dem Zweck und Inhalt der Systemkatalogtabellen
- Allgemeinen Konzepten zur Anwendungsoptimierung

Sie können auch unter Verwendung von SQL-Anweisungen auf die EXPLAIN-Tabellen zugreifen. Damit werden eine einfache Bearbeitung der Ausgabe, der Vergleich verschiedener Abfragen und Vergleiche derselben Abfrage über einen bestimmten Zeitraum hinweg gewährleistet.

Analysieren des Auswertungsortes einer Abfrage mit der Serveroption DB2_MAXIMAL_PUSHDOWN

Sie können die Serveroption DB2_MAXIMAL_PUSHDOWN zusammen mit den EXPLAIN-Dienstprogrammen verwenden, um zu ermitteln, warum ein bestimmter Operator nicht im Pushdown-Modus zur Ausführung an die Datenquelle weitergeleitet wurde: aufgrund einer kostenbasierten Entscheidung des Optimierungsprogramms oder weil die Pushdown-Analyse feststellte, dass dies nicht möglich war.

Vorgehensweise

Gehen Sie wie folgt vor, um die EXPLAIN-Tools mit der Serveroption DB2_MAXIMAL_PUSHDOWN für eine Abfrage auszuführen:

1. Setzen Sie die Serveroption DB2_MAXIMAL_PUSHDOWN auf 'N'. Dies ist die Standardeinstellung für diese Option. Die Pushdown-Analyse bestimmt, welche Abschnitte der SQL im Pushdown-Modus weitergeleitet werden können. Das Abfrageoptimierungsprogramm generiert anschließend alle alternativen Pläne, die gegen keine der von der Pushdown-Analyse festgelegten Bedingungen verstoßen. Das Abfrageoptimierungsprogramm schätzt den Aufwand für jeden Plan und wählt den Plan mit dem geringsten Aufwand aus. Sie können die Operatoren analysieren, die im Pushdown-Modus an die Datenquelle gesendet wurden, indem Sie die Details des entsprechenden SHIP-Operators anzeigen. Wenn ein Operator wider Erwarten nicht im Pushdown-Modus weitergeleitet wurde, fahren Sie mit Schritt 2 fort.
2. Setzen Sie die Serveroption DB2_MAXIMAL_PUSHDOWN auf 'Y'. Verwenden Sie die EXPLAIN-Tools, um die SQL-Anweisung erneut zu analysieren. Der in der Ausgabe von EXPLAIN angezeigte Plan zeigt alle SQL-Operationen, die an die Datenquelle weitergeleitet werden können.
 - Wenn der Operator im Pushdown-Modus weitergeleitet wird, nachdem die Option auf 'Y' gesetzt wurde, hatte das Optimierungsprogramm ermittelt, dass die lokale Ausführung des Operators kosteneffizienter war als die ferne Ausführung. Wird der Operator nicht im Pushdown-Modus weitergeleitet, nachdem die Option auf 'Y' gesetzt wurde, hat die Pushdown-Analyse die ferne Ausführung des Operators vermutlich nicht zugelassen.
 - Wenn das Optimierungsprogramm kostenbasiert entschieden hat, den Operator nicht im Pushdown-Modus weiterzuleiten, sollten Sie die Kurznamenstatistiken überprüfen, um sicherzustellen, dass sie korrekt sind. Wenn die Pushdown-Analyse entschieden hat, den Operator nicht im Pushdown-Modus weiterzuleiten, sollten Sie die Serveroptionen, die Datentypzuordnungen und die Funktionszuordnungen überprüfen.

Entscheidungen für die Auswertung von Zugriffsplänen

Die Themen in diesem Abschnitt enthalten typische Fragen zur Analyse des Zugriffsplans sowie Informationen zu Bereichen, die Sie zur Erweiterung der Push-down-Möglichkeiten untersuchen können.

Warum wird dieses Vergleichselement nicht fern ausgewertet?

Diese Frage stellt sich bei einem selektiven Vergleichselement, das zum Filtern von Zeilen und somit zur Reduzierung des Netzwerkverkehrs verwendet werden könnte. Eine ferne Auswertung des Vergleichselements bestimmt auch, ob ein Join von zwei Tabellen derselben Datenquelle fern ausgewertet werden kann.

Zu den zu untersuchenden Bereichen gehören:

- Serveroptionen. Wie wirken sich die Einstellungen für die Serveroptionen `COLLATING_SEQUENCE` und `VARCHAR_NO_TRAILING_BLANKS` darauf aus, wo das Vergleichselement ausgewertet wird?
- Vergleichselemente für Unterabfragen. Enthält dieses Vergleichselement eine Unterabfrage, die zu einer anderen Datenquelle gehört? Enthält dieses Vergleichselement eine Unterabfrage mit einem SQL-Operator, der von dieser Datenquelle nicht unterstützt wird? Nicht alle Datenquellen unterstützen Gruppenoperatoren in einem Vergleichselement.
- Funktionen von Vergleichselementen. Enthält dieses Vergleichselement eine Funktion, die von dieser fernen Datenquelle nicht ausgewertet werden kann? Relationale Operatoren werden als Funktionen klassifiziert.
- Einbindungsanforderungen von Vergleichselementen. Erfordert dieses Vergleichselement bei ferner Auswertung die Einbindung von Werten? Würde dies zur Verletzung von SQL-Einschränkungen in dieser Datenquelle führen?
- Globale Optimierung. Das Optimierungsprogramm hat festgestellt, dass eine lokale Verarbeitung weniger kostenintensiv ist.

Warum wird der Operator `GROUP BY` nicht fern ausgewertet?

Sie können eine Reihe von Bereichen überprüfen, um festzustellen, warum ein `GROUP BY`-Operator nicht fern ausgewertet wird.

Hierzu gehören folgende Bereiche:

- Wird die Eingabe für den Operator `GROUP BY` fern ausgewertet? Wenn nicht, überprüfen Sie die Eingabe.
- Bestehen in der Datenquelle Einschränkungen für diesen Operator? Beispiele:
 - Begrenzte Anzahl von `GROUP BY`-Elementen
 - Begrenzte Anzahl der Bytes bei kombinierten `GROUP BY`-Elementen
 - Spaltenspezifikation nur für `GROUP BY`-Liste
- Unterstützt die Datenquelle diesen SQL-Operator?
- Globale Optimierung. Das Optimierungsprogramm hat festgestellt, dass eine lokale Verarbeitung weniger kostenintensiv ist.

Warum wird der Operator `SET` nicht fern ausgewertet?

Sie können die Operanden und die Einschränkungen der Datenquelle überprüfen, um festzustellen, warum der Operator `SET` nicht fern ausgewertet wird.

Überlegungen:

- Werden beide Operanden des Operators SET vollständig in derselben fernen Datenquelle ausgewertet? Wenn nicht, überprüfen Sie alle Operanden.
- Bestehen in der Datenquelle Einschränkungen für diesen Operator SET? Sind z. B. große Objekte oder lange Felder für diesen bestimmten Operator SET eine gültige Eingabe?

Warum wird die Operation ORDER BY nicht fern ausgewertet?

Sie können die Operationseingabe, den Inhalt der Klausel und Einschränkungen der Datenquelle überprüfen, um festzustellen, warum der Operator ORDER BY nicht fern ausgewertet wird.

Überlegungen:

- Serveroptionen. Wie wirken sich die Einstellungen für die Serveroptionen COLLATING_SEQUENCE und VARCHAR_NO_TRAILING_BLANKS darauf aus, wo das Vergleichselement ausgewertet wird?
- Wird die Eingabe für den Operator ORDER BY fern ausgewertet? Wenn nicht, überprüfen Sie die Eingabe.
- Enthält die Klausel ORDER BY einen Zeichenausdruck? Wenn ja, verfügt die ferne Datenquelle über eine andere Sortierfolge als der Server mit föderierten Datenbanken?
- Bestehen in der Datenquelle Einschränkungen für diesen Operator? Ist beispielsweise die Anzahl der ORDER BY-Elemente begrenzt? Bestehen in der Datenquelle Einschränkungen in Bezug auf Spaltenspezifikationen für die Liste ORDER BY?

Warum wird eine ferne Anweisung INSERT mit Fullselect nicht vollständig fern ausgewertet?

Sie können eine Reihe von Elementen des Subselects überprüfen, um festzustellen, warum eine ferne Anweisung INSERT mit Fullselect nicht vollständig fern ausgewertet wird.

Überlegungen:

- Könnte der Subselect vollständig in der fernen Datenquelle ausgewertet werden? Wenn nicht, überprüfen Sie den Subselect.
- Enthält der Subselect einen Gruppenoperator? Wenn ja, unterstützt diese Datenquelle Gruppenoperatoren als Eingabe für eine Anweisung INSERT?
- Verweist der Subselect auf die Zieltabelle? Wenn ja, lässt die Datenquelle diese Syntax zu?

Warum wird eine ferne Anweisung INSERT mit der Klausel VALUES nicht vollständig fern ausgewertet?

Sie können die Klausel VALUES sowie den Ausdruck überprüfen, um festzustellen, warum eine ferne Anweisung INSERT mit der Klausel VALUES nicht vollständig fern ausgewertet wird.

Überlegungen:

- Kann die Klausel VALUES vollständig in der fernen Datenquelle ausgewertet werden? Anders ausgedrückt, enthält ein Ausdruck eine Funktion, die von der fernen Datenquelle nicht unterstützt wird?
- Schließt der Ausdruck eine skalare Unterabfrage ein? Wird diese Syntax unterstützt?
- Verweist der Ausdruck auf die Zieltabelle? Wird diese Syntax unterstützt?

Warum wird eine ferne Anweisung UPDATE für eine Aktualisierung mit Suche nicht vollständig fern ausgewertet?

Sie können Elemente der Klausel SET und der Suchbedingung überprüfen, um festzustellen, warum eine ferne Anweisung UPDATE für eine Aktualisierung mit Suche nicht vollständig fern ausgewertet wird.

Überlegungen:

- Kann die Klausel SET vollständig in der fernen Datenquelle ausgewertet werden? Anders ausgedrückt, enthält ein Aktualisierungsausdruck eine Funktion, die von der fernen Datenquelle nicht unterstützt wird?
- Schließt die Klausel SET eine skalare Unterabfrage ein? Unterstützt die Datenquelle diese Syntax?
- Kann die Suchbedingung vollständig in der fernen Datenquelle ausgewertet werden? Wenn nicht, überprüfen Sie die Suchbedingung.
- Verweisen die Suchbedingung oder die Klausel SET auf die Zieltabelle? Unterstützt die Datenquelle diese Syntax?
- Verweisen die Suchbedingung oder die Klausel SET mit Korrelation auf die Zieltabelle? Unterstützt die Datenquelle diese Syntax?

Warum wird eine Anweisung UPDATE für eine positionierte Aktualisierung nicht vollständig fern ausgewertet?

Dies ist der Fall, wenn die föderierte Datenbank den Aktualisierungsausdruck lokal auswertet und erst dann die Anweisung UPDATE an die Datenquelle sendet. Diese Methode beeinflusst die Leistung unerheblich.

Überlegungen:

- Kann die Klausel SET vollständig in der fernen Datenquelle ausgewertet werden? Anders ausgedrückt, enthält ein Aktualisierungsausdruck eine Funktion, die von der fernen Datenquelle nicht unterstützt wird?
- Schließt die Klausel SET eine skalare Unterabfrage ein? Unterstützt die Datenquelle diese Syntax?

Warum wird eine ferne Anweisung DELETE für einen gezielten Löschvorgang nicht vollständig fern ausgewertet?

Sie können Elemente der Suchbedingung überprüfen, um festzustellen, warum eine ferne Anweisung DELETE für einen gezielten Löschvorgang nicht vollständig fern ausgewertet wird.

Überlegungen:

- Kann die Suchbedingung vollständig in der fernen Datenquelle ausgewertet werden? Wenn nicht,überprüfen Sie die Suchbedingung.
- Verweist die Suchbedingung auf die Zieltabelle? Unterstützt die Datenquelle diese Syntax?
- Verweist die Suchbedingung mit Korrelation auf die Zieltabelle? Unterstützt die Datenquelle diese Syntax?

Aktualisieren und Anpassen von Datenquellen

Wenn für Datenquellen ein Upgrade oder eine Anpassung durchgeführt wird, müssen die Informationen im globalen Katalog aktualisiert werden.

Der SQL-Compiler bezieht sich auf die im globalen Katalog gespeicherten Informationen über das SQL-Leistungsspektrum der jeweiligen Datenquelle. Diese Informationen müssen in regelmäßigen Abständen aktualisiert werden. Das SQL-Leistungsspektrum der Datenquellen kann sich mit neuen Versionen der einzelnen Datenquellen ändern. Werden Datenquellen aktualisiert oder angepasst, aktualisieren Sie die Informationen im globalen Katalog, um sicherzustellen, dass der SQL-Compiler die neuesten Informationen verwendet.

Verwenden Sie zum Aktualisieren des Katalogs SQL-DDL-Anweisungen, wie z. B. CREATE FUNCTION MAPPING und ALTER SERVER.

Pushdown-Verarbeitung von Vergleichselementen mit Funktionsschablonen

In einem föderierten System sind jeder fernen Datenquelle eigene Funktionen zugeordnet. Die meisten dieser Funktionen verfügen über semantisch äquivalente DB2-Funktionen und standardmäßig über entsprechende Funktionszuordnungen. Für einige ferne Quellenfunktionen gibt es jedoch möglicherweise keine äquivalenten Funktionen auf dem Server mit föderierten Datenbanken. Diese Funktionen können daher nur von der fernen Datenquelle ausgeführt werden. Um Abfragen zu schreiben, die diese Funktionen verwenden, müssen Sie eine Funktionsschablone auf dem Server mit föderierten Datenbanken erstellen.

Eine Funktionsschablone fungiert als lokale Beschreibung der fernen Funktion. Eine Funktionsschablone wird mit der Anweisung CREATE FUNCTION und der Klausel AS TEMPLATE erstellt. Der Funktionsschablone auf dem Server mit föderierten Datenbanken ist kein ausführbarer Code zugeordnet. Nach der Definition der Schablone können Sie sie verwenden, um eine Funktionszuordnung zu erstellen, mit der die Funktionsschablone der entsprechenden fernen Funktion zugeordnet wird. Es ist dann möglich, in den SQL-Anweisungen, die an den Server mit föderierten Datenbanken gesendet werden, auf die Funktionsschablone zu verweisen und die Funktion in der Datenquelle auszuwerten.

Das Abfrageoptimierungsprogramm entscheidet kostenbasiert, wo ein Vergleichselement ausgewertet werden kann. Wenn möglich, erstellt das Optimierungsprogramm einen Plan zur Auswertung eines Vergleichselements mit einer Funktionsschablone auf dem entsprechenden fernen Server. In manchen Fällen kann das Optimierungsprogramm möglicherweise keinen Plan erstellen, mit dem die Funktionsschablone in der Datenquelle ausgewertet wird. In diesem Fall wird die Fehlermeldung SQL0142N mit folgendem Text ausgegeben:

Die SQL-Anweisung wird nicht unterstützt.

Um diesen Fehler zu vermeiden, kann die Abfrage so umgeschrieben werden, dass eine Weiterleitung im Pushdown-Modus ermöglicht, die Semantik der ursprünglichen Abfrage jedoch beibehalten wird.

Damit eine Funktionsschablone im Pushdown-Modus weitergeleitet werden kann, muss sie mit den Klauseln DETERMINISTIC und NO EXTERNAL ACTION erstellt werden.

Kapitel 24. Parallelität bei Abfragen, die auf Kurznamen verweisen

Abfragen, die Kurznamen enthalten, können an drei Arten der abfrageinternen Parallelität beteiligt sein.

Die drei Arten der abfrageinternen Parallelität lauten:

- Partitionsinterne Abfrageparallelität bei Einzelpartitionskonfigurationen mit Multiprozessoren
- Partitionsübergreifende Abfrageparallelität bei Mehrpartitionskonfigurationen
- Gemischte Abfrageparallelität, die aus partitionsinterner und -übergreifender Parallelität besteht, wobei jede Partition auf einem SMP-Computer ausgeführt wird

Partitionsinterne Parallelität bei Abfragen, die auf Kurznamen verweisen

Die partitionsinterne Parallelität bezieht sich auf den Prozess der Teilung einer Abfrage in mehrere gleichzeitig ausführbare Teile, die von mehreren Prozessen in einer einzelnen Datenbankpartition parallel ausgeführt werden.

Bei Abfragen für föderierte Datenbanken kann der Teil einer Abfrage mit lokalen Daten parallel ausgeführt werden, während der Teil mit Kurznamen seriell ausgeführt wird, wobei ein einzelner Agentenprozess verwendet wird.

Wenn mehrere Prozessoren die lokalen Teile der Abfrage bearbeiten können, kann die Leistung bei Abfragen, die auf lokale Tabellen und Kurznamen verweisen, verbessert werden.

Der Datenbankkonfigurationsparameter `DFT_DEGREE` und das Sonderregister `CURRENT DEGREE` steuern den Grad der partitionsinternen Parallelität.

Aktivieren der partitionsinternen Parallelität bei Abfragen, die auf Kurznamen verweisen

Für Abfragen, die auf lokale Tabellen und Kurznamen in einer Multiprozessorumgebung verweisen, können Sie die partitionsinterne Parallelität aktivieren. Der Server mit föderierten Datenbanken kann dann die lokalen Tabellen parallel verarbeiten.

Einschränkungen

Das föderierte System kann nur den Teil einer Abfrage parallel verarbeiten, der auf lokale Tabellen verweist. Die Koordinatorpartition verarbeitet alle Operationen für den fernen Teil einer Abfrage seriell.

Vorgehensweise

Gehen Sie wie folgt vor, um die partitionsinterne Parallelität zu aktivieren:

1. Setzen Sie den Datenbankkonfigurationsparameter `INTRA_PARALLEL` auf YES (JA).

2. Setzen Sie den Datenbankkonfigurationsparameter MAX_QUERYDEGREE auf einen Wert, der größer als 1 ist.
3. Setzen Sie den Datenbankkonfigurationsparameter DFT_DEGREE auf einen Wert, der größer als 1 ist, oder setzen Sie das Sonderregister CURRENT DEGREE. Wenn Sie den Parameter DFT_DEGREE auf ANY setzen, ist die Standardstufe der partitionsinternen Parallelität gleich der Anzahl der Prozessoren auf dem Computer.

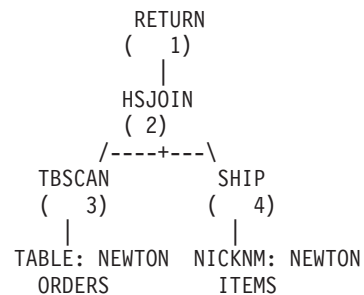
Partitionsinterne Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne

Sie können die EXPLAIN-Funktion von DB2 verwenden, um den Zugriffsplan anzuzeigen, den das Optimierungsprogramm bei der Abfrageverarbeitung verwendet. Die folgenden Beispiele zeigen, wie das Optimierungsprogramm in einer Umgebung mit partitionsinterner Parallelität auf Kurznamendaten zugreift.

Beispiel 1: Ohne Unterstützung der Parallelität

Bei diesem Beispiel verarbeitet der Server mit föderierten Datenbanken den Join zwischen der lokalen Tabelle ORDERS und dem Kurznamen ITEMS seriell. Die partitionsinterne Parallelität wird nicht verwendet.

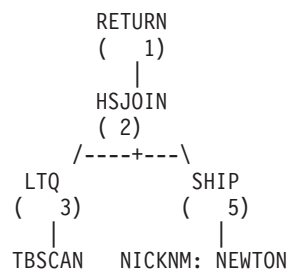
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Beispiel 2: Mit Unterstützung der Parallelität

Bei diesem Beispiel eines Joins kann die Abfrage schneller ausgeführt werden, da die lokale Tabelle vor dem seriellen Join mit dem Kurznamen parallel eingelesen wird. Der Operator LTQ gibt an, an welcher Stelle die Parallelität in den Plan eingefügt wird.

```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```





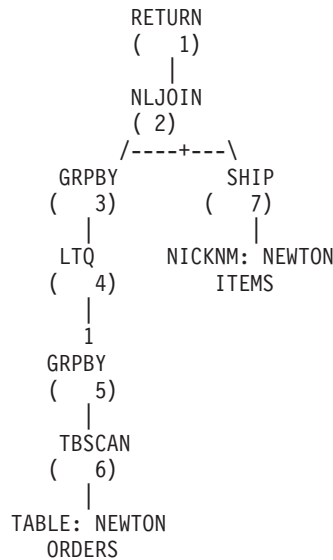
Beispiel 3: Partitionsinterne Parallelität bei Spaltenberechnung

Bei diesem Beispiel werden die lokalen Tabellendaten von der Datenbank in der Partition parallel zusammengefasst, wodurch die Ausführung der Spaltenberechnung verbessert wird. Der Join der lokalen Tabelle und des Kurznamens erfolgt seriell.

```

SELECT *
FROM ITEMS A
WHERE ID =
  (SELECT MAX(ID)
   FROM ORDERS
   WHERE NUMBER = 10)

```



Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen

Die partitionsübergreifende Parallelität bezieht sich auf den Prozess der Teilung einer einzelnen Abfrage in mehrere Teile, die in verschiedenen Partitionen einer partitionierten Datenbank parallel ausgeführt werden.

Bei Abfragen, die auf lokale und ferne Daten verweisen, kann der Server mit föderierten Datenbanken die fernen Daten an jede lokale Partition verteilen. In Abb. 10 auf Seite 256 und Abb. 11 auf Seite 256 ist das Konzept der partitionsübergreifenden Parallelität mit lokalen und fernen Datenquellen dargestellt.

Abb. 10 auf Seite 256 zeigt, wie dieser Abfragetyp ohne partitionsübergreifende Parallelität verarbeitet wird. Die fernen Kurznamendaten und die lokalen partitionierten Daten werden seriell in einer Koordinatorpartition verarbeitet. Dieses Verfahren nutzt das Parallelverarbeitungspotenzial der Datenbankpartitionen nicht, da der Hauptteil der Verarbeitung auf einer einzelnen Partition stattfindet. Bei sehr großen Datenvolumen führt dieses Verfahren in vielen Fällen zu Abfragen mit langer Laufzeit.

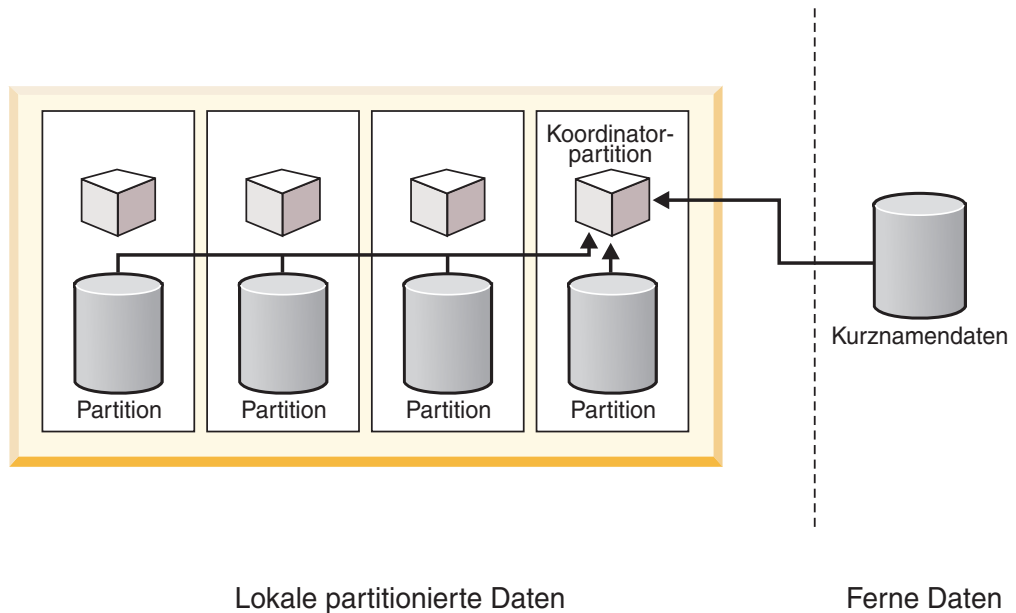


Abbildung 10. Abfrage ohne partitionsübergreifende Parallelität lokaler und ferner Datenquellen

Abb. 11 zeigt, wie die Verarbeitung erfolgt, wenn das Optimierungsprogramm die Kurznamendaten an die Partitionen verteilt. Die Koordinatorpartition ruft die Kurznamendaten ab und verteilt sie an die Datenbankpartitionen zur Parallelverarbeitung. Nach Abschluss der Parallelverarbeitung werden die Ergebnisse zur abschließenden Verarbeitung zurück an die Koordinatorpartition gesendet, bevor sie an die Anwendung zurückgegeben werden.

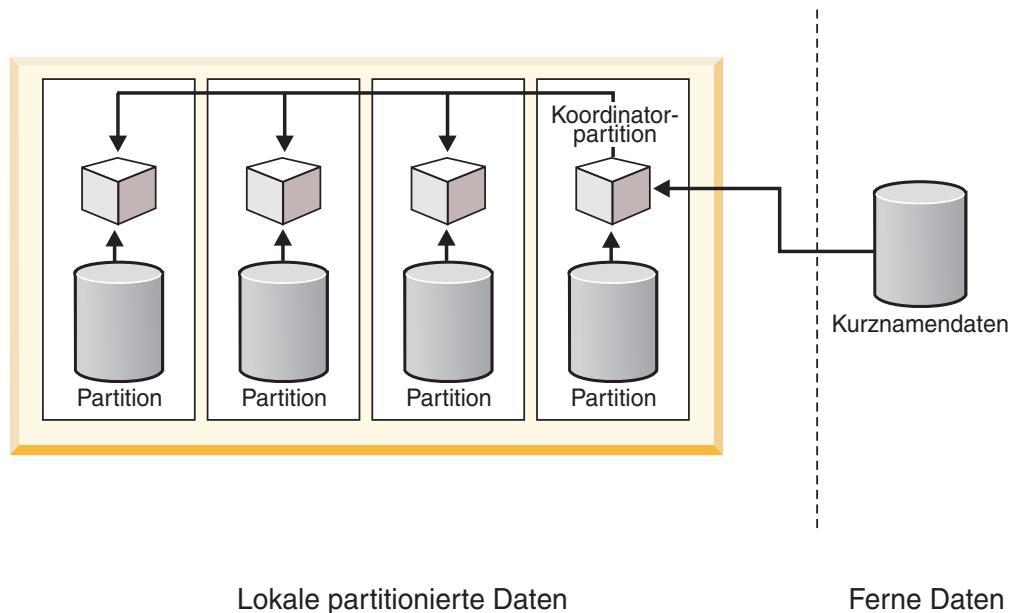


Abbildung 11. Abfrage mit partitionsübergreifende Parallelität lokaler und ferner Datenquellen

Abb. 12 auf Seite 257 und Abb. 13 auf Seite 258 zeigen das Konzept der partitionsübergreifenden Parallelität, an der nur ferne Datenquellen beteiligt sind.

Abb. 12 zeigt die serielle Verarbeitung der fernen Kurznamendaten in der Koordinatorpartition. Die Koordinatorpartition, die auch als Server mit föderierten Datenbanken fungiert, ruft die Kurznamendaten ab und verarbeitet sie seriell.

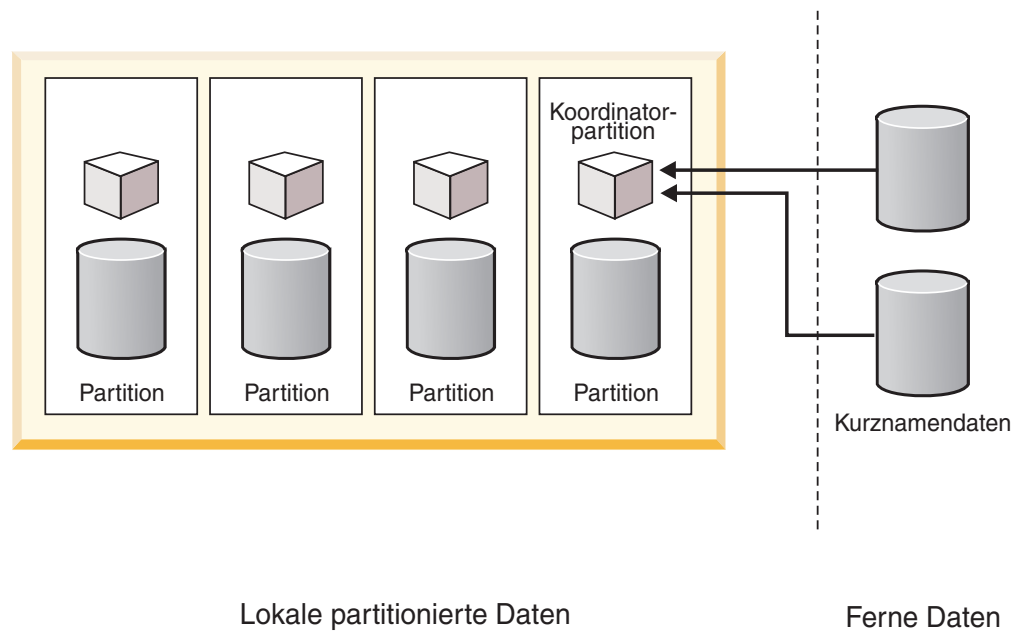


Abbildung 12. Abfrage ohne partitionsübergreifende Parallelität, die nur auf ferne Datenquellen verweist.

Abb. 13 auf Seite 258 zeigt, wie die Koordinatorpartition die Daten über eine Rechenpartitionsgruppe verteilt. Rechenpartitionsgruppen ermöglichen es dem Optimierungsprogramm, Zugriffspläne zu erstellen, die Kurznamendaten zur Parallelverarbeitung an die Partitionen eines Servers mit partitionierten Datenbanken verteilen.

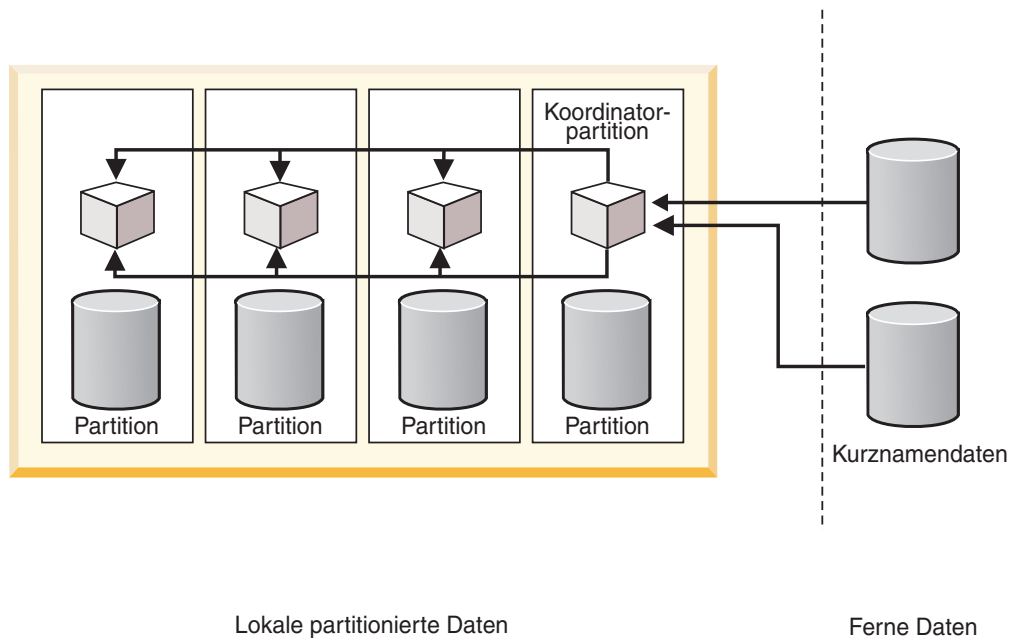


Abbildung 13. Abfrage mit partitionsübergreifender Parallelität, die nur auf ferne Datenquellen verweist.

Unabhängig vom Plan, den das Optimierungsprogramm auswählt, erfolgt der Zugriff auf die Kurznamendaten stets seriell über die Koordinatorpartition.

Aktivieren von partitionsübergreifender Parallelität bei Abfragen, die auf Kurznamen verweisen

Sie können einen partitionierten Server mit föderierten Datenbanken so konfigurieren, dass Abfragen mit Kurznamen gegebenenfalls parallel auf mehreren Partitionen ausgeführt werden können. Durch die parallele Ausführung kann die für föderierte Abfragen in einer partitionierten Umgebung benötigte Zeit deutlich reduziert werden.

Einschränkungen

Es können nur die Teile einer Abfrage, die auf Kurznamen mit abgeschirmten Wrappern verweisen, parallel ausgeführt werden. Die Teile einer Abfrage, die auf Kurznamen mit gesicherten Wrappern verweisen, können nicht parallel ausgeführt werden.

Informationen zu dieser Task

In einer partitionierten Datenbankumgebung können Abfragen für föderierte Datenbanken unter folgenden Bedingungen parallel ausgeführt werden:

- Sie enthalten eine Kombination aus Kurznamen, die mithilfe eines abgeschirmten Wrappers definiert wurden, und lokalen partitionierten Tabellen.
- Sie enthalten Kurznamen, die mithilfe eines abgeschirmten Wrappers definiert wurden, und es wurde eine Rechenpartitionsgruppe definiert.

Sie müssen keine Datenbankparameter bzw. Datenbankkonfigurationsparameter in einer partitionierten Umgebung definieren, um die partitionsübergreifende Parallelität für föderierte Abfragen verfügbar zu machen.

Vorgehensweise

Gehen Sie wie folgt vor, um die partitionsübergreifende Parallelität zu aktivieren:

1. Setzen Sie die Anweisung CREATE WRAPPER bzw. ALTER WRAPPER ab, und setzen Sie die Option DB2_FENCED auf Y.
2. **Optional:** Definieren Sie eine Rechenpartitionsgruppe, wenn Sie die Parallelität für die Abschnitte von Abfragen aktivieren möchten, in denen nur Kurznamen verwendet werden. Wenn Abfragen eine Kombination aus Kurznamen und lokalen partitionierten Tabellen enthalten, muss keine Rechenpartitionsgruppe definiert werden.

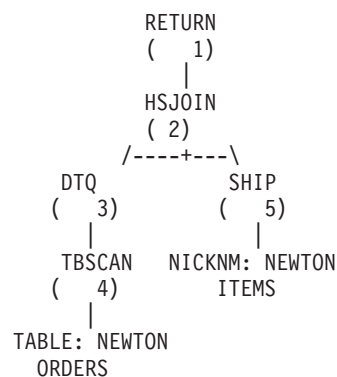
Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne

Sie können die EXPLAIN-Funktion von DB2 verwenden, um den Zugriffsplan anzuzeigen, den das Optimierungsprogramm bei der Abfrageverarbeitung verwendet. Die folgenden Beispiele zeigen, wie das Optimierungsprogramm in einer Umgebung mit partitionsübergreifender Parallelität auf Kurznamendaten zugreift.

Beispiel 1: Gesicherter Modus

In diesem Beispiel verwendet der Kurzname einen gesicherten Wrapper. Die Datenbank führt den Join zwischen der lokalen Tabelle und dem Kurznamen in der Koordinatorpartition seriell durch. Die Datenbank sendet die über zwei Partitionen verteilten lokalen Daten an die Koordinatorpartition. Der Server mit föderierten Datenbanken verknüpft dann die lokalen Daten mit den Kurznamendaten. Die Datenbank verknüpft Kurznamen, die mithilfe eines gesicherten Wrappers in der Koordinatorpartition definiert wurden, seriell. Die Datenbank kann die Daten nicht über mehrere Partitionen verteilen, um einen parallelen Join zu erstellen.

```
SELECT *  
FROM ORDERS A, ITEMS B  
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

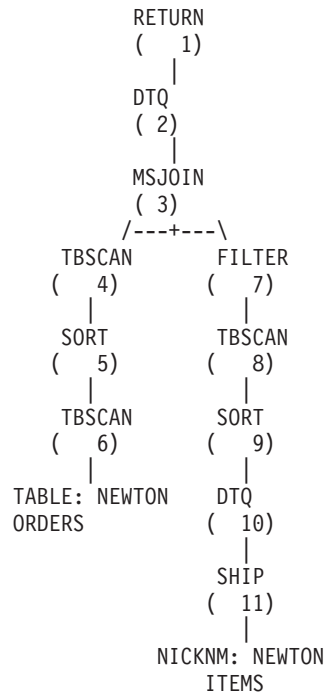


Beispiel 2: Abgeschirmter Modus

In diesem Beispiel verwendet der Kurzname einen abgeschirmten Wrapper. Der Server mit föderierten Datenbanken verteilt die Kurznamendaten an andere Partitionen und führt den Join mit den lokalen Daten parallel durch. Der Operator DTQ (Distributed Table Queue) oberhalb des Operators SHIP gibt an, dass die Kurznamendaten mithilfe der Hashpartitionierung an die lokalen Partitionen verteilt werden, um einen zusammengefassten parallelen Join zu erreichen. Bei einem zusammengefassten parallelen Join werden die Kurznamendaten so an die lokalen

Partitionen verteilt, dass sich übereinstimmende Kurznamendaten und lokale Daten für den Join stets auf derselben Partition befinden.

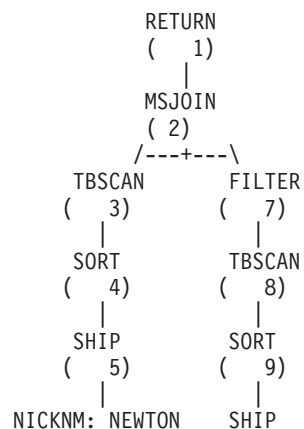
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

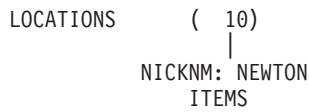


Beispiel 3: Abgeschirmter Modus ohne Rechenpartitionsgruppe

In diesem Beispiel verwenden die beiden Kurznamen einen abgeschirmten Wrapper; eine Rechenpartitionsgruppe wird nicht definiert. Der Server mit föderierten Datenbanken führt den Join in der Koordinatorpartition durch. Der Server mit föderierten Datenbanken verteilt die Daten für die Verarbeitung nicht an andere Partitionen. Die Tatsache, dass keinem SHIP-Operator ein TQ-Operator vorangeht, zeigt, dass die Kurznamendaten nicht partitionsübergreifend verteilt werden.

```
SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID1 = B.ID1
```





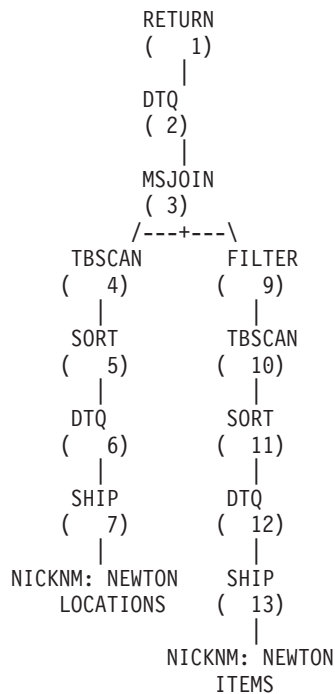
Beispiel 4: Abgeschirmter Modus mit Rechenpartitionsgruppe

In diesem Beispiel verwenden die Kurznamen abgeschirmte Wrapper; eine Rechenpartitionsgruppe wird definiert. In diesem Fall wählt das Optimierungsprogramm einen Plan aus, bei dem die Daten von der Koordinatorpartition an die anderen Partitionen der Rechenpartitionsgruppe verteilt werden. Die DTQ-Operatoren oberhalb beider Kurznamen bewirken eine Hashpartitionierung der ankommenden ferneren Daten, sodass sich übereinstimmende Joinschlüssel auf derselben Partition der Rechenpartitionsgruppe befinden. Der Join wird parallel auf jeder Partition durchgeführt, und die Ergebnisse werden auf der Koordinatorpartition gesammelt.

```

SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID = B.ID

```



Rechenpartitionsgruppen

Eine Rechenpartitionsgruppe definiert eine Reihe von Datenbankpartitionen, die das Optimierungsprogramm dazu verwenden kann, Kurznamendaten dynamisch umzuverteilen. Eine Rechenpartitionsgruppe ist für die Abschnitte von Abfragen relevant, in denen nur Kurznamen verwendet werden.

Die Koordinatorpartition ruft Kurznamendaten seriell ab und verteilt die Daten anschließend auf die Partitionen innerhalb der Rechenpartitionsgruppe, wobei Parallelverarbeitung stattfindet. Die Verwendung von Rechenpartitionsgruppen durch das Optimierungsprogramm führt in vielen Fällen zu Leistungsverbesserungen, vor allem dann, wenn große Mengen an Kurznamendaten verarbeitet werden oder die Abfragen komplex sind.

Eine Rechenpartitionsgruppe ist eine Datenbankpartitionsgruppe (nicht IBM-CAT-NODEGROUP), die im Systemkatalog SYSCAT.DBPARTITIONGROUPS angegeben wird.

Verwenden Sie die Registrierdatenbankvariable DB2_COMPPARTITIONGROUP, um die Rechenpartitionsgruppe anzugeben.

Definieren einer Rechenpartitionsgruppe

Die Definition einer Rechenpartitionsgruppe ermöglicht es dem Optimierungsprogramm, einen Plan zu verwenden, mit dem Kurznamendaten an die Partitionen der Rechenpartitionsgruppe verteilt werden. Sie definieren eine Rechenpartitionsgruppe, um die partitionsübergreifende Abfrageparallelität für Abfragen oder Teile von Abfragen zu aktivieren, die nur auf Kurznamen verweisen.

Vorbereitung

Alle Partitionsgruppen, die zur Darstellung der Rechenpartitionsgruppe in allen Datenbanken der Instanz verwendet werden, müssen denselben Namen haben. Sie können diese Partitionsgruppen zwar in jeder Datenbank unterschiedlich definieren, der Name muss jedoch identisch sein. Beispiel: Die drei Datenbanken DB1, DB2 und DB3 definieren eine Rechenpartitionsgruppe (Computational Partition Group, CPG), die unterschiedliche Knoten enthält:

- DB1: Die Rechenpartitionsgruppe (CPG) enthält die Knoten 1, 2, 3 und 4.
- DB2: Die Rechenpartitionsgruppe (CPG) enthält die Knoten 49, 50 und 53.
- DB3: Die Rechenpartitionsgruppe (CPG) enthält die Knoten 78 und 96.

Sie können für die Variable 'db2set' den Namen 'CPG' angeben. Der Name 'CPG' ist für alle Datenbanken gleich, der Inhalt der Rechenpartitionsgruppe jedoch für die einzelnen Datenbanken unterschiedlich.

Einschränkungen

Das Optimierungsprogramm verwendet Rechenpartitionsgruppen nur für die Teile einer Abfrage, die auf Kurznamen verweisen, ohne dabei auf lokale Daten zu verweisen.

Vorgehensweise

Geben Sie zur Definition einer Rechenpartitionsgruppe den folgenden Befehl in der DB2-Befehlszeile ein.

```
db2set DB2_COMPPARTITIONGROUP=partitionsgruppenname
```

Dabei ist 'partitionsgruppenname' der Name der Partitionsgruppe, die Sie als Rechenpartitionsgruppe definieren möchten. Die Partitionsgruppe muss bereits definiert sein.

Im folgenden Beispiel sehen Sie, wie die Rechenpartitionsgruppe FINANCE3 mit der Registrierdatenbankvariablen DB2_COMPPARTITIONGROUP definiert wird.

```
db2set DB2_COMPPARTITIONGROUP=FINANCE3
```

Partitionsübergreifende Parallelität bei Abfragen, die auf Kurznamen verweisen - erwartetes Leistungsverhalten

Bei Abfragen, die auf eine Kombination lokaler partitionierter Tabellen und Kurznamen verweisen, kann das Optimierungsprogramm einen Ausführungsplan auswählen, der Kurznamendaten auf die geeigneten Partitionen umverteilt.

Mithilfe von Umverteilungsplänen können Abfragen schneller ausgeführt werden, wenn die Menge an Kurznamendaten im Join kleiner als die Menge an lokalen partitionierten Daten ist. Wenn die Menge der Kurznamendaten im Join erheblich größer als die Menge der lokalen Daten ist, ist die Verwendung eines parallelen Plans zur Umverteilung der Kurznamendaten unwahrscheinlich. Wenn das Optimierungsprogramm keinen parallelen Plan auswählt, führt der Server mit föderierten Datenbanken die Joins zwischen Kurznamen und lokalen Tabellen in der Koordinatorpartition seriell durch.

Bei Joins zwischen zwei Kurznamen kann ein Ausführungsplan von Vorteil sein, bei dem die Daten unter allen Partitionen einer Rechenpartitionsgruppe verteilt werden, sofern hierbei eine große Datenmenge verarbeitet wird. Die Vorteile einer parallelen Verarbeitung von Joins mit großen Datenmengen überwiegt den zusätzlichen Aufwand der Umverteilung der Daten über mehrere Partitionen. Wenn die Menge an Kurznamendaten relativ gering ist, ist der Join nicht aufwendig genug, dass sich der Extraaufwand einer Umverteilung der Daten über die Partitionen lohnt. Im Allgemeinen wählt das Optimierungsprogramm Rechenpartitionsgruppenpläne aus, wenn die Menge der beteiligten Kurznamendaten groß ist. Andernfalls verknüpft der Server mit föderierten Datenbanken die Kurznamen in der Koordinatorpartition seriell.

Gemischte Parallelität bei Abfragen, die auf Kurznamen verweisen

Bei Abfragen mit lokalen Tabellen und Kurznamen in einer partitionierten Umgebung kann das Optimierungsprogramm sowohl mit partitionsinterner als auch mit partitionsübergreifender Parallelität arbeiten. Partitionsübergreifende Parallelität ist eine Option für das Optimierungsprogramm in einer partitionierten Umgebung. Partitionsinterne Parallelität ist dann eine Option, wenn sie in der Datenbankkonfiguration oder in der Konfiguration des Datenbankmanagers aktiviert wurde.

Bei der partitionsübergreifenden Parallelität kann der Server mit föderierten Datenbanken ferne Daten an Partitionen verteilen und die Daten parallel in den einzelnen Partitionen verarbeiten.

Bei der partitionsinternen Parallelität werden mehrere Subagentenprozesse innerhalb einer Partition verwendet, um lokale Daten parallel zu verarbeiten.

Aktivieren von gemischter Parallelität bei Abfragen, die auf Kurznamen verweisen

Mithilfe der partitionsinternen und der partitionsübergreifenden Parallelität können Sie die Leistung von Abfragen verbessern, die auf lokale und ferne Daten verweisen.

Vorgehensweise

Gehen Sie wie folgt vor, um die partitionsübergreifende Parallelität für einen partitionierten Server mit föderierten Datenbanken zu aktivieren:

1. Setzen Sie die Anweisung CREATE WRAPPER bzw. ALTER WRAPPER ab, und setzen Sie die Option DB2_FENCED auf Y.
2. **Optional:** Definieren Sie eine Rechenpartitionsgruppe, wenn Sie die Parallelität für Joins aktivieren möchten, in denen nur Kurznamen verwendet werden.

Gehen Sie wie folgt vor, um die partitionsinterne Parallelität für einen Server mit föderierten Datenbanken zu aktivieren:

1. Setzen Sie den Datenbankkonfigurationsparameter MAX_QUERYDEGREE auf einen Wert, der größer als 1 ist.
2. Setzen Sie den Datenbankkonfigurationsparameter DFT_DEGREE auf einen Wert, der größer als 1 ist, oder setzen Sie das Sonderregister CURRENT DEGREE. Wenn Sie den Parameter DFT_DEGREE auf ANY setzen, ist die Standardstufe der partitionsinternen Parallelität gleich der Anzahl der SMP-Prozessoren auf dem Computer.

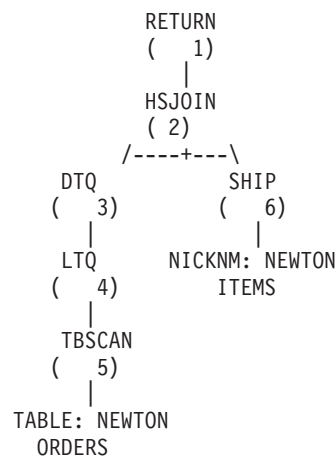
Gemischte Parallelität bei Abfragen, die auf Kurznamen verweisen - Beispiele für Zugriffspläne

Sie können die EXPLAIN-Funktion von DB2 verwenden, um den Zugriffsplan anzuzeigen, den das Optimierungsprogramm bei der Abfrageverarbeitung verwendet. Die folgenden Beispiele zeigen, wie das Optimierungsprogramm in einer Umgebung mit partitionsinterner und partitionsübergreifender Parallelität auf Kurznamendaten zugreift.

Beispiel 1: Gesicherter Modus

Das folgende Beispiel zeigt einen Join zwischen einer lokalen Tabelle und einem Kurznamen im gesicherten Modus. Der Server mit föderierten Datenbanken verarbeitet die lokalen Daten in jeder Partition parallel, bevor er diese Daten mit den Kurznamendaten in der Koordinatorpartition verknüpft. Der Server mit föderierten Datenbanken verarbeitet die Kurznamendaten nicht parallel auf den verschiedenen Partitionen oder den Prozessoren der einzelnen Partitionen.

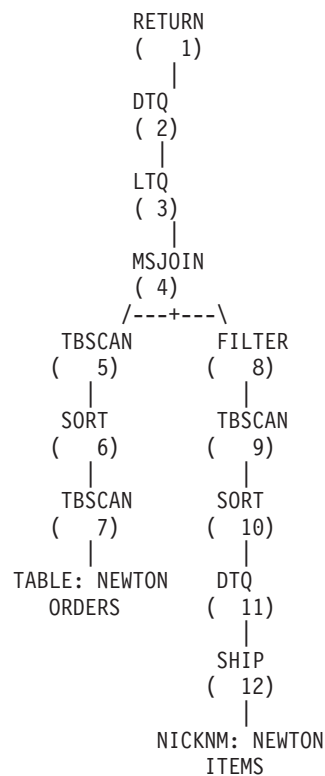
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Beispiel 2: Abgeschirmter Modus

Das folgende Beispiel zeigt einen Join zwischen einer lokalen, partitionierten Tabelle und einem Kurznamen im abgeschirmten Modus. Der Server mit föderierten Datenbanken ruft die Kurznamendaten seriell auf die Koordinatorpartition ab und verteilt diese Daten anschließend an die anderen Partitionen in der Datenbank. Die Daten werden dann parallel mit den lokalen Daten auf allen entsprechenden Datenbankpartitionen verknüpft. In jeder Partition lesen mehrere Subagenten die lokale Tabelle und führen Joins mit den Kurznamendaten aus. Diese Operation wird als partitionsinterne Parallelität bezeichnet und im Plan durch den LTQ-Operator gekennzeichnet. Das Ergebnis wird an die Koordinatorpartition zur abschließenden Verarbeitung gesendet und dann an die Anwendung zurückgegeben.

```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Kapitel 25. Asynchrone Verarbeitung föderierter Abfragen

Asynchronität ist eine Methode zur Verbesserung der Abfrageleistung. Dabei werden mehrere Abschnitte eines Zugriffsplans gleichzeitig ausgeführt, um die für eine bestimmte Abfrage benötigte Zeit zu reduzieren.

In einem föderierten System sind die Daten systemübergreifend auf eine Vielzahl von Datenquellen verteilt, wobei jedes System über eigene Ressourcen verfügt. Bei der Asynchronität überschneiden sich die Operationen, die diese Ressourcen nutzen, sodass mehrere Systeme gleichzeitig aktiv sind. Durch die sich überschneidenden Operationen können mehrere Abschnitte eines Zugriffsplans gleichzeitig anstatt seriell ausgeführt werden.

Komplexe Abfragen, die zeitaufwendige Operationen in fernen Datenquellen umfassen, können von der Asynchronität profitieren. Asynchronität ermöglicht die gleichzeitige Ausführung der folgenden Operationsarten:

- Zwei oder mehr Operationen in fernen Datenquellen
- Operationen auf dem Server mit föderierten Datenbanken und in mindestens einer fernen Datenquelle

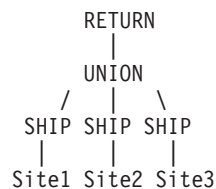
Da Abfrageoperationen Ressourcen belegen, ist die Asynchronität für die Systeme von Vorteil, in denen Ressourcen inaktiv sind oder in denen zu einem bestimmten Zeitpunkt nur eine der Datenquellen bzw. das föderierte System aktiv ist.

Asynchrone Verarbeitung föderierter Abfragen - Beispiele

Beispiele für Abfragen mit einer Union-Verknüpfung und einem Mischjoin veranschaulichen den Unterschied zwischen Abfrageoperationen mit und ohne asynchrone Verarbeitung.

Beispiel: Abfrage mit einer Union-Verknüpfungsoperation

Eine einfache Abfrage führt eine Union-Verknüpfungsoperation für Daten aus drei verschiedenen Datenquellen aus. Die Berechnung, die erforderlich ist, um die Daten in den einzelnen Datenquellen zu generieren, nimmt viel Zeit in Anspruch. Der Zugriffsplan sieht wie folgt aus:

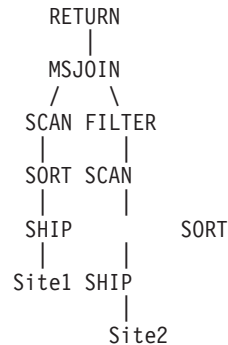


Ohne Asynchronität liest die Union-Verknüpfungsoperation die Daten nacheinander aus den einzelnen Abfrageverzweigungen, wobei sie jeweils eine Abfrageverzweigung nach der anderen von links nach rechts verarbeitet. Während die Daten vom Server 'Site1' gelesen werden, sind die Server 'Site2' und 'Site3' inaktiv. In diesem Beispiel benötigt jede Verzweigung der Union-Verknüpfung etwa zwei Stunden, um die Ergebniszeilen von einer Site zurückzugeben. Die gesamte Ausführungszeit der drei Verzweigungen beträgt ungefähr die Summe der für die Verarbeitung der einzelnen Verzweigungen benötigten Zeit, in diesem Beispiel also etwa sechs Stunden.

Mit Asynchronität wird die Verarbeitung jeder Verzweigung innerhalb der Union-Verknüpfung zur selben Zeit gestartet, und die drei fernen Server sind gleichzeitig aktiv. Die Ausführungszeit der Abfrage entspricht ungefähr der Ausführungszeit der langsamsten Verzweigung innerhalb der Union-Verknüpfung. Im vorliegenden Beispiel wird die Ausführungszeit auf ca. zwei Stunden reduziert (etwa 66% schneller), gegenüber sechs Stunden ohne Asynchronität.

Beispiel: Abfrage mit einer Mischjoinoperation

Eine Abfrage, bei der Daten aus zwei unterschiedlichen Datenquellen verknüpft werden, verwendet eine Mischjoinoperation (MSJOIN). Der Zugriffsplan des Optimierungsprogramms sieht wie folgt aus:



Ohne Asynchronität verarbeitet der Mischjoinoperator zunächst die äußere (linke) Verzweigung; die innere (rechte) Verzweigung wird erst verarbeitet, wenn die linke Verzweigung damit beginnt, Zeilen zurückzugeben. In diesem Beispiel führt jede Verzweigung eine komplexe Abfrage aus, die viel Zeit in Anspruch nimmt. Die ungefähre Gesamtzeit für die Ausführung des Mischjoins ist die Summe der für die einzelnen Verzweigungen benötigten Zeit.

Mit Asynchronität werden beide Verzweigungen des Mischjoins gleichzeitig gestartet, wodurch die Gesamtausführungszeit für die Abfrage reduziert wird.

Asynchronitätsoptimierung

Das Abfrageoptimierungsprogramm trifft Entscheidungen hinsichtlich der asynchronen Verarbeitung von fernen Operationen mithilfe eines Abfrageausführungsplans. Die *Asynchronitätsoptimierung* ist der Prozess, anhand dessen das Optimierungsprogramm einen vorhandenen Abfrageausführungsplan analysiert und die Möglichkeiten zur gleichzeitigen Ausführung ferner Operationen ermittelt.

Zugriffspläne ohne Asynchronität

In einem Ausführungsplan definiert der Operator SHIP bzw. RPD einen Teil des Plans, der in einer fernen Datenquelle ausgeführt wird.

Ohne Asynchronität wird der Operator SHIP bzw. RPD nur dann aktiv und initiiert die ferne Verarbeitung nur dann, wenn die entsprechenden Daten von anderen Operatoren benötigt werden, die sich oberhalb des Operators SHIP bzw. RPD im Ausführungsplan befinden.

Für Asynchronität optimierte Zugriffspläne

Das Optimierungsprogramm kann festlegen, dass ferne Operationen, die durch den SHIP- oder RPD-Operator definiert werden, asynchron ausgeführt werden.

Bei einer asynchronen Operation wird ein Tabellenwarteschlangenoperator (TQ-Operator) unmittelbar oberhalb des SHIP- bzw. RPD-Operators in den Ausführungsplan eingefügt. Der TQ-Operator definiert einen Abschnitt des Plans, den so genannten *Subplan*. Ein separater Prozess, oder Thread, mit eigenem Speicherbereich, führt den Subplan aus. Ein Subplan wird sofort nach dem Starten der Abfrage initiiert.

Der TQ-Operator kann als Pipe zwischen dem SHIP- bzw. RPD-Operator (Datenproduzent) und dem im Plan unmittelbar darüber stehenden Operator (Datenkonsument) betrachtet werden. Diese Pipe entkoppelt die Ausführung des SHIP-Operators im darauf folgenden Subplan vom Hauptplan und ermöglicht so den asynchronen Austausch von Daten zwischen den beiden Abschnitten des Plans.

Ein TQ-Operator, der unmittelbar oberhalb eines SHIP- oder RPD-Operators im Plan steht, ermöglicht es, dass die fernen Operationen, die durch den SHIP- bzw. RPD-Operator definiert werden, zu Beginn der Abfrage initiiert werden und ihre Ergebnisse asynchron an den Server mit föderierten Datenbanken liefern. Wenn Asynchronität für eine bestimmte ferne Operation von Vorteil ist, fügt das Optimierungsprogramm einen TQ-Operator unmittelbar oberhalb des entsprechenden SHIP- bzw. RPD-Operators im Plan ein.

TQ-Operatoren werden in Ausführungsplänen zu unterschiedlichen Zwecken verwendet. Normalerweise kennzeichnet ein TQ-Operator parallele Operationen in partitionierten Datenbanken oder in Datenbanken, für die die partitionsinterne Parallelität aktiviert ist. Ein anderer TQ-Operatortyp, der die asynchrone Ausführung eines Subplans ermöglicht, wird als ATQ-Operator (Asynchrony Table Queue - Asynchrony-Tabellenwarteschlange) bezeichnet.

Das Optimierungsprogramm definiert einen bestimmten SHIP- oder RPD-Operator als asynchron, wenn Folgendes zutrifft:

- Die Abfrageleistung wird verbessert.
- Die Anzahl der ATQs liegt unterhalb der pro Server und pro Abfrage gültigen Grenzwerte.
- Der Operator ist nicht bereits aufgrund eines anderen Optimierungsverfahrens asynchron.
- Es werden keine Einschränkungen für die Asynchronität verletzt.
- Die Semantik der Abfrage wird nicht verändert.

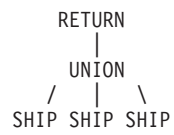
Zugriffspläne - Beispiele

Die Beispiele für Zugriffspläne veranschaulichen den Unterschied zwischen der Planausführung mit Asynchronitätsoptimierung und der Planausführung ohne Asynchronitätsoptimierung.

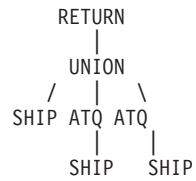
Die ersten beiden Beispiele zeigen, wie die Pläne für Union-Verknüpfungen und Mischjoins in „Asynchrone Verarbeitung föderierter Abfragen - Beispiele“ auf Seite 267 aussehen, wenn die Asynchronität aktiviert ist.

Der Einfachheit halber sind in den Beispielen nur Pläne mit SHIP-Operatoren dargestellt. Ein Plan mit RPD-Operatoren wird durch die Asynchronitätsoptimierung auf dieselbe Weise umgesetzt wie ein Plan mit SHIP-Operatoren. Die SHIP- und RPD-Operatoren sind austauschbar; wo dies nicht der Fall ist, wird explizit darauf hingewiesen.

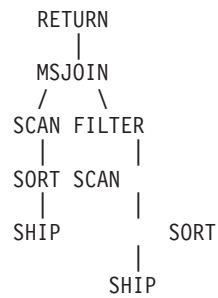
Beispiel 1a: Plan ohne Asynchronität



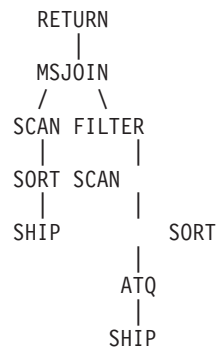
Beispiel 1b: Plan mit Asynchronität



Beispiel 2a: Plan ohne Asynchronität



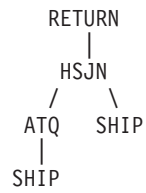
Beispiel 2b: Plan mit Asynchronität



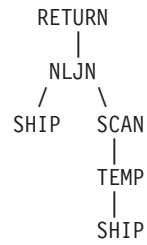
Beispiel 3a: Plan ohne Asynchronität



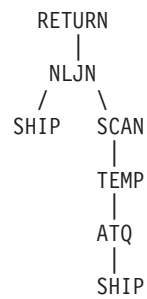
Beispiel 3b: Plan mit Asynchronität



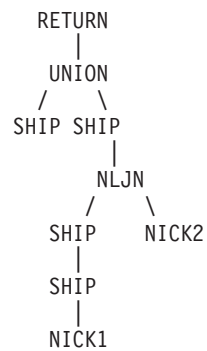
Beispiel 4a: Plan ohne Asynchronität



Beispiel 4b: Plan mit Asynchronität

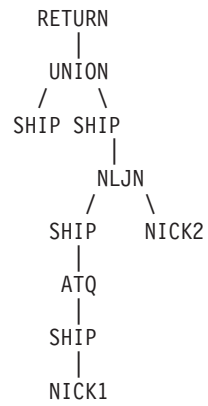


Beispiel 5a: Plan ohne Asynchronität

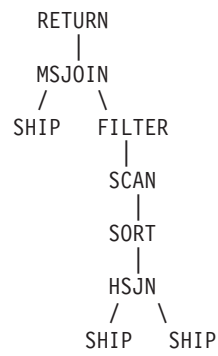


RPDs cannot replace the SHIP-SHIP pair in this plan.

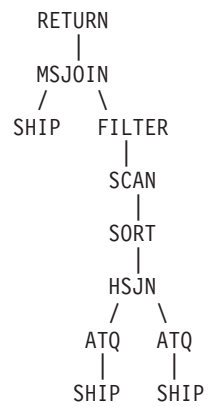
Beispiel 5b: Plan mit Asynchronität



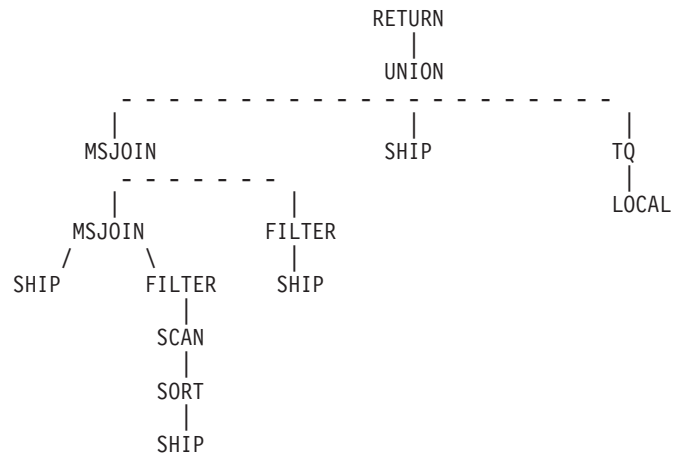
Beispiel 6a: Plan ohne Asynchronität



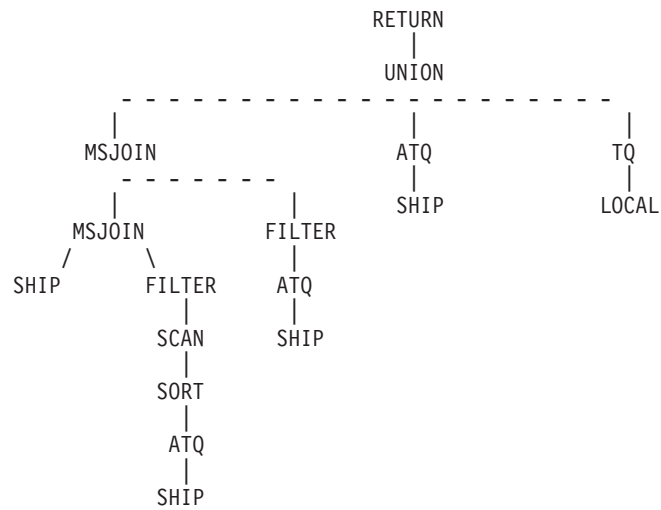
Beispiel 6b: Plan mit Asynchronität



Beispiel 7a: Plan ohne Asynchronität



Beispiel 7b: Plan mit Asynchronität



Überwachen der Ressourcennutzung

Der ATQ-Operator aktiviert nicht nur die asynchrone Ausführung einer fernen Abfrage, sondern hat auch Auswirkungen auf den Server mit föderierten Datenbanken und die fernen Datenquellen.

Da jeder ATQ-Operator einen neuen Prozess erstellt und eine gewisse Menge an Hauptspeicher belegt (als Puffer), kann das Einfügen einer großen Anzahl von ATQs in einen Ausführungsplan bewirken, dass zu viele Systemressourcen auf dem Server mit föderierten Datenbanken belegt werden. Darüber hinaus werden, wenn eine Reihe von SHIP- oder RPD-Operatoren in einer Abfrage auf einer bestimmten fernen Datenquelle ausgeführt und einige der Operatoren als asynchron definiert werden, mehrere Cursor auf dieser Datenquelle gleichzeitig geöffnet, was zu einer nicht akzeptablen Systembelastung führen kann.

Sie können Konfigurationsparameter definieren, um die Ressourcenauslastung im föderierten System oder in den Datenquellen zu steuern. Die Parameter legen Grenzwerte für die Gesamtzahl der ATQs fest, die innerhalb einer Abfrage zulässig sind, sowie für die Anzahl der ATQs, die für die einzelnen Server innerhalb einer Abfrage zulässig sind.

Aktivieren der Asynchronitätsoptimierung

Geben Sie zum Aktivieren der Asynchronitätsoptimierung die Anzahl der ATQ-Operatoren für eine bestimmte Abfrage an, und legen Sie eine Serveroption für die Datenquelle fest.

Einschränkungen

Für die Asynchronitätsoptimierung ist Folgendes erforderlich:

- Ein föderiertes System mit der Datenbankpartitionierungsfunktion (DPF) und mehr als einer logischen Datenbankpartition.
- Zugriff auf Datenquellen über abgeschirmte Wrapper.

Diese Optimierung unterstützt die folgenden Objekte nicht:

- Kurznamen einer Datenquelle, auf die über einen gesicherten Wrapper zugegriffen wird.
- Abfragen mit Einfüge-, Aktualisierungs- oder Löschoperation.

Vorgehensweise

Gehen Sie wie folgt vor, um die asynchrone Verarbeitung zu aktivieren:

1. Definieren Sie einen oder mehrere der folgenden Parameter:
 - Legen Sie für den Konfigurationsparameter `FEDERATED_ASYNC` des Datenbankmanagers einen Wert zwischen 0 und 32767 oder `ANY` fest. `MAXAGENTS` ist ein Datenbankkonfigurationsparameter, der die maximale Anzahl der pro Anfrage zulässigen ATQs angibt. Der Wert `ANY` lässt zu, dass das Optimierungsprogramm die Anzahl der ATQs für einen bestimmten Zugriffsplan festlegt. Der Standardwert ist 0.

- Definieren Sie optional die Binde- und Vorkompilierungsoption `FEDERATED_ASYNC` für die statischen Anweisungen im Paket so, dass die Einstellung des Konfigurationsparameters für die Abfrage überschrieben wird. Der Standardwert ist 0.
- Definieren Sie optional das Sonderregister `CURRENT FEDERATED ASYNCHRONY` so, dass die Einstellung für den Konfigurationsparameter des Datenbankmanagers und die Bindeoption für die Abfrage dynamisch überschrieben werden.

Diese Parameter weisen die folgende Hierarchie auf:

- a. Sonderregister
- b. Bindeoption
- c. Konfigurationsparameter des Datenbankmanagers

Der Wert des Sonderregisters hat, wenn er angegeben wird, Priorität gegenüber der Bindeoption, die ihrerseits Priorität gegenüber dem Konfigurationsparameter des Datenbankmanagers hat.

2. Definieren Sie für die Serveroption `DB2_MAX_ASYNC_REQUESTS_PER_QUERY` einen numerischen Wert. Die Serveroption gibt die maximale Anzahl asynchroner Anforderungen an, die der Server für die Abfrage zulässt.

Der Wertebereich für die Serveroption ist -1 bis 64000.

- Der Standardwert ist 1. Somit wird ein SHIP-Operator oder ein RPD-Operator, der einem Server zugeordnet ist, für die Asynchronität in einer Abfrage berücksichtigt.
- Nur für die ODBC-Datenquelle ist der Standardwert 0. Für die Asynchronität sind mehrere Cursor pro Verbindung erforderlich. Dieser Wert muss für den ODBC-Wrapper 0 sein, es sei denn, die Datenquelle unterstützt mehrere Cursor pro Verbindung.

Optimierungsaspekte für die Asynchronitätsoptimierung

Wenn die Asynchronität aktiviert ist, müssen Sie eine Reihe von Faktoren berücksichtigen, die Auswirkungen auf die Leistung haben.

Wenn auf Ihrem System Prozess-, Speicher- und CPU-Ressourcen verfügbar sind, kann die Aktivierung der Asynchronität die Leistung von föderierten Abfragen verbessern. Darüber hinaus kann durch die Aktivierung der Asynchronität die Nutzung von Fernquellensystemen durch den Server mit föderierten Datenbanken erhöht werden, da eine ferne Quelle potenziell mehrere Anforderungen für eine föderierte Abfrage gleichzeitig verarbeiten kann. Wenn auf Ihrem System ein Ressourcenengpass besteht, kann die Asynchronität zu einer Beeinträchtigung der Leistung führen.

Sie können Ihr System für die Asynchronität optimieren, indem Sie die Konfigurationsparameter so ändern, dass Sie den für Ihr System am besten geeigneten Grad der Asynchronität erreichen.

Jeder ATQ-Operator, den die Asynchronitätsoptimierung in einen Plan einfügt, benötigt einen zusätzlichen Subagenten. Wenn im System eine ausreichende Anzahl von Subagenten vorhanden ist, können Sie den Konfigurationsparameter `MAXAGENTS` des Datenbankmanagers entsprechend anpassen.

Einschränkungen bei der Asynchronitätsoptimierung

Wenn das Optimierungsprogramm die Asynchronitätsoptimierung für eine bestimmte Abfrage anwendet, gelten für die Anzahl der ATQs, die im Abfrageausführungsplan verwendet werden können, bestimmte Einschränkungen.

Die Anzahl der SHIP- bzw. RPD-Operatoren, die in einem Plan für das Koppeln mit einem ATQ in Frage kommen und damit von der Asynchronität profitieren könnten, kann höher sein als der durch den Parameter FEDERATED_ASYNC definierte Höchstwert oder der in der Serveroption DB2_MAX_ASYNC_REQUESTS_PER_QUERY angegebene Grenzwert pro Server. In diesem Fall wählt das Optimierungsprogramm die mit einem ATQ zu koppelnden SHIP- bzw. RPD-Operatoren in der folgenden Weise aus:

- Die Gesamtzahl der ATQs im Plan muss kleiner-gleich dem Wert sein, der für die folgenden Parameter - in der angegebenen Reihenfolge - festgelegt ist:
 1. Sonderregister FEDERATED ASYNCHRONY, falls angegeben
 2. Binde- oder Vorkompilierungsoption FEDERATED_ASYNC, falls angegeben
 3. Parameter FEDERATED_ASYNC
- Die Gesamtzahl der ATQs für einen bestimmten Server muss kleiner-gleich dem Wert der Serveroption DB2_MAX_ASYNC_REQUESTS_PER_QUERY für diesen Server sein.
- Die in Frage kommenden SHIP- bzw. RPD-Operatoren können das Koppeln mit einem ATQ nutzen und auf diese Weise die Abfrageleistung verbessern.

Ermitteln, ob die Asynchronitätsoptimierung für eine Abfrage angewendet wird

Um festzustellen, ob die Asynchronitätsoptimierung für eine bestimmte Abfrage angewendet wird, können Sie eine von mehreren möglichen Methoden zur Überprüfung des Zugriffsplans für die Abfrage sowie zur Überprüfung auf einen bestimmten Operator verwenden.

Sie können die betreffende Abfrage anhand einer der folgenden Ausgaben überprüfen:

- db2exfmt-Ausgabe
- Visual Explain-Ausgabe
- dynexpln-Ausgabe

Jede Ausgabe zeigt asynchrone Abfrageanforderungen im Zugriffsplan als ATQs. Da Merkmal 'origin' in der Beschreibung des ATQ-Operators gibt 'Asynchrone' an.

Das folgende Planfragment zeigt die Verwendung des ATQ-Operators sowie dessen detaillierte Merkmale.

```

                                1.6e+06
                                HSJOIN
                                ( 2)
                                4213.74
                                131
                                /-----+-----\
                                40000 1000
                                HSJOIN SHIP
                                ( 3) ( 10)
                                1122.26 427.733
                                117 14
```



```

          /-----+-----\
1000 1000 1000
ATQ          ATQ          NICKNM: NEWTON
( 4) ( 7) S1_NN07
511.795 532.669
16 101
|
1000 1000
SHIP SHIP
( 5)          ( 8)
478.773 499.887
16 101
|
1000 1000
NICKNM: NEWTON NICKNM: NEWTON
S2_NN02 S3_NN15

```

- show the origin of the ATQ as ASYNCHRONY:

```

4) TQ : (Table Queue)
Cumulative Total Cost: 511.795
Cumulative CPU Cost: 2.79486e+06
Cumulative I/O Cost: 16
Cumulative Re-Total Cost: 68.9489
Cumulative Re-CPU Cost: 1.72372e+06
Cumulative Re-I/O Cost: 0
Cumulative First Row Cost: 30.8308
Cumulative Comm Cost: 18.2176
Cumulative First Comm Cost: 0
Estimated Bufferpool Buffers: 16
Remote communication cost: 538.297

```

Arguments:

```

-----
JN INPUT: (Join input leg)
OUTER
LISTENER: (Listener Table Queue type)
FALSE
TQMERGE : (Merging Table Queue flag)
FALSE
TQORIGIN: (Table Queue Origin type)
ASYNCHRONY
TQREAD : (Table Queue Read type)
READ AHEAD
TQSEND : (Table Queue Write type)
DIRECTED
UNIQUE : (Uniqueness required flag)
FALSE

```

Wenn Sie keine ATQ-Operatoren im Zugriffsplan für eine Abfrage finden, stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind:

- Das System ist ein föderiertes System, das für DPF aktiviert ist.
- Die Abfrage greift auf Kurznamen in einer Datenquelle zu, auf die Zugriff über einen abgeschirmten Wrapper besteht.
- Die Asynchronität ist über den Konfigurationsparameter `FEDERATED_ASYNC` des Datenbankmanagers, das Sonderregister `CURRENT FEDERATED ASYNCHRONY` oder die Bindeoption `FEDERATED_ASYNC` aktiviert.
- Für die Serveroption `DB2_MAX_ASYNC_REQUESTS_PER_CONNECTION` ist ein Wert ungleich null für die Kurznamen auf den einzelnen Servern definiert.

Kapitel 26. Globale Optimierung

Der SQL-Compiler durchläuft drei Phasen bei der Erstellung einer optimalen Zugriffsstrategie für die Auswertung einer auf eine ferne Datenquelle verweisende Abfrage. Bei diesen Phasen handelt es sich um die Pushdown-Analyse, die globale Optimierung und die Generierung von fernem SQL.

Die Zugriffsstrategie für eine an die föderierte Datenbank übergebene Abfrage kann darin bestehen, die ursprüngliche Abfrage in mehrere Abfragefragmente zu zerlegen und dann die Ergebnisse zu kombinieren.

Das Abfrageoptimierungsprogramm verwendet die Ausgabe der Pushdown-Analysephase, um zu entscheiden, wo die Operation ausgewertet wird. Eine Operation kann lokal auf dem Server mit föderierten Datenbanken oder fern in der Datenquelle ausgewertet werden. Diese Entscheidung ist abhängig von der Ausgabe des hoch entwickelten Kostenmodells, das das Optimierungsprogramm verwendet. Dieses Modell ermittelt folgende Aufwände:

- Den Aufwand für die Auswertung der Operation.
- Den Aufwand für die Übertragung der Daten oder Nachrichten zwischen dem Server mit föderierten Datenbanken und den Datenquellen.

Das Ziel der globalen Optimierung ist die Erstellung eines Zugriffsplans, der die Abfrageoperationen für alle Datenquellen im föderierten System global optimiert. Ein global optimierter Zugriffsplan weist die geringsten Gesamtkosten für die Ausführung von Operationen in einem föderierten System auf. In der Phase der Generierung von fernem SQL wird der global optimierte Plan in Abfragefragmente rückübersetzt, die von den individuellen Datenquellen ausgeführt werden.

Der SQL-Compiler verfügt über eine Wissensbasis, die Merkmale der unterstützten Datenquellen sowie Metadaten zu den Daten in diesen Datenquellen enthält. Das Optimierungsprogramm generiert kein SQL, keine Abfragefragmente und keine Planhinweise, die die ferne Datenquelle nicht verstehen oder akzeptieren kann.

Viele Faktoren können die Ausgabe der globalen Optimierung und somit auch die Abfrageleistung beeinflussen. Die Hauptfaktoren sind Servermerkmale und Kurznamenmerkmale.

Zwischen relationalen und nicht relationalen Wrappern bestehen Unterschiede hinsichtlich der Details der Erstellung eines Zugriffsplans. Das Konzept und das Endergebnis sind jedoch identisch.

Servermerkmale mit Auswirkungen auf die globale Optimierung

Beim Erstellen oder Ändern einer Serverdefinition können einige der ausgewählten Optionen Auswirkungen auf die Abfrageleistung haben.

Das Abfrageoptimierungsprogramm erhält Informationen über die Servermerkmale der Datenquelle. Die Einstellungen für die Serveroptionen sind ein Teil der Serverdefinition der Datenquelle. Sie können bei der anfänglichen Erstellung der Serverdefinition die Einstellungen für die Serveroptionen in der Anweisung `CREATE SERVER` festlegen. Verwenden Sie die Anweisung `ALTER SERVER`, um zu einer

vorhandenen Serverdefinition Serveroptionen hinzuzufügen. Die Einstellungen für die Serveroptionen werden im Katalog der föderierten Datenbank gespeichert.

Diese Optionen sind in Optionen für die Speicherposition (z. B. für den Rechnernamen der Datenquelle), Sicherheitsoptionen (z. B. Authentifizierungsinformationen) und Leistungsoptionen (z. B. das CPU-Verhältnis) unterteilt.

Mithilfe der Leistungsoptionen kann das Optimierungsprogramm ermitteln, ob die Auswertung einer Operation in einer Datenquelle erfolgen kann und ob die Auswertung einer Operation in der Datenquelle die Ausführung beschleunigt. Folgende Serveroptionen beeinflussen die Leistung, sodass eventuell eine Optimierung erforderlich ist:

- CPU_RATIO
- IO_RATIO
- COMM_RATE
- COLLATING_SEQUENCE
- PLAN_HINTS
- VARCHAR_NO_TRAILING_BLANKS
- NO_EMPTY_STRING

Gehen Sie bei der Optimierung der Serveroption CPU_RATIO, IO_RATIO oder COMM_RATE behutsam vor, da unerwartete Fehler auftreten können, wenn die Aufwandskalkulation für eine Abfrage Über- bzw. Unterläufe erzeugt. In den meisten Fällen sind die Standardwerte für diese Optionen ausreichend. Normalerweise ist es wichtiger, sicherzustellen, dass die Statistikdaten zu den Objekten, auf die in Ihren Abfragen verwiesen wird, korrekt sind, als zu versuchen, die Werte dieser Serveroptionen zu optimieren.

Relatives Verhältnis der CPU-Geschwindigkeit

Dieser Wert gibt das Verhältnis zwischen der CPU-Geschwindigkeit des Servers mit föderierten Datenbanken und der CPU-Geschwindigkeit des Servers mit der fernen Datenquelle an.

Dieser Wert wird als CPU-Geschwindigkeit des Servers mit föderierten Datenbanken dividiert durch die CPU-Geschwindigkeit des Servers für die ferne Datenquelle definiert. Beispiel: Wenn die CPU-Geschwindigkeit des Servers mit föderierten Datenbanken das Doppelte der CPU-Geschwindigkeit des fernen Servers beträgt, muss für CPU_RATIO der Wert 2 angegeben werden. Wenn die CPU-Geschwindigkeit des Servers mit föderierten Datenbanken nur ein Drittel der CPU-Geschwindigkeit für den fernen Server beträgt, muss für CPU_RATIO der Wert 0,33 angegeben werden.

Wenn Sie die Serveroption für das CPU-Verhältnis nicht explizit definieren, verwendet das Optimierungsprogramm des föderierten Systems den Standardwert 1; dieser Wert gibt an, dass die CPU-Geschwindigkeit des Servers mit föderierten Datenbanken und die CPU-Geschwindigkeit der Datenquelle gleich sind.

Ein niedriger Wert für das Verhältnis gibt an, dass die CPU-Geschwindigkeit des Datenquellenservers höher ist als die des Servers mit föderierten Datenbanken. Bei einem niedrigen Verhältnis versucht das Optimierungsprogramm möglicherweise, CPU-intensive Verarbeitungsoperationen im Pushdown-Modus an die Datenquelle weiterzuleiten. Ein niedriges Verhältnis ist ein Wert kleiner als 1.

Relatives Verhältnis der E/A-Geschwindigkeit

Dieser Wert gibt das Verhältnis zwischen der E/A-Geschwindigkeit des Servers mit föderierten Datenbanken und der E/A-Geschwindigkeit des Servers mit der fernen Datenquelle an.

Dieser Wert wird als E/A-Geschwindigkeit des Servers mit föderierten Datenbanken dividiert durch die E/A-Geschwindigkeit des fernen Servers definiert. Beispiel: Wenn die E/A-Geschwindigkeit des Servers mit föderierten Datenbanken das Doppelte der E/A-Geschwindigkeit des fernen Servers beträgt, muss für IO_RATIO der Wert 2 angegeben werden. Wenn die E/A-Geschwindigkeit des Servers mit föderierten Datenbanken dagegen die Hälfte der E/A-Geschwindigkeit für den fernen Server beträgt, muss für IO_RATIO der Wert 0,5 angegeben werden.

Wenn Sie die Serveroption für das E/A-Verhältnis nicht explizit definieren, verwendet das Optimierungsprogramm des föderierten Systems den Standardwert 1; dieser Wert gibt an, dass die E/A-Geschwindigkeit des Servers mit föderierten Datenbanken und die E/A-Geschwindigkeit des fernen Servers gleich sind.

Ein Wert kleiner als 1 für IO_RATIO gibt an, dass der ferne Server eine höhere E/A-Geschwindigkeit hat als der Server mit föderierten Datenbanken. In diesem Fall wird das Optimierungsprogramm E/A-intensive Operationen wahrscheinlich im Pushdown-Modus an die ferne Datenquelle weiterleiten. Ein niedriges Verhältnis ist ein Wert kleiner als 1.

Übertragungsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und der Datenquelle

Eine niedrige Übertragungsgeschwindigkeit gibt eine langsame Netzwerkkommunikation zwischen dem Server mit föderierten Datenbanken und der Datenquelle an.

Die Einstellung der Serveroption COMM_RATE bestimmt die Übertragungsgeschwindigkeit. COMM_RATE gibt die Geschwindigkeit der Netzwerkverbindung zwischen dem Datenquellenserver und dem Server mit föderierten Datenbanken an. Die Geschwindigkeit wird in Megabyte pro Sekunde gemessen. Der Standardwert ist 2 MB/s.

Bei niedrigen Übertragungsgeschwindigkeiten reduziert das Abfrageoptimierungsprogramm die Anzahl der Nachrichten, die an die bzw. von der betreffenden Datenquelle gesendet werden. Ist die Serveroption COMM_RATE sehr niedrig eingestellt, erstellt das Optimierungsprogramm eine Abfrage, die minimalen Netzwerkverkehr erfordert.

Sortierfolge der Datenquelle

Die ausgewählte Sortierfolge kann Auswirkungen auf die Leistung der föderierten Datenbank haben. Sie können die Serveroption COLLATING_SEQUENCE verwenden, um anzugeben, ob die Sortierfolge einer Datenquelle mit der Sortierfolge der lokalen föderierten Datenbank übereinstimmt.

Der Server mit föderierten Datenbanken kann reihenfolgeabhängige Verarbeitungsvorgänge, in denen Zeichendaten verarbeitet werden, im Pushdown-Modus an die Datenquelle weiterleiten, falls die Serveroption COLLATING_SEQUENCE angibt, dass die Sortierfolge der Datenquelle und die Sortierfolge der föderierten Datenbank übereinstimmen. Stimmen die Sortierfolgen der Datenquelle und der föderierten Datenbank nicht überein, betrachtet das Optimierungsprogramm die von dieser

Datenquelle abgerufenen Daten als unsortiert. Die föderierte Datenbank ruft die relevanten Daten ab und führt alle reihenfolgeabhängigen Verarbeitungsvorgänge für Zeichendaten lokal aus; hierdurch kann die Abfrage verlangsamt und die Leistung beeinträchtigt werden.

Ferne Planhinweise

Planhinweise sind Anweisungsfragmente, die Optimierungsprogrammen von Datenquellen zusätzliche Informationen zur Verfügung stellen.

Verwenden Sie die Serveroption `PLAN_HINTS`, um ferne Planhinweise zu generieren. Bei bestimmten Abfragetypen kann mithilfe dieser Informationen die Abfrageleistung verbessert werden. Die Planhinweise können das Optimierungsprogramm der Datenquelle bei der Entscheidung unterstützen, ob ein Index, und wenn ja, welcher, verwendet werden soll oder nach welcher Reihenfolge beim Join von Tabellen vorzugehen ist.

Sie sollten einige Tests durchführen, um festzustellen, ob diese Serveroption die Leistung Ihrer Abfragen verbessert.

Sie können in einer Abfrage keine eigenen Planhinweise codieren.

Ist die Funktion für Planhinweise aktiviert, enthält die an die Datenquelle gesendete Abfrage zusätzliche Informationen. Eine Anweisung, die an ein Oracle-Optimierungsprogramm mit Planhinweisen gesendet wird, könnte z. B. wie folgt aussehen:

```
SELECT /*+ INDEX (table1, t1index)*/  
  col1  
FROM table1
```

Der Planhinweis ist die Zeichenfolge `/*+ INDEX (table1, t1index)*/`

Kurznamenmerkmale mit Auswirkungen auf die globale Optimierung

Es gibt verschiedene kurznamenspezifische Faktoren, die die globale Optimierung beeinflussen können. Dazu gehören die Indexinformationen und die Statistikdaten des globalen Katalogs.

Die Indexinformationen und die Statistikdaten des globalen Katalogs, die dem SQL-Compiler zur Verfügung stehen, müssen aktuell sein.

Indexspezifikationen

Der SQL-Compiler verwendet zur Optimierung von Abfragen Indexinformationen.

Eine Datenquellentabelle erhält nur dann Indexinformationen, wenn der Kurzname für diese Tabelle erstellt wird. Nach der Erstellung des Kurznamens werden alle am Index für diese Datenquellentabelle vorgenommenen Änderungen nicht auf dem Server mit föderierten Datenbanken aktualisiert. Wenn sich die ferneren Indexinformationen ändern, können Sie die auf dem Server mit föderierten Datenbanken gespeicherten Indexinformationen ändern, indem Sie den Kurznamen für die Tabelle löschen und erneut erstellen. Wenn ein neuer Index für die Datenquellentabelle hinzugefügt wurde, können Sie alternativ auf dem Server mit föderierten Datenbanken eine Indexspezifikation für den Kurznamen definieren.

Für Kurznamen von Objekten, die keine Indizes haben (wie z. B. Sichten, Synonyme oder Objekte von nicht relationalen Datenquellen), werden keine Indexinformationen erfasst.

Wenn ein Objekt, für das ein Kurzname definiert ist, nicht über einen Index verfügt, können Sie eine Indexspezifikation dafür definieren. Indexspezifikationen bilden eine Indexdefinition im globalen Katalog. Die Indexspezifikation selbst ist kein Index. Verwenden Sie die Anweisung `CREATE INDEX` mit der Klausel `SPECIFICATION ONLY`, um eine Indexspezifikation zu erstellen. Die Syntax zum Erstellen einer Indexspezifikation für einen Kurznamen ist ähnlich wie die Syntax zum Erstellen einer Indexspezifikation für eine lokale Tabelle.

Ziehen Sie in folgenden Fällen das Erstellen von Indexspezifikationen in Betracht:

- Eine Tabelle erhält einen neuen Index.
- Sie erstellen einen Kurznamen für ein Datenquellenobjekt, das keine Indizes enthält, wie z. B. eine Sicht oder ein Synonym.

Wenn Sie eine Indexspezifikation (`SPECIFICATION ONLY`) für einen Kurznamen erstellen und angeben, dass der Index eindeutig ist, überprüft die föderierte Datenbank nicht, ob die Spaltenwerte in der fernen Tabelle eindeutig sind. Wenn die fernen Spaltenwerte nicht eindeutig sind, geben Abfragen auf den Kurznamen, die diese Indexspalte enthalten, möglicherweise falsche Daten zurück oder verursachen Fehler.

Bevor Sie Anweisungen des Typs `CREATE INDEX...SPECIFICATION ONLY` für einen Kurznamen für eine Datenquellsicht eingeben, sollten Sie sich über Ihre Anforderungen bewusst sein:

- Handelt es sich bei der fernen Sicht um eine einfache Anweisung `SELECT` für eine Datenquellentabelle mit Index, kann das Erstellen einer Indexspezifikation für den Kurznamen, die mit dem Index für die Datenquellentabelle übereinstimmt, die Abfrageleistung erheblich verbessern.
- Wenn eine Indexspezifikation für eine ferne Sicht erstellt wird, die nicht aus einer einfachen `SELECT`-Anweisung besteht (z. B. eine durch den Join zweier Tabellen erstellte Sicht), wird die Abfrageleistung unter Umständen beeinträchtigt.

Angenommen, es wird eine Indexspezifikation für eine ferne Sicht erstellt, die einen Join zweier Tabellen ist. Das Optimierungsprogramm kann diese Sicht als inneres Element eines Joins mit Verschachtelungsschleife auswählen. Die Abfrage verfügt möglicherweise über ein schlechtes Leistungsverhalten, da der Join mehrmals ausgewertet wird. Eine alternative Methode besteht darin, für jede Tabelle, auf die in der Sicht der Datenquelle verwiesen wird, einen Kurznamen zu erstellen und dann eine Sicht für föderierte Datenbanken zu erstellen, die auf beide Kurznamen verweist.

Statistikdaten des globalen Katalogs

Der globale Katalog auf dem Server mit föderierten Datenbanken enthält statistische Informationen, die zur Abfrageoptimierung verwendet werden.

Der Server mit föderierten Datenbanken benötigt die Statistikdaten für Datenquellenobjekte, für die Kurznamen definiert wurden, um Abfragen zu optimieren, in denen diese Kurznamen verwendet werden. Diese Statistiken werden aus der Datenquelle abgerufen, wenn Sie unter Verwendung der Anweisung `CREATE NICKNAME` für ein Datenquellenobjekt einen Kurznamen erstellen. Die föderierte

Datenbank überprüft die Existenz des Objekts in der Datenquelle und versucht dann, statistische Daten zu bestehenden Datenquellen zu sammeln. Informationen, die für das Optimierungsprogramm nützlich sind, werden aus den Katalogen der Datenquellen gelesen und in den globalen Katalog auf dem Server mit föderierten Datenbanken aufgenommen. Da einige oder alle der Kataloginformationen zu Datenquellen u. U. vom Optimierungsprogramm verwendet werden können, empfiehlt es sich, Statistiken (unter Verwendung des Datenquellenbefehls, der RUNSTATS entspricht) in der Datenquelle zu aktualisieren, bevor ein Kurzname erstellt wird.

Die Katalogstatistik beschreibt die Gesamtgröße der Tabellen und Sichten sowie den Wertebereich in zugeordneten Spalten. Die abgerufenen Informationen umfassen unter anderem Folgendes:

- Die Anzahl Zeilen in einem Kurznamenobjekt
- Die Anzahl Seiten, die ein Kurzname belegt
- Die Anzahl eindeutiger Werte in jeder Spalte einer Tabelle
- Die Anzahl eindeutiger Werte in Spalten eines Index
- Die höchsten/niedrigsten Werte einer Spalte

Die föderierte Datenbank kann zwar die statistischen Daten einer Datenquelle abrufen, sie kann jedoch Aktualisierungen von vorhandenen statistischen Daten in Datenquellen nicht automatisch ermitteln. Außerdem verfügt die föderierte Datenbank über keinen Mechanismus zur Bearbeitung von Objektdefinitionen oder zur Ausführung von strukturellen Änderungen der Objekte in der Datenquelle (wie z. B. beim Hinzufügen einer Spalte zu einer Tabelle).

Wenn sich die statistischen Daten oder die strukturellen Merkmale eines fernen Objekts, für das ein Kurzname definiert ist, ändern, haben Sie die folgenden Auswahlmöglichkeiten für die Aktualisierung der Statistik:

- Manuelle Statistikerfassung
 - Führen Sie das RUNSTATS entsprechende Programm in der Datenquelle aus. Löschen Sie anschließend den aktuellen Kurznamen, und erstellen Sie den Kurznamen erneut. Dies ist die empfohlene Methode zur Aktualisierung von Statistikdaten.

Der Vorteil dieser Methode ist, dass zusätzlich zu den aktualisierten Statistikdaten alle Informationen zu neuen Indizes oder strukturelle Änderungen des fernen Objekts im neuen Kurznamen reflektiert sind. Ein Nachteil dieser Methode ist, dass alle neuen Sichten oder Pakete, die auf dem alten Kurznamen basieren, ungültig werden.
 - Verwenden Sie die Aktualisierungsfunktion für Kurznamenstatistiken in der DB2-Steuerzentrale. Alternativ dazu können Sie auch die zugrunde liegende gespeicherte Prozedur SYSPROC.NNSTAT() verwenden, die über den Befehlszeilenprozessor verfügbar ist.

Die Aktualisierungsfunktion für Kurznamenstatistiken (oder SYSPROC.NNSTAT()) aktualisiert nur die Kurznamenstatistik; sie ändert den Kurznamen nicht, um eventuelle strukturelle Änderungen des fernen Objekts zu reflektieren. Wenn beispielsweise das ferne Objekt über eine neue Spalte verfügt, fügt die Aktualisierungsfunktion für Kurznamenstatistiken keine Spalte zum Kurznamen hinzu.
 - Aktualisieren Sie die Statistik in der Katalogsicht SYSSTAT.TABLES manuell. Verwenden Sie diese Methode nur dann, wenn Sie wissen, dass die Statistikdaten in der fernen Datenquelle falsch oder unvollständig sind.
- Automatische Statistikerfassung

Diese Funktion wird standardmäßig ausgeführt, um die Leistung zu verbessern, indem aktuelle Tabellen- und Kurznamenstatistikdaten automatisch erfasst werden.

Aktualisieren von Zeilenänderungen

Wenn in einem Objekt in der Datenquelle eine große Anzahl Zeilen hinzugefügt oder gelöscht wurden, ist die föderierte Datenbank nicht über diese Änderungen informiert, da in der Katalogstatistik für den Kurznamen weiterhin die ursprüngliche Anzahl von Zeilen angegeben ist.

Es kann jedoch zu einer Verschlechterung der Leistung kommen, da das Optimierungsprogramm seine Entscheidungen weiterhin auf der Basis der Kurznamenstatistikdaten trifft, die nicht mehr korrekt sind. Sie können nach der Aktualisierung der Statistikdaten für das ferne Objekt in der Datenquelle die entsprechenden Statistikdaten für den Kurznamen aktualisieren, um sicherzustellen, dass das Optimierungsprogramm bei der Generierung und Auswahl von Zugriffsplänen zur Verarbeitung von Abfragen in der Datenquelle präzise Statistikdaten verwenden kann.

Aktualisierung von Statistikdaten bei Spaltenänderungen

Wenn an einem Datenquellenobjekt strukturelle Änderungen vorgenommen werden, z. B. wenn eine Spalte zu einer Tabelle hinzugefügt wird, müssen Sie eine Reihe von Schritten ausführen, um die Statistikdaten für das betreffende Objekt im Katalog der föderierten Datenbank zu aktualisieren.

Informationen zu dieser Task

Werden in der Datenquelle Spalten hinzugefügt, gelöscht oder geändert, erhalten Sie unter Umständen fehlerhafte Ergebnisse oder eine Fehlernachricht. Nehmen Sie beispielsweise an, dass der Kurzname *EUROSALES* auf die Tabelle *europa* in einer Sybase-Datenbank verweist. Wenn eine neue Spalte mit der Bezeichnung *CZECH* zur Tabelle hinzugefügt wird, wird diese von der föderierten Datenbank nicht erkannt. Für Abfragen, die auf diese Spalte verweisen, wird eine Fehlernachricht ausgegeben.

Vorgehensweise

Gehen Sie wie folgt vor, um beim Auftreten von Spaltenänderungen die Statistikdaten für das betreffende Objekt zu aktualisieren:

1. Führen Sie in der Datenquelle das Dienstprogramm aus, das dem DB2-Dienstprogramm *RUNSTATS* entspricht. Dadurch werden die im Katalog der Datenquelle gespeicherten Statistikdaten aktualisiert.
2. Löschen Sie den aktuellen Kurznamen für das Datenquellenobjekt mit der Anweisung *DROP NICKNAME*.
3. Erstellen Sie den Kurznamen mit der Anweisung *CREATE NICKNAME* neu.

Analysieren der globalen Optimierung

Detaillierte Informationen zu Zugriffsplänen, einschließlich bestimmter Informationen, die der Global Optimizer für die Auswahl des optimalen Plans verwendet, werden in *EXPLAIN*-Tabellen separat vom eigentlichen Plan gespeichert.

Mithilfe dieser Informationen ist eine intensive Analyse eines Zugriffsplans möglich. Auf die *EXPLAIN*-Tabellen kann von allen unterstützten Betriebssystemen

zugegriffen werden. Sie enthalten Informationen über statische und dynamische SQL-Anweisungen. Sie können unter Verwendung von SQL-Anweisungen auf die EXPLAIN-Tabellen zugreifen. Damit werden eine einfache Bearbeitung der Ausgabe, der Vergleich verschiedener Abfragen und Vergleiche derselben Abfrage über einen bestimmten Zeitraum hinweg gewährleistet.

Es gibt mehrere Möglichkeiten, Informationen zum globalen Zugriffsplan aus den EXPLAIN-Tabellen abzurufen:

- Sie können das Tool db2exfmt für das EXPLAIN-Tabellenformat verwenden, um die Informationen aus EXPLAIN-Tabellen in einem vordefinierten Format anzuzeigen.
- Sie können auch die in die DB2-Steuerzentrale integrierte EXPLAIN-Funktion zusammen mit Visual Explain verwenden, um den für eine bestimmte SQL-Anweisung ausgewählten Zugriffsplan zu verstehen.

Mit EXPLAIN können sowohl dynamische als auch statische SQL-Anweisungen bearbeitet werden. Ein Unterschied zwischen Visual Explain und db2exfmt besteht darin, dass Visual Explain die Informationen in einem grafischen Format darstellt, db2exfmt dagegen in einem Textformat. Die beiden Methoden bieten denselben Detaillierungsgrad.

- Sie können die Tools db2expln und dynexpln verwenden, um den Zugriffsplan zu verstehen, der für eine bestimmte SQL-Anweisung ausgewählt wurde.

Um die Ausgabe von db2exfmt, Visual Explain, db2expln oder dynexpln im Detail zu verstehen, benötigen Sie Kenntnisse zu Folgendem:

- Den unterschiedlichen, unterstützten SQL-Anweisungen und der zu diesen Anweisungen gehörenden Terminologie (z. B. Vergleichselemente in einer Anweisung SELECT)
- Dem Zweck eines Pakets (Zugriffsplan)
- Dem Zweck und Inhalt der Systemkatalogtabellen
- Zu grundlegenden Abfrageverarbeitungsoperatoren, wie z. B. Joins, Gruppierungen, Spaltenberechnungen und Sortierungen.

Durch den Zugriffsplan bestimmte Möglichkeiten der Optimierung

In diesem Abschnitt werden typische Fragen zur Optimierung sowie zu untersuchende Bereiche zur Verbesserung der Leistung aufgeführt.

Warum wird ein Join zwischen zwei Kurznamen derselben Datenquelle nicht fern ausgewertet?

Sie können die Elemente der Joinoperation, die Joinvergleichselemente und die Anzahl der Zeilen im Ergebnis überprüfen, um zu ermitteln, warum ein Join zwischen zwei Kurznamen derselben Datenquelle nicht fern ausgewertet wird.

Zu den zu untersuchenden Bereichen gehören:

- Joinoperationen. Kann die Datenquelle einen Join unterstützen?
- Joinvergleichselemente. Kann das Joinvergleichselement in der fernen Datenquelle ausgewertet werden?
- Anzahl der Zeilen im Joinergebnis. Die Anzahl der Zeilen kann mit Visual Explain ermittelt werden. Ist die durch den Join erzeugte Zeilenmenge sehr viel größer als die Kombination der beiden Kurznamen? Sind die Zahlen realistisch? Wenn dies zu verneinen ist, sollten Sie die Kurznamenstatistik mithilfe der gespeicherten Prozedur SYSPROC.NNSTAT() aktualisieren.

Warum wird der Operator GROUP BY nicht fern ausgewertet?

Zu den zu untersuchenden Bereichen gehören:

- Syntax des Operators. Überprüfen Sie, ob der Operator in der fernen Datenquelle ausgewertet werden kann.
- Anzahl der Zeilen. Überprüfen Sie unter Verwendung von Visual Explain die in der Ein- und Ausgabe des Operators GROUP BY geschätzte Anzahl der Zeilen. Liegen diese beiden Zahlen nah beieinander? Wenn ja, wertet das Optimierungsprogramm diesen Operator GROUP BY aus Gründen der Effizienz eventuell lokal aus. Sind diese Zahlen realistisch? Wenn dies zu verneinen ist, sollten Sie die Kurznamenstatistik mithilfe der gespeicherten Prozedur SYSPROC.NNS-TAT() aktualisieren.

Warum wird die Anweisung nicht vollständig fern ausgewertet?

Der Server mit föderierten Datenbanken versucht, sicherzustellen, dass die Abfragesemantik und die für föderierte Abfragen erzielten Ergebnisse exakt so aussehen, als seien sie durch DB2 Database für Linux, UNIX und Windows ausgewertet worden. In der Pushdown-Analysephase des Abfragecompilers wird entschieden, ob bei der Pushdown-Weiterleitung der Verarbeitung an ferne Quellen die DB2-Semantik beibehalten wird. Föderierte Abfrageoperationen können nur dann problemlos im Pushdown-Verfahren weitergeleitet werden, wenn die entsprechenden Operationen in der fernen Quelle dieselbe Bedeutung und dasselbe Ergebnis haben. Der häufigste Grund dafür, dass die Verarbeitung einer Abfrage nicht vollständig im Pushdown-Verfahren an eine einzelne ferne Quelle weitergeleitet wird, ist, dass für eine oder mehrere Operationen innerhalb der Abfrage geringe Funktionalitätsunterschiede zwischen dem Server mit föderierten Datenbanken und der fernen Quelle bestehen.

Das Optimierungsprogramm führt eine aufwandsabhängige Optimierung durch. Auch wenn die Pushdown-Analyse angibt, dass alle Operatoren in der fernen Datenquelle ausgewertet werden können, nimmt das Optimierungsprogramm seine Aufwandsschätzung als Basis für die Generierung eines global optimalen Zugriffsplans. Zur Entscheidung für einen bestimmten Plan trägt eine Vielzahl von Faktoren bei. Angenommen, die ferne Datenquelle kann alle Operationen in der ursprünglichen Abfrage verarbeiten. Die CPU-Geschwindigkeit der fernen Datenquelle ist jedoch sehr viel niedriger als die des Servers mit föderierten Datenbanken. In diesem Fall kann es von Vorteil sein, die Verarbeitungen stattdessen auf dem Server mit föderierten Datenbanken auszuführen. Wird die gewünschte Leistung nicht erreicht, überprüfen Sie die Serverstatistik in der Katalogtabelle SYSSTAT.SERVEROPTIONS.

Warum verfügt ein vom Optimierungsprogramm generierter und vollständig fern ausgewerteter Zugriffsplan über eine schlechtere Leistung als die ursprüngliche, direkt in der fernen Datenquelle ausgeführte Abfrage?

Zu den zu untersuchenden Bereichen gehören:

- Die vom Abfrageoptimierungsprogramm generierte ferne SQL-Anweisung. Neben dem Ersetzen von Kurznamen durch die entsprechenden Namen der fernen Tabellen weist die generierte SQL-Anweisung in der Regel die folgenden Unterschiede gegenüber der ursprünglichen föderierten Anweisung auf:
 - Die Reihenfolge der Vergleichselemente in der Abfrage wurde möglicherweise geändert.

- Vergleichselemente in der ursprünglichen Abfrage wurden möglicherweise entfernt, durch äquivalente Vergleichselemente ersetzt oder durch zusätzliche Vergleichselemente ergänzt.
- Unterabfragen wurden möglicherweise als Joins neu geschrieben.
- Zusätzliche Funktionen, die Konvertierungen durchführen oder Zeichenfolgen abschneiden, wurden möglicherweise hinzugefügt, um die DB2-Semantik beizubehalten.

Mit Ausnahme des zuletzt aufgeführten Punkts haben diese Änderungen normalerweise positive Auswirkungen auf die Leistung. In einigen Fällen können die Änderungen jedoch bewirken, dass das ferne Abfrageoptimierungsprogramm einen anderen (langsameren) Plan generiert, als dies für die ursprüngliche Abfrage der Fall gewesen wäre.

Ein gutes Abfrageoptimierungsprogramm sollte unabhängig von der Sortierfolge des Vergleichselements einer Abfrage arbeiten. Nicht alle DBMS-Optimierungsprogramme sind jedoch identisch. Das Optimierungsprogramm der fernen Datenquelle generiert auf der Basis der Sortierfolge der Vergleichselemente für die Eingabe einen anderen Zugriffsplan. Trifft dies zu, ist dieses Problem dem fernen Optimierungsprogramm zuzuordnen. Sie können in diesem Fall entweder die Reihenfolge der Vergleichselemente ändern oder die Serviceorganisation der fernen Datenquelle um Unterstützung bitten.

Überprüfen Sie auch die Ersetzung von Vergleichselementen. Ein gutes Abfrageoptimierungsprogramm sollte unabhängig von äquivalenten Ersetzungen von Vergleichselementen arbeiten. Möglicherweise generiert das Optimierungsprogramm der fernen Datenquelle auf der Basis der Vergleichselemente für die Eingabe einen anderen Zugriffsplan. Einige Optimierungsprogramme können beispielsweise keine transitiven Schließungsanweisungen für Vergleichselemente generieren.

- Anzahl der zurückgegebenen Zeilen. Sie können diese Zahl von Visual Explain abrufen. Gibt die Abfrage eine große Zeilenanzahl zurück, kann im Netzwerkverkehr ein Engpass auftreten.
- Zusätzliche Funktionen. Enthält die ferne SQL-Anweisung im Vergleich zur ursprünglichen Abfrage zusätzliche Funktionen? Einige dieser Zusatzfunktionen können zum Konvertieren von Datentypen generiert werden. Stellen Sie sicher, dass diese Funktionen erforderlich sind.

Kapitel 27. Systemmonitorelemente, die die Leistung beeinflussen

Der Monitor für Systeme föderierter Datenbanken stellt statistische Informationen zum aktuellen Status des Datenbankmanagers, Aktivitätsinformationen, wie z. B. Zähler, und andere Maße der Datenbankverarbeitung zusammen.

In einem föderierten System können Sie den Datenbanksystemmonitor dazu verwenden, Informationen zur Datenbankaktivität, zur Systemleistung und zur Anwendungsleistung zusammenzustellen.

Der Zeitmarkenmonitorschalter wird verwendet, um die Antwortzeiten der Interaktionen zwischen der föderierten Datenbank und einer Datenquelle zu verfolgen. Die folgenden föderierten Datenelemente werden durch den Zeitmarkenschalter verfolgt:

- Antwortzeit zum Erstellen eines Kurznamens
- Antwortzeit zum Löschen
- Antwortzeit zum Einfügen
- Zeit für einen Durchgriff
- Antwortzeit für Abfragen
- Zeit für ferne Sperren
- Zeit für gespeicherte Prozeduren
- Antwortzeit zum Aktualisieren

Die Standardeinstellung für den Zeitmarkenmonitorschalter ist ON.

Empfehlung: Sie können die Leistung erhöhen, indem Sie die Einstellung für den Zeitmarkenmonitorschalter für alle Anwendungen in OFF ändern. Wenn der Zeitmarkenschalter für eine Anwendung auf ON gesetzt ist, fährt das System mit dem Erfassen der Antwortzeiten fort. Sie verbessern die Leistung deshalb nicht, wenn Sie den Zeitmarkenschalter nur für einige Ihrer Anwendungen ausschalten.

Das Ausschalten des Schalters hat andere Auswirkungen.

- Wenn Sie den Zeitmarkenmonitorschalter für alle Anwendungen ausschalten, müssen Sie die DB2-Instanz stoppen und erneut starten, um die Änderung zu implementieren.
- Das Ausschalten des Zeitmarkenmonitorschalters inaktiviert das Zusammenstellen von Zeitmarkeninformationen für Anwendungen mit und ohne föderierte Datenbanken. Auch die lokale Datenbank empfängt keine Zeitmarkeninformationen.

Wenn Sie Zeitmarkeninformationen für lokale Anwendungen ohne föderierte Datenbanken benötigen, sollten Sie den Zeitmarkenmonitorschalter nicht ausschalten.

Sie können den Zeitmarkenschalter für alle Anwendungen auf OFF setzen, indem Sie den folgenden Befehl verwenden:

```
update dbm cfg using dft_mon_timestamp off
```

Setzen Sie danach folgende Befehle ab:

```
db2stop  
db2start
```

Durch das Stoppen und Starten des Server mit föderierten Datenbanken wird sichergestellt, dass der Schalter für alle Anwendungen ausgeschaltet ist.

Individuelle Informationen zu den Elementen, die durch den Zeitmarkenschalter protokolliert werden, werden in einem separaten Abschnitt besprochen.

Kapitel 28. MQTs (Materialized Query Tables)

Eine MQT (Materialized Query Table, gespeicherte Abfragetabelle) ist eine Tabelle, die die Ergebnisse einer Abfrage im Cache speichert. Wenn Sie die Abfrage erneut übergeben, kann die Datenbanksteuerkomponente die Daten aus der MQT zurückgeben. Sie können MQTs mit Kurznamen verwenden, um die Leistung einer Abfrage zu verbessern.

MQTs werden bei der Erstellung einer Cachetabelle verwendet. Die Cachetabelle speichert lokale Daten, die von den zugehörigen MQTs definiert werden.

MQTs (Materialized Query Tables) und föderierte Systeme - Übersicht

Eine MQT (Materialized Query Table, gespeicherte Abfragetabelle) ist eine Tabelle, die die Ergebnisse einer Abfrage im Cache speichert. Wenn Sie die Abfrage erneut übergeben, kann die Datenbanksteuerkomponente die Daten aus der MQT zurückgeben und muss die Berechnung der Abfrage nicht wiederholen.

Sie können MQTs mit Kurznamen verwenden, um die Leistung einer Abfrage zu verbessern und um eine Logikkomponente einzubinden. MQTs werden bei der Erstellung von Cachetabellen verwendet.

Das SQL-Optimierungsprogramm stellt fest, ob eine Abfrage mit einer MQT effizienter ausgeführt werden kann als mit den Basistabellen oder Kurznamen. Das Optimierungsprogramm berücksichtigt für die Auswahl der MQT die folgenden Faktoren:

- Die MQT muss teilweise oder ganz mit der Abfrage übereinstimmen.
- Die Bedingung für das Aktualisierungsalter muss erfüllt werden.
- Der Zugriffsplan, der eine MQT verwendet, muss kostengünstiger als der Zugriffsplan sein, der die Basistabellen oder Kurznamen verwendet.

Es werden MQTs unterstützt, die Kurznamen für Objekte aus den folgenden Datenquelle verwenden:

- Relationale Datenquellen
 - DRDA
 - Informix
 - JDBC
 - ODBC
 - Oracle
 - Sybase
 - Microsoft SQL Server
 - Teradata
- Nicht relationale Datenquellen
 - BioRS
 - Excel
 - Dateien mit Tabellenstruktur
 - Web-Services
 - XML

Erstellen einer MQT in föderierten Systemen

Sie verwenden MQTs (Materialized Query Tables, gespeicherte Abfragetabellen), um Daten lokal im Cache zu speichern und um die Leistung Ihrer Abfragen zu verbessern. Sie können Kurznamen relationaler und nicht relationaler Datenquellen verwenden, um MQTs zu erstellen.

Einschränkungen

- „Datenquellenspezifische Einschränkungen für MQTs“
- Wenn eine Abfrage über eine Funktionsschablone in einem Vergleichselement oder einer Auswahlliste verfügt, muss die Funktionsschablone Teil der MQT sein.
- „Einschränkungen bei der Verwendung von MQTs mit Kurznamen“ auf Seite 293

Vorgehensweise

Setzen Sie zur Erstellung einer MQT eine Anweisung `CREATE TABLE` ab, die auf die Kurznamen verweist, die die zu verwendenden fernen Datenquellenobjekte darstellen.

Sie können eine benutzerverwaltete MQT füllen, indem Sie eine Anweisung `INSERT` in einer Subselectanweisung verwenden. Beispiel:

```
insert into my_mqt (select ..from n1, n2 where ..)
```

Dabei stimmt der `SELECT`-Abschnitt der Abfrage mit der MQT-Definition überein. Das Optimierungsprogramm verwendet möglicherweise `'my_mqt'`, um den `SELECT`-Abschnitt der Abfrage zu ersetzen. In diesem Fall sieht die Anweisung wie folgt aus:

```
insert into my_mqt (select .. from my_mqt);
```

In diesem Fall wird die MQT zur Quelle der Einfügeoperation. Um dies zu verhindern, können Sie einen der folgenden Befehle eingeben und damit die MQT vorübergehend inaktivieren:

```
SET CURRENT REFRESH AGE 0  
SET CURRENT MAINTAINED TABLE TYPE FOR OPTIMIZATION SYSTEM
```

Datenquellenspezifische Einschränkungen für MQTs

Bei der Erstellung von MQTs (Materialized Query Tables, gespeicherten Abfragetabellen) müssen die für bestimmte Datenquellen gültigen Einschränkungen beachtet werden.

In diesem Abschnitt werden die Einschränkungen bei der Erstellung von MQTs (Materialized Query Tables, gespeicherten Abfragetabellen) beschrieben, die bei den folgenden Datenquellen gelten:

- BioRS
- Dateien mit Tabellenstruktur
- Web-Services
- XML

Einschränkungen für BioRS-Suchoperatoren

Für den BioRS-Wrapper ist mindestens ein Vergleichselement in der WHERE-Klausel erforderlich. Sie müssen eine MQT erstellen, die den Vergleichselementanforderungen des Wrappers gerecht wird. Wenn Sie kein Vergleichselement angeben, schlägt die Aktualisierung der MQT fehl.

Einschränkungen für Dateien mit Tabellenstruktur

Wenn Sie mit der Option DOCUMENT einen Kurznamen für eine Datei mit Tabellenstruktur definieren, muss die MQT über ein Vergleichselement verfügen, das den Dateipfad angibt. Wenn Sie kein Vergleichselement angeben, schlägt die Aktualisierung der MQT fehl.

Einschränkungen für Web-Services

Sie können eine MQT nur über eine Übersichtsanzeige einer Hierarchie von Kurznamen erstellen. Sie können nicht für jeden Kurznamen einer Hierarchie eine MQT erstellen.

Einschränkungen für XML

Sie können keine gespeicherte Tabelle für eine untergeordnete Tabelle erstellen.

Wenn Sie mit der Option DOCUMENT einen Kurznamen für eine XML-Tabelle definieren, fordert die MQT ein Vergleichselement an, das den Dateipfad angibt. Wenn Sie kein Vergleichselement angeben, schlägt die Aktualisierung der MQT fehl.

Einschränkungen bei der Verwendung von MQTs mit Kurznamen

Bei der Optimierung des föderierten Systems müssen diese Einschränkungen für MQTs (Materialized Query Tables, gespeicherte Abfragetabellen), die auf Kurznamen verweisen, beachtet werden.

Kennsatzbasierte Zugriffssteuerung (LBAC) für Datenquellenobjekte

Kurznamen für Datenquellenobjekte mit kennsatzbasierter Zugriffssteuerung (LBAC) oder Oracle Label Security können nicht im Cache zwischengespeichert werden, und es können keine MQTs für sie erstellt werden.

Systemverwaltete MQTs

Das föderierte System unterstützt keine vom System verwalteten MQTs (Materialized Query Tables, gespeicherten Abfragetabellen), die auf Kurznamen in einer Umgebung mit partitionierten Datenbanken verweisen.

Der föderierte Wrapper muss abgeschirmt sein, damit die föderierte Umschreibungsfunktion die Abfrage an die MQT weiterleiten kann. Sie können den Wrapper entweder mit der Angabe FENCED erstellen oder mithilfe der Anweisung ALTER WRAPPER ändern. Beispiel:

```
CREATE WRAPPER <wrappername>
  LIBRARY <libname>
  OPTIONS (DB2_FENCED 'N');
ALTER WRAPPER <wrappername> OPTIONS (SET DB2_FENCED 'Y');
```

Sie müssen für die Typen der verwalteten Tabellen für die Optimierung ALL oder USER definieren. Verwenden Sie hierzu den Datenbankkonfigurationsparameter DFT_MTTB_TYPES oder die vor der SQL-Abfrage ausgeführte Definition des aktuellen Sonderregisters.

Geben Sie Folgendes ein, um den Wert des Konfigurationsparameters in USER zu ändern:

```
update db cfg for <dbalias> using DFT_MTTB_TYPES USER
```

Geben Sie Folgendes ein, um die Definition des aktuellen Sonderregisters zu verwenden:

```
set current maintained table types for optimization ALL
```

Um diese Einschränkung zu umgehen, können Sie benutzerverwaltete MQTs verwenden.

Sie können z. B. für den nicht relationalen Kurznamen DEPART die folgenden Befehle zur Simulation einer vom System verwalteten MQT eingeben.

```
SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION ALL;
```

```
CREATE TABLE AST1(C1, C2)
  AS (SELECT EMPNO, FIRSTNME FROM DEPART WHERE EMPNO>'000000')
  DATA INITIALLY DEFERRED REFRESH DEFERRED
  ENABLE QUERY OPTIMIZATION MAINTAINED BY USER;
```

```
SET INTEGRITY FOR AST1 ALL IMMEDIATE UNCHECKED;
```

```
INSERT INTO AST1 (SELECT EMPNO, FIRSTNME FROM DEPART WHERE EMPNO>'000000');
```

```
SET CURRENT REFRESH AGE ANY;
```

Die folgende Anweisung SELECT kann von der zuvor definierten MQT beantwortet werden:

```
SELECT EMPNO, FIRSTNME FROM DEPART
  WHERE EMPNO > '000000' AND FIRSTNME LIKE 'AN%';
```

Kapitel 29. Cachetabellen

Cachetabellen werden zur Speicherung von Daten verwendet, die häufig geöffnet, aber selten geändert werden.

Durch eine Cachetabelle kann die Abfrageleistung verbessert werden, da die Daten lokal gespeichert werden und nicht direkt von der Datenquelle aus auf die Daten zugegriffen wird.

Daten aus den folgenden Datenquellen können in den Cache gestellt werden:

- DB2-Produktfamilie
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

Eine Cachetabelle besteht aus den folgenden Komponenten:

- Einem Kurznamen auf dem föderierten System. Der Kurzname weist dieselben Spaltendefinitionen und Datenzugriffsmöglichkeiten auf wie die Datenquellentabelle.
- Mindestens einer MQT (Materialized Query Table), die für den Kurznamen definiert wird. Der Typ der MQT ist FEDERATED_TOOL. Die MQT enthält normalerweise eine Untergruppe mit häufig verwendeten Daten aus der Datenquellentabelle.
- Einem Replikationsplan für jede MQT. Der Replikationsplan sorgt dafür, dass die lokalen MQTs mit den Datenquellentabellen aktualisiert werden. Der Replikationsplan wird von Ihnen definiert.

Die folgende Abbildung zeigt eine Cachetabelle.

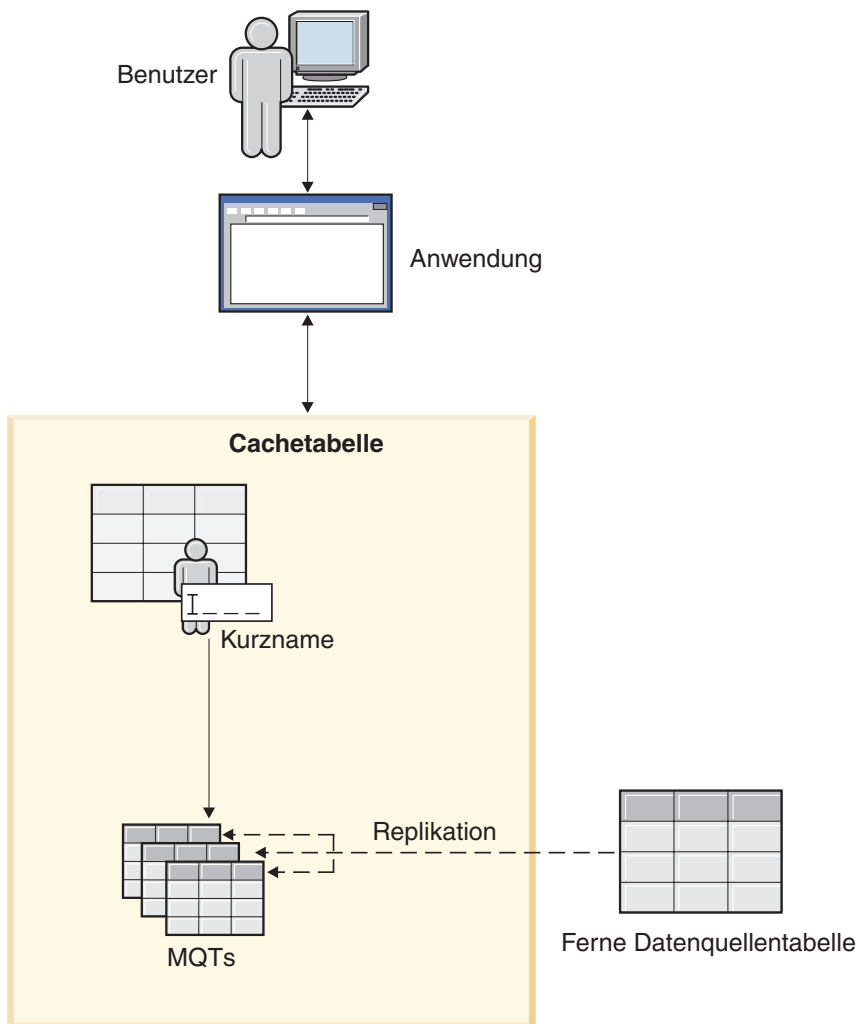


Abbildung 14. Cachetabelle.

Die Cachetabelle hat denselben Namen wie der Kurzname. Sie können eine Cachetabelle nur einer Datenquellentabelle zuordnen.

Wenn eine Cachetabelle aktiviert ist, leitet das Abfrageoptimierungsprogramm Abfragen an die Cachetabelle weiter, wenn die von der Abfrage angeforderten Daten in der MQT enthalten sind.

Erstellen von Cachetabellen

Eine Cachetabelle kann mithilfe eines Assistenten der Steuerzentrale erstellt werden. Der Assistent erstellt den Kurznamen, die MQT und den Replikationsplan, die für die Cachetabelle erforderlich sind.

Vorbereitung

- Setzen Sie den Parameter FEDERATED auf dem Server mit föderierten Datenbanken auf YES. Der Parameter FEDERATED ist ein Konfigurationsparameter des Datenbankmanagers.

- Installieren und konfigurieren Sie für den Zugriff auf Informix-Datenquellen das SDK (Software Development Kit) des Informix-Clients auf dem Server mit föderierten Datenbanken.
- Konfigurieren Sie die DB2-Datenbank für die Archivprotokollierung, um Daten aus Tabellen von DB2 Database für Linux, UNIX und Windows in den Cache zu stellen.
- Die föderierte Datenbank bzw. die Quelldatenbank muss sich auf dem Computer befinden, von dem aus Sie die Cachetabellen erstellen. Wenn die föderierte Datenbank bzw. die Quelldatenbank nicht lokal gespeichert ist, müssen Sie die Datenbanken auf dem lokalen Computer katalogisieren. Der Aliasname, den Sie beim Katalogisieren der Datenbank verwenden, muss mit dem Datenbanknamen identisch sein.
- Die Benutzer-ID, die für die Benutzerzuordnung zwischen den Datenbanken definiert wurde, muss über die Berechtigung zum Erstellen von Tabellen in der Quelldatenbank verfügen.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Cachetabelle zu erstellen:

1. Erweitern Sie in der Steuerzentrale den Ordner **Cacheobjekte**.
2. Klicken Sie mit der rechten Maustaste den Ordner **Cachetabellen** an, und klicken Sie **Erstellen** an.
3. Führen Sie die Schritte im Cachetabellenassistenten aus, um die Cachetabelle zu erstellen. Sie können die Cachetabelle schnell erstellen, indem Sie nur die Werte für die erforderlichen Felder angeben und für die übrigen Felder die Standardeinstellungen verwenden. Gehen Sie wie folgt vor, um die Standardeinstellungen zu ändern:
 - a. Klicken Sie auf der Seite **MQT** die Option **Erweiterte MQT-Einstellungen** an, um die Standardeinstellungen zu ändern oder eine Untergruppe von Spalten für die MQT auszuwählen.
 - b. Klicken Sie auf der Seite **Replikation** die Option **Erweiterte Einstellungen** an, um die Standardeinstellungen für das Replizieren von Daten von der Datenquellentabelle in die MQT zu ändern.

In einigen Fällen ist das Caching nach Beendigung des Assistenten nicht aktiviert. Sie müssen das Caching aktivieren, damit das Capture- und das Apply-Programm für die Replikation gestartet werden.

Der Cachetabellenassistent erstellt eine einzige MQT, wenn Sie die Cachetabelle erstellen. Sie können zusätzliche MQTs erstellen, um weitere Daten aus derselben Datenquelle zu speichern.

Modifizieren der Einstellungen für MQTs

Die Einstellungen für MQTs können nicht direkt modifiziert werden. Zum Ändern der Replikations- und MQT-Einstellungen müssen alternative Methoden verwendet werden.

Vorgehensweise

Gehen Sie wie folgt vor, um die Einstellungen für eine MQT zu modifizieren:

1. Rufen Sie das Fenster **MQT-Details** auf, um die Einstellungen für die MQT und den Replikationsplan anzuzeigen.
 - a. Erweitern Sie in der Steuerzentrale den Ordner **Cacheobjekte**.
 - b. Klicken Sie die Cachetabelle mit der rechten Maustaste an, und klicken Sie **Merkmale** an.
 - c. Wählen Sie die MQT aus, und klicken Sie **Details** an, um die aktuellen Einstellungen anzuzeigen.
2. Änderungen an den Replikationseinstellungen können über die Replikationszentrale vorgenommen werden. Die Replikationseinstellungen für die MQT können über das Fenster **MQT-Details** nicht geändert werden.
3. Wenn Sie die Einstellungen für eine MQT ändern müssen, löschen Sie die MQT, und erstellen Sie eine neue MQT. Wenn Sie der MQT zum Beispiel eine weitere Spalte hinzufügen möchten, löschen Sie die MQT, und erstellen Sie eine MQT mit den neuen Einstellungen.

Hinzufügen von MQTs zu einer Cachetabelle

Für eine Cachetabelle können zusätzliche MQTs erstellt werden. Mit diesen zusätzlichen MQTs können weitere Daten aus derselben Datenquelle gespeichert werden.

Informationen zu dieser Task

Wenn Sie eine Cachetabelle erstellen, speichert der Server mit föderierten Datenbanken die Daten separat von der Datenquelle in einer MQT (Materialized Query Table) auf der lokalen Einheit. Die Kriterien, die Sie im Cachetabellenassistenten angeben, bestimmen, welche Daten in der MQT gespeichert werden sollen.

Beispiel: Ihre erste MQT enthält Informationen zu Kunden in Asien. Sie können eine weitere MQT erstellen, die Informationen zu Kunden in Südamerika enthält.

Vorgehensweise

Gehen Sie wie folgt vor, um einer vorhandenen Cachetabelle eine MQT hinzuzufügen:

1. Erweitern Sie in der Steuerzentrale den Ordner **Cacheobjekte** und den Ordner **Cachetabellen**.
2. Klicken Sie mit der rechten Maustaste die entsprechende Cachetabelle an, und klicken Sie **Merkmale** an.
3. Klicken Sie auf **Hinzufügen**.
4. Führen Sie die Schritte im Cachetabellenassistenten aus, um die zusätzliche MQT zu erstellen.

Das Fenster **Merkmale der Cachetabelle** kann auch über ein Kurznamenobjekt geöffnet werden. Klicken Sie den Kurznamen mit der rechten Maustaste an, und klicken Sie **In den Cache stellen** an.

Weiterleiten von Abfragen an Cachetabellen

Abfragen können an die MQT für die Cachetabelle oder an die Datenquelle weitergeleitet werden. Dazu werden die Abfragen an den Kurznamen weitergeleitet.

Vorbereitung

- Für den Kurznamen, der abgefragt wird, muss eine Cachetabelle vorhanden sein.
- Aktivieren Sie das Caching für die MQT.
- Setzen Sie die folgenden Datenbankkonfigurationsparameter:
 - MQT-Typ für Optimierungen (DFT_MTTB_TYPES)
 - Abfrageoptimierungsklasse (DFT_QUERYOPT)

Verwenden Sie das Notizbuch **Datenbank konfigurieren** in der Steuerzentrale oder die Befehlszeile, um diese Parameter zu setzen.

Vorgehensweise

Gehen Sie wie folgt vor, um die Weiterleitung für Abfragen zu ändern:

1. Erweitern Sie in der Steuerzentrale den Ordner **Cacheobjekte** und den Ordner **Cachetabellen**.
2. Klicken Sie mit der rechten Maustaste die entsprechende Cachetabelle an, und klicken Sie **Merkmale** an.
3. Wählen Sie die MQT aus, und klicken Sie **Status überprüfen** an.
4. Wählen Sie die Weiterleitungsziele für die Abfragen aus:
 - Wählen Sie **An MQT weiterleiten** aus. Alle Abfragen für die Datenquelle werden an die MQT gesendet. Wenn die Daten in der MQT für die Abfrage nicht ausreichend sind, wird die Abfrage unter Verwendung des Kurznamens an die Datenquelle weitergeleitet.
 - Wählen Sie **An Kurznamen weiterleiten** aus. Alle Abfragen werden unter Verwendung des Kurznamens an die Datenquelle gesendet. Die MQT wird in eine reguläre Datenbanktabelle geändert. Die Replikation für die MQT wird fortgesetzt, falls das Caching aktiviert bleibt.
5. Wählen Sie das Markierungsfeld **Alle in den Cache gestellten dynamischen SQL-Anweisungen entfernen** aus, um die Weiterleitung sofort zu ändern. Die aktuell im Paketcache vorhandenen dynamischen SQL-Anweisungen werden gelöscht.
6. Klicken Sie **OK** an.

Das Fenster **Merkmale der Cachetabelle** kann auch über ein Kurznamenobjekt geöffnet werden. Klicken Sie den Kurznamen mit der rechten Maustaste an, und klicken Sie **In den Cache stellen** an.

Aktivieren und Inaktivieren der Cacheeinstellungen für die Replikation

Durch Änderung der Cacheeinstellungen kann die Datenreplikation für die MQT gestartet und gestoppt werden.

Vorbereitung

Setzen Sie die Datenbankprotokollierung für Datenquellen unter DB2 Database für Linux, UNIX und Windows auf Archivprotokollierung.

Informationen zu dieser Task

Wenn Sie das Caching aktivieren, werden das Capture- und das Apply-Programm gestartet, sofern diese Programme nicht bereits aktiv sind. Gleichzeitig wird der Subskriptionsgruppeneintrag aktiviert. Die Aktivierung des Subskriptionsgruppeneintrags weist das Apply-Programm an, die Daten in der MQT mit den Daten in der Datenquellentabelle zu synchronisieren.

Bei Inaktivierung des Caching wird der Subskriptionsgruppeneintrag inaktiviert. Die Daten aus der Datenquelle werden nicht in die MQT repliziert.

Wichtig: Wenn Sie die Einstellung für die Weiterleitung nicht ändern, werden die Abfragen an die MQT weitergeleitet, obwohl die Daten nicht in die MQT repliziert werden.

Vorgehensweise

Gehen Sie wie folgt vor, um die Cacheeinstellungen für die Replikation für eine MQT zu aktivieren oder zu inaktivieren:

1. Erweitern Sie in der Steuerzentrale den Ordner **Cacheobjekte** und den Ordner **Cachetabellen**.
2. Klicken Sie mit der rechten Maustaste die entsprechende Cachetabelle an, und klicken Sie **Merkmale** an.
3. Wählen Sie die MQT aus, und klicken Sie **Status überprüfen** an.
4. Wählen Sie entweder **Caching aktivieren** oder **Caching inaktivieren** aus. Klicken Sie **Anweisungen anzeigen** an, um die Befehle zum Aktivieren oder Inaktivieren anzuzeigen.
5. Klicken Sie **OK** an.

Das Fenster **Merkmale der Cachetabelle** kann auch über ein Kurznamenobjekt geöffnet werden. Klicken Sie den Kurznamen mit der rechten Maustaste an, und klicken Sie **In den Cache stellen** an.

Löschen von MQTs aus einer Cachetabelle

Wenn Sie keine Daten mehr lokal in einer MQT (Materialized Query Table) speichern möchten, können Sie die MQT aus der Cachetabelle löschen.

Informationen zu dieser Task

Wenn eine Cachetabelle nur über eine MQT verfügt, wird durch das Löschen der MQT auch die Cachetabelle gelöscht.

Wenn Sie sicherstellen möchten, dass die MQT vollständig aus dem System entfernt wird, verwenden Sie die Steuerzentrale, um die MQT aus der Cachetabelle zu löschen.

Vorgehensweise

Gehen Sie wie folgt vor, um eine MQT aus einer Cachetabelle zu löschen:

1. Erweitern Sie in der Steuerzentrale in der Objektbaumstruktur den Ordner **Cacheobjekte** und den Ordner **Cachetabellen**.
2. Klicken Sie mit der rechten Maustaste die entsprechende Cachetabelle an, und klicken Sie **Merkmale** an.
3. Wählen Sie die entsprechende MQT aus, und klicken Sie **Entfernen** an.

Löschen von Cachetabellen

Wenn Sie keine Daten mehr lokal in einer Cachetabelle speichern möchten, können Sie die Cachetabelle löschen.

Informationen zu dieser Task

Wenn Sie eine Cachetabelle löschen, führt der Server mit föderierten Datenbanken die folgenden Aktionen aus:

- Die für die Cachetabelle erstellten MQTs werden gelöscht.
- Der Replikationsplan zwischen den Datenquellen und den MQTs wird entfernt.
- Der Kurzname für die Datenquelle wird gelöscht, wenn der Kurzname bei der Erstellung der Cachetabelle erstellt wurde. Wenn Sie bei der Erstellung der Cachetabelle einen vorhandenen Kurznamen verwendet haben, löscht der Server mit föderierten Datenbanken den Kurznamen nicht.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Cachetabelle zu löschen:

1. Erweitern Sie in der Steuerzentrale in der Objektbaumstruktur den Ordner **Cacheobjekte** und den Ordner **Cachetabellen**.
2. Klicken Sie mit der rechten Maustaste die entsprechende Cachetabelle an, und klicken Sie **Löschen** an.

Kapitel 30. Sicherheitseinrichtungen für Server mit föderierten Datenbanken

Der Server mit föderierten Datenbanken unterstützt SSL für die Datenverschlüsselung sowie HTTP- und SOCKS-Proxys für bestimmte Datenquellen.

Verschlüsselung

Die Verschlüsselung sorgt für eine Sicherheitsstufe, die über die durch einen Namen und ein Kennwort bereitgestellte Sicherheitsstufe hinausgeht. Den Internet-Standard für die Verschlüsselung zwischen Verbindungsendpunkten bilden Secure Socket Layer (SSL) und Transport Layer Security (TLS). SSL sorgt mithilfe signierter Zertifikate für eine sichere Kommunikation. Ein *Zertifikat* ist ein digitales Dokument, das eine Bestätigung der Identifikationsdaten eines Benutzers oder Servers liefert. Ein Zertifikat ist von einer Zertifizierungsstelle, wie zum Beispiel VeriSign, signiert oder kann vom Sender selbst signiert sein. Jeder Kommunikationspartner bestimmt selbst, ob er ein bestimmtes Zertifikat als authentisch akzeptiert.

Jeder Kommunikationspartner besitzt einen Zertifikatsspeicher bzw. Schlüssel-speicher (Keystore). Im Schlüsselspeicher sind die Zertifikate enthalten, die der Partner von anderen akzeptiert sowie die Zertifikate, die sich selbst darstellen. Beim Öffnen einer Kommunikationsverbindung bestimmt jeder Endpunkt, ob er ein Zertifikat sendet oder nicht und ob er eine Datenübertragung von einem Partner akzeptiert, der kein Zertifikat bereitstellt.

Im Hinblick auf die Zwecke von Wrappern und Funktionen sind die folgenden SSL-Funktionen relevant:

- Serverseitige Identifikation eines zu sendenden Zertifikats
- Serverseitige Prüfung eines Clientzertifikats
- Clientseitige Prüfung eines Serverzertifikats
- Clientseitige Identifikation eines zu sendenden Zertifikats
- Unterstützungstyp, Position und Zugriff auf einen lokalen Schlüsselspeicher für Client und Server

Die Interaktion zwischen SSL und einem Proxy hängt vom Typ des Proxys ab. Im Allgemeinen erfolgen SSL-Datenübertragungen im Tunnelungsverfahren oder werden über einen Proxy weitergeleitet. Die Proxysitzung wird im Klartextformat eingerichtet.

IBM Global Security Kit (GSKit) stellt Verschlüsselungsservices für Wrapper und benutzerdefinierte Funktionen zur Verfügung.

Proxys

Viele Unternehmen implementieren eine Firewall zu dem Zweck, die zahlreichen verschiedenen Arten von Internetangriffen abzuwehren. Eine *Firewall* ist eine Netz-konfiguration, die in der Regel sowohl Hardware- als auch Softwarekomponenten umfasst und häufig an einer Kommunikationsgrenze, zum Beispiel zwischen dem Intranet eines Unternehmens und dem Internet aufgebaut wird. Die Firewall fungiert als Torwächter und regelt den die Kommunikationsgrenze überschreitenden Datenverkehr. In den meisten Fällen verhindert die Firewall, dass unerwünschte

Datenübertragungen über die Grenze hinweg stattfinden. Gelegentlich kann es jedoch vorkommen, dass die Firewall auch legitimen Datenverkehr abblockt.

Zur Sicherstellung, dass alle legitimen Datenübertragungen die Firewall passieren, wird ein Proxy implementiert. Ein *Proxy* ist ein Serverprogramm, das zur Kommunikation durch die Firewall hindurch autorisiert ist. Wenn ein Benutzerprogramm eine Verbindung zu einem fernen Server herstellen muss, sendet es eine Anfordernungen an den Proxy, der wiederum die Verbindung zu dem fernen Server herstellt. Nach dem Herstellen der Verbindung steuert der Proxy den Datenverkehr zwischen dem Benutzerprogramm und dem fernen Server. Dieses Verfahren stellt sicher, dass das Benutzerprogramm durch die Firewall hindurch kommunizieren kann und gleichzeitig die Sicherheit aufrechterhalten wird: Der ferne Server kennt nur die Adresse des Proxys, nicht jedoch die Adresse des Benutzerprogramms.

SOCKS- und HTTP-Proxy-Server steuern den Datenverkehr zwischen Benutzerprogrammen und fernen Servern. Ein SOCKS-Proxy arbeitet in der Transportschicht und leitet beliebige Transportnachrichten (TCP oder UDP) zwischen zwei Adressen weiter. Der Server mit föderierten Datenbanken unterstützt SOCKS4 und SOCKS5. Die SOCKS4-Spezifikation, die IPv4 unterstützt, unterstützt keine Benutzerauthentifizierung. Das heißt, jeder Benutzer kann über einen SOCKS4-Proxy kommunizieren, ohne Berechtigungsnachweise angeben zu müssen. Die SOCKS5-Spezifikation, die IPv6 unterstützt, bietet Unterstützung für verschiedene Modi der Benutzerauthentifizierung. SOCKS5 wurde von der Internet Engineering Task Force (IETF) als Standard freigegeben. Weitere Informationen zu diesem Thema finden Sie auf der Website 'www.ietf.org' unter RFC1928, RFC1929 und RFC1961.

Zur Verwendung eines SOCKS-Proxys muss die Transportschicht im Voraus zur Verwendung eines Proxys konfiguriert werden. Nach dem Öffnen einer Verbindung zum Proxy fordert die Transportschicht von dem Proxy das Öffnen einer Verbindung zum fernen Server an. Wenn eine Authentifizierung als erforderlich konfiguriert wurde, fordert der SOCKS-Proxy das Programm unter Umständen dazu auf, eine ID und ein Kennwort zu liefern, bevor die Verbindung zum fernen Server geöffnet wird.

HTTP-Proxys arbeiten mit dem HTTP-Protokoll, bei dem es sich um ein Protokoll der Anwendungsschicht handelt. Nach dem Öffnen einer TCP/IP-Verbindung zum Proxy sendet das Benutzerprogramm eine Anforderung, die den Namen des fernen Servers enthält. Anschließend fährt das Benutzerprogramm fort, Anforderungen über den Proxy-Server zu übergeben. Dieses Verfahren ändert sich geringfügig, wenn der ferne Server eine Authentifizierung erfordert. Erfordert der ferne Server eine Authentifizierung, sendet der Proxy-Server eine Antwortnachricht, die den Header der Authentifizierungsanforderung für die Proxyberechtigung enthält. Dieser Header enthält Informationen zu der Art der auszuführenden Authentifizierung. Das Benutzerprogramm übergibt anschließend die Anforderung erneut und fügt einen Header zur Proxyberechtigung ein, der die Antwort auf die Authentifizierungsanforderung enthält.

Kapitel 31. Gesicherte Kontexte und Verbindungen föderierter Systeme

Verbessern Sie die Systemleistung und reduzieren oder eliminieren Sie die Verwendung und Verwaltung von Benutzerzuordnungen.

Bei einem *gesicherten Kontext* handelt es sich um ein DB2-Datenbankobjekt, das eine Vertrauensbeziehung zwischen einem Client und einer Datenquelle, z. B. zwischen einem Anwendungsserver und einem Server mit föderierten Datenbanken oder einem Server mit föderierten Datenbanken und einem fernen Datenbankserver definiert. Die Vertrauensbeziehung wird von dem gesicherten Kontext über *Vertrauensattribute* definiert. Es gibt drei Typen von Vertrauensattributen:

- Die Systemberechtigungs-ID, über die die Datenbankverbindung zunächst angefordert wird
- Die IP-Adresse bzw. der Domänenname, von der/dem die Verbindung ausgeht
- Die Verschlüsselungseinstellung für den Datenaustausch zwischen Datenbankserver und Datenbankclient

Eine *gesicherte Verbindung* liegt vor, wenn alle Attribute einer Verbindungsanforderung mit den Vertrauensattributen übereinstimmen, die in einem auf dem Server definierten gesicherten Kontextobjekt angegeben sind. Nach dem Einrichten einer expliziten gesicherten Verbindung können die Benutzer-IDs für diese physische Verbindung mit oder ohne Authentifizierung gewechselt werden. Darüber hinaus können Benutzern für die Verbindung Rollen erteilt werden, die Berechtigungen enthalten, die ausschließlich für die gesicherte Verbindung gelten.

Bei dem folgenden Beispiel wird ein gesichertes Kontextobjekt für den Benutzer BOSS erstellt:

```
CREATE TRUSTED CONTEXT MYCTX
  BASED UPON CONNECTION USING SYSTEM AUTHID BOSS
  ATTRIBUTES (ADDRESS '9.26.111.111')
  WITH USE FOR MARY WITH AUTHENTICATION ROLE MANAGER,
  PUBLIC WITHOUT AUTHENTICATION
  DEFAULT ROLE AUDITOR
  ENABLE
```

Bei diesem Beispiel kann nur der Benutzer mit der ID BOSS eine gesicherte Verbindung von der IP-Adresse 9.26.111.111 aus aufbauen. Die Verbindung kann von der Benutzerin Mary wiederverwendet werden. Sie muss dazu jedoch zunächst eine Authentifizierung vornehmen. Anschließend erhält diese Benutzerin die zusätzliche Rolle MANAGER, die ihre Berechtigungen innerhalb dieser gesicherten Verbindung vorgibt. Andere Benutzer, die als PUBLIC angegeben sind, können die Verbindung ohne Authentifizierung wiederverwenden. Diese Benutzer erhalten die zusätzliche Rolle AUDITOR, die bestimmte Berechtigungen innerhalb dieser gesicherten Verbindung vorgibt. Diese zusätzlichen Berechtigungen sind für die Benutzer verfügbar, während sie die gesicherte Verbindung aktiv verwenden.

Bei gesicherten Verbindungen werden explizite und implizite Verbindungen unterschieden. Der Typ der Verbindung bestimmt, ob die betreffende Verbindung wiederverwendet werden kann und ob Benutzer zusätzliche Rollen erhalten können.

Eine *implizite gesicherte Verbindung* wird eingerichtet, wenn eine gesicherte Verbindung nicht explizit angefordert wird, die Verbindungsattribute der Verbindung jedoch mit den Vertrauensattributen eines gesicherten Kontextobjekts auf dem Server übereinstimmen. Nach dem Einrichten einer impliziten gesicherten Verbindung kann nur der Ersteller der gesicherten Verbindung Rollen erben, die ihm ansonsten nicht zur Verfügung stehen. Eine implizite gesicherte Verbindung kann nicht von anderen Benutzern wiederverwendet werden.

Eine *explizite gesicherte Verbindung* wird eingerichtet, wenn eine Anwendung eine gesicherte Verbindung über eine API anfordert. Stimmen die Verbindungsattribute mit den Vertrauensattributen eines gesicherten Kontexts überein, wird eine gesicherte Verbindung eingerichtet. Andernfalls wird eine reguläre Verbindung eingerichtet. Nach dem Einrichten einer expliziten gesicherten Verbindung können andere Benutzer die Verbindung wiederverwenden. Sowohl der Ersteller der Verbindung als auch die Benutzer, die die Verbindung wiederverwenden, können zusätzliche Rollen erben, die ihnen ansonsten nicht zur Verfügung stehen.

Vorteile gesicherter Verbindungen föderierter Systeme

In einem vielschichtigen Anwendungsmodell wird eine einzelne physische Verbindung von verschiedenen gesicherten Verbindungen föderierter Systeme wiederverwendet, um die Identität der einzelnen Benutzer über die verschiedenen Schichten an den Datenbankserver weiterzugeben.

Um verstehen zu können, welche Vorteile gesicherte Verbindungen föderierter Systeme bieten, muss man sich die Probleme vor Augen halten, die in einem typischen vielschichtigen Anwendungsmodell auftreten. Ein vielschichtiges Anwendungsmodell besteht aus den Benutzern (Schicht 1), die mit einer Anwendung auf dem Anwendungsserver (Schicht 2) interagieren, der alle Datenbankzugriffe über den Server mit föderierten Datenbanken (Schicht 3) weiterleitet, der wiederum den Datenaustausch mit verschiedenen Datenbankservern (Schicht 4) verwaltet. Bei diesem Modell erfolgt die Authentifizierung der Benutzer über den Anwendungsserver, der auch die Interaktion mit dem Server mit föderierten Datenbanken verwaltet. Der Server mit föderierten Datenbanken setzt die Benutzeranforderungen in datenquellenspezifische Formate um, stellt Verbindungen zu den fernen Datenquellen her und sendet Anforderungen an diese Datenquellen.

Zum Herstellen einer Verbindung zum Datenbankserver werden bei diesem Modell die Anwendungsserver-ID und das zugehörige Kennwort verwendet. Der Server mit föderierten Datenbanken übergibt die ID und das Kennwort lediglich vom Anwendungsserver an den Datenbankserver. Der Datenbankserver entscheidet anhand der Datenbankzugriffsrechte, die dieser ID zugeordnet sind, über die Ausführung der Transaktionen, die der Anwendungsserver ausführen soll, und protokolliert ausgeführte Transaktionen. Dies schließt alle Transaktionen ein, die der Anwendungsserver für Benutzer ausführt.

Die Verwendung der Anwendungsserver-ID führt zu folgenden Problemen:

- Die Identität des Benutzers, der eine Transaktion ausführt, ist nicht bekannt, da alle Transaktionen vom Anwendungsserver ausgeführt werden.
- Benutzer können nicht für Transaktionen verantwortlich gemacht werden, weil sie nicht in den Protokollen erfasst werden können.
- Es wird gegen das Prinzip des niedrigstmöglichen Zugriffsrechts verstoßen, da der Anwendungsserver-ID eine Obermenge aller Zugriffsrechte zugeordnet sein muss, die von den einzelnen Benutzer benötigt werden.

- Die Daten sind schlecht geschützt, wenn ein unbefugter Zugriff auf die Anwendungsserver-ID erfolgt.

Mit gesicherten Verbindungen föderierter Systeme lassen sich diese Probleme lösen. Diese Verbindungen weisen verschiedene Vorzüge auf, die Sicherheit und Leistung des Systems erhöhen können:

Die Benutzeridentität ist bekannt.

Da Benutzer-IDs innerhalb einer Verbindung gewechselt werden können, ist die Identität der Benutzer, die jeweils auf die Datenbank zugreifen, bekannt.

Benutzer sind für ihre Transaktionen verantwortlich.

Die Prüfprotokolle für die föderierte Datenbank und für die ferne Datenquellendatenbank weisen die Transaktionen aus, die der Anwendungsserver für interne Zwecke ausführt, sowie die Transaktionen, die einzelne Benutzer ausführen. Die Benutzer, die bestimmte Transaktionen ausgeführt haben, können daher ermittelt werden.

Die Zugriffsrechte sind begrenzt.

Wenn Sie einen gesicherten Kontext erstellen, können Sie allen Benutzern eine Standarddatenbankrolle zuordnen und bestimmten Benutzern mit Grant bestimmte Rollen erteilen. Nur gesicherte Datenbankverbindungen, die der Definition des jeweiligen gesicherten Kontexts entsprechen, können anschließend die Zugriffsrechte nutzen, die der jeweiligen Rolle zugeordnet sind.

Daten sind besser geschützt.

In einem System, in dem gesicherte Verbindungen föderierter Systeme verwendet werden, muss der Anwendungsserver-ID keine Obermenge der Zugriffsrechte zugeordnet werden, die für die Benutzer erforderlich sind. Sollte ein unbefugter Zugriff auf die Anwendungsserver-ID erfolgen, sind Daten deshalb besser geschützt, da dieser ID nicht mehr die Obermenge aller für die Benutzer erforderlichen Zugriffsrechte zugeordnet ist.

Der Verwaltungsaufwand wird minimiert.

Die Notwendigkeit zum Erstellen und Verwalten von Benutzerzuordnungen ist erheblich geringer.

Die Leistung verbessert sich.

Nach dem Einrichten einer expliziten gesicherten Verbindung kann der Server mit föderierten Datenbanken von der aktuellen Benutzer-ID der Verbindung zu einer anderen Benutzer-ID wechseln. Dies kann mit oder ohne Benutzerauthentifizierung erfolgen. Die Wiederverwendung einer einzelnen physischen Leitung durch verschiedene Benutzer kann zu einer Leistungserhöhung führen.

Typen gesicherter Verbindungen föderierter Systeme

Bei gesicherten Verbindungen föderierter Systeme handelt es sich entweder um gesicherte End-to-End-Verbindungen oder gesicherte, abgehende Verbindungen. Der verwendete Verbindungstyp richtet sich nach der Konfiguration des Systems und danach, ob die Anforderung der eingehenden Verbindung gesichert ist.

Eine typische föderierte Konfiguration ist vielschichtig: Sie enthält einen Anwendungsserver, einen Server mit föderierten Datenbanken und einen fernen Server, der als Datenquelle dient. Innerhalb dieser Konfiguration empfängt der Ser-

ver mit föderierten Datenbanken Anforderungen für eingehende Verbindungen und sendet Anforderungen für abgehende Verbindungen an den fernen Datenquellenserver.

Gesicherte End-to-End-Verbindungen föderierter Systeme

Gesicherte End-to-End-Verbindungen föderierter Systeme ermöglichen eine Wiederverwendung von Verbindungen sowie eine Identitätsprüfung für eingehende und abgehende Verbindungen. Ist z. B. eine eingehende Verbindung auf dem Server mit föderierten Datenbanken implizit oder explizit gesichert, fordert der Server automatisch eine gesicherte, abgehende Verbindung an. Verfügt die Datenquelle über die erforderliche Funktionalität für Identitätsprüfungen, wird eine gesicherte, abgehende Verbindung hergestellt und die Identität des Benutzers wird an die ferne Datenquelle weitergegeben. Die eingehenden und abgehenden Verbindungen werden wiederverwendet, sobald ein anderer Benutzer die Wiederverwendung der gesicherten Verbindung anfordert. In diesem Fall wird die Identität des neuen Benutzers über das System weitergegeben.

Für Datenquellen, die nicht über die Funktionalität zur Identitätsprüfung verfügen, durchführen können, stellt der Server mit föderierten Datenbanken eine durchgängige Identitätsprüfung, jedoch keine Möglichkeit zur Wiederverwendung von Verbindungen bereit. In diesen Fällen schließt der Server mit föderierten Datenbanken bei jedem Benutzerwechsel die abgehende Verbindung des vorangehenden Benutzers und stellt eine neue Verbindung für den neuen Benutzer her. Dadurch wird die Identität des Benutzers über das System weitergegeben, auch wenn die abgehende Verbindung nicht wiederverwendet wird.

Gesicherte, abgehende Verbindungen föderierter Systeme

Gesicherte, abgehende Verbindungen föderierter Systeme nutzen die Möglichkeiten der Datenquellen für eine Wiederverwendung gesicherter Verbindungen ohne Authentifizierung, um auf das Speichern von für die Datenquellen definierten Kennwörtern in Benutzerzuordnungen verzichten zu können. Ist eine eingehende Verbindung gesichert, fordert der Server mit föderierten Datenbanken automatisch eine gesicherte, abgehende Verbindung an, die ohne Authentifizierung wiederverwendet werden kann. Bei nicht gesicherten, eingehenden Verbindungen ermöglichen gesicherte, abgehende Verbindungen des föderierten Systems die Wiederverwendung von Verbindungen ohne eine Authentifizierung der Benutzer.

Bei dieser Konfiguration geben Sie in der Serverdefinition mithilfe der Option `FED_PROXY_USER` die Berechtigungs-ID an, über die die abgehende Verbindung ursprünglich eingerichtet wurde. Für die von Ihnen angegebene Berechtigungs-ID muss eine Benutzerzuordnung vorliegen, die sowohl die Option `REMOTE_AUTHID` als auch die Option `REMOTE_PASSWORD` enthält.

Je nach Konfiguration des gesicherten Kontexts auf dem fernen Datenquellenserver können Sie die Anzahl der Benutzerzuordnungen im Datenbankkatalog auf eine einzige Zuordnung reduzieren. Wenn Sie beispielsweise `PUBLIC` die Berechtigung einräumen, Verbindungen ohne Authentifizierung herzustellen, benötigt nur der Proxybenutzer des föderierten Systems eine Benutzerzuordnung. Gibt der gesicherte Kontext des föderierten Systems auf der fernen Datenquelle jedoch an, dass bei bestimmten Benutzern eine Authentifizierung erforderlich ist, oder verwenden Benutzer unterschiedliche Benutzer-IDs auf dem Server mit föderierten Datenbanken und der fernen Datenquelle, müssen Sie eine Benutzerzuordnung so erstellen bzw. die vorhandene Benutzerzuordnung so ändern, dass die Option `REMOTE_`

AUTHID und/oder die Option REMOTE_PASSWORD verwendet wird/werden. Darüber hinaus müssen Sie die Option USE_TRUSTED_CONTEXT mit 'Y' definieren.

Da gesicherte, abgehende Verbindungen des föderierten Systems einen beträchtlichen Zusatzaufwand bei Konfigurations- und Verwaltungstasks mit sich bringen, empfiehlt es sich, das föderierte System so zu gestalten, dass soweit möglich gesicherte End-to-End-Verbindungen implementiert werden. Verwenden Sie gesicherte, abgehende Verbindungen föderierter Systeme nur, wenn absolut notwendig.

APIs für gesicherte Verbindungen föderierter Systeme

Die APIs, die Sie zum Anfordern und Wiederverwenden gesicherter Verbindungen verwenden, richten sich nach dem Typ der Anwendung.

Je nach Anwendung müssen folgende APIs zum Anfordern gesicherter Verbindungen föderierter Systeme angegeben werden:

CLI/ODBC-Anwendungen

Geben Sie 'SQLSetConnectAttr' mit dem Attribut SQL_ATTR_USE_TRUSTED_CONTEXT an, um auszuweisen, ob der Client eine gesicherte Verbindung anfordert. Setzen Sie anschließend eine Anweisung 'SQLConnect' ab.

CLI/ODBC-Anwendungen in einer XA-Umgebung

Definieren Sie zum Anfordern einer gesicherten Verbindung das Attribut TCTX in der Anforderung 'xa_open string' mit 'true'.

Java-Anwendungen

Fordern Sie eine gesicherte Verbindung mit 'getDB2TrustedPooledConnection' oder 'getDB2TrustedXAConnection' an.

Je nach Anwendung müssen zur Wiederverwendung einer Verbindung für einen anderen Benutzer mit oder ohne Authentifizierung des Benutzers die folgenden APIs angegeben werden:

CLI/ODBC-Anwendungen

Definieren Sie SQL_ATTR_TRUSTED_CONTEXT_USERID und SQL_ATTR_TRUSTED_CONTEXT_PASSWORD in Verbindung mit 'SQLSetConnectAttr' an, um Benutzer-ID und Kennwort des Benutzers anzugeben, zu dem in der Verbindung gewechselt werden soll.

CLI/ODBC-Anwendungen in einer XA-Umgebung

Definieren Sie SQL_ATTR_TRUSTED_CONTEXT_USERID und SQL_ATTR_TRUSTED_CONTEXT_PASSWORD in Verbindung mit 'SQLSetConnectAttr' an, um Benutzer-ID und Kennwort des Benutzers anzugeben, zu dem in der Verbindung gewechselt werden soll.

Java-Anwendungen

Verwenden Sie 'getDB2Connection' und 'reuseDB2Connection'.

Szenarios für die Implementierung gesicherter, abgehender Verbindungen bei föderierten Systemen

Föderierte Systeme können sehr unterschiedlich sein. Für die Implementierung gesicherter Kontexte für föderierte Systeme kann es deshalb keine schrittweise Anleitung geben. Mithilfe der Szenarios können Sie sich jedoch ein umfassendes Wissen über gesicherte Verbindungen föderierter Systeme aneignen. Planen und implementieren Sie anschließend Ihre eigene Lösung.

Jedem Szenario liegt dasselbe vielschichtige föderierte System zugrunde, das mehrere Benutzer, eine auf einem Anwendungsserver ausgeführte Anwendung, einen Server mit föderierten Datenbanken und einen fernen DB2-Datenbankserver beinhaltet. Die Szenarios veranschaulichen die Konfiguration von fernen Quellen und Servern mit föderierten Datenbanken für die Verwendung gesicherter Verbindungen für föderierte Systeme.

Bei dem ersten Szenario, das keine Benutzerzuordnungen erfordert, handelt es sich um das Szenario, das am einfachsten zu implementieren und zu verwalten ist. Dieses Szenario wird deshalb für die Implementierung gesicherter Verbindungen in einem neuen System empfohlen.

Das zweite Szenario verdeutlicht die Implementierung gesicherter Verbindungen in einem System mit Benutzerzuordnungen. Das dritte Szenario zeigt die Möglichkeiten gesicherter Verbindungen föderierter Systeme auf, mit deren Hilfe auf Benutzerzuordnungen verzichtet werden kann. Dabei ist zu beachten, dass die beiden zuletzt genannten Szenarios komplexer sind und einen größeren Konfigurations- und Verwaltungsaufwand erfordern als das erste Szenario.

Szenario: Gesicherte End-to-End-Verbindungen föderierter Systeme ohne Benutzerzuordnungen

Benutzerzuordnungen erfordern einen relativ hohen Verwaltungsaufwand. Das vorliegende Szenario veranschaulicht, wie Sie gesicherte Verbindungen föderierter Systeme konfigurieren können, ohne dass dazu für das föderierte System Benutzerzuordnungen erforderlich werden.

Anforderungen an Benutzer-IDs und Kennwörter

Das Einrichten gesicherter Kontexte mit föderierten Systemen ohne die Verwendung von Benutzerzuordnungen setzt voraus, dass die Benutzer-IDs und Kennwörter folgende Anforderungen erfüllen:

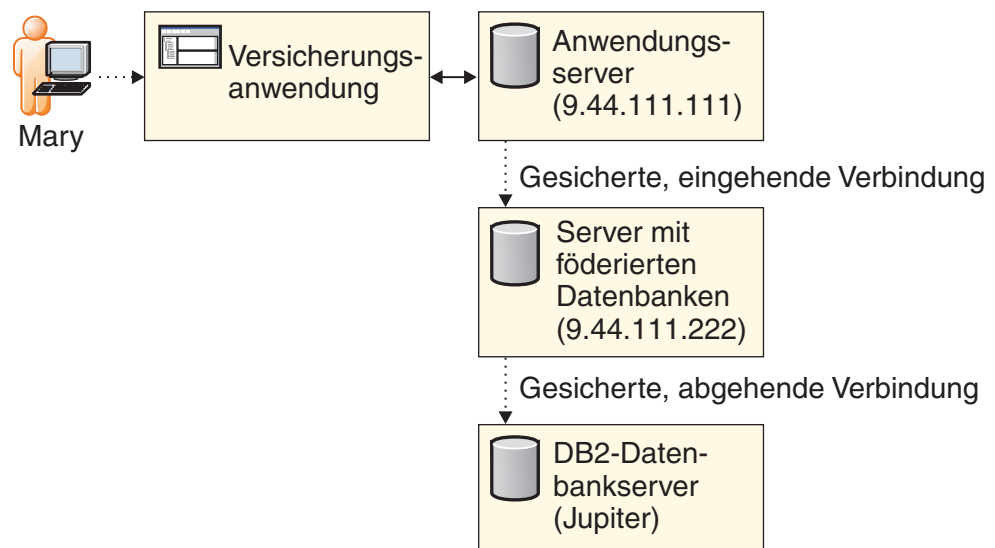
- Die Benutzer-ID und das Kennwort des Erstellers der Verbindung müssen auf dem Server mit föderierten Datenbanken verfügbar sein und mit den entsprechenden Angaben der Datenquelle übereinstimmen. Die Berechtigungsnachweise werden in einer Anweisung CONNECT, die der Ersteller der Verbindung absetzt, oder in einem API-Aufruf der Anwendung angegeben. Bei Verwendung der Anweisung CONNECT wird eine implizite gesicherte Verbindung hergestellt, und die Verbindung kann nicht wiederverwendet werden. Bei einem API-Aufruf der Anwendung wird, sofern eine gesicherte Verbindung explizit angefordert, eine explizite gesicherte Verbindung hergestellt und die Verbindung kann wiederverwendet werden.
- Ermöglicht der ferne gesicherte Kontext die Wiederverwendung von Verbindungen ohne Authentifizierung, müssen Benutzer, die die Verbindung wiederverwenden, auf dem Server mit föderierten Datenbanken und der fernen Datenquelle über dieselbe Benutzer-ID verfügen.

- Ermöglicht der ferne gesicherte Kontext die Wiederverwendung von Verbindungen mit Authentifizierung, müssen Benutzer, die die Verbindung wiederverwenden, auf dem Server mit föderierten Datenbanken und der fernen Datenquelle über denselben Benutzernamen und dasselbe Kennwort verfügen.

Szenario

Die folgende Abbildung veranschaulicht ein typisches vielschichtiges, föderiertes System, das für die Verwendung gesicherter End-to-End-Verbindungen föderierter Systeme konfiguriert ist. Dieses Szenario beinhaltet einen Anwendungsserver, eine gesicherte Verbindung kann jedoch von einem beliebigen Datenbankclient hergestellt werden.

Ersteller der Verbindung: BOSS



In diesem Szenario sind zwei Benutzer vorgesehen, für die keine Benutzerzuordnungen erforderlich sind:

- BOSS verfügt auf dem Server mit föderierten Datenbanken und der Datenquelle über dieselbe Benutzer-ID und dasselbe Kennwort. Für den Benutzer BOSS ist deshalb keine Benutzerzuordnung erforderlich.
- Mary verwendet auf dem Server mit föderierten Datenbanken und der Datenquelle dieselbe Benutzer-ID, und der föderierte Kontext ermöglicht es ihr, eine Verbindung ohne Authentifizierung herzustellen. Auch für die Benutzerin Mary ist aus diesem Grund keine Benutzerzuordnung erforderlich.

In dem Szenario sind drei Server vorgesehen:

- Der Anwendungsserver mit der Versicherungsanwendung und der IP-Adresse 9.44.111.111
- Der Server mit föderierten Datenbanken mit der IP-Adresse 9.44.111.222
- Der ferne DB2-Datenbankserver, der auf dem Server mit föderierten Datenbanken als JUPITER katalogisiert ist.

Die für dieses Szenario erforderlichen Konfigurationsschritte werden im Folgenden erläutert.

Anmerkung: In den Befehlen werden variable Objektnamen kursiv dargestellt. Bei der Implementierung gesicherter Kontexte müssen Sie Variablennamen angeben, die für Ihre spezielle Systemkonfiguration gelten.

1. Erstellen Sie auf dem fernen DB2-Datenbankserver das folgende gesicherte Kontextobjekt:

```
CREATE TRUSTED CONTEXT MY_DB2_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer BOSS der Ersteller der gesicherten Verbindung ist und die Verbindung von der IP-Adresse 9.44.111.222 (dem Server mit föderierten Datenbanken) angefordert werden muss. Der gesicherte Kontext gibt an, dass keine Authentifizierung erforderlich ist (WITH USE FOR PUBLIC WITHOUT AUTHENTICATION). Nach dem Einrichten der gesicherten Verbindung kann die Verbindung demzufolge über eine beliebige, auf der fernen Datenquelle gültige Benutzer-ID wiederverwendet werden, indem lediglich die betreffende Benutzer-ID eingegeben wird.

2. Erstellen Sie auf dem Server mit föderierten Datenbanken folgendes gesichertes Kontextobjekt:

```
CREATE TRUSTED CONTEXT MY_WFS_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.111')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer BOSS der Ersteller der gesicherten Verbindung ist und die Verbindung von der IP-Adresse 9.44.111.111 (dem Anwendungsserver) angefordert werden muss. Nach dem Einrichten der gesicherten Verbindung kann die Verbindung über eine beliebige, auf dem Server mit föderierten Datenbanken gültige Benutzer-ID wiederverwendet werden, indem lediglich die betreffende Benutzer-ID eingegeben wird.

3. Erstellen Sie auf dem Server mit föderierten Datenbanken folgende Serverdefinition:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb', ...);
```

Diese Serverdefinition enthält die Informationen, die der Server mit föderierten Datenbanken für die Verbindung zu der fernen DB2-Datenbank 'remotedb' benötigt.

Schrittweiser Ablauf des Szenarios

Im Folgenden wird das Herstellen gesicherter Verbindung und Wechseln von Benutzer-IDs innerhalb dieses Szenarios kurz in einzelnen Schritten beschrieben. Der für das Szenario verwendete Code beinhaltet Kommentare, die die Ausführung der Tasks durch die Anwendung erläutern.

1. Der Anwendungsserver fordert eine gesicherte, eingehende Verbindung für den Benutzer BOSS an.
2. Der Benutzer BOSS führt eine Task aus, und der Server mit föderierten Datenbanken richtet eine explizite gesicherte, abgehende Verbindung für BOSS ein. Die Benutzer-ID BOSS wird vom Anwendungsserver über den Server mit föderierten Datenbanken an den DB2-Datenbankserver weitergegeben, auf dem die von BOSS ausgeführten Aktionen protokolliert werden können.

3. Die Benutzerin Mary meldet sich bei der auf ihrem Laptop gespeicherten Versicherungsanwendung an. Der Anwendungsserver wechselt bei der eingehenden Verbindung zu dem Server mit föderierten Datenbanken vom Benutzer BOSS zur Benutzerin Mary.
4. Die Benutzerin Mary führt eine Task innerhalb der Anwendung aus.
5. Der Server mit föderierten Datenbanken wechselt bei der abgehenden Verbindung von BOSS zu Mary, und die ID der Benutzerin Mary wird über den Server mit föderierten Datenbanken an den DB2-Server weitergegeben, auf dem die von Mary ausgeführten Aktionen protokolliert werden können.

Beispielcode für Szenarios mit gesicherten End-to-End-Verbindungen föderierter Systeme

Dieser Beispielcode veranschaulicht die Verwendung von APIs in Anwendungen, bei denen gesicherte End-to-End-Verbindungen föderierter Systeme genutzt werden.

Anwendungen benötigen APIs zum Anfordern expliziter gesicherter, eingehender Verbindungen und zum Wechseln von Benutzer-IDs für eine Verbindung. Der Beispielcode veranschaulicht die für diese Tasks verantwortlichen Bestandteile der Anwendung. In den beiden Szenarios mit gesicherten End-to-End-Kontexten wird dieselbe Anwendung verwendet.

Bei dem Anwendungsauszug werden die APIs der Befehlszeilenschnittstelle eingesetzt. APIs sind auch für Anwendungen verfügbar, die Java oder ODBC/CLI in einer XA-Umgebung verwenden.

```
//Definieren Sie das Attribut für gesicherte Verbindungen.
SQLSetConnectAttr(h1, SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER);

//Richten Sie eine gesicherte, eingehende Verbindung für BOSS ein.
SQLConnect(h1, "testdb", SQL_NTS, "BOSS", SQL_NTS, "*****", SQL_NTS);

//Richten Sie eine gesicherte, abgehende Verbindung für BOSS ein.
Arbeiten Sie mit der ID BOSS.
SQLExecDirect(hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS);
...
//Führen Sie ein Commit für die Aktionen durch.
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);

//Wechseln Sie an der Transaktionsgrenze innerhalb der gesicherten
Verbindung bei der Benutzer-ID
//der eingehenden Verbindung zur Benutzerin Mary.
SQLSetConnectAttr(h1, SQL_ATTR_TRUSTED_CONTEXT_USERID, "Mary", SQL_IS_POINTER);

//Wechseln Sie innerhalb der gesicherten Verbindung bei der Benutzer-ID
der abgehenden Verbindung zur Benutzerin Mary.
//Arbeiten Sie mit der Benutzer-ID der Benutzerin Mary.
SQLExecDirect(*hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS)

...
//Führen Sie ein Commit für die Aktionen durch.
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);

//Trennen Sie die Verbindung zu der Datenbank.
SQLDisconnect(h1);
```

Szenario: Gesicherte End-to-End-Verbindungen föderierter Systeme mit Benutzerzuordnungen

Benutzerzuordnungen sind erforderlich, wenn Benutzer auf dem Server mit föderierten Datenbanken und der fernen Datenquelle nicht dieselbe Benutzer-ID und dasselbe Kennwort verwenden. Im vorliegenden Szenario erstellen Sie gesicherte Benutzerzuordnungen und konfigurieren gesicherte Kontexte.

Anforderungen an Benutzerzuordnungen

Der Server mit föderierten Datenbanken empfängt Anforderungen für eingehende Verbindungen und sendet Anforderungen für abgehende Verbindungen an eine ferne Datenquelle. Wenn Benutzer auf dem Server mit föderierten Datenbanken und dem fernen DB2-Datenbankserver dieselbe Benutzer-ID und dasselbe Kennwort verwenden, ist keine Benutzerzuordnung erforderlich. Stimmen die Benutzerberechtigungs-nachweise jedoch nicht überein, ist eine Benutzerzuordnung erforderlich. Benutzerzuordnungen ordnen die Benutzer-ID eines Benutzers, die auf dem Server mit föderierten Datenbanken verwendet wird, zu der Benutzer-ID des Benutzers (und dem zugehörigen Kennwort, soweit angegeben) auf dem fernen Datenbankserver zu.

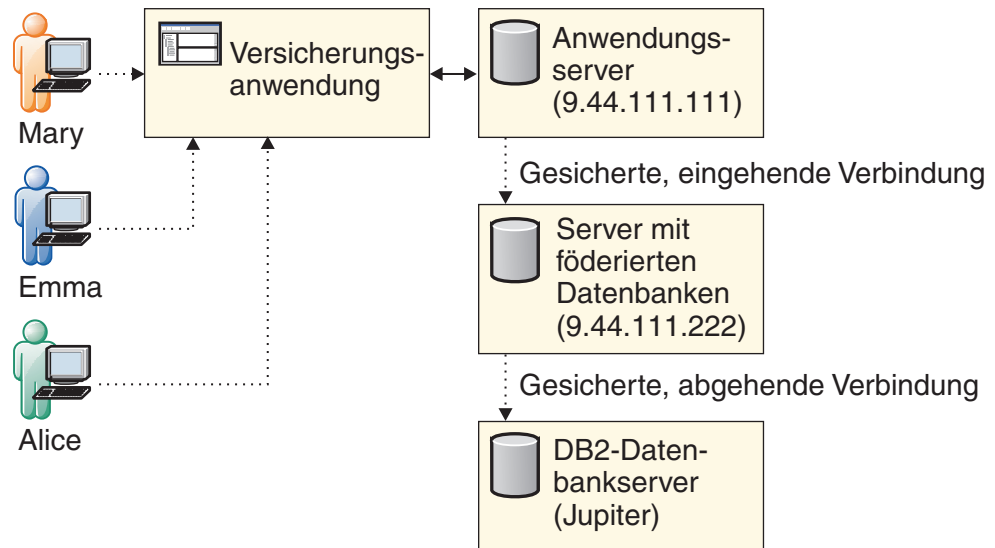
In einem föderierten System, das gesicherte End-to-End-Kontexte verwendet, sind *gesicherte Benutzerzuordnungen* für alle Benutzer erforderlich, die auf dem Server mit föderierten Datenbanken und dem fernen DB2-Datenbankserver unterschiedliche Namen und Kennwörter verwenden. Gesicherte Benutzerzuordnungen geben an, dass der jeweilige Benutzer über die Berechtigung zur Verwendung eines gesicherten Kontexts verfügt. Definieren Sie zum Erstellen einer gesicherten Benutzerzuordnung oder zum Ändern einer vorhandenen Benutzerzuordnung die Benutzerzuordnungsoption `USE_TRUSTED_CONTEXT` mit 'Y'.

Die Gruppe der Benutzer, die gesicherte Benutzerzuordnungen erstellen und ändern können, ist genau eingegrenzt. Es können nur Benutzer mit der Berechtigung `SECADM` gesicherte Benutzerzuordnungen erstellen und löschen sowie vorhandene Benutzerzuordnungen durch Hinzufügen, Definieren oder Löschen der Benutzerzuordnungsoption `USE_TRUSTED_CONTEXT` ändern. Benutzer mit einer gesicherten Benutzerzuordnung können lediglich die Option `REMOTE_PASSWORD` der eigenen Benutzerzuordnung ändern.

Szenario

Diese einfache Grafik veranschaulicht ein typisches vielschichtiges, föderiertes System, bei dem Benutzerzuordnungen verwendet werden. Dieses Szenario beinhaltet einen Anwendungsserver, eine gesicherte Verbindung kann jedoch von einem beliebigen Datenbankclient hergestellt werden.

Ersteller der Verbindung: BOSS



In diesem Szenario gibt es vier Benutzer:

- Der Benutzer BOSS, der Ersteller der Verbindung, verwendet auf dem Server mit föderierten Datenbanken und dem fernen DB2-Datenbankserver dieselbe Benutzer-ID und dasselbe Kennwort. Der Benutzer BOSS verfügt deshalb nicht über eine Benutzerzuordnung.
- Die Benutzerin Mary verfügt nicht über eine Benutzerzuordnung. Sie verwendet auf dem Server mit föderierten Datenbanken und dem DB2-Datenbankserver dieselbe Benutzer-ID. Da der gesicherte Kontext angibt, dass PUBLIC die Verbindung ohne Authentifizierung wiederverwenden kann, muss die Benutzerin Mary kein Kennwort eingeben.
- Die Benutzerin Alice verfügt über eine gesicherte Benutzerzuordnung, in der die Option REMOTE_AUTHID angegeben ist. Die Benutzerin Alice verwendet auf dem Server mit föderierten Datenbanken und dem DB2-Datenbankserver unterschiedliche Benutzer-IDs. Da der gesicherte Kontext angibt, dass die Verbindung von allen Benutzern ohne Authentifizierung wiederverwendet werden kann, muss die Benutzerin Alice kein Kennwort eingeben.
- Auf dem Server mit föderierten Datenbanken verwendet die Benutzerin Emma die Benutzer-ID EMMA. Diese Benutzer-ID ist auf dem DB2-Datenbankserver der Benutzer-ID EGREENE und dem Kennwort MYPASS zugeordnet. Da der gesicherte Kontext eine Authentifizierung der Benutzerin Emma verlangt, verfügt Emma über eine gesicherte Benutzerzuordnung, in der sowohl die Option REMOTE_AUTHID als auch die Option REMOTE_PASSWORD angegeben ist.

In dem Szenario sind drei Server vorgesehen:

- Der Anwendungsserver mit der Versicherungsanwendung und der IP-Adresse 9.44.111.111
- Der Server mit föderierten Datenbanken mit der IP-Adresse 9.44.111.222
- Der ferne DB2-Datenbankserver, der auf dem Server mit föderierten Datenbanken als JUPITER katalogisiert ist

Der Benutzer mit der ID SECADM führt zur Konfiguration dieses Szenarios folgende Schritte aus:

1. Erstellen des gesicherten Kontextobjekts auf dem fernen DB2-Datenbankserver:

```
CREATE TRUSTED CONTEXT MY_DB2_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR EMMA WITH AUTHENTICATION,
PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer BOSS der Ersteller der gesicherten Verbindung ist und die Verbindung von dem Server mit föderierten Datenbanken mit der IP-Adresse 9.44.111.222 angefordert werden muss. Nach dem Einrichten der gesicherten Verbindung kann die Verbindung über eine beliebige, zur Verwendung der Datenbanken berechnigte Benutzer-ID wiederverwendet werden, indem lediglich die betreffende Benutzer-ID eingegeben wird.

2. Erstellen des gesicherten Kontextobjekts auf dem Server mit föderierten Datenbanken:

```
CREATE TRUSTED CONTEXT MY_WFS_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.111')
WITH USE FOR EMMA WITH AUTHENTICATION,
PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer BOSS der Ersteller der gesicherten Verbindung ist und die Verbindung von der IP-Adresse 9.44.111.111 (dem Anwendungsserver) angefordert werden muss. Nach dem Einrichten der gesicherten Verbindung kann die Verbindung von der Benutzerin Emma wiederverwendet werden. Sie muss dazu jedoch eine Authentifizierung vornehmen. Bei allen zur Datenbankverwendung berechtigten Benutzer-IDs ist für die Wiederverwendung der Verbindung lediglich die Eingabe der Benutzer-ID erforderlich.

3. Erstellen der folgenden Serverdefinition auf dem Server mit föderierten Datenbanken:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda
OPTIONS(DBNAME 'remotedb', ...);
```

Diese Serverdefinition enthält die Informationen, die der Server mit föderierten Datenbanken für die Verbindung zu der fernen DB2-Datenbank 'remotedb' benötigt.

4. Erstellen der folgenden gesicherten Benutzerzuordnung für die Benutzerin Alice auf dem Server mit föderierten Datenbanken:

```
CREATE MAPPING FOR USER ALICE
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'AJACKSON', USE_TRUSTED_CONTEXT 'Y');
```

Diese Benutzerzuordnung gibt an, dass gesicherte Verbindungen über die Benutzer-ID ALICE, die der Benutzer-ID AJACKSON auf dem fernen DB2-Datenbankserver zugeordnet ist, wiederverwendet werden können.

5. Erstellen der folgenden gesicherten Benutzerzuordnung für die Benutzerin Emma auf dem Server mit föderierten Datenbanken:

```
CREATE MAPPING FOR USER EMMA
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'EGREENE', REMOTE_PASSWORD 'MYPASS', USE_TRUSTED_CONTEXT 'Y');
```


Diese Benutzerzuordnung gibt an, dass gesicherte Verbindungen über die Benutzer-ID EMMA, die der Benutzer-ID EGREENE und dem Kennwort MYPASS auf dem fernen DB2-Datenbankserver zugeordnet ist, wiederverwendet werden können.

Schrittweiser Ablauf des Szenarios

1. Der Anwendungsserver fordert eine gesicherte, eingehende Verbindung für den Benutzer BOSS an.
2. Der Benutzer BOSS führt eine Task aus, und die Benutzer-ID BOSS wird über den Server mit föderierten Datenbanken an den DB2-Datenbankserver weitergegeben, auf dem die von BOSS ausgeführten Aktionen protokolliert werden können.
3. Die Benutzerin Emma meldet sich bei der vom Anwendungsserver bereitgestellten Versicherungsanwendung an. Der Anwendungsserver fordert nach der Authentifizierung der Benutzerin Emma einen Benutzerwechsel bei der eingehenden Verbindung des föderierten Systems vom Benutzer BOSS zur Benutzerin Emma an.
4. Die Benutzerin Emma führt eine Task innerhalb der Anwendung aus.
5. Der Server mit föderierten Datenbanken wechselt bei der abgehenden Verbindung des föderierten Systems vom Benutzer BOSS zur Benutzerin Emma, und die ID der Benutzerin Emma wird über den Server mit föderierten Datenbanken an den DB2-Server weitergegeben, auf dem die von der ID EGREENE (der fernen Benutzer-ID der Benutzerin Emma) ausgeführten Aktionen protokolliert werden können.
6. Die Benutzerin Alice meldet sich bei der Versicherungsanwendung an. Der Anwendungsserver fordert ohne vorherige Authentifizierung der Benutzerin Alice einen Benutzerwechsel bei der eingehenden Verbindung des föderierten Systems von der Benutzerin Emma zur Benutzerin Alice an.
7. Die Benutzerin Alice führt eine Task innerhalb der Anwendung aus.
8. Der Server mit föderierten Datenbanken wechselt bei der abgehenden Verbindung des föderierten Systems von der Benutzerin Emma zur ID AJACKSON (der der Benutzerin Alice zugeordneten Benutzer-ID), und die ID der Benutzerin Alice wird über den Server mit föderierten Datenbanken an den DB2-Datenbankserver weitergegeben, auf dem die von AJACKSON ausgeführten Aktionen protokolliert werden können.
9. Die Benutzerin Mary meldet sich bei der Versicherungsanwendung an. Die Benutzerin Mary muss sich nicht authentifizieren. Der Anwendungsserver wechselt deshalb bei der gesicherten, eingehenden Verbindung des föderierten Systems zur Benutzerin Mary, ohne ein Kennwort bereitzustellen.
10. Die Benutzerin Mary führt eine Task innerhalb der Anwendung aus.
11. Der Server mit föderierten Datenbanken wechselt bei der abgehenden Verbindung des föderierten Systems von AJACKSON zur Benutzerin Mary, und die ID der Benutzerin Mary wird über den Server mit föderierten Datenbanken an den DB2-Datenbankserver weitergegeben, auf dem die von der Benutzerin Mary ausgeführten Aktionen protokolliert werden können.

Szenario: Gesicherte, abgehende Verbindungen föderierter Systeme

Eine gesicherte, abgehende Verbindung eines föderierten Systems stellt eine gesicherte Umgebung für den Datenaustausch zwischen dem Server mit föderierten

Datenbanken und einem fernen DB2-Datenbankserver bereit. Da für derartige Verbindungen keine Authentifizierung erforderlich ist, erübrigt sich das Speichern und Verwalten von Benutzerkennwörtern.

In einigen Konfigurationen muss der Server mit föderierten Datenbanken eingehende Verbindungen zulassen, die nicht gesichert sind. Eine Verbindung gilt als nicht gesichert, wenn eine der folgenden Aussagen zutrifft:

- Die Attribute der eingehenden Verbindungsanforderung entsprechen nicht den Attributen eines gesicherten Kontextobjekts auf dem Server mit föderierten Datenbanken.
- Der Server mit föderierten Datenbanken gibt keinen gesicherten Kontext für den Server an, von dem die Verbindungsanforderung ausgeht. Alle Benutzer erhalten nicht gesicherte, eingehende Verbindungen.

Anforderungen an gesicherte Kontexte, Serverdefinitionen und Benutzerzuordnungen

Wenn Sie den gesicherten Kontext auf dem Server erstellen, der als ferne DB2-Datenquelle verwendet wird, müssen Sie `WITH USE FOR PUBLIC WITHOUT AUTHENTICATION` angeben. Dies bewirkt, dass alle Benutzer, die auf dem Server mit föderierten Datenbanken und dem DB2-Datenbankserver dieselbe Benutzer-ID verwenden, den gesicherten Kontext für abgehende Verbindungen ohne Authentifizierung verwenden können.

Geben Sie zum Herstellen einer gesicherten, abgehenden Verbindung des Servers mit föderierten Datenbanken die Option `FED_PROXY_USER` in der Serverdefinition für den fernen DB2-Datenquellenserver an. Diese Option kennzeichnet die Berechtigungs-ID des Benutzers, von dem die gesicherte, abgehende Verbindung ausgeht. Erstellen Sie darüber hinaus eine Benutzerzuordnung für den Proxybenutzer des föderierten Systems. Diese Benutzerzuordnung muss die Option `REMOTE_AUTHID` und die Option `REMOTE_PASSWORD` enthalten, da der gesicherte Kontext bei der Datenquelle erfordert, dass die Berechtigung des Erstellers der Verbindung abgefragt wird.

Es können nur Benutzer mit der Berechtigung `SECADM` Serverdefinitionen ändern, bei denen die Option `FED_PROXY_USER` definiert ist. Darüber hinaus gilt, dass die Anweisung `SET SERVER OPTION` nicht für die Serveroption `FED_PROXY_USER` verwendet werden kann.

Es können Situationen vorkommen, in denen Sie unterschiedliche Gruppen von Benutzern konfigurieren möchten, die eine Verbindung über unterschiedliche Proxybenutzer-IDs herstellen. Möglicherweise möchten Sie z. B. unterschiedliche Rollen für verschiedene Benutzer konfigurieren. Eine einfache Möglichkeit, dies zu erreichen, besteht darin, bei den einzelnen Benutzerzuordnungen, die den Standardproxybenutzer des föderierten Systems nicht verwenden, die Option `FED_PROXY_USER` anzugeben und die Option `USE_TRUSTED_CONTEXT` mit 'Y' zu definieren. Ist die Option `FED_PROXY_USER` sowohl in der Serverdefinition als auch in der Benutzerzuordnung angegeben, überschreibt die Benutzerzuordnung den Wert in der Serverdefinition.

Die Option `FED_PROXY_USER` kann nur von Benutzern mit der Berechtigung `SECADM` hinzugefügt, gelöscht oder definiert werden. Das Gleiche gilt für das Erstellen und Löschen von Benutzerzuordnungen, die die Option `FED_PROXY_USER` enthalten.

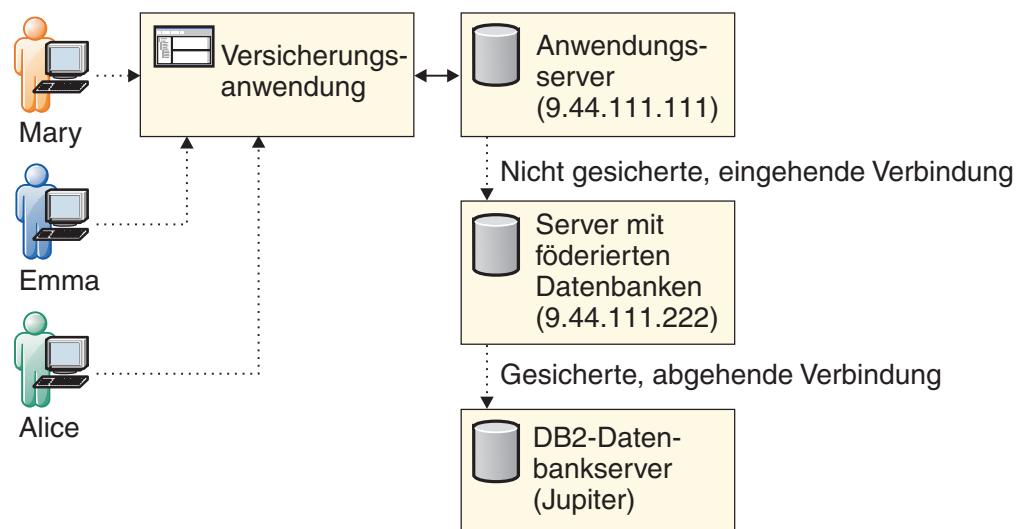
Wichtig:

Es gibt nur zwei Situationen, in denen für den Benutzer einer gesicherten, abgehenden Verbindung eine Benutzerzuordnung erforderlich ist: Der Benutzer verwendet unterschiedliche Benutzer-IDs auf dem Server mit föderierten Datenbanken und dem Datenbankserver, oder der Benutzer muss eine Verbindung über einen Proxybenutzer des föderierten Systems herstellen, bei dem es sich nicht um den Standardproxybenutzer des föderierten Systems handelt.

Szenario

Die folgende Abbildung veranschaulicht ein Szenario, für das gesicherte, abgehende Verbindungen des föderierten Systems erforderlich sind. Dieses Szenario beinhaltet einen Anwendungsserver, eine gesicherte Verbindung kann jedoch von einem beliebigen Datenbankclient hergestellt werden.

Proxybenutzer des föderierten Systems: BOSS und ADM



In diesem Szenario gibt es fünf Benutzer:

- BOSS, der Standardproxybenutzer des föderierten Systems, verfügt über eine Benutzerzuordnung.
- ADM, ein Proxybenutzer des föderierten Systems, verfügt über eine Benutzerzuordnung.
- Mary, die die Versicherungsanwendung verwenden möchte, verfügt nicht über eine Benutzerzuordnung.
- Emma und Alice, die ebenfalls die Versicherungsanwendung verwenden möchten, verfügen über Benutzerzuordnungen.

In dem Szenario sind drei Server vorgesehen:

- Der Anwendungsserver mit der Versicherungsanwendung und der IP-Adresse 9.44.111.111. Im Szenario wird ein Anwendungsserver verwendet, es kann jedoch ein auch ein beliebiger Client verwendet werden.
- Der Server mit föderierten Datenbanken mit der IP-Adresse 9.44.111.222
- Der ferne DB2-Datenbankserver, der auf dem Server mit föderierten Datenbanken als JUPITER katalogisiert ist

Die Konfiguration beinhaltet folgende Schritte:

Anmerkung: In den Befehlen werden variable Objektnamen kursiv dargestellt. Bei der Implementierung gesicherter Kontexte müssen Sie Variablennamen angeben, die für Ihre spezielle Systemkonfiguration gelten.

1. Erstellen Sie auf dem fernen DB2-Datenbankserver die folgenden gesicherten Kontextobjekte:

```
CREATE TRUSTED CONTEXT MY_DB2_TXC
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer *BOSS* der Ersteller der Verbindung ist und die gesicherte, abgehende Verbindung von dem Server mit föderierten Datenbanken mit der IP-Adresse *9.44.111.222* angefordert werden muss. Nach dem Einrichten der gesicherten, abgehenden Verbindung kann die Verbindung von einem beliebigen Benutzer ohne Authentifizierung verwendet werden.

```
CREATE TRUSTED CONTEXT MY_DB2_TXC_ALICE
BASED UPON CONNECTION USING
SYSTEM AUTHID ADM
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION ROLE Manager
ENABLE
```

Dieser gesicherte Kontext gibt an, dass der Benutzer *ADM* der Ersteller der Verbindung ist und die gesicherte, abgehende Verbindung von dem Server mit föderierten Datenbanken mit der IP-Adresse *9.44.111.222* angefordert werden muss. Nach dem Einrichten der gesicherten, abgehenden Verbindung kann die Verbindung von der Benutzerin *Alice* ohne Authentifizierung verwendet werden. *Alice* erhält für diese gesicherte Verbindung die Rolle 'Manager'.

2. Erstellen Sie auf dem Server mit föderierten Datenbanken folgende Serverdefinition:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb',FED_PROXY_USER 'BOSS');
```

Diese Serverdefinition enthält die Informationen, die der Server mit föderierten Datenbanken für die Verbindung zu der fernen DB2-Datenbank '*remotedb*' benötigt. Die Definition gibt an, dass der Standardproxybenutzer des föderierten Systems *BOSS* die gesicherte, abgehende Verbindung einrichtet, wenn die eingehende Verbindung zum Server mit föderierten Datenbanken nicht gesichert ist.

3. Erstellen Sie auf dem Server mit föderierten Datenbanken die Benutzerzuordnungen für die Proxybenutzer des föderierten Systems:

```
CREATE USER MAPPING FOR BOSS SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'BOSS',REMOTE_PASSWORD 'MYPASS');
CREATE USER MAPPING FOR ADM SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'ADM', REMOTE_PASSWORD 'PWD');
```

4. Erstellen Sie auf dem Server mit föderierten Datenbanken die folgenden gesicherten Benutzerzuordnungen für die Benutzerinnen *Emma* und *Alice*.

```
CREATE USER MAPPING FOR EMMA SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'EGREENE', USE_TRUSTED_CONTEXT 'Y')
CREATE USER MAPPING FOR ALICE SERVER JUPITER
OPTIONS (USE_TRUSTED_CONTEXT 'Y',FED_PROXY_USER 'ADM');
```

Bei beiden Benutzerzuordnungen ist die Option `USE_TRUSTED_CONTEXT` mit '*Y*' definiert, sodass sie den gesicherten Kontext verwenden können. Darüber hinaus sind für die Benutzerinnen *Emma* und *Alice* folgende zusätzliche Optionen erforderlich:

- Für Emma ist eine Benutzerzuordnung erforderlich, da sie auf dem Server mit föderierten Datenbanken und dem DB2-Datenbankserver über unterschiedliche Benutzer-IDs verfügt. Ihre Benutzerzuordnung enthält deshalb die Option `REMOTE_AUTHID`.
- Die Benutzerin Alice erfordert eine Benutzerzuordnung, da der gesicherte Kontext angibt, dass sie als Proxybenutzer-ID des föderierten Systems die ID `ADM` verwendet. Ihre Benutzerzuordnung enthält deshalb die Option `FED_PROXY_USER`.

In diesem Szenario ist für die Benutzerin Mary keine Benutzerzuordnung erforderlich. Die Benutzerzuordnungen für die Benutzerinnen Emma und Alice enthalten das ferne Kennwort nicht. Diese Benutzerzuordnungen müssen demzufolge nicht kontinuierlich aktualisiert werden, um das ferne Kennwort auf dem aktuellen Stand zu halten.

Schrittweiser Ablauf des Szenarios

Der nachfolgend skizzierte Ablauf beschreibt, wie die gesicherten, abgehenden Verbindungen in diesem Szenario eingesetzt werden.

1. Die Anwendung stellt eine nicht gesicherte, eingehende Verbindung zum Server mit föderierten Datenbanken her:

```
CONNECT TO FEDSVR USER MARY USING '****'
```
2. Die erste Anforderung an das föderierte System, die auf den Server JUPITER zugreift, stellt eine abgehende Verbindung zu JUPITER her:

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```
3. Da für JUPITER `FED_PROXY_USER=BOSS` angegeben ist und die Benutzerzuordnung der Benutzerin Mary keinen Proxybenutzer für das föderierte System angibt, wird die abgehende Verbindung für `BOSS` hergestellt und anschließend sofort zu der aktuellen Benutzerin der eingehenden Verbindung, in diesem Fall also Mary, gewechselt.
4. Die Anforderung der Benutzerin Mary an das föderierte System wird bearbeitet und im Prüfprotokoll unter dem Namen `MARY` aufgezeichnet. Die Verbindung wird anschließend zurückgesetzt.
5. Für die Benutzerin Emma wird eine nicht gesicherte, eingehende Verbindung zu dem Server mit föderierten Datenbanken hergestellt:

```
CONNECT TO FEDSVR USER EMMA USING '****'
```
6. Die erste Anforderung an das föderierte System über die aktuelle Verbindung, die auf JUPITER zugreift, stellt eine abgehende Verbindung des föderierten Systems zu JUPITER her.

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```
7. Die abgehende Verbindung wird mit der ID `BOSS` hergestellt. Anschließend wird sofort zu der aktuellen Benutzerin der eingehenden Verbindung, in diesem Fall also Emma, gewechselt, deren ID `EGREENE` zugeordnet ist.
8. Die Anforderung der Benutzerin Emma an das föderierte System wird bearbeitet und im Prüfprotokoll unter dem Namen `EGREENE` aufgezeichnet. Die Verbindung wird anschließend zurückgesetzt.
9. Für die Benutzerin Alice wird eine nicht gesicherte eingehende Verbindung zu dem Server mit föderierten Datenbanken hergestellt:

```
CONNECT TO FEDSVR USER ALICE USING '****'
```
10. Die erste Anforderung an das föderierte System über die aktuelle Verbindung, die auf JUPITER zugreift, stellt eine abgehende Verbindung des föderierten

Systems zu JUPITER her. Da die gesicherte Benutzerzuordnung der Benutzerin Alice angibt, dass sie anstelle der Standardproxybenutzer-ID BOSS die Proxybenutzer-ID ADM des föderierten Systems verwendet, wird die abgehende Verbindung mit der ID ADM hergestellt. Anschließend wird sofort zu der ID 'Alice', die die aktuelle ID für die eingehende Verbindung darstellt, gewechselt.

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```

- Die Anforderung der Benutzerin Alice an das föderierte System wird bearbeitet und kann im Prüfprotokoll aufgezeichnet werden. Die Verbindung wird anschließend zurückgesetzt.

Benutzerzuordnungen und gesicherte Verbindungen föderierter Systeme

Die folgenden Tabellen erläutern, wie Benutzerzuordnungen mit oder ohne ferne IDs und ferne Kennwörter in gesicherten End-to-End-Verbindungen und gesicherten, abgehenden Verbindungen föderierter Systeme verwendet werden.

Die Tabellen beschreiben die möglichen Benutzerzuordnungen für den Benutzer BOSS, den Ersteller der Verbindung, und die Benutzerin Mary, die die Verbindung wiederverwendet. In den Tabellen bezieht sich der Begriff *nicht gesicherte Verbindung* auf eine reguläre Verbindung, die sowohl als eingehende Verbindung als auch als abgehende Verbindung nicht gesichert ist.

In einigen Zellen der Tabelle wird die Angabe *soweit verfügbar* in Bezug auf Kennwörter verwendet. Kennwörter sind auf dem Server mit föderierten Datenbanken verfügbar, wenn sowohl eine Benutzer-ID als auch ein Kennwort explizit im Rahmen der Verbindung angegeben sind. Wenn Sie Verbindungen zum Server mit föderierten Datenbanken über API-Verbindungsaufrufe herstellen, werden Kennwörter explizit übergeben. Bei CLI/ODBC z. B. wird das Kennwort über das Verbindungsattribut `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` angegeben. Wenn Sie Verbindungen zum Server mit föderierten Datenbanken über die Anweisung `CONNECT` herstellen, können Sie mit der folgenden Syntax sicherstellen, dass das Kennwort explizit übergeben wird:

```
CONNECT TO datenbankname USER benutzer-id USING kennwort
```

Zuordnungen für Ersteller von Verbindungen und Proxybenutzer föderierter Systeme

Tabelle 23. Benutzerzuordnungen für BOSS (Ersteller der Verbindung)

Benutzerzuordnung für BOSS	Nicht gesicherte Verbindung	Gesicherte End-to-End-Verbindung des föderierten Systems (wobei BOSS die gesicherte, eingehende Verbindung einrichtet)	Gesicherte, abgehende Verbindung des föderierten Systems (wobei die Server- bzw. Benutzerzuordnungsoption <code>FED_PROXY_USER='BOSS'</code> definiert ist)
Keine Benutzerzuordnung	Übergeben Sie die Benutzer-ID des föderierten Systems für BOSS mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	Übergeben Sie die Benutzer-ID des föderierten Systems für BOSS mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	FEHLER SQL1101N

Tabelle 23. Benutzerzuordnungen für BOSS (Ersteller der Verbindung) (Forts.)

Benutzerzuordnung für BOSS	Nicht gesicherte Verbindung	Gesicherte End-to-End-Verbindung des föderierten Systems (wobei BOSS die gesicherte, eingehende Verbindung einrichtet)	Gesicherte, abgehende Verbindung des föderierten Systems (wobei die Server- bzw. Benutzerzuordnungsoption FED_PROXY_USER='BOSS' definiert ist)
Benutzerzuordnung, die lediglich eine ferne Benutzer-ID angibt	Übergeben Sie die ferne Benutzer-ID für BOSS mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	Übergeben Sie die ferne Benutzer-ID für BOSS mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	FEHLER SQL1101N
Benutzerzuordnung, die eine ferne Benutzer-ID und ein fernes Kennwort angibt	Übergeben Sie die ferne Benutzer-ID für BOSS mit dem zugehörigen fernem Kennwort an die ferne Datenquelle.	Übergeben Sie die ferne Benutzer-ID für BOSS mit dem zugehörigen fernem Kennwort an die ferne Datenquelle.	Übergeben Sie die ferne Benutzer-ID für BOSS mit dem fernem Kennwort an die ferne Datenquelle.

Zuordnungen für andere Benutzer (Wiederverwendung der Verbindung)

Tabelle 24. Benutzerzuordnungen für Mary (Wiederverwendung der Verbindung)

Benutzerzuordnung für Mary	Gesicherte End-to-End-Verbindung des föderierten Systems (wobei BOSS die gesicherte, eingehende Verbindung einrichtet und anschließend zur Benutzerin Mary gewechselt wird)	Gesicherte, abgehende Verbindung des föderierten Systems (wobei die Server- bzw. Benutzerzuordnungsoption FED_PROXY_USER='BOSS' definiert ist)
Keine Benutzerzuordnung	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary an die ferne Datenquelle.
Nicht gesicherte Benutzerzuordnung, die lediglich eine ferne Benutzer-ID angibt	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary an die ferne Datenquelle.
Nicht gesicherte Benutzerzuordnung, die eine ferne Benutzer-ID und ein fernes Kennwort angibt	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary mit dem zugehörigen Kennwort des föderierten Systems (soweit verfügbar) an die ferne Datenquelle.	Übergeben Sie die Benutzer-ID des föderierten Systems für Mary an die ferne Datenquelle.
Gesicherte Benutzerzuordnung, die lediglich eine ferne Benutzer-ID angibt	Übergeben Sie die ferne Benutzer-ID für Mary an die ferne Datenquelle.	Übergeben Sie die ferne Benutzer-ID für Mary an die ferne Datenquelle.

Tabelle 24. Benutzerzuordnungen für Mary (Wiederverwendung der Verbindung) (Forts.)

Benutzerzuordnung für Mary	Gesicherte End-to-End-Verbindung des föderierten Systems (wobei BOSS die gesicherte, eingehende Verbindung einrichtet und anschließend zur Benutzerin Mary gewechselt wird)	Gesicherte, abgehende Verbindung des föderierten Systems (wobei die Server- bzw. Benutzerzuordnungsoption FED_PROXY_USER='BOSS' definiert ist)
Gesicherte Benutzerzuordnung, die eine ferne Benutzer-ID und ein fernes Kennwort angibt	Übergeben Sie die ferne Benutzer-ID für Mary mit dem zugehörigen fernen Kennwort an die ferne Datenquelle.	Übergeben Sie die ferne Benutzer-ID für Mary mit dem zugehörigen fernen Kennwort an die ferne Datenquelle.

Kapitel 32. Kennsatzbasierte Zugriffssteuerung (LBAC) und Systeme mit föderierten Datenbanken

Stellen Sie sicher, dass nur Benutzer mit den entsprechenden Berechtigungen in Tabellen enthaltene Daten einsehen können.

Die kennsatzbasierte Zugriffssteuerung (LBAC) ermöglicht es Ihnen, Sicherheitsrichtlinien auf Zeilen und Spalten einer Tabelle anzuwenden. Die einzelnen Sicherheitsrichtlinien geben die Berechtigungsnachweise an, die den einzelnen Benutzer-IDs und Sitzungs-IDs mit Grant erteilt wurden. Angenommen z. B., dass die Tabelle 'Prices' die Spalten 'Wholesale', 'Retail' und 'Sale' enthält. Ist die Benutzerin 'Alice' zu einem Zugriff auf die Spalten 'Retail' und 'Sale' berechtigt, kann die Abfrage `SELECT RETAIL, SALE FROM PRICES` erfolgreich ausgeführt werden. Die Abfrage `SELECT * WHOLESale` schlägt jedoch fehl.

Wenn Sie einen Kurznamen für ein Objekt erstellen, erkennt der Server mit föderierten Datenbanken automatisch, ob die Datenquelle die kennsatzbasierte Zugriffssteuerung verwendet. Ist dies der Fall, wird der Kurzname nicht in den Cache-Speicher gestellt. Für Kurznamen, die vor der Verfügbarkeit der kennsatzbasierten Zugriffssteuerung erstellt wurden, müssen Sie die Anweisung `ALTER NICKNAME` verwenden, um ein Caching zu ermöglichen bzw. zu verhindern. Wenn Sie z. B. einen Kurznamen für ein Datenquellenobjekt erstellt haben, bevor die föderierte Unterstützung für die kennsatzbasierte Zugriffssteuerung verfügbar war, können Sie das Caching durch eine Änderung des Kurznamens verhindern.

Die einzelnen Sicherheitsrichtlinien verfügen jeweils über einen eindeutigen Kennsatz, der in der Kennsatzspalte von Tabellen gespeichert wird. Diese Spalte mit den Kennsätzen kann von einem Datenbankadministrator ausgeblendet werden, um zu verhindern, dass Benutzer von der Existenz dieser Spalte wissen. Kurznamen mit ausgeblendeten Kennsatzspalten werden nicht in den Cache gestellt.

Kapitel 33. Externe Repositories für Benutzerzuordnungen

Durch das Speichern von Benutzerzuordnungen in einem einzelnen externen Repository, das viele Server mit föderierten Datenbanken gemeinsam nutzen, können Sie den Verwaltungsaufwand reduzieren, der ansonsten durch das Speichern von Benutzer-IDs und Kennwörtern auf den einzelnen Servern mit föderierten Datenbanken entstehen würde.

Vom Verwaltungsaufwand her ist das Speichern von Benutzerzuordnungen in einem externen Repository günstiger als das Speichern von Zuordnungen im Katalog. Ferne Kennwörter laufen in regelmäßigen Abständen ab. Wenn Sie Benutzerzuordnungen im Katalog speichern, müssen Kennwörter in der fernen Datenquelle und im Katalog aktualisiert werden, wenn sie abgelaufen sind. Wenn Sie ein externes Repository verwenden, müssen Sie Kennwörter dagegen lediglich einmal, nämlich in dem externen Repository, aktualisieren, wenn sie ablaufen.

Erstellen Sie zum Verwenden eines externen Repositories für Benutzerzuordnungen ein Plug-in für Benutzerzuordnungen. Das Plug-in muss dieselben Sicherheitseinstellungen wie das externe Repository verwenden. Schützen Sie zwischen dem Server mit föderierten Datenbanken und dem externen Repository ausgetauschte Daten mit SSL (Secure Sockets Layer). Geben Sie anschließend beim Erstellen der Konfigurationsdatei für das Plug-in für Benutzerzuordnungen an, dass das Plug-in SSL verwendet. Schränken Sie auch den Zugriff auf den Quellcode des Plug-ins ein, damit die Informationen geschützt bleiben.

Ist die Prüfung aktiviert, wird bei jedem Zugriff des Servers mit föderierten Datenbanken auf das Plug-in für Benutzerzuordnungen ein VALIDATE-Prüfsatz erstellt. Konfigurieren Sie die Funktion 'db2audit' zum Erfassen dieser VALIDATE-Datensätze.

Es stehen folgende Beispiel-Plug-ins mit ausführlichen Anweisungen zum Erstellen, Testen und Implementieren eigener angepasster Plug-ins zur Verfügung:

Plug-in für Benutzerzuordnungen (Programmiersprache C)

Das C-Plug-in besteht aus fünf Funktionen, über die eine Schnittstelle zum Abrufen von Benutzerzuordnungen aus einem externen Repository bereitgestellt wird.

Die Funktionsaufrufe erfolgen in der folgenden Reihenfolge:

1. FSUMPluginInit
2. FSUMconnect
3. FSUMfetchUM
4. FSUMdisconnect
5. FSUMPluginTerm

Initialisieren – FSUMPluginInit

Unmittelbar nach dem Laden der Plug-in-Bibliothek durch den Server mit föderierten Datenbanken ruft der Server die Funktion 'FSUMPluginInit' auf, über die das Plug-in initialisiert wird und die Zeiger für die anderen Funktionen an den Server mit föderierten Datenbanken übergeben werden. Diese Funktionen sind global und

müssen extern aufgelöst werden können. Ist das Plug-in in C++ geschrieben, müssen diese Funktionen mit 'extern "C"' deklariert werden. Der Server mit föderierten Datenbanken übergibt nach Bedarf die Zeiger zu einer Gruppe von Dienstprogrammfunktionen, die das Plug-in anfordern kann.

Das Plug-in wird in einen db2fmp-Threadprozess geladen. Alle Anwendungen, die das Plug-in verwenden, nutzen dieselbe Plug-in-Bibliothek gemeinsam. Diese Bibliothek muss threadsicher sein. Jede Anwendung verwendet einen einzelnen Thread im db2fmp-Prozess. Wenn die Anwendung eine Benutzerzuordnung erfolgreich abgerufen und eine Bereinigung durchgeführt hat, gibt der Server mit föderierten Datenbanken den Thread zur weiteren Verwendung in den Thread-Pool zurück. Da der Server mit föderierten Datenbanken mehrere Anwendungen gleichzeitig ausführen kann, können gleichzeitig mehrere Threads, die dieselbe Plug-in-Bibliothek verwenden, aktiviert sein. Jeder einzelne Thread verfügt über eine separate Verbindungskennung zu dem Repository mit den Benutzerzuordnungen. Die API 'FSUMPluginInit' ermöglicht es den einzelnen Threads darüber hinaus, globale Plug-in-Ressourcen zu bearbeiten, z. B. um die Plug-in-Referenzanzahl zu erhöhen.

Verbindung zum Repository herstellen – FSUMconnect

Der Server mit föderierten Datenbanken ruft die Funktion 'FSUMconnect' auf, um eine Verbindung zum Repository mit Benutzerzuordnungen herzustellen. Das Plug-in muss einen Deskriptor enthalten, in dem alle erforderlichen Informationen für die Verbindung zum Repository gespeichert sind. Diese Informationen können z. B. eine Kennung für eine geöffnete Datei oder eine Verbindungskennung für die Verbindung zu einem LDAP-Server enthalten. Je nachdem, wie die Sicherheit für das externe Repository implementiert ist, kann auch eine Benutzer-ID und ein Kennwort zu diesen Informationen gehören. Sind Berechtigungsnachweise erforderlich, müssen Sie die Verwaltung der Berechtigungsnachweise konfigurieren. Wenn Sie die Berechtigungsnachweise z. B. in einer Konfigurationsdatei speichern, liest das Plug-in die Berechtigungsnachweise in dieser Datei, wenn es versucht, eine Verbindung zum externen Repository herzustellen.

Benutzerzuordnung abrufen – FSUMfetchUM

Das Plug-in ruft zunächst die Funktion 'FSUMfetchUM' auf, um die Benutzerzuordnung aus dem externen Repository abzurufen, und anschließend die Dienstprogrammfunktion 'FSUMaddUMOption', um die ferne ID und das ferne Kennwort an den Server mit föderierten Datenbanken zu senden. Im Repository werden die einzelnen Benutzerzuordnungen durch den Namen der Serverinstanz mit föderierten Datenbanken, den Datenbanknamen, den Namen des fernen Servers sowie die lokale Authentifizierungs-ID gekennzeichnet. Zusätzlich muss jede Benutzerzuordnung die Optionen REMOTE_AUTHID und REMOTE_PASSWORD enthalten. Ist ein fernes Kennwort verschlüsselt, muss das Plug-in dieses Kennwort vor dem Senden des Kennworts an den Server mit föderierten Datenbanken entschlüsseln.

Verbindung zum Repository trennen – FSUMdisconnect

Der Server mit föderierten Datenbanken ruft die Funktion 'FSUMdisconnect' auf, um die Verbindung zum Repository mit Benutzerzuordnungen zu trennen. Zum Trennen der Verbindung hebt diese Funktion die Zuordnung des Threads zum Repository mit Benutzerzuordnungen auf. Dazu schließt die Funktion z. B. möglicherweise eine geöffnete Datei oder eine Verbindung zu einem LDAP-Server.

Globale Ressourcen freigeben – FSUMPluginTerm

Als letzte Funktion ruft der Server mit föderierten Datenbanken die Funktion 'FSUMPluginTerm' auf, um von der Funktion 'FSUMPluginInit' zugeordnete globale Ressourcen, soweit vorhanden, freizugeben. Anschließend hebt diese Funktion die Zuordnung zwischen Thread und Plug-in auf.

Unterstützte Plattformen für das Plug-in für Benutzerzuordnungen (Programmiersprache C)

Vergewissern Sie sich vor dem Erstellen des Plug-ins, dass Sie eine unterstützte Plattform verwenden.

Die nachstehende Tabelle enthält die für das Plug-in für Benutzerzuordnungen unterstützten Plattformen.

Plattform	Dateiname
AIX, 64 Bit	plugin_file_name.a
Linux AMD 64, Power PC 64, Power PC 390	plugin_file_name.so
Microsoft Windows, 32 Bit und 64 Bit	plugin_file_name.dll

Bei der Entwicklung von Plug-ins für Benutzerzuordnungen zu beachtende Einschränkungen (Programmiersprache C)

Bei der Entwicklung von Plug-Ins für Benutzerzuordnungen in der Programmiersprache C sind verschiedene Einschränkungen zu beachten.

Option 'extern "C"'

Die Bibliothek des Plug-ins muss mit einer speziellen Option der Programmiersprache C verknüpft werden. Kopfdatendateien, die die erforderlichen Prototypen, Datenstrukturen und Fehlercodedefinitionen für die Implementierung des Plug-ins bereitstellen, stehen nur für C/C++ zur Verfügung. Funktionen, die beim Laden aufgelöst werden, müssen mit der Option 'extern "C"' deklariert werden, wenn die Bibliothek des Plug-ins als C++ kompiliert ist.

.NET CLR wird nicht unterstützt

.NET CLR (Common Language Runtime) wird beim Kompilieren und Verknüpfen des Quellcodes für die Bibliothek des Plug-ins nicht unterstützt.

Signalroutinen

Die Plug-in-Bibliothek darf keine Signalroutinen installieren und die Signalmaske nicht verändern, da dies zu Überschneidungen mit der Berichterstellung und der Wiederherstellung nach dem Auftreten von Fehlern führen würde. Die Plug-in-Bibliothek darf keine C++-Ausnahmebedingungen auslösen.

Threadsicher

Die Plug-in-Bibliothek muss threadsicher und simultan verwendbar sein. Nur bei der Plug-in-Initialisierung ist eine simultane Verwendbarkeit nicht erforderlich. Die Initialisierungsfunktion für das Plug-in kann mehrfach über verschiedene Threads aufgerufen werden. Das Plug-in bereinigt in diesem Fall jeweils alle verwendeten Ressourcen und initialisiert sich selbst erneut.

Überschreiben von C-Standardbibliotheksaufrufen und -Betriebssystemaufrufen

Die Plug-in-Bibliothek darf C-Standardbibliotheksaufreufe und -Betriebssystemaufrufe nicht überschreiben.

32-Bit- und 64-Bit-Anwendungen

32-Bit-Server mit föderierten Datenbanken müssen ein 32-Bit-Plug-in verwenden. 64-Bit-Server mit föderierten Datenbanken müssen ein 64-Bit-Plug-in verwenden. Bei gemischten Instanzen mit 32-Bit-Client und 64-Bit-Server muss ein 64-Bit-Plug-in verwendet werden.

Textzeichenfolgen

Eingabezeichenfolgen enden nicht zwingend auf Null, und Ausgabezeichenfolgen müssen nicht unbedingt auf Null enden. Es werden stattdessen ganzzahlige Längenangaben für alle Eingabezeichenfolgen vorgegeben sowie Zeiger auf ganze Zahlen für zurückzugebende Längen.

Kopfdatendatei 'fsumplugin.h' für das Plug-in für Benutzerzuordnungen (Programmiersprache C)

Die Kopfdatendatei 'fsumplugin.h' enthält Datenstrukturen, Funktionen und Fehlercode.

Das Plug-in für Benutzerzuordnungen muss diese Kopfdatendatei beinhalten. Die Datei befindet sich im Verzeichnis `sqllib/include`.

```
/* Definition der Namen der Benutzerzuordnungsoptionen */
#define FSUM_REMOTE_AUTHID_OPTION "REMOTE_AUTHID"
#define FSUM_REMOTE_PASSWORD_OPTION "REMOTE_PASSWORD"

/* Definition der Typen der Optionswerte */
#define FSUM_OPTION_VALUE_BINARY_TYPE 1
#define FSUM_OPTION_VALUE_STRING_TYPE 2

/* Datenstruktur zum Beschreiben einer Benutzeroption */

typedef struct _FSUMOption
{
    const char* optionName;
    size_t optionNameLen;
    char* optionValue;
    size_t optionValueLen;
    size_t optionValueType;
    struct _FSUMOption* nextOption;
} FSUMOption;

/* Datenstruktur zum Beschreiben eines Benutzerzuordnungseintrags.
   Ein Benutzerzuordnungseintrag kann verschiedene Benutzerzuordnungsoptionen,
   z. B. REMOTE_AUTHID und REMOTE_PASSWORD, enthalten.
   Diese Optionen werden in eine verkettete Liste gestellt. Die Datenstruktur
   enthält den Zeiger zu der ersten Option der Liste. */

typedef struct _FSUMEntry
{
    const char* fsInstanceName;
    size_t fsInstanceNameLen;
    const char* fsDatabaseName;
    size_t fsDatabaseNameLen;
    const char* fsServerName;
    size_t fsServerNameLen;
    const char* fsAuthID;
    size_t fsAuthIDLen;
    FSUMOption* firstOption;
} FSUMEntry;
```

```

/* Die Funktionen, die zusätzlich zur Funktion 'FSUMPluginInit',
   die nicht in der Struktur 'FSUMPluginAPIs' enthalten ist,
   zu implementieren sind. */

typedef struct _FSUMPluginAPIs
{
    size_t version;
    SQL_API_RC (SQL_API_FN * FSUMconnect)
    (void** a_FSUMRepository, const char* a_cfgFilePath);

    SQL_API_RC (SQL_API_FN * FSUMfetchUM)
    (void* a_FSUMRepository, FSUMEntry* a_entry);

    SQL_API_RC (SQL_API_FN * FSUMdisconnect)
    (void* a_FSUMRepository);

    SQL_API_RC (SQL_API_FN * FSUMPluginTerm) ();
} FSUMPluginAPIs;

/* Der Server mit föderierten Datenbanken stellt die folgenden Dienstprogramme als
   Rückruffunktionen für das Plug-in bereit. */

typedef SQL_API_RC (SQL_API_FN FSUMallocateFP)
(size_t a_blkSize, void** a_pblkPtr);
typedef void(SQL_API_FN FSUMdeallocateFP)
(void* a_blkPtr);
typedef SQL_API_RC (SQL_API_FN FSUMloadFP)
(const char* a_libName, void** a_lib);
typedef SQL_API_RC (SQL_API_FN FSUMgetFunctionFP)
(const char* a_functionName, void* a_lib, void** a_pFuncAddress);
typedef SQL_API_RC (SQL_API_FN FSUMunloadFP)
(void* a_lib);
typedef SQL_API_RC (SQL_API_FN FSUMlogErrorMsgFP)
(sqlint32 a_level, const char* a_msg, size_t a_length);

typedef SQL_API_RC (SQL_API_FN FSUMaddUMOptionFP)
(FSUMEntry *a_entry,
 const char* optionName,
 size_t optionNameLen,
 const char* optionValue,
 size_t optionValueLen);

/* Struktur zur Aufnahme der vom Server mit föderierten Datenbanken
   bereitgestellten Dienstprogrammfunktionen */

typedef struct _FSUMPluginUtilities
{
    FSUMallocateFP    *allocate;
    FSUMdeallocateFP *deallocate;
    FSUMloadFP        *load;
    FSUMgetFunctionFP *getFunction;
    FSUMunloadFP      *unload;
    FSUMlogErrorMsgFP *logErrorMsg;
    FSUMaddUMOptionFP *addUMOption;
} FSUMPluginUtilities;

/* Typ des Einstiegspunkts für das Plug-in für Benutzerzuordnungen */
typedef SQL_API_RC (SQL_API_FN *FSUMPluginInitType)
(sqlint32, FSUMPluginAPIs*, FSUMPluginUtilities*);

/* Typ des Einstiegspunkts für die C-Schnittstelle des Plug-ins
   für Benutzerzuordnungen */
typedef SQL_API_RC (*fsum_plugin_hook_type)
(const char*, FSUMPluginInitType*, FSUMPluginUtilities*);

/* Definition des Rückkehrcodes für Dienstprogrammfunktionen */
#define FSUM_PLUGIN_UTIL_OK 0

```

```

#define FSUM_PLUGIN_UTIL_FAILED -1

/* Definition für 'extern "C"'. */
#define FSUM_PLUGIN_EXT_C extern "C"

/* Fehlerbewertungen der Funktion 'logErrorMsg' */
#define FSUM_LOG_NONE      0 /* No logging */
#define FSUM_LOG_CRITICAL  1 /* Severe error encountered */
#define FSUM_LOG_ERROR     2 /* Error encountered */
#define FSUM_LOG_WARNING   3 /* Warning */
#define FSUM_LOG_INFO      4 /* Informational */

/* Von der Funktion zurückzugebende Fehlercodes */
#define FSUM_PLUGIN_OK      0
#define FSUM_INITIALIZE_ERROR  1
#define FSUM_PLUGIN_VERSION_ERROR  2
#define FSUM_CONNECTION_ERROR  3
#define FSUM_LOOKUP_ERROR     4
#define FSUM_DECRYPTION_ERROR  5
#define FSUM_DISCONNECT_ERROR  6
#define FSUM_INVALID_PARAMETER_ERROR  7
#define FSUM_UNAUTHORIZED_CALLER  8
#define FSUM_AUTHENTICATION_ERROR  9
#define FSUM_TERMINATION_ERROR  10

/* Maximale Länge von Namen */
#define FSUM_MAX_NAME_LEN      128

/* Maximale Länge von Dateipfaden */
#define FSUM_MAX_PATH_LEN     256

/* Maximale Länge von Optionswerten */
#define FSUM_MAX_OPTION_VALUE_LEN  (2048+1)

/* Maximale Länge von Fehlermeldungen */
#define FSUM_MAX_ERROR_MSG_SIZE  2048

```

Funktion 'FSUMPluginInit' (Programmiersprache C)

Bei der Funktion 'FSUMPluginInit' handelt es sich um die Initialisierungsfunktion des Plug-ins für Benutzerzuordnungen.

Diese Funktion führt folgende Tasks aus:

- Die Zeiger für die vier weiteren erforderlichen Funktionen werden an den Server mit föderierten Datenbanken übergeben.
- Die vom Server mit föderierten Datenbanken übergebenen Dienstprogramm-funktionen werden abgerufen.
- Die globalen Ressourcen werden auf Plug-in-Ebene eingerichtet.

Syntax

```

SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
FSUMPluginAPIs* a_pluginAPIs,
FSUMPluginUtilities* a_pluginUtils);

```

Eingaben

sqlint32 a_version

Die Versionsnummer der Schnittstelle des Plug-ins für Benutzerzuordnungen

FSUMPluginUtilities *a_pluginUtils

Der Zeiger zu der Struktur, die die Dienstprogrammfunktionen enthält. Das Plug-in fordert die Zeiger der einzelnen Dienstprogrammfunktionen nach Bedarf an.

Ausgaben

FSUMPluginAPIs *a_pluginAPIs

Das Plug-in verwendet diese Struktur, um die Funktionszeiger für FSUMconnect, FSUMfetchUM, FSUMdisconnect und FSUMPluginTerm an den Server mit föderierten Datenbanken zu übergeben.

Funktion 'FSUMconnect' (Programmiersprache C)

Der Server mit föderierten Datenbanken ruft die Funktion 'FSUMconnect' auf, um eine Verbindung zum externen Benutzerzuordnungsrepository herzustellen.

Syntax

```
SQL_API_RC SQL_API_FN FSUMconnect  
(void** a_FSUMRepository,  
const char* a_cfgFilePath)
```

Eingaben

const char* a_cfgFilePath

Der vollständige Pfad zur Plug-in-Bibliothek. Wenn sich die Konfigurationsdatei in dem Verzeichnis mit der Plug-in-Bibliothek befindet, kann das Plug-in diese Pfadinformation zur Lokalisierung der Konfigurationsdatei verwenden.

Ausgaben

void a_FSUMRepository**

Der Zeiger auf den Verbindungsdeskriptor wird vor dem Senden an den Server mit föderierten Datenbanken als typenloser Zeiger umgesetzt. In nachfolgenden API-Funktionsaufrufen wird der Zeiger vom Server mit föderierten Datenbanken an das Plug-in übergeben, und das Plug-in muss den Zeiger erneut als Zeiger auf die reale Struktur des Verbindungsdeskriptors umsetzen.

Funktion 'FSUMfetchUM' (Programmiersprache C)

Der Server mit föderierten Datenbanken ruft die Funktion 'FSUMfetchUM' zum Abrufen von Benutzerzuordnungsoptionen aus einem externen Repository auf.

Die einzelnen Benutzerzuordnungseinträge werden über den Namen der Instanz mit föderierten Datenbanken (fsInstanceName), den Datenbanknamen (fsDatabaseName), den Namen des fernen Servers (fsServerName) und die Berechtigungs-ID des lokalen Benutzers (fsAuthID) gekennzeichnet. Die Benutzerzuordnung kann darüber hinaus die Optionen REMOTE_AUTHID und REMOTE_PASSWORD enthalten, die vom Plug-in abgerufen werden.

Syntax

```
SQL_API_RC SQL_API_FN FSUMfetchUM (void* a_FSUMRepository,  
FSUMEntry* a_entry);
```

Eingaben

void* a_FSUMRepository

Die Verbindungskennung für das externe Repository. Diese Kennung muss für die reale Struktur umgesetzt werden, die im Plug-in definiert ist.

FSUMEntry* a_entry

Bei diesem Parameter handelt es sich sowohl um einen Eingabe- als auch um einen Ausgabeparameter.

Als Eingabe übergibt dieser Parameter Informationen, die zur Kennzeichnung des Benutzerzuordnungseintrags im Benutzerzuordnungsrepository erforderlich sind, z. B. fsInstanceName, fsDatabaseName, fsServerName und fsAuthID.

Als Ausgabe werden diesem Parameter die Benutzerzuordnungsoptionen hinzugefügt, die aus dem Benutzerzuordnungsrepository abgerufen werden. Das Plug-in muss die Dienstprogrammfunktion 'FSUMaddUMOption' verwenden, um eine Option zum Benutzerzuordnungseintrag (a_entry) hinzuzufügen. Enthält die Benutzerzuordnung die Option REMOTE_PASSWORD und ist das jeweilige Kennwort verschlüsselt, muss das Plug-in das Kennwort entschlüsseln, bevor die Funktion 'FSUMaddUMOption' aufgerufen wird. Der Server mit föderierten Datenbanken ist nicht für die Freigabe des Speicherplatzes für die Zeichenfolgen zuständig, die an die Funktion 'FSUMaddUMOption' übergeben werden.

Funktion 'FSUMdisconnect' (Programmiersprache C)

Der Server mit föderierten Datenbanken ruft die Funktion 'FSUMdisconnect' auf, um die Verbindung zum externen Benutzerzuordnungsrepository zu trennen.

Syntax

```
SQL_API_RC SQL_API_FN FSUMdisconnect  
(void* a_FSUMRepository);
```

Eingaben

void* a_FSUMRepository

Die Verbindungskennung für das externe Repository. Diese Kennung muss für die reale Struktur umgesetzt werden, die im Plug-in definiert ist.

Funktion 'FSUMPluginTerm' (Programmiersprache C)

Diese Funktion wird vom Plug-in für Benutzerzuordnungen aufgerufen, um globale Ressourcen freizugeben, die über die Funktion 'FSUMPluginInit' zugeordnet wurden.

Für diese Funktion werden keine Parameter verwendet.

```
SQL_API_RC SQL_API_FN FSUMPluginTerm ()
```

Entwickeln eines Beispiel-Plug-ins für Benutzerzuordnungen (Programmiersprache C)

Das von Ihnen entwickelte Plug-in muss eine Verbindung zu dem externen Repository herstellen, Benutzerzuordnungen abrufen und ferne Kennwörter entschlüsseln. Die Codierung des Plug-ins richtet sich nach dem verwendeten Repository.

Wichtig: Denken Sie bei der Entwicklung und Verwendung des Plug-ins daran, dass Sie Benutzer-IDs und Kennwörter zwischen verschiedenen Quellen übertragen. Um diese Informationen zu schützen, müssen Sie deshalb den Zugriff auf den Quellcode des Plug-ins einschränken und die DB2-Prüffunktion so konfigurieren,

dass bei jeder Verwendung des Plug-ins durch den Server mit föderierten Datenbanken ein VALIDATE-Datensatz in der Datei mit dem Diagnoseprogrammprotokoll erfasst wird. Die Datei mit dem Diagnoseprogrammprotokoll ist nicht nur für die Protokollierung der Plug-in-Verwendung, sondern auch bei der Behebung von eventuell auftretenden Fehlern nützlich.

Führen Sie zum Entwickeln eines Plug-ins für Benutzerzuordnungen die folgenden Tasks aus:

Deklarieren von externen Funktionen für das Plug-in für Benutzerzuordnungen (Programmiersprache C)

Das Plug-in implementiert die Funktionen. Wenn die Funktion 'FSUMPluginInit' aufgerufen wird, übergibt die Funktion die Funktionszeiger an den Server mit föderierten Datenbanken.

Die Initialisierungsfunktion 'FSUMPluginInit' ist die einzige Funktion, die genau diesen Funktionsnamen aufweisen muss. Alle anderen Funktionen können mit einem beliebigen gültigen C-Funktionsnamen bezeichnet werden. Diese Funktionen sind global und müssen extern aufgelöst werden. Wenn Sie das Plug-in in C++ schreiben, müssen Sie die Funktionen mit FSUM_PLUGIN_EXT_C deklarieren.

Das Beispiel-Plug-in implementiert die folgenden APIs:

- API 'FSUMconnect' in der Funktion 'myConnect'
- 'FSUMfetchUM' in der Funktion 'myFetchUM'
- 'FSUMdisconnect' in der Funktion 'myDisconnect'
- 'FSUMPluginTerm' in der Funktion 'myPluginTerm'

Im Beispiel-Plug-in werden bei Aufruf der Funktion 'FSUMPluginInit' die Zeiger auf 'myConnect', 'myFetchUM', 'myDisconnect' und 'myPluginTerm' an den Server mit föderierten Datenbanken übergeben.

Der folgende Code ist in der Datei fsumplugin_file.c enthalten.

```
SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint23 version,
FSUMPluginAPIs *pluginAPIs,
FSUMPluginUtilities* pluginUtils)
```

```
SQL_API_RC SQL_API_FN myConnect
(void** FSUMRepository)
```

```
SQL_API_RC SQL_API_FN myFetchUM
(void* FSUMRepository,
FSUMEntry* entry)
```

```
SQL_API_RC SQL_API_FN myDisconnect
(void* FSUMRepository)
```

```
SQL_API_RC SQL_API_FN myPluginTerm ()
```

Deklarieren von Dienstprogrammfunktionen für das Plug-in für Benutzerzuordnungen (Programmiersprache C)

Sie müssen vier Dienstprogrammoptionen für das Plug-in für Benutzerzuordnungen deklarieren.

Für das Plug-in für Benutzerzuordnungen sind die Dienstprogrammfunktionen 'FSUMlogErrorMsg', 'FSUMaddUMOption', 'FSUMallocate' und 'FSUMdeallocate' erforderlich. Die optionalen Dienstprogrammfunktionen können bei der Entwick-

lung eines Plug-ins für Benutzerzuordnungen hilfreich sein. Die Zeiger auf die Dienstprogrammfunktionen können Sie aus der Struktur 'FSUMPluginUtilities' in der Funktion 'FSUMPluginInit' abrufen.

Erforderliche Dienstprogrammfunktionen

```
FSUMlogErrorMsgFP *FSUMlogErrorMsg=NULL;
FSUMaddUMOptionFP *FSUMaddUMOption=NULL;
FSUMallocateFP *FSUMallocate=NULL;
FSUMdeallocateFP *FSUMdeallocate=NULL;
```

Optionale Dienstprogrammfunktionen

```
FSUMgetFunctionFP *FSUMgetFunction=NULL;
FSUMloadFP *FSUMload=NULL;
FSUMunloadFP *FSUMunload=NULL;
```

Definieren von Funktionszeigern für das Plug-in für Benutzerzuordnungen (Programmiersprache C)

Übergeben Sie die Zeiger auf die erforderlichen Funktionen an den Server mit föderierten Datenbanken, und rufen Sie die Zeiger auf die Dienstprogrammfunktionen vom Server mit föderierten Datenbanken ab.

```
/* Initialisierungsfunktion des Plug-ins. Diesen Namen nicht ändern. */

SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
 FSUMPluginAPIs* a_pluginAPIs,
 FSUMPluginUtilities* a_pluginUtils)
{
    SQL_API_RC rc = FSUM_PLUGIN_OK ;

    /* Funktionszeiger an den Server mit föderierten Datenbanken übergeben */

    a_pluginAPIs->FSUMconnect = &myConnect
    a_pluginAPIs->FSUMfetchUM = &myFetchUM
    a_pluginAPIs->FSUMdisconnect = &myDisconnect
    a_pluginAPIs->FSUMPluginTerm = &myPluginTerm

    /* Zeiger auf die erforderlichen Dienstprogrammfunktionen abrufen */

    FSUMallocate = a_pluginUtils->allocate;
    FSUMdeallocate = a_pluginUtils->deallocate;
    FSUMlogErrorMsg = a_pluginUtils->logErrorMsg;
    FSUMaddUMOption = a_pluginUtils->addUMOption;
    /* Bei Bedarf auch die Zeiger auf die optionalen Dienstprogrammfunktionen abrufen */

    /** Bei Fehlern wie folgt vorgehen:
        1. Optional --- 'FSUMlogErrorMsg' zum Protokollieren des Fehlers aufrufen
        2. Zwingend erforderlich --- Fehlercode zurückgeben: rc = FSUM_INITIALIZE_ERROR
    **/

    return rc;
}
```

Fehlerbehandlung beim Plug-in für Benutzerzuordnungen (Programmiersprache C)

Das Plug-in für Benutzerzuordnungen zeichnet mithilfe der Funktion 'FSUMlogErrorMsg' alle Fehler- und Informationsnachrichten in der Datei db2diag.log auf.

Bei erfolgreichen Funktionsaufrufen gibt das Plug-in FSUM_PLUGIN_OK zurück. Ist ein Aufruf nicht erfolgreich, gibt das Plug-in einen entsprechenden Fehlercode zurück. Beschreibungen zu den Fehlern können Sie der folgenden Tabelle entnehmen.

Tabelle 25. Fehlermeldungen für das Plug-in für Benutzerzuordnungen

Fehlernummer	Konstantenname	Fehlermeldung	Rückgabe des Fehlers durch	Fehlercode
0	FSUM_PLUGIN_OK	Das Plug-in wurde erfolgreich ausgeführt.	<ul style="list-style-type: none"> • FSUMPluginInit • FSUMconnect • FSUMfetchUM • FSUMdisconnect • FSUMPluginTerm 	SQL_SUCCESS (Kein Fehler)
1	FSUM_INITIALIZE_ERROR	Das Plug-in konnte nicht initialisiert werden.	<ul style="list-style-type: none"> • FUMPluginInit 	SQL20349N , Ursachencode 1
2	FSUM_PLUGIN_VERSION_ERROR	Das Plug-in liegt nicht in der richtigen Version vor.	<ul style="list-style-type: none"> • FUMPluginInit 	SQL20349N, Ursachencode 2
3	FSUM_CONNECTION_ERROR	Es kann keine Verbindung zum externen Repository hergestellt werden.	<ul style="list-style-type: none"> • FSUMPluginInit 	SQL20349N, Ursachencode 3
4	FSUM_LOOKUP_ERROR	Die Suchfunktion für das Repository ist fehlgeschlagen.	<ul style="list-style-type: none"> • FSUMfetchUM 	SQL20349N, Ursachencode 4
5	FSUM_DECRYPTION_ERROR	Die Entschlüsselung ist fehlgeschlagen.	<ul style="list-style-type: none"> • FSUMfetchUM 	SQL20349N, Ursachencode 5
6	FSUM_DISCONNECT_ERROR	Die Verbindung zum Repository kann nicht getrennt werden.	<ul style="list-style-type: none"> • FSUMdisconnect 	SQL20349N, Ursachencode 6
7	FSUM_INVALID_PARAMETER_ERROR	Ungültiger Parameter.	<ul style="list-style-type: none"> • FSUMfetchUM • FSUMdisconnect • FSUMPluginTerm 	SQL20349N, Ursachencode 7
8	FSUM_UNAUTHORIZED_CALLER	Der Aufruf des Plug-ins erfolgte ohne die erforderlichen Berechtigungen.	<ul style="list-style-type: none"> • FSUMPluginInit 	SQL20349N, Ursachencode 8
9	FSUM_AUTHENTICATION_ERROR	Es kann keine Authentifizierung für das Repository durchgeführt werden.	<ul style="list-style-type: none"> • FSUMconnect 	SQL20350N, kein Ursachencode
10	FSUM_TERMINATION_ERROR	Eine Ressource auf Plug-in-Ebene konnte nicht bereinigt werden.	<ul style="list-style-type: none"> • FSUMPluginTerm 	SQL20349N, Ursachencode 9
Eine Zahl außerhalb des Bereichs von 0 bis 10.		Unbekannter Fehler.	<ul style="list-style-type: none"> • FSUMPluginInit • FSUMconnect • FSUMfetchUM • FSUMdisconnect • FSUMPluginTerm 	SQL20349N, Ursachencode 10

Das Plug-in kann Nachrichten auch mithilfe des Dienstprogramms 'FSUMlogErrorMsg' in der Datei db2diag.log aufzeichnen. Geben Sie die Bewertung für die

Nachrichten mithilfe von FSUM_LOG_CRITICAL, FSUM_LOG_ERROR, FSUM_LOG_WARNING und FSUM_LOG_INFO an.

Wenn Sie das Testprogramm zum Testen des Plug-ins verwenden, werden die Nachrichten direkt auf dem Bildschirm ausgegeben.

Build des Plug-ins für Benutzerzuordnungen (Programmiersprache C)

Erstellen Sie mit dem Script 'bldplugin' einen Build des Plug-ins für Benutzerzuordnungen.

Erstellen Sie wie folgt einen Build des Plug-ins für Benutzerzuordnungen:

1. Überprüfen Sie das Script 'bldplugin', und aktualisieren Sie die Pfadinformation, sodass sie für Ihre Installation geeignet ist.
2. Setzen Sie den folgenden Befehl ab, um einen Build des Plug-ins zu erstellen:

```
bldplugin name_der_plugin_datei
```

Weitere Informationen hierzu können Sie der Datei 'bldplugin' entnehmen.

3. Richten Sie das externe Repository ein. Das Repository muss die Benutzerzuordnungsinformationen enthalten und ein Verschlüsselungsschema vorgeben. Öffnen Sie die Datei 'fsumplugin_file.txt', und erstellen Sie einen neuen Benutzereintrag für Ihr System, wenn Sie einen Build des Beispiel-Plug-ins für Benutzerzuordnungen erstellen. Das Beispiel-Plug-in verwendet ein einfaches Verschlüsselungsschema, bei dem lediglich die Bytes des fernen Kennworts umgekehrt werden.

Testen des Plug-ins für Benutzerzuordnungen (Programmiersprache C)

Testen und laden Sie das Beispiel-Plug-in für Benutzerzuordnungen bzw. Ihr eigenes angepasstes Plug-in für Benutzerzuordnungen mithilfe des Testprogramms, bevor Sie es auf dem Server mit föderierten Datenbanken implementieren.

Das Konfigurationsprogramm 'fsumsetup_file' (unter Microsoft Windows 'fsumsetup_file.exe') sowie das Testprogramm 'fsumlookup' (unter Microsoft Windows 'fsumlookup.exe') werden in demselben Verzeichnis wie die Dateien für das Beispiel-Plug-in installiert.

Das Programm 'fsumsetup_file' erzeugt eine Konfigurationsdatei mit der Bezeichnung 'fsumplugin_file.cfg'. In dieser Datei wird der vollständige Pfad zu der Textdatei gespeichert, die die Benutzerzuordnungseinträge enthält. Das Programm 'fsumlookup' (unter Microsoft Windows 'fsumlookup.exe') ist zum Testen des Plug-ins vorgesehen.

1. Führen Sie das Programm 'fsumsetup_file' aus, um den vollständigen Dateipfad und den Dateinamen der Datei, in der die Benutzerzuordnungen gespeichert werden, zu konfigurieren.
2. Führen Sie das Programm 'fsumlookup' aus, um das Plug-in zu testen:

```
fsumlookup plug-in-name fs-instanz fs-datenbank ferner_fs-server fs_berechtigungs-id
```

Dabei gilt Folgendes:

- *plug-in-name* ist der Name des Plug-ins.
- *fs-instanz* ist der Name der Serverinstanz mit föderierten Datenbanken.
- *fs-datenbank* ist der Name der föderierten Datenbank.
- *ferner_fs-server* ist der Name des fernen Datenquellenservers, wie in der Anweisung CREATE SERVER angegeben.
- *fs-berechtigungs-id* ist die lokale Benutzer-ID.

Wenn das Plug-in ordnungsgemäß funktioniert, gibt das Programm 'fsumlookup' die Parameter, die die Benutzerzuordnungseinträge kennzeichnen, sowie die Optionsnamen und Optionswerte auf dem Bildschirm aus. Es wird z. B. eine Ausgabe ähnlich der folgenden angezeigt:

```
fsumlookup fsumplugin_file.a FSinst1 FSdb remoteDB localID
fsInstanceName: FSinst1
fsDatabaseName: FSdb
fsServerName: remoteDB
fsAuthID: localID
optionName:REMOTE_PASSWORD
optionValue:p4ssw0rd
optionName:REMOTE_AUTHID
optionValue:remoteID
```

Sollten Fehler auftreten, gibt das Testprogramm die entsprechenden Fehlermeldungen auf dem Bildschirm aus.

Implementieren des Plug-ins für Benutzerzuordnungen (Programmiersprache C)

Sie können für den Build und die Kompilierung des Plug-ins für Benutzerzuordnungen ein beliebiges Verzeichnis wählen. Kopieren Sie das Plug-in anschließend zum Implementieren in das dafür vorgesehene Verzeichnis auf dem Server mit föderierten Datenbanken.

Kopieren Sie das Plug-in für die Implementierung auf dem Server mit föderierten Datenbanken in das folgende Verzeichnis des Servers mit föderierten Datenbanken:

- UNIX und Linux: *inst_home*/sqllib/function, wobei *inst_home* das Ausgangsverzeichnis der Instanz ist.
- Windows: %DB2PATH%\function, wobei %DB2PATH% das Verzeichnis ist, in dem das DB2-Datenbanksystem installiert ist.

Ermöglichen des Zugriffs auf das externe Benutzerzuordnungsrepository

Definieren Sie nach dem Erstellen eines Plug-ins für Benutzerzuordnungen die Optionen DB2_UM_PLUGIN und DB2_UM_PLUGIN_LANG, um dem Server mit föderierten Datenbanken den Zugriff auf die Benutzerzuordnungen im externen Repository zu ermöglichen.

Sie können die Optionen auf Wrapper- oder Serverebene definieren, die Ebene muss bei beiden Optionen jedoch übereinstimmen. DB2_UM_PLUGIN gibt den Namen des Plug-ins an und DB2_UM_PLUGIN_LANG die Sprache, in der das Plug-in geschrieben ist. Die beiden Optionen sind voneinander abhängig. Sie müssen deshalb DB2_UM_PLUGIN vor DB2_UM_PLUGIN_LANG hinzufügen. Wenn Sie beide Optionen in derselben Anweisung hinzufügen, ist die Reihenfolge der Optionen nicht von Bedeutung. Wenn Sie die Optionen löschen möchten, müssen Sie DB2_UM_PLUGIN_LANG vor DB2_UM_PLUGIN löschen.

Bei den folgenden Anweisungen wird z. B. der Name des Beispiel-Plug-ins verwendet, um Wrapper 'w1' und Server 's1' so zu erstellen bzw. Wrapper 'w2' und Server 's2' so zu ändern, dass das Plug-in für Benutzerzuordnungen verwendet wird:

```
CREATE WRAPPER w1 LIBRARY 'libdb2drda.a'
OPTIONS
(DB2_UM_PLUGIN 'fsumplugin_file.a', DB2_UM_PLUGIN_LANG 'C');

ALTER WRAPPER w2
OPTIONS
(ADD DB2_UM_PLUGIN 'fsumplugin_file.a', ADD DB2_UM_PLUGIN_LANG 'C');
```

```

CREATE SERVER s1 TYPE db2/cs VERSION 10 WRAPPER w2
AUTHORIZATION remoteID PASSWORD p4ssw0rd
OPTIONS
(NODE 'node1',
DBNAME 'db1',
DB2_UM_PLUGIN 'fsumplugin_file.a',
DB2_UM_PLUGIN_LANG 'C');

ALTER SERVER s2
OPTIONS
(ADD DB2_UM_PLUGIN 'fsumplugin_file.a', ADD DB2_UM_PLUGIN_LANG 'C');

```

Aktualisieren des Plug-ins für Benutzerzuordnungen (Programmiersprache C)

Nach dem Modifizieren eines vorhandenen Plug-ins für Benutzerzuordnungen müssen Sie die aktualisierte Version auf den Server mit föderierten Datenbanken kopieren.

Gehen Sie zum Aktualisieren des Plug-ins für Benutzerzuordnungen wie folgt vor:

1. Kopieren Sie das aktualisierte Plug-in in das Verzeichnis 'sqlib/function'.
2. Geben Sie zum Stoppen und erneuten Starten des Servers mit föderierten Datenbanken die folgenden Befehle ein:

```

db2stop
db2start

```

Beispiel-Plug-in für Benutzerzuordnungen (Programmiersprache C)

Orientieren Sie sich beim Entwickeln eines Plug-ins, das eine C-Schnittstelle zu einem externen Repository mit Benutzerzuordnungen bereitstellt, an dem Beispielquellcode.

Das Beispiel-Plug-in für Benutzerzuordnungen verwendet eine Datei als externes Repository. Die Quellendateien für das Beispiel werden automatisch auf Ihrem System installiert. Das Installationsverzeichnis richtet sich nach dem verwendeten Betriebssystem:

- Windows: %DB2PATH%\samples\federated\umplugin\datei. Dabei ist %DB2PATH% das Verzeichnis, in dem das DB2-Datenbanksystem installiert ist.
- Linux und UNIX: *inst_home*/sqlib/samples/federated/umplugin/datei. Dabei ist *inst_home* das Ausgangsverzeichnis der Instanz.

Das Beispiel enthält folgende Dateien:

fsumplugin_file.h

Die Kopfdatendatei mit den Datenstrukturen und Funktionsdefinitionen für das Beispiel-Plug-in

fsumplugin_file.c

Der Quellcode für das Beispiel-Plug-in, das eine Datei als externes Repository verwendet

fsumplugin_file.txt

Eine Datei mit Beispieleinträgen für Benutzerzuordnungen in unverschlüsseltem Text

fsumlookup

Ein Standalone-Programm, mit dem das Beispielprogramm außerhalb der

Serverumgebung mit föderierten Datenbanken getestet werden kann. Unter Microsoft Windows weist diese Datei die Standarderweiterung für ausführbare Dateien auf: `fsumlookup.exe`.

fsumsetup_file

Ein Standalone-Programm, das den vollständigen Pfad und den Dateinamen der Datei, in der die Benutzerzuordnungen gespeichert werden, definiert. Unter Microsoft Windows weist diese Datei die Standarderweiterung für ausführbare Dateien auf: `fsumsetup_file.exe`.

blplugin

Das Script für den Build zum Kompilieren und Verknüpfen des Plug-ins. Unter Microsoft Windows weist diese Datei die Standarderweiterung für Batchdateien auf: `blplugin.bat`.

README Eine Datei mit den Anweisungen für die Kompilierung, den Build, das Testen und die Implementierung des Beispiel-Plug-ins

In der Datei `fsumplugin_file.txt` nimmt jeder Benutzerzuordnungseintrag eine einzelne Zeile ein. Die Kennwörter sind durch eine einfache Umkehrung der Bytes verschlüsselt. Dies stellt ein sehr einfaches Verschlüsselungsverfahren dar. Wenn Sie das Beispiel-Plug-in für den Einsatz in einer Produktionsumgebung modifizieren, müssen Sie ein Verschlüsselungsverfahren verwenden, das den Sicherheitsanforderungen für diese Umgebung entspricht.

Die folgende Syntax verdeutlicht das Format der Einträge anhand eines Beispiel-eintrags für Benutzerzuordnungen.

Anmerkung: Dieser Code besteht aus mehreren Zeilen. Sie müssen den gesamten Code in Ihrer Anwendung jedoch in einer einzigen Zeile eingeben. Dabei ist zu beachten, dass die einzelnen Bestandteile der Benutzerzuordnung durch ein Semikolon getrennt und die Optionsnamen `REMOTE_AUTHID` und `REMOTE_PASSWORD` durch einen Doppelpunkt von den zugehörigen Einstellungen abgesetzt sind.

```
fs_instanz;fs_datenbank;ferner_fs_server;  
fs-berechtigungs-id;REMOTE_AUTHID:ferne_berechtigungs-id;  
REMOTE_PASSWORD:fernes_kennwort
```

Beispiel:

```
DB2INST1;TESTDB;MSSQL2K;ALICE;REMOTE_AUTHID:TESTUSR1;REMOTE_PASSWORD:drowssap
```

Plug-in für Benutzerzuordnungen (Programmiersprache Java)

Die Architektur des Java-Plug-ins beinhaltet zwei Schnittstellenklassen ('interface') und drei Dienstprogrammklassen ('utility'). Mit diesen Klassen können Sie ein Plug-in entwickeln, das Benutzerzuordnungen aus einem externen Repository abrufen.

Standardmäßig werden Benutzerzuordnungen für eine Datenquelle lokal auf den einzelnen Servern mit föderierten Datenbanken gespeichert. LDAP-Server speichern Objekte, z. B. Benutzerzuordnungseinträge, in einer Verzeichnisstruktur. Diese Objekte können über Attribute, z. B. Kennwörter, verfügen. Darüber hinaus speichern LDAP-Server in der Regel zusätzliche Informationen zu Benutzern, z. B. E-Mail-Adressen und Telefonnummern.

Die folgende Abbildung veranschaulicht die Architektur des Beispiel-Plug-ins für Benutzerzuordnungen:

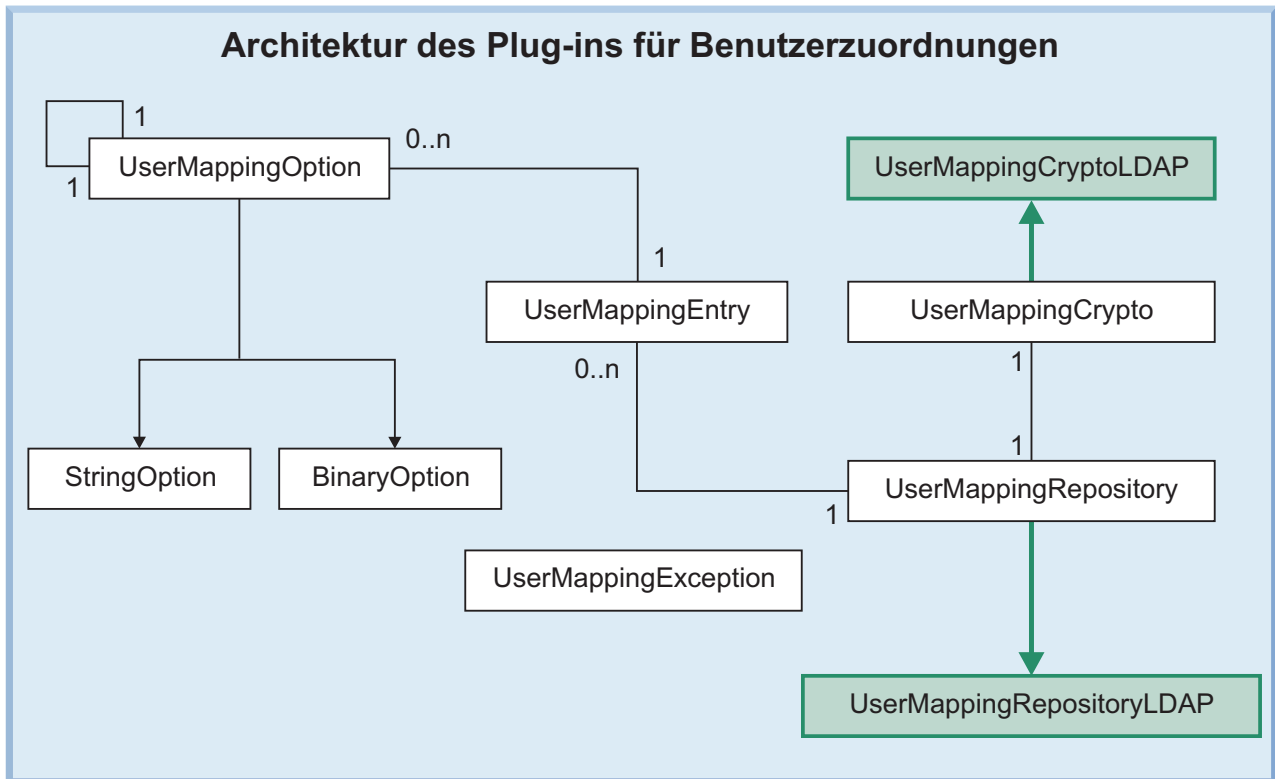


Abbildung 15. Architektur des Plug-ins für Benutzerzuordnungen

Bei den Klassen 'UserMappingCrypto' und 'UserMappingRepository' handelt es sich um Schnittstellenklassen. Wenn Sie ein Plug-in für Benutzerzuordnungen erstellen möchten, das Benutzerzuordnungen aus dem in Ihrem Unternehmen verwendeten externen Repository abrufen, müssen Sie diese Schnittstellenklassen erweitern. Bei den Klassen 'UserMappingEntry', 'UserMappingOption' und 'UserMappingException' handelt es sich um Dienstprogrammklassen. Sie können diese Dienstprogrammklassen ohne zusätzliche Änderungen verwenden. Einige in den Schnittstellenklassen enthaltene Funktionen und Methoden verhalten sich wie Dienstprogrammfunktionen. Sie können z. B. die Funktionen 'getChars()' und 'getBytes()' der Klasse 'UserMappingCrypto' unverändert einsetzen.

In der Abbildung gibt die Angabe *0..n* an, dass beliebig viele oder auch überhaupt keine dieser Objekte vorhanden sein können. Ein Objekt 'UserMappingEntry' kann z. B. viele Objekte 'UserMappingOption' enthalten, es kann jedoch nur ein einziges Objekt 'UserMappingRepository' geben. Die Klassen 'UserMappingCryptoLDAP' und 'UserMappingRepositoryLDAP' stellen erweiterte Klassen der zugehörigen übergeordneten Klassen dar.

Klassen für das Plug-in für Benutzerzuordnungen (Programmiersprache Java)

Die Architektur des Java-Plug-ins beinhaltet zwei Schnittstellenklassen ('interface') und drei Dienstprogrammklassen ('utility'). Mit diesen Klassen können Sie ein Plug-in entwickeln, das Benutzerzuordnungen aus einem externen Repository abrufen.

Klasse 'UserMappingRepository' (Programmiersprache Java)

Die Klasse 'UserMappingRepository' ist eine abstrakte Klasse, die keinen Konstruktor hat. Zum Erstellen eines eigenen Plug-ins für Benutzerzuordnungen müssen Sie eine Unterklasse der Klasse 'UserMappingRepository' erstellen oder die Unterklasse im in Java programmierten Beispiel-Plug-in ändern.

Die Klasse 'UserMappingRepository' enthält die folgenden allgemein zugänglichen Methoden: `getVersionNumber()`, `getCrypto()`, `connect()`, `disconnect()`, `fetchUM()` und `lookupUM()`. Sie müssen Ihre eigene Unterklasse der Klasse 'UserMappingRepository' erstellen, die diese Funktionen enthält. Schreiben Sie in diese Funktionen den Code, den das Plug-in für die Interaktion mit dem externen Repository verwendet.

Ein in Java programmiertes Beispiel-Plug-in verdeutlicht die Implementierung dieser Funktionen. Dieses Beispiel-Plug-in ruft Benutzerzuordnungen von einem LDAP-Server ab. Die Dateien befinden sich im Verzeichnis `'sql1lib/samples/federated/umplugin/ldap/'`. Die Funktionen aus dieser Klasse werden in den Dateien 'UserMappingRepositoryLDAP.java' und 'UserMappingLookupLDAP.java' verwendet.

Allgemein zugängliche Methoden

int getVersionNumber()

Gibt die Development Kit-Versionsnummer zurück, die das Plug-in verwendet.

UserMappingCrypto getCrypto()

Gibt das Objekt 'UserMappingCrypto' zurück, das diesem Objekt 'UserMappingRepository' zugeordnet ist.

abstract void connect()

Sie müssen innerhalb dieser Funktion Ihre eigene Methode für das Herstellen einer Verbindung zu Ihrem Repository implementieren.

abstract void disconnect()

Sie müssen innerhalb dieser Funktion Ihre eigene Methode für das Trennen der Verbindung zu Ihrem Repository implementieren.

abstract void fetchUM(UserMappingEntry um)

Sie müssen innerhalb dieser Funktion Ihre eigene Methode für das Abrufen der Benutzerzuordnung aus dem Repository implementieren. Der Parameter 'um' enthält die ausführlichen Abfrageinformationen, mit denen festgelegt wird, welche Benutzerzuordnung abgerufen werden soll.

UserMappingEntry lookupUM(UserMappingRepository repository, String iiInstanceName, String iiDatabaseName, String iiRemoteServerName, String iiAuthid)

Diese Funktion wird in erster Linie zum Testen des Plug-ins verwendet. Die Funktion verwendet die Parameter `iiInstanceName`, `iiDatabaseName`, `iiRemoteServerName` und `iiAuthid` als Eingabe zum Erstellen und Initialisieren der Klasse 'UserMappingEntry'. Die Funktion verwendet die Methoden `connect`, `fetchUM` und `disconnect`.

Klasse 'UserMappingCrypto' (Programmiersprache Java)

Wenn ferne Kennwörter in Ihrem externen Repository verschlüsselt oder codiert werden, müssen Sie Ihre eigene Unterklasse der Klasse 'UserMappingCrypto' schreiben. Der Konstruktor der von Ihnen erstellten Unterklasse wird zum Aufbau des Verschlüsselungsobjekts verwendet. Die Methoden der Verschlüsselungsklasse werden von anderen Klassen aufgerufen, wenn die Kennwörter der Benutzerzuordnungen verschlüsselt, entschlüsselt, codiert oder decodiert werden müssen.

Die Klasse 'UserMappingCrypto' enthält die folgenden allgemein zugänglichen Methoden: encrypt(), decrypt(), encode() und decode(). In diesen Funktionen müssen Sie Ihren Code zum Verschlüsseln, Entschlüsseln, Codieren und Decodieren des fernen Kennworts schreiben. Bei den Funktionen 'getBytes()' und 'getChars()' handelt es sich um Dienstprogrammfunktionen, die übernommen und ohne Änderungen verwendet werden können. Die von Ihnen codierten Methoden zum Verschlüsseln, Entschlüsseln, Codieren und Decodieren müssen den Verschlüsselungs- und Codierungsmethoden entsprechen, die das externe Repository zum Schutz der gespeicherten Kennwörter verwendet.

Ein in Java programmiertes Beispiel-Plug-in verdeutlicht die Implementierung dieser Funktionen. Dieses Beispiel-Plug-in ruft Benutzerzuordnungen von einem LDAP-Server ab. Die Dateien befinden sich im Verzeichnis 'sql1lib/samples/federated/umplugin/ldap/'. Die Funktionen aus dieser Klasse werden in den Beispieldateien 'UserMappingRepositoryLDAP.java' und 'UserMappingSetupLDAP.java' verwendet.

Allgemein zugängliche Methoden

abstract byte[] encrypt(byte[]plainValue)

Diese Methode implementiert den Verschlüsselungsalgorithmus, der dem vom externen Repository verwendeten Verschlüsselungsalgorithmus entspricht.

abstract byte[] decrypt(byte[]encryptedValue)

Diese Methode implementiert den Entschlüsselungsalgorithmus, der den vom externen Repository verwendeten Verschlüsselungsalgorithmus umkehrt und das Kennwort anschließend zurückgibt.

abstract string encode(byte[]bytes)

Diese Methode schreibt oder implementiert eine Funktion, die den Parameter **bytes** in eine Zeichenfolge codiert. Diese Funktion codiert den verschlüsselten Bytewert in eine Zeichenfolge.

abstract byte[] decode(String[]string)

Diese Methode schreibt oder implementiert eine Funktion, die den Parameter **string** in Bytes decodiert. Diese Funktion decodiert das abgerufene Kennwort, das als Zeichenfolge vorliegt, in Bytes, sodass der Wert entschlüsselt werden kann.

byte[] getBytes(char[]chars)

Diese Funktion wird übernommen und kann ohne Änderungen verwendet werden. Die Funktion setzt jedes Zeichen einer Zeichenfolge in ein Byte um.

char[] getChars(byte[]bytes)

Diese Funktion wird übernommen und kann ohne Änderungen verwendet werden. Die Funktion setzt jedes Byte in ein Zeichen um.

Geschützte Attribute

SecretKey schlüssel

Der geheime Schlüssel, der zum Verschlüsseln und Entschlüsseln der fernen Kennwörter verwendet wird.

Cipher chiffrierwert

Der vom geheimen Schlüssel verwendete Algorithmus für die Kennwortverschlüsselung.

Klasse 'UserMappingEntry' (Programmiersprache Java)

Die Klasse 'UserMappingEntry' ist eine Dienstprogrammklasse, die die Benutzerzuordnungsoptionen erstellt und verwaltet. Die Methoden in der Klasse 'UserMap-

'pingEntry' werden von den Funktionen 'fetchUM()' und 'lookupUM()' über die Klasse 'UserMappingRepository' aufgerufen.

Allgemein zugängliche Methoden

Ein in Java programmiertes Beispiel-Plug-in verdeutlicht die Verwendung dieser Funktionen. Dieses Beispiel-Plug-in ruft Benutzerzuordnungen von einem LDAP-Server ab. Die Dateien befinden sich im Verzeichnis 'sql11ib/samples/federated/umplugin/ldap/'. Die Funktionen aus dieser Klasse werden in den Dateien 'UserMappingRepositoryLDAP.java' und 'UserMappingLookupLDAP.java' verwendet.

UserMappingEntry(UserMappingRepository repository, String iiInstanceName, String iiDatabaseName, String iiRemoteServerName, String iiAuthID)

Dieser Konstruktor wird verwendet, um das Objekt 'UserMappingEntry' mit den folgenden Eingabeparametern zu instanziiieren:

- **iiInstance** - Der Name der Instanz des Servers mit föderierten Datenbanken
- **iiDatabase** - Der Name der Datenbank auf dem Server mit föderierten Datenbanken
- **iiRemoteServerName** - Der Name des fernen Servers für die Datenquelle
- **iiAuthid** - Die lokale Benutzer-ID, die der Benutzerzuordnung zugeordnet ist

UserMappingRepository getRepository()

Gibt das Objekt 'UserMappingRepository' zurück, das diesem Objekt 'UserMappingEntry' zugeordnet ist.

String getiiInstanceName()

Gibt den Name der Instanz zurück.

String getiiDatabaseName()

Gibt den Namen der Datenbank zurück.

String getiiRemoteServerName()

Gibt den Namen des fernen Servers zurück.

String getiiAuthID()

Gibt den Namen der lokalen Benutzer-ID zurück, die der Benutzerzuordnung zugeordnet ist.

UserMappingOption getFirstOption()

Gibt das erste Objekt 'UserMappingOption' zurück, das zu diesem Objekt 'UserMappingEntry' gehört.

void addOption(UserMappingOption newOption)

Fügt diesem Objekt 'UserMappingEntry' ein neues Objekt 'UserMappingOption' hinzu.

Klasse 'UserMappingOption' (Programmiersprache Java)

Die Klasse 'UserMappingOption' stellt eine Dienstprogrammklasse dar, die die erforderlichen Funktionen enthält, um dem Server mit föderierten Datenbanken die ferne Benutzer-ID und das ferne Kennwort zu liefern.

Allgemein zugängliche Methoden

Ein in Java programmiertes Beispiel-Plug-in verdeutlicht die Verwendung dieser Funktionen. Dieses Beispiel-Plug-in ruft Benutzerzuordnungen von einem LDAP-Server ab. Die Dateien befinden sich im Verzeichnis 'sql11ib/samples/federated/umplugin/ldap/'. Die Funktionen aus dieser Klasse werden in den Dateien 'UserMappingRepositoryLDAP.java' und 'UserMappingLookupLDAP.java' verwendet.

UserMappingEntry getEntry()

Gibt das Objekt 'UserMappingEntry' zurück, das diesem Objekt 'UserMappingOption' zugeordnet ist.

String getName()

Gibt den Namen der Option zurück.

void setName()

Definiert den Namen der Option.

UserMappingOption getNextOption()

Gibt die nächste Option zurück.

void setNextOption(UserMappingOptionnextOption)

Definiert die nächste Option.

abstract object getValue()

Gibt den Wert der Option zurück. Diese Funktion muss einen Zeichenfolge-wert oder einen binären Wert zurückgeben. Vgl. hierzu die nachstehend aufgeführten Methoden 'StringOption' und 'BinaryOption'.

Klasse 'StringOption'

Die Klasse 'StringOption', eine Erweiterung auf Grundlage der Klasse 'UserMappingOption', enthält die folgenden allgemein zugänglichen Methoden: 'getValue()' und 'setValue()'.

Allgemein zugängliche Methoden**object getValue()**

Gibt den Wert der Zeichenfolgeoption (StringOption) zurück, wenn es sich bei der Option um eine Zeichenfolge handelt.

void setValue(String value)

Definiert den Wert der Zeichenfolgeoption.

Klasse 'BinaryOption'

Die Klasse 'BinaryOption', eine Erweiterung auf Grundlage der Klasse 'UserMappingOption', enthält die folgenden allgemein zugänglichen Methoden: 'getValue()' und 'void setValue()'.

Allgemein zugängliche Methoden**object getValue()**

Gibt den Wert der Binäroption (BinaryOption) zurück.

void setValue(byte[] value)

Definiert den Wert der Binäroption.

Klasse 'UserMappingException' (Programmiersprache Java)

Das Java-Plug-in für Benutzerzuordnungen verwendet die Klasse 'UserMappingException', eine Unterklasse der Klasse 'java.lang.Exception', zum Melden von Fehlern.

Allgemein zugängliche Methoden

Ein in Java programmiertes Beispiel-Plug-in verdeutlicht die Verwendung dieser Funktionen. Dieses Beispiel-Plug-in ruft Benutzerzuordnungen von einem LDAP-

Server ab. Die Dateien befinden sich im Verzeichnis 'sql/lib/samples/federated/umplugin/ldap/'. Die Funktionen aus dieser Klasse werden in jeder Java-Beispiel-datei verwendet.

UserMappingException(int errorNumber)

Der Konstruktor, der zum Instanzieren des Objekts 'UserMappingException' verwendet wird, das für die Fehlermeldung zuständig ist. Der Parameter **errorNumber**, der an das Objekt 'UserMappingException' gesendet wird, definiert den Typ des gemeldeten Fehlers.

int getErrorNumber()

Gibt die Fehlernummer der Ausnahmebedingung zurück.

Die Datei 'UserMappingRepositoryLDAP.java' enthält diese Funktion zum Erfassen und Melden von Fehlern für den Test eines LDAP-Plug-ins.

String getErrorMessage()

Gibt die Fehlermeldung der Ausnahmebedingung zurück.

Die Datei 'UserMappingRepositoryLDAP.java' enthält eine Methode zum Erfassen und Melden von Fehlern für den Test eines LDAP-Plug-ins.

Fehlernachrichten

Die folgende Tabelle enthält eine Liste der Fehlernummern, Namen von Konstanten und Fehlermeldungen.

Tabelle 26. Fehlernummern und -nachrichten

Fehlernummer	Konstantenname	Fehlermeldung
1	INITIALIZE_ERROR	Das Plug-in konnte nicht initialisiert werden.
2	CONNECTION_ERROR	Es kann keine Verbindung zum Repository hergestellt werden.
3	AUTHENTICATION_ERROR	Es kann keine Authentifizierung für das Repository durchgeführt werden.
4	LOOKUP_ERROR	Die Suchfunktion für das Repository ist fehlgeschlagen.
5	DECRYPTION_ERROR	Die Entschlüsselung ist fehlgeschlagen.
6	DISCONNECT_ERROR	Die Verbindung zum Repository kann nicht getrennt werden.
7	INVALID_PARAMETER_ERROR	Ungültiger Parameter.
8	UNAUTHORIZED_CALLER	Das aufrufende Modul ist nicht zum Aufruf des Plug-ins berechtigt.

Beispiel-Plug-in für Benutzerzuordnungen (Programmiersprache Java)

Das in Java geschriebene Plug-in ruft Benutzerzuordnungseinträge von einem LDAP-Server ab. Passen Sie das Beispiel-Plug-in an die Einstellungen Ihres LDAP-Servers an, um es in Ihrer Umgebung verwenden zu können.

Die Dateien für das Plug-in befinden sich im Verzeichnis 'sql/lib/samples/federated/umplugin/ldap/'. Die folgende Tabelle enthält Beschreibungen zu den einzelnen Dateien. Kopieren Sie die Dateien vor dem Ändern in ein leeres Arbeitsverzeichnis. Bearbeiten Sie dann die Kopien.

Tabelle 27. Beschreibung der Dateien des Beispiel-Plug-ins für Benutzerzuordnungen (Java)

Dateiname	Beschreibung
README.txt	Diese Datei enthält eine komprimierte Version der Anweisungen und Dokumentation zum Testen und Verwenden des Beispiel-Plug-ins.
UserMappingCryptoLDAP.java	Diese Java-Klasse enthält den Code, der die Sicherheitsmaßnahmen für das Verschlüsseln, Entschlüsseln, Codieren und Decodieren der vom LDAP-Server abgerufenen Benutzerzuordnungen implementiert. Sie müssen diese Datei modifizieren, um sie mit Ihrem LDAP-Server einsetzen zu können.
UserMappingSetupLDAP.java	Diese Java-Klasse erstellt die Konfigurationsdatei, in der LDAP-Verbindungsinformationen und andere Konfigurationsparameter, wie IP-Adresse bzw. Hostname, SSL- bzw. Nicht-SSL-Informationen, Benutzer-ID und Kennwort, gespeichert werden.
UserMappingRepositoryLDAP.java	Diese Java-Klasse enthält den Code zum Herstellen und Trennen einer Verbindung zum LDAP-Server sowie zum Abrufen von Benutzerzuordnungen von diesem LDAP-Server. Der Code für diese Datei verwendet das in der Datei <code>schema.ldif</code> definierte Schema. Wenn Sie das Schema ändern wollen, müssen Sie auch einen Abschnitt dieser Datei ändern.
UserMappingLookupLDAP.java	Diese Java-Klasse enthält den Code zum Durchführen eines LDAP-Suchfunktionstests. Mit dieser Datei können Sie das Plug-in unabhängig vom Server mit föderierten Datenbanken testen. Testen Sie das Plug-in anschließend auf dem Server mit föderierten Datenbanken.
schema.ldif	Diese Datei wird auf den LDAP-Server geladen, um das Schema zu definieren. Die LDIF-Datei enthält Objekte und Attribute, die dem LDAP-Server hinzugefügt werden.
entry.ldif	Die Datei <code>entry.ldif</code> fügt dem LDAP-Server Benutzereinträge hinzu. Die Benutzereinträge verwenden die Objekte und Attribute aus der Datei <code>schema.ldif</code> zum Speichern der Benutzerzuordnungen.

Entwickeln eines Beispiel-Plug-ins für Benutzerzuordnungen (Programmiersprache Java)

Nutzen Sie den Beispielcode als Ausgangspunkt für die Entwicklung eines Java-Plug-ins, das Benutzerzuordnungen aus einem externen Repository abrufen. Der Beispielcode ruft Zuordnungen aus einem LDAP-Repository ab, Sie können den Code jedoch so ändern, dass auf ein beliebiges externes Repository zugegriffen werden kann.

Überprüfen Sie Folgendes:

- JDK (Java Development Kit) Version 1.4 oder höher ist installiert.
- Die Datei `db2umplugin.jar` ist vorhanden. Diese JAR-Datei (Java Archive) wird bei der DB2-Server- oder DB2-Clientinstallation installiert.

- Die Beispieldateien für das Plug-in für Benutzerzuordnungen sind installiert. Diese Dateien werden bei der DB2-Clientinstallation im Verzeichnis `sql1lib/samples/federated/umplugin/ldap/` installiert.
- Der Parameter `java_heap_sz` ist mit 2048 definiert.

Das von Ihnen entwickelte Plug-in muss in der Lage sein, eine Verbindung zu dem externen Repository herzustellen, Benutzerzuordnungen abzurufen und ferne Kennwörter zu entschlüsseln. Die Codierung des Plug-ins richtet sich nach dem verwendeten Repository. Wenn Sie beispielsweise ein LDAP-Repository verwenden, in dem verschlüsselte Kennwörter gespeichert werden, muss das Plug-in das Verschlüsselungsschema und den geheimen Schlüssel für die Entschlüsselung der Kennwörter enthalten.

Denken Sie bei der Entwicklung und Verwendung des Plug-ins daran, dass Sie Benutzer-IDs und Kennwörter zwischen verschiedenen Quellen übertragen. Um diese Informationen zu schützen, müssen Sie deshalb den Zugriff auf den Quellcode des Plug-ins einschränken und die Funktion `'db2audit'` so konfigurieren, dass bei jeder Verwendung des Plug-ins durch den Server mit föderierten Datenbanken ein VALIDATE-Datensatz in der Datei `db2diag.log` mit dem Diagnoseprogrammprotokoll erfasst wird. Die Datei mit dem Diagnoseprogrammprotokoll ist nicht nur für die Protokollierung der Plug-in-Verwendung, sondern auch bei der Behebung von eventuell auftretenden Fehlern nützlich.

Führen Sie zum Entwickeln eines Plug-ins die folgenden Tasks aus:

Modifizieren der Plug-in-Beispieldateien für Benutzerzuordnungen (Programmiersprache Java)

Das in Java geschriebene Plug-in ruft Benutzerzuordnungen von einem LDAP-Server ab. Sie können das Plug-in modifizieren, um Benutzerzuordnungen aus anderen Typen externer Repositories abzurufen. Verwenden Sie das Beispiel-Plug-in als Ausgangspunkt für die Entwicklung eines angepassten Plug-ins, das mit Ihrem externen Repository arbeitet.

Die einzelnen Beispieldateien erfüllen unterschiedliche Tasks beim Abrufen von Benutzerzuordnungen. Sie können Funktionen und Klassen im Beispielcode modifizieren, sodass sie Ihr externes Repository verwenden. Die folgende Tabelle enthält die wichtigsten Funktionen.

Tabelle 28. Zu ändernde Funktionen und Klassen

Dateiname	Zu modifizierendes Element	Referenzklasse
UserMappingCryptoLDAP.java	UserMappingCryptoLDAP() encrypt() decrypt() getKey() decode() encode()	UserMappingCrypto (Klasse)UserMappingException (Klasse)
UserMappingRepositoryLDAP.java	class UserMappingRepositoryLDAP(String configFile) connect() disconnect() fetchUM()	UserMappingRepository (Klasse)UserMappingEntry (Klasse)UserMappingOption (Klasse)UserMappingException (Klasse)

Tabelle 28. Zu ändernde Funktionen und Klassen (Forts.)

Dateiname	Zu modifizierendes Element	Referenzklasse
UserMappingSetupLDAP.java	Modifizieren Sie diese Datei, um eine Konfigurationsdatei zu erstellen, in der die Werte gespeichert werden, die Ihr Plug-in benötigt, um eine Verbindung zum externen Repository herzustellen und von dort Benutzerzuordnungen abzurufen. Wenn Sie eine Konfigurationsdatei manuell erstellen wollen, achten Sie darauf, dass alle in der Datei gespeicherten Kennwörter verschlüsselt werden. Der Name der Konfigurationsdatei muss dem Namen der Repositoryklasse entsprechen (beispielsweise Klasse 'UserMappingRepositoryXXXX' und Datei 'UserMappingRepositoryXXXX.cfg').	

Ändern der Beispieldatei 'UserMappingCryptoLDAP' (Programmiersprache Java)

Ändern Sie zum Implementieren der vom LDAP-Server verwendeten Sicherheitsverfahren Funktionen, die ferne Kennwörter verschlüsseln, entschlüsseln, codieren oder decodieren.

Da Verschlüsselungsverfahren geheim und einzigartig sein müssen, werden im Folgenden die zu ändernden Abschnitte und Funktionen der Datei aufgeführt. Es wird erläutert, wie Sie den zu implementierenden Code an die Sicherheitsverfahren anpassen können, die der LDAP-Server verwendet.

Gehen Sie zum Ändern der Sicherheitsfunktionen in der Datei 'UserMappingCryptoLDAP' wie folgt vor:

1. Öffnen Sie die Datei UserMappingCryptoLDAP.java mit einem Texteditor.
2. Importieren Sie unter dem Abschnitt über das Copyright und den Haftungsausschluss von IBM die Pakete, auf die Ihr Code verweist. Das Beispiel-Plug-in verwendet die Java-Pakete 'javax.crypto' und 'javax.crypto.spec', die die Klassen zum Ver- und Entschlüsseln sowie die Schlüssel- und die Algorithmusparameter enthalten. Ersetzen Sie diese Java-Pakete durch eigene Pakete.
3. Aktualisieren Sie die folgenden Funktionen:

public UserMappingCryptoLDAP()

Ersetzen Sie den für die Verschlüsselung verwendeten Code durch den Code, der der vom LDAP-Server verwendeten Kennwortverschlüsselung entspricht.

public byte[] encrypt(byte[] plainValue)

Diese Funktion stellt den Code bereit, mit dem die Kennwörter verschlüsselt werden, damit sie auf dem LDAP-Server gespeichert werden können. Diese Funktion verschlüsselt auch das LDAP-Verbindungs-kennwort, das in der Konfigurationsdatei gespeichert ist.

Ersetzen Sie den Code dieser Funktion durch Ihren eigenen Code, der den Parameter **plainValue** verschlüsselt.

public byte[] decrypt(byte[] encryptedValue)

Ersetzen Sie den Code dieser Funktion durch Ihren eigenen Code, der den Parameter **encryptedValue** entschlüsselt.

private SecretKey getKey()

Ersetzen Sie den Code dieser Funktion durch den Code, der dem Plug-in den Schlüssel bereitstellt, der zum Verschlüsseln und Entschlüsseln Ihrer Kennwörter verwendet wird.

public byte[] decode(String string)

Die Kennwörter werden zunächst verschlüsselt und anschließend codiert. Diese Funktion stellt den Code zum Decodieren der Kennwörter vor deren Entschlüsselung bereit. Die verschlüsselten Kennwörter werden codiert, um die Binärausgabe des verschlüsselten Kennworts in ASCII-Zeichen umzusetzen.

Ersetzen Sie den Code dieser Funktion durch Ihren eigenen Code, der den Parameter **string** entschlüsselt.

public String encode(byte[] bytes)

Die Kennwörter werden zunächst verschlüsselt und anschließend codiert. Diese Funktion stellt den Code zum Codieren der Binärausgabe der verschlüsselten Kennwörter bereit. Die verschlüsselten Kennwörter werden codiert, um die Binärausgabe des verschlüsselten Kennworts in ASCII-Zeichen umzusetzen.

Ersetzen Sie den Code dieser Funktion durch Ihren eigenen Code, der den Parameter **bytes** verschlüsselt.

Modifizieren der Beispieldatei 'UserMappingRepositoryLDAP' (Programmiersprache Java)

Die Datei `UserMappingRepositoryLDAP.java` für das Java-Plug-in enthält die Funktionen zum Herstellen einer Verbindung zum LDAP-Server, zum Abrufen der Benutzerzuordnungen und zum Trennen der Verbindung. Modifizieren Sie den Beispielcode entsprechend den Objektklassen, die in der Verzeichnisstruktur des LDAP-Servers definiert sind.

Um die Benutzerzuordnungen vom LDAP-Server abzurufen, muss das Plug-in das Verzeichnis nach den Benutzereinträgen mit den Attributen durchsuchen, die die einzelnen Benutzerzuordnungen definieren. Die folgenden Informationen werden als Attribute eines Benutzers gespeichert:

- Name des fernen Servers
- Instanzname
- Datenbankname
- Name des fernen Benutzers
- Kennwort des fernen Benutzers

Beim Beispielcode wird davon ausgegangen, dass der Benutzereintrag über die Objektklasse `inetOrgPerson` und der Benutzerzuordnungseintrag über die Objektklasse `IUserMapping` gekennzeichnet ist. Die LDIF-Beispieldateien (Lightweight Directory Interchange Format) `schema.ldif` und `entry.ldif` werden zum Laden des Beispielschemas und der Beispielinträge in den LDAP-Server verwendet.

Beim Code der Datei `UserMappingRepositoryLDAP.java` wird davon ausgegangen, dass die Datei `schema.ldif` den LDIF-Code für Objekte und Attribute mit den folgenden Namen enthält:

- `IUserMapping` (Objekt)
 - `IIRemoteServerName` (Attribut)
 - `IIInstanceName` (Attribut)
 - `IIDatabaseName` (Attribut)
 - `IIRemotePassword` (Attribut)
 - `uid` (Attribut)

Sie müssen die Datei `UserMappingRepositoryLDAP.java` entsprechend dem Schema des LDAP-Servers modifizieren. Die Datei `UserMappingRepositoryLDAP.java` sucht die Benutzerzuordnungseinträge auf dem LDAP-Server und ruft sie von diesem Server ab. Die LDIF-Dateien werden als Beispielschema und Beispielmethode für das Speichern von Benutzerzuordnungseinträgen bereitgestellt.

Gehen Sie zum Modifizieren des Schemas, das von der Datei `UserMappingRepositoryLDAP.java` verwendet wird, wie folgt vor:

1. Suchen Sie den Code `private String UserObjectClassName = "inetOrgPerson"`, und ersetzen Sie den Wert `inetOrgPerson` durch den Namen der Objektklasse für Benutzereinträge, die von Ihrem LDAP-Server für Benutzereinträge verwendet wird.
2. Optional: Ändern Sie die Attributnamen, die vom Plug-in verwendet werden. Ersetzen Sie die Werte der Variablen `IIRemoteServerAttrName`, `IIInstanceAttrName`, `IIDatabaseAttrName` und `IIRemotePasswordAttrName` durch die von Ihnen gewählten Attributnamen.
3. Wenn Sie die LDIF-Dateien verwenden, stellen Sie sicher, dass das Schema in den LDIF-Dateien der Struktur entspricht, die von der Datei `UserMappingRepositoryLDAP.java` durchsucht wird.

Kompilieren der Dateien für das Plug-in für Benutzerzuordnungen (Programmiersprache Java)

Nach dem Modifizieren müssen Sie die Quellendateien für das Java-Plug-in kompilieren.

Enthält der vollständige Pfad Leerzeichen, muss er in den nachstehenden Befehlen in Anführungszeichen gesetzt werden. Beispiel: `"C:\program files\sql11b\java\db2umplugin.jar"`. `%DB2PATH%` gibt das Verzeichnis an, in dem DB2 installiert ist, z. B. `C:\ProgramFiles\IBM\sql11b`. `inst_home` steht für das Ausgangsverzeichnis der Instanz.

Bei den nachstehenden Befehlen wird davon ausgegangen, dass die Dateinamenskennung verwendet wird, die auf den Namen der Klassen basiert. Ersetzen Sie `xxxx` durch den von Ihnen verwendeten Namen.

Gehen Sie zum Kompilieren der Quellendateien wie folgt vor:

1. Setzen Sie den folgenden Kompilierbefehl ab:

Windows:

```
javac -classpath "%DB2PATH%\java\db2umplugin.jar; ^
%CLASSPATH%" -d . ^
.\UserMappingRepositoryxxxx.java ^
.\UserMappingCryptoxxxx.java ^
.\UserMappingSetupxxxx.java ^
.\UserMappingLookupXXXX.java
```

UNIX:

```
javac -classpath inst_home/sql11b/java/db2umplugin.jar:\
$CLASSPATH -d . \
./UserMappingRepositoryxxxx.java \
./UserMappingCryptoxxxx.java \
./UserMappingSetupxxxx.java \
./UserMappingLookupxxxx.java
```

2. Archivieren Sie die Java-Klassendateien in einer einzigen Java-Archivdatei (JAR-Datei): Der Punkt nach dem Namen der Ausgabedatei weist den Befehl an, die Dateien in demselben Verzeichnis zu suchen und dort zu erstellen.

Wenn Sie das Verzeichnis ändern, verwenden Sie den entsprechenden Dateipfad für Ihr Betriebssystem (beispielsweise '/home/user/folder' oder 'C:\test\folder').

```
jar -cfM0 UserMappingRepositoryXXXX.jar .
```

Erstellen der Konfigurationsdatei für das Plug-in für Benutzerzuordnungen (Programmiersprache Java)

In der Konfigurationsdatei sind die Verbindungsinformationen gespeichert, die vom Java-Plug-in für die Verbindung zum LDAP-Server verwendet werden.

Führen Sie zum Erstellen der Konfigurationsdatei das Konfigurationsprogramm aus. Sie werden zur Eingabe der folgenden Angaben aufgefordert:

- Hostname oder IP-Adresse des LDAP-Servers
- Portnummer des LDAP-Servers (Standardwert: 389)
- Definierter Name der LDAP-Unterverzeichnisstruktur (beispielsweise ou=ii,o=ibm,c=us)
- Benutzer-ID für die Verbindung zum LDAP-Server
- Kennwort für die Verbindung zum LDAP-Server
- SSL-Konfiguration

Die von Ihnen eingegebenen Angaben werden zum Erstellen der Datei UserMappingRepositoryLDAP.cfg verwendet, in der die Konfigurationsdaten gespeichert werden. Ist zum Herstellen der Verbindung zum LDAP-Server ein Kennwort erforderlich, wird das Kennwort anhand des von Ihnen in der Datei UserMappingCryptoLDAP.java angegebenen Algorithmus verschlüsselt.

Enthält der vollständige Pfad Leerzeichen, dann muss er in den nachstehenden Befehlen in Anführungszeichen gesetzt werden. Beispiel: "C:\program files\sqllib\java\db2umplugin.jar". %DB2PATH% gibt das Verzeichnis an, in dem DB2 installiert ist, z. B. C:\ProgramFiles\IBM\sqllib. *inst_home* steht für das Ausgangsverzeichnis der Instanz.

Geben Sie zum Erstellen der Konfigurationsdatei für das LDAP-Beispiel-Plug-in Folgendes ein:

Windows:

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^  
.\UserMappingRepositoryLDAP.jar;%CLASSPATH%" UserMappingSetupLDAP
```

UNIX:

```
java -classpath inst_home/sqllib/java/db2umplugin.jar: \  
./UserMappingRepositoryLDAP.jar:$CLASSPATH UserMappingSetupLDAP
```

Testen des Plug-ins für Benutzerzuordnungen (Programmiersprache Java)

Entwickeln Sie zum Testen des Java-Plug-ins unabhängig vom Server mit föderierten Datenbanken eine Anwendung, die eine Verbindung zu dem externen Repository herstellt und von dort Benutzerzuordnungen abrufen.

Sie können ein einfaches Programm entwickeln, das die Methode lookupUM() aufruft, die die Klasse 'UserMappingRepositoryXXXX' von der Klasse 'UserMappingRepository' übernommen hat, um eine Verbindung zu dem externen Repository herzustellen und von dort Benutzerzuordnungen abzurufen. Sie können die Datei

'UserMappingLookupLDAP.java' anzeigen, die sich im Verzeichnis `sqllib/samples/federated/umplugin/ldap/` befindet.

Enthält der vollständige Pfad Leerzeichen, dann muss er in den nachstehenden Befehlen in Anführungszeichen gesetzt werden. Beispiel: `"C:\program files\sqllib\java\db2umplugin.jar"`. `%DB2PATH%` gibt das Verzeichnis an, in dem DB2 installiert ist, z. B. `C:\ProgramFiles\IBM\sqllib`. `inst_home` steht für das Ausgangsverzeichnis der Instanz.

Das Testprogramm muss die Parameter verwenden, die zur Identifizierung der Benutzerzuordnung erforderlich sind:

- `remoteServerName` - Der Name des fernen Servers für die Datenquelle
- `iiAuthid` - Die lokale Benutzer-ID, die der Benutzerzuordnung zugeordnet ist
- `iiInstance` - Der Name der Instanz des Servers mit föderierten Datenbanken
- `iiDatabase` - Der Name der Datenbank auf dem Server mit föderierten Datenbanken

Gehen Sie zum Testen des Plug-ins für Benutzerzuordnungen wie folgt vor:

Windows:

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^
.\UserMappingRepositoryXXXX.jar;%CLASSPATH%" ^
UserMappingLookupXXXX -server remoteServerName ^
-authid iiAuthid ^
-instance iiInstance -database iiDatabase
```

UNIX:

```
java -classpath inst_home/sqllib/java/db2umplugin.jar: ^
./UserMappingRepositoryXXXX.jar:$CLASSPATH \
UserMappingLookupXXXX -server remoteServerName \
-authid iiAuthid \
-instance iiInstance -database iiDatabase
```

Implementieren der Plug-in-Dateien für Benutzerzuordnungen (Programmiersprache Java)

Implementieren Sie nach dem Kompilieren und Testen des Java-Plug-ins die Dateien auf dem Server mit föderierten Datenbanken.

In den nachstehenden Befehlen ist `sqllib` der vollständige Pfad zu Ihrer DB2-Installation. Bei den Befehlen wird davon ausgegangen, dass eine Dateinamenskennung verwendet wird, die auf den Namen der Klassen basiert. Ersetzen Sie `xxxx` durch den von Ihnen verwendeten Namen.

Gehen Sie zum Implementieren der kompilierten Dateien wie folgt vor:

Kopieren Sie die Dateien `UserMappingRepositoryxxxx.jar` und `UserMappingRepositoryxxxx.cfg` in das Verzeichnis `sqllib/function/`.

Das Speichern der Dateien in diesem Verzeichnis ermöglicht es dem Server mit föderierten Datenbanken, die Klassen in diesen Dateien zu laden und aufzurufen.

Konfigurieren des Zugriffs auf das Plug-in für Benutzerzuordnungen (Programmiersprache Java)

Sie müssen die Option `DB2_UM_PLUGIN` definieren, um den Server mit föderierten Datenbanken für die Verwendung eines Java-Plug-ins zum Abrufen von Benutzerzuordnungen zu konfigurieren.

Vorbereitung

Bevor Sie den Server mit föderierten Datenbanken für den Zugriff auf die Benutzerzuordnungen in einem externen Repository konfigurieren, müssen Sie folgende Tasks ausführen:

- Ein Plug-in für Benutzerzuordnungen entwickeln
- Das Plug-in auf dem Server mit föderierten Datenbanken implementieren
- Die Konfiguration des Datenbankmanagers aktualisieren

```
db2 update dbm cfg using JDK_PATH ihr_jdk-pfad
db2 terminate
db2stop
db2start
```

Die Option DB2_UM_PLUGIN muss den vollständigen Pfad der Klasse enthalten, einschließlich des Paketnamens. Wenn Sie ein Paket entwickeln, müssen Sie den Paketnamen vor dem Klassennamen einschließen. Beispiel: 'paket.klassenname'. Wenn Sie das LDAP-Beispiel-Plug-in verwenden, geben Sie den Wert 'UserMappingRepositoryLDAP' in der Option DB2_UM_PLUGIN an. Das Beispiel-Plug-in ist nicht als Paket entwickelt.

Gehen Sie wie folgt vor, um den Server mit föderierten Datenbanken für den Zugriff auf das externe Repository zu konfigurieren:

Wählen Sie aus, wie das Plug-in für Benutzerzuordnungen implementiert werden soll:

Methode	SQL-Anweisung
Angeben des Plug-ins für Benutzerzuordnungen beim Erstellen eines Wrappers	CREATE WRAPPER <i>wrappername</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');
Ändern eines vorhandenen Wrappers zum Angeben des Plug-ins für Benutzerzuordnungen	ALTER WRAPPER <i>wrappername</i> (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');
Angeben des Plug-ins für Benutzerzuordnungen beim Erstellen einer Serverdefinition	CREATE SERVER <i>servername</i> TYPE <i>datenquellentyp</i> VERSION <i>versionsnummer</i> WRAPPER <i>wrappername</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');
Ändern einer vorhandenen Serverdefinition zum Angeben des Plug-ins für Benutzerzuordnungen	ALTER SERVER <i>servername</i> OPTIONS (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');

Nach Definition der Option DB2_UM_PLUGIN ruft der Server mit föderierten Datenbanken Benutzerzuordnungen anhand der in der Datei UserMappingRepositoryXXXX.cfg von Ihnen angegebenen Verbindungsinformationen aus dem externen Repository ab. XXXX ist hierbei der Name des Plug-ins.

Ändern des Wrappers, um ein anderes Plug-in zu verwenden:

```
ALTER WRAPPER wrappername (  
    SET DB2_UM_PLUGIN 'com.package_name.um.UserMappingRepositoryXXXX'  
);
```

Ändern einer Serverdefinition, um ein anderes Plug-in zu verwenden:

```
ALTER SERVER servername
  OPTIONS (
    SET DB2_UM_PLUGIN 'UserMappingRepositoryXXX'
  );
```

Kapitel 34. Oracle-Sicherheit in einem System föderierter Datenbanken

Federation unterstützt eine Reihe von Oracle-Sicherheitsfunktionen.

Wenn Sie diese Funktionen in Oracle implementieren, können sie vom Server mit föderierten Datenbanken genutzt werden.

Oracle Label Security

Der Server mit föderierten Datenbanken unterstützt Oracle Label Security, das dazu verwendet werden kann, Daten zu schützen und sicherzustellen, dass nur Benutzer mit der entsprechenden Berechtigung diese Daten anzeigen können.

Administratoren können Oracle Label Security nutzen, um Sicherheitsrichtlinien für die einzelnen Zeilen einer Tabelle anzuwenden. Die Sicherheitsrichtlinien bestimmen die Zugriffsebene eines Benutzers für die Daten auf der Basis der Berechtigungen, die der Benutzer-ID oder der Sitzungs-ID erteilt wurden. Wenn Sie einen Kurznamen für ein Oracle-Datenquellenobjekt erstellen, ermittelt der Server mit föderierten Datenbanken automatisch, ob die Datenquelle Oracle Label Security verwendet. Wird Oracle Label Security verwendet, wird der Kurzname nicht in den Cache gestellt.

Mit der Anweisung ALTER NICKNAME können Sie das Caching zulassen bzw. nicht zulassen. Beispiel: Wenn Sie einen Kurznamen für ein Datenquellenobjekt mit Oracle Label Security erstellt haben, bevor die Unterstützung für föderierte Datenbanken für diese Funktion verfügbar war, können Sie den Kurznamen so ändern, dass das Caching nicht zugelassen wird. Wenn Sie einen Kurznamen für ein Datenquellenobjekt mit Oracle Label Security erstellt haben und die Oracle Label Security-Funktion entfernt wird, können Sie den Kurznamen so ändern, dass das Caching zugelassen wird.

Ein Datenbankadministrator kann festlegen, dass ein Kennsatz verdeckt wird, um zu verhindern, dass bestimmte Benutzer eine bestimmte Zeile anzeigen können. In diesem Fall wird die Spalte in der Tabelle verdeckt. Kurznamen, für die die Kennsatzspalten ausgeblendet sind, werden nicht in den Cache gestellt.

Oracle-Proxyauthentifizierung und föderierter gesicherter Kontext

Stellen Sie eine physische Verbindung zu einer Oracle-Datenquelle her, und wechseln Sie dann zu einem anderen Benutzer mit derselben Verbindung.

Die Verwendung der Oracle-Proxyauthentifizierung und des föderierten gesicherten Kontexts reduziert den Systemaufwand im Netz, der erforderlich wäre, um für jeden Benutzer eine separate Netzverbindung vom Server mit föderierten Datenbanken zur Oracle-Datenbank herzustellen und dabei die Identität des jeweils verbundenen Benutzers für die Oracle-Datenquelle zu bestätigen. Die Anwendung kann je nach Bedarf von Benutzer zu Benutzer wechseln, um Transaktionen für diese Benutzer zu verarbeiten.

Führen Sie die folgenden Tasks aus, um dieses Szenario zu konfigurieren:

1. Registrieren Sie auf dem Oracle-Server jeden Proxybenutzer mithilfe der Oracle-Anweisung ALTER USER. In diesem Beispiel erhält der Proxybenutzer, in diesem Fall Mary, die Berechtigung, den Proxy BOSS zu verwenden, und bekommt für die Dauer der Verbindung die Rolle CLERK:

```
ALTER USER MARY GRANT  
CONNECT THROUGH BOSS  
WITH ROLE CLERK
```

2. Erstellen Sie auf dem Server mit föderierten Datenbanken das Objekt für gesicherten Kontext:

```
CREATE TRUSTED CONTEXT MY_FED_TCX  
BASED UPON CONNECTION USING SYSTEM AUTHID BOSS  
ATTRIBUTES (ENCRYPTION 'NONE')  
WITH USE FOR MARY WITHOUT AUTHENTICATION  
ENABLE
```

Mit dieser Konfiguration kann der Server mit föderierten Datenbanken eine gesicherte End-to-End-Verbindung vom Client über den Server mit föderierten Datenbanken zur Oracle-Datenquelle einrichten. BOSS kann eine gesicherte Verbindung herstellen, und MARY kann diese wiederverwenden.

Verwenden Sie für die Einrichtung und Wiederverwendung von gesicherten Verbindungen die von DB2 bereitgestellte API.

Kapitel 35. Datenquellenunterstützung für föderierte Funktionen

Anhand dieser Tabelle können Sie ermitteln, ob eine Datenquelle eine bestimmte föderierte Funktion unterstützt oder nicht.

Bevor Sie die hier aufgeführten Funktionen verwenden können, müssen Sie bestimmte Wrapper- oder Serveroptionen festlegen oder andere Tasks ausführen, damit die Funktionalität aktiviert ist. Weitere Informationen finden Sie im speziellen Abschnitt für die jeweilige Funktion.

Tabelle 29. Funktionen und unterstützte Datenquellen

Funktion	Datenquellen
Anwendungssicherungspunkte mit Operationen WRITE für Kurznamen	DB2 für Linux, UNIX und Windows
Asynchronitätsoptimierung	Alle Datenquellen
Cachetabellen	DB2-Produktfamilie Informix Microsoft SQL Server Oracle Sybase
Import von Daten in Kurznamen	DB2-Produktfamilie Informix Microsoft SQL Server Oracle Sybase Teradata
Fehlertoleranz in verschachtelten Tabellenausdrücken	DB2-Produktfamilie Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Teradata
Externes Repository für die Benutzerzuordnung	Alle Datenquellen
Diagnoseanzeiger von föderierten Systemen	DB2-Produktfamilie Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Dateien mit Tabellenstruktur Teradata XML (nur Rootkurznamen)
Föderierte Prozeduren	DB2-Produktfamilie im gesicherten Modus Oracle im gesicherten Modus Microsoft SQL Server im gesicherten Modus Sybase im abgeschirmten Modus bei unter UNIX installiertem Server mit föderierten Datenbanken Sybase im abgeschirmten oder gesicherten Modus bei unter Linux oder Microsoft Windows installiertem Server mit föderierten Datenbanken

Kapitel 36. Referenz für Datenquellenoptionen

Jede Datenquelle unterstützt bestimmte Wrapper-, Server-, Benutzerzuordnungs-, Kurznamen- und Spaltenoptionen.

Referenz für BioRS-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen aufgelistet, die für diese Datenquelle gelten, und die erforderlichen Optionen gezeigt, die Sie in den Anweisungen CREATE WRAPPER und CREATE SERVER angeben müssen.

Tabelle 30. Wrapperoptionen für BioRS

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Diese Option ist erforderlich, wenn der Wert für PROXY_TYPE HTTP oder SOCKS lautet. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Diese Option ist erforderlich, wenn der Wert für PROXY_TYPE HTTP oder SOCKS lautet. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Servicenamen.

Tabelle 30. Wrapperoptionen für BioRS (Forts.)

Name	Beschreibung
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE , HTTP und SOCKS . Der Standardwert ist NONE . Wenn Sie diese Option auf HTTP oder SOCKS gesetzt haben, müssen Sie auch PROXY_SERVER_NAME und PROXY_SERVER_PORT angeben.

Serveroptionen

Tabelle 31. Serveroptionen für BioRS

Name	Beschreibung
CASE_SENSITIVE	Gibt an, ob der BioRS-Server bei Namen die Groß-/Kleinschreibung beachtet. Gültige Werte sind Y und N . Der Standardwert ist Y . Hierbei wird bei Namen die Groß-/Kleinschreibung beachtet. Im Produkt BioRS steuert ein Konfigurationsparameter die Beachtung bzw. Nichtbeachtung der Groß-/Kleinschreibung für die Daten, die auf dem BioRS-Server gespeichert sind. Die Option CASE_SENSITIVE und der Konfigurationsparameter müssen dieselbe Einstellung dafür aufweisen, ob die Groß-/Kleinschreibung beachtet wird oder nicht. Wenn Sie den Wert der Option CASE_SENSITIVE ändern müssen, nachdem Sie die Serverdefinition erstellt haben, müssen Sie die Serverdefinition löschen und die Definition sowie alle Kurznamen erneut erstellen.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.

Tabelle 31. Serveroptionen für BioRS (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN_LANG	Gibt die Sprache für das Plug-in für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
NODE	Erforderlich. Gibt den DNS-Hostnamen oder die IP-Adresse des Systems an, auf dem das BioRS-Abfragetool verfügbar ist. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist. Der Standardwert ist localhost.
PORT	Gibt den Port für Verbindungen zum BioRS-Server an. Gültige Werte sind ein numerischer Port oder ein TCP/IP-Service-Name. Der Standardwert ist 5014.
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an.
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Service-Name für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Service-Name.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE , HTTP und SOCKS . Der Standardwert ist NONE .
TIMEOUT	Gibt an, wie viele Minuten der Server mit föderierten Datenbanken maximal auf eine Antwort vom fernen Server wartet. Der Standardwert ist 10.

Benutzerzuordnungsoptionen

Tabelle 32. Benutzerzuordnungsoptionen für BioRS

Option	Beschreibung
GUEST	Gibt an, dass die BioRS-Authentifizierungs-ID GUEST verwendet wird, um Operationen auszuführen. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird die BioRS-ID GUEST nicht verwendet. Diese Option ist nicht gültig, wenn Sie die Optionen REMOTE_AUTHID und REMOTE_PASSWORD angeben.
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an. Das Kennwort wird verschlüsselt, wenn es im Katalog für föderierte Datenbanken gespeichert wird.
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Kurznamenoptionen

Tabelle 33. Kurznamenoptionen für BioRS

Option	Beschreibung
REMOTE_OBJECT	Gibt den Namen der BioRS-Databank an, die dem Kurznamen zugeordnet ist. Dieser Name bestimmt das Schema und die BioRS-Databank für den Kurznamen. Der Name gibt außerdem die Beziehung des Kurznamens zu anderen Kurznamen an. Ob bei dieser Option die Groß-/Kleinschreibung beachtet werden muss, hängt von den entsprechenden Eigenschaften des BioRS-Servers und vom Wert der Serveroption CASE_SENSITIVE ab. Sie können die Anweisung ALTER NICKNAME nicht verwenden, um diesen Namen zu ändern oder zu löschen. Wenn sich der Name der BioRS-Databank ändert, müssen Sie den Kurznamen löschen und dann erneut erstellen.
TIMEOUT	Gibt an, wie viele Minuten maximal auf eine Antwort vom Datenquellenserver gewartet wird. Der Standardwert ist 10.

Spaltenoptionen

Tabelle 34. Spaltenoptionen für BioRS

Option	Beschreibung
ELEMENT_NAME	Gibt den BioRS-Elementnamen an. Ob bei diesem Namen die Groß-/Kleinschreibung beachtet werden muss, hängt von den entsprechenden Eigenschaften des BioRS-Servers und vom Wert der Serveroption CASE_SENSITIVE ab. Sie müssen den BioRS-Elementnamen nur dann angeben, wenn er vom Spaltennamen abweicht.
IS_INDEXED	Gibt an, ob die entsprechende Spalte indiziert ist und daher in einem Vergleichselement angegeben werden kann. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei ist die Spalte nicht indiziert.
REFERENCED_OBJECT	Gibt den Namen der BioRS-Datenbank an, auf die von der aktuellen Spalte verwiesen wird. Ob bei diesem Namen die Groß-/Kleinschreibung beachtet werden muss, hängt von den entsprechenden Eigenschaften des BioRS-Servers und vom Wert der Serveroption CASE_SENSITIVE ab. Diese Option ist nur für Spalten gültig, die den BioRS-Datentyp für Verweise (Reference) aufweisen.

Referenz für Optionen der DB2-Datenbank

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen sind die Optionen, die für DB2-Datenquellen gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 35. Wrapperoptionen für DB2-Datenquellen

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.

Tabelle 35. Wrapperoptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 36. Serveroptionen für DB2-Datenquellen

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.

Tabelle 36. Serveroptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
DATE_COMPAT	Gibt an, ob der Parameter 'date_compat' auf die Datenbank angewendet wird. Gültige Werte sind Y und N. Der Standardwert ist N. Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.7 oder höher gültig.
DBNAME	Erforderlich. Gibt die spezielle Datenbank an, die für die einleitende ferne DB2-Datenbankverbindung verwendet werden soll. Diese spezielle Datenbank ist der Datenbankaliasname für die ferne DB2-Datenbank, die auf dem Server mit föderierten Datenbanken mit dem Befehl CATALOG DATABASE oder dem DB2-Konfigurationsassistenten katalogisiert wird.
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_TWO_PHASE_COMMIT	Gibt an, ob der Server mit föderierten Datenbanken über ein Protokoll für zweiphasige Commits oder über ein Protokoll für einphasige Commits eine Verbindung zur Datenquelle herstellt. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stellt der Server mit föderierten Datenbanken die Verbindung über ein Protokoll für einphasige Commits her. Y gibt an, dass der Server mit föderierten Datenbanken die Verbindung über das Protokoll für zweiphasige Commits herstellt.

Tabelle 36. Serveroptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
FED_PROXY_USER	Gibt die Berechtigungs-ID an, die verwendet werden soll, um alle abgehenden gesicherten Verbindungen herzustellen, wenn die eingehende Verbindung nicht gesichert ist. Der Benutzer, dessen ID in dieser Option angegeben ist, muss eine Benutzerzuordnung haben, die sowohl die Option REMOTE_AUTHID als auch die Option REMOTE_PASSWORD angibt. Einschränkung: Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.5 oder höher und DB2 für z/OS Version 9 oder höher gültig.
FOLD_ID	Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.
FOLD_PW	Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.

Tabelle 36. Serveroptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
IO_RATIO	<p>Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.</p>
NO_EMPTY_STRING	<p>Gibt an, ob der Server der fernen Datenquelle leere Zeichenfolgen enthalten kann. Gültige Werte sind Y und N. Der Standardwert variiert abhängig von der fernen Datenquelle. Bei fernen Oracle-Datenquellen lautet der Standardwert Y; hierbei werden alle leeren Zeichenfolgewerte in Nullwerte konvertiert. Bei allen anderen fernen Datenquellen lautet der Standardwert N; hierbei kann die Datenquelle leere Zeichenfolgen enthalten.</p> <p>Sie können die Systemleistung verbessern, indem Sie diese Option bei Systemkonfigurationen auf Y setzen, bei denen der föderierte Server einen mit VARCHAR2 kompatiblen Modus aufweist, die ferne Datenquelle jedoch nicht mit VARCHAR2 kompatibel ist.</p>
NUMBER_COMPAT	<p>Gibt an, ob der Datenquellenserver den Datentyp NUMBER unterstützt. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei unterstützt der Datenquellenserver den Datentyp NUMBER nicht. In Systemen, bei denen der Server mit föderierten Datenbanken den Datentyp NUMBER nicht unterstützt, der Datenquellenserver hingegen schon, müssen Sie die Option NUMBER_COMPAT auf Y setzen, da die Möglichkeit besteht, dass der Datenquellenserver Ergebnisse des Typs DECFLOAT zurückgibt, die außerhalb des Bereichs für den Datentyp DECIMAL liegen und den Fehler SQLSTATE 560BD verursachen.</p> <p>Einschränkung: Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.7 und höher gültig.</p>

Tabelle 36. Serveroptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
OLD_NAME_GEN	<p>Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.</p>
PUSHDOWN	<p>Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N. Der Standardwert ist Y. Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.</p>
SAME_DECFLT_ROUNDING	<p>Gibt an, ob der Server mit föderierten Datenbanken und der Datenquellenserver beim Rundungsmodus dieselben Rundungsmoduseinstellungen für DECFLOAT verwenden. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei verwenden der Server mit föderierten Datenbanken und der ferne Server unterschiedliche Rundungsmoduseinstellungen für DECFLOAT.</p> <p>Wichtig: Falls Sie diese Option auf Y setzen und die Rundungsmodi des Servers mit föderierten Datenbanken und des Datenquellenservers unterschiedlich sind, erhalten Sie möglicherweise falsche Rundungsergebnisse für DECFLOAT.</p> <p>Um einen vorhandenen Server mit föderierten Datenbanken und einen Datenquellenserver mit derselben Rundungsmoduseinstellung für DECFLOAT zu konfigurieren, verwenden Sie die Anweisung ALTER SERVER.</p> <p>Einschränkung: Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.5 und höher gültig.</p>

Tabelle 36. Serveroptionen für DB2-Datenquellen (Forts.)

Name	Beschreibung
VARCHAR2_COMPAT	<p>Gibt an, ob die ferne Datenquelle mit VARCHAR2 kompatibel ist. Gültige Werte sind Y und N. Der Standardwert variiert abhängig von der fernen Datenquelle. Bei fernen Oracle-Datenquellen lautet der Standardwert Y; dies bedeutet, dass die Datenquelle mit VARCHAR2 kompatibel ist. Bei allen anderen fernen Datenquellen lautet der Standardwert N; hierbei ist die Datenquelle nicht mit VARCHAR2 kompatibel.</p> <p>Sie müssen diese Serveroption auf Y setzen, falls die ODBC-, JDBC- oder DB2 Database für Linux, UNIX und Windows-Datenquelle im VARCHAR2-Kompatibilitätsmodus konfiguriert ist.</p>

Benutzerzuordnungsoptionen

Tabelle 37. Benutzerzuordnungsoptionen für DB2-Datenquellen

Option	Beschreibung
FED_PROXY_USER	<p>Gibt die Berechtigungs-ID an, die verwendet werden soll, um alle abgehenden gesicherten Verbindungen herzustellen, wenn die eingehende Verbindung nicht gesichert ist. Der Benutzer, dessen ID in dieser Option angegeben ist, muss eine Benutzerzuordnung haben, die sowohl REMOTE_AUTHID als auch REMOTE_PASSWORD angibt. Wenn Sie die Benutzerzuordnungsoption FED_PROXY_USER angeben, müssen Sie auch die Serveroption FED_PROXY_USER angeben.</p> <p>Einschränkung: Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.5 oder höher und DB2 für z/OS Version 9 oder höher gültig.</p>
ACCOUNTING_STRING	<p>Diese Option ist erforderlich, wenn Abrechnungsdaten übergeben werden müssen. Sie gibt eine DRDA-Abrechnungszeichenfolge an. Gültige Werte sind beliebige Zeichenfolgen mit maximal 255 Zeichen.</p>
REMOTE_AUTHID	<p>Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.</p>
REMOTE_PASSWORD	<p>Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.</p>

Tabelle 37. Benutzerzuordnungsoptionen für DB2-Datenquellen (Forts.)

Option	Beschreibung
USE_TRUSTED_CONTEXT	<p>Gibt an, ob die Benutzerzuordnung gesichert ist. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei ist die Benutzerzuordnung nicht gesichert und kann nur in nicht gesicherten abgehenden Verbindungen föderierter Systeme verwendet werden. Y gibt an, dass die Benutzerzuordnung gesichert ist und sowohl in gesicherten als auch in nicht gesicherten abgehenden Verbindungen föderierter Systeme verwendet werden kann.</p> <p>Einschränkung: Diese Serveroption ist nur bei DB2 Database für Linux, UNIX und Windows Version 9.5 oder höher und DB2 für z/OS Version 9 oder höher gültig.</p>

Spaltenoptionen

Tabelle 38. Spaltenoptionen für DB2-Datenquellen

Option	Beschreibung
NUMERIC_STRING	<p>Gibt an, wie numerische Zeichenfolgen behandelt werden. Der Standardwert ist N. Wenn die Zeichenfolgespalte für Datenquellen nur numerische Zeichenfolgen und keine anderen Zeichen, einschließlich Leerzeichen, enthält, setzen Sie die Option NUMERIC_STRING auf Y. Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.</p>
NO_EMPTY_STRING	<p>Gibt an, ob der Server der fernen Datenquelle leere Zeichenfolgen enthalten kann. Gültige Werte sind Y und N. Der Standardwert variiert abhängig von der fernen Datenquelle. Bei fernen Oracle-Datenquellen lautet der Standardwert Y; hierbei werden alle leeren Zeichenfolgewerte in Nullwerte konvertiert. Bei allen anderen fernen Datenquellen lautet der Standardwert N; hierbei kann die Datenquelle leere Zeichenfolgen enthalten.</p>

Tabelle 38. Spaltenoptionen für DB2-Datenquellen (Forts.)

Option	Beschreibung
XML_ROOT	Gibt das XML-Stammelement an, das zu den Werten einer XML-Spalte hinzugefügt werden soll, die auf eine XML-Sequenz verweist. Diese Option stellt sicher, dass die Werte der XML-Spalte ein korrekt formatiertes XML-Dokument darstellen.

Referenz für Excel-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server und Kurzname, um zu konfigurieren, wie der Server mit föderierten Datenbanken sowie die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 39. Wrapperoptionen für Excel

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.

Serveroptionen

Tabelle 40. Serveroptionen für Excel

Name	Beschreibung
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.

Kurznamenoptionen

Tabelle 41. Kurznamenoptionen für Excel

Option	Beschreibung
FILE_PATH	Erforderlich. Gibt den vollständig qualifizierten Verzeichnispfad und den Dateinamen des Excel-Arbeitsblatts an, auf das Sie zugreifen wollen.

Tabelle 41. Kurznamenoptionen für Excel (Forts.)

Option	Beschreibung
RANGE	Gibt den zu verwendenden Zellenbereich an, wie z. B. A1:C100. Der Wert vor dem Doppelpunkt gibt die oberste linke Zelle des Bereichs an. Der Wert hinter dem Doppelpunkt gibt die unterste rechte Zelle des Bereichs an.

Referenz für Informix-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen sind die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 42. Wrapperoptionen für Informix

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 43. Serveroptionen für Informix

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass bei dem Wert die Groß-/Kleinschreibung nicht beachtet werden muss. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.
DBNAME	Erforderlich. Gibt den Namen der Informix-Datenbank an, auf die Sie zugreifen wollen.
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet. Wenn mehrere Zugriffspläne dieses Kriterium erfüllen, wird der Plan mit dem geringsten Aufwand ausgewählt.

Tabelle 43. Serveroptionen für Informix (Forts.)

Name	Beschreibung
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_TWO_PHASE_COMMIT	Gibt an, ob der Server mit föderierten Datenbanken über ein Protokoll für zweiphasige Commits oder über ein Protokoll für einphasige Commits eine Verbindung zur Datenquelle herstellt. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stellt der Server mit föderierten Datenbanken die Verbindung über ein Protokoll für einphasige Commits her. Y gibt an, dass der Server mit föderierten Datenbanken die Verbindung über das Protokoll für zweiphasige Commits herstellt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
FOLD_ID	Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.

Tabelle 43. Serveroptionen für Informix (Forts.)

Name	Beschreibung
FOLD_PW	Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.
INFORMIX_CLIENT_LOCALE	Gibt den Wert für die Umgebungsvariable <code>CLIENT_LOCALE</code> an, der für die Verbindung zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver verwendet werden soll. Der Wert kann eine beliebige gültige Informix-Ländereinstellung sein. Wenn Sie diese Option nicht angeben, wird die Umgebungsvariable <code>CLIENT_LOCALE</code> auf den Wert gesetzt, der in der Datei <code>db2dj.ini</code> angegeben ist. Falls die Umgebungsvariable <code>CLIENT_LOCALE</code> in der Datei <code>db2dj.ini</code> nicht definiert ist, wird als Wert für <code>INFORMIX_CLIENT_LOCALE</code> diejenige Informix-Ländereinstellung verwendet, die am ehesten mit der Codepage und dem Gebiet der föderierten Datenbank übereinstimmt.
INFORMIX_DB_LOCALE	Gibt den Informix-Wert für die Umgebungsvariable <code>DB_LOCALE</code> an, der für die Verbindung zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver verwendet werden soll. Falls die Option <code>INFORMIX_DB_LOCALE</code> nicht angegeben ist, wird die Informix-Umgebungsvariable <code>DB_LOCALE</code> auf den Wert gesetzt, der in der Datei <code>db2dj.ini</code> angegeben ist. Ist in der Datei <code>db2dj.ini</code> kein Wert definiert, wird die Informix-Umgebungsvariable <code>DB_LOCALE</code> nicht festgelegt.

Tabelle 43. Serveroptionen für Informix (Forts.)

Name	Beschreibung
INFORMIX_LOCK_MODE	<p>Gibt den Sperrmodus an, der für eine Informix-Datenquelle festgelegt werden soll. Der Informix-Wrapper setzt den Befehl SET LOCK MODE unmittelbar nach dem Herstellen der Verbindung zu einer Informix-Datenquelle ab. Gültige Werte sind W, N und eine Zahl. Der Standardwert ist W. Hierbei wartet der Wrapper mit unbegrenzter Dauer auf die Freigabe der Sperre. Der Wert N gibt an, dass der Wrapper nicht auf die Freigabe warten soll; hierbei wird sofort ein Fehler zurückgegeben. Mit einer Zahl geben Sie die Höchstdauer des Wartezeitraums in Sekunden an. Falls ein Deadlock oder eine Zeitlimitüberschreitung auftritt, können Sie den Wert für die Option INFORMIX_LOCK_MODE mit der Anweisung ALTER SERVER ändern. Beispiel:</p> <pre>ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')</pre>
IO_RATIO	<p>Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.</p>
IUD_APP_SVPT_ENFORCE	<p>Gibt an, ob der Server mit föderierten Datenbanken die Verwendung von Sicherungspunktanweisungen für Anwendungen unterstützt. Gültige Werte sind Y und N. Der Standardwert ist Y. Falls die Datenquelle Sicherungspunktanweisungen für Anwendungen nicht umsetzt und während einer Einfüge-, Aktualisierungs- oder Löschoperation ein Fehler auftritt, macht der Server mit föderierten Datenbanken die Transaktion rückgängig und der SQL-Fehlercode SQL1476N wird zurückgegeben. Es wird empfohlen, die Standardeinstellung zu verwenden.</p>

Tabelle 43. Serveroptionen für Informix (Forts.)

Name	Beschreibung
NODE	Erforderlich. Gibt den Namen an, unter dem die Datenquelle als Instanz für ihr Managementsystem für relationale Datenbanken definiert ist.
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N . Der Standardwert ist Y . Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.

Benutzerzuordnungsoptionen

Tabelle 44. Benutzerzuordnungsoptionen für Informix

Name	Beschreibung
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Spaltenoptionen

Tabelle 45. Spaltenoptionen für Informix

Name	Beschreibung
NUMERIC_STRING	Gibt an, wie numerische Zeichenfolgen behandelt werden. Der Standardwert ist N . Wenn die Zeichenfolgespalte für Datenquellen nur numerische Zeichenfolgen und keine anderen Zeichen, einschließlich Leerzeichen, enthält, setzen Sie die Option NUMERIC_STRING auf Y . Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.

Referenz für JDBC-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 46. Wrapperoptionen für JDBC

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Der einzige gültige Wert ist Y , weil der DB2-Server das Laden der JVM nur im abgeschirmten Modus unterstützt. Der Standardwert ist Y . Hierbei wird der Wrapper im abgeschirmten Modus ausgeführt.

Tabelle 46. Wrapperoptionen für JDBC (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 47. Serveroptionen für JDBC

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.

Tabelle 47. Serveroptionen für JDBC (Forts.)

Name	Beschreibung
DATEFORMAT	Gibt das Datumsformat an, das von der Datenquelle verwendet wird. Verwenden Sie 'TT', 'MM' und 'JJ' bzw. 'JJJJ', um das Datumsformat anzugeben. Sie können ein Begrenzungszeichen angeben, wie z. B. ein Leerzeichen, einen Bindestrich oder ein Komma. Das Format 'JJJJ-MM-TT' gibt beispielsweise ein Datum wie 1958-10-01 an. Der Wert kann Nullwerte enthalten.
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 0. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Tabelle 47. Serveroptionen für JDBC (Forts.)

Name	Beschreibung
DRIVER_CLASS	<p>Gibt die JDBC-Treiberbibliothek an. Sie können den Server für mehrere JDBC-Datenquellen registrieren, falls der JDBC-Treiber der Version 3.0 oder höher der JDBC-Spezifikation entspricht. Angaben zu den JDBC-Spezifikationen sowie Informationen zum Festlegen der Serveroption DRIVER_CLASS finden Sie in der Dokumentation des JDBC-Treibers.</p> <p>Beispiel</p> <p>Im folgenden Beispiel wird die JDBC-Treiberbibliothek <code>com.ibm.db2.jcc.DB2Driver</code> angegeben:</p> <pre>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</pre> <p>Wichtig: Falls Sie diese Option angeben, müssen Sie auch die Serveroption URL angeben.</p>
DRIVER_PACKAGE	<p>Gibt die JDBC-Treiberpakete an. Verwenden Sie ein Pfadtrennzeichen, um mehrere Treiberklassenpakete anzugeben. Verwenden Sie bei Windows-Betriebssystemen hierzu ein Semikolon; bei Linux- und Unix-Betriebssystemen einen Doppelpunkt.</p> <p>Beispiel</p> <p>Im folgenden Beispiel geben Sie mehrere Treiberpakete bei Linux-Betriebssystemen mit einem Doppelpunkt an:</p> <pre>DRIVER_PACKAGE '/path1/file1.jar: /path2/file2.jar'</pre>
FOLD_ID	<p>Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>

Tabelle 47. Serveroptionen für JDBC (Forts.)

Name	Beschreibung
FOLD_PW	<p>Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>
IO_RATIO	<p>Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.</p>
JDBC_LOG	<p>Gibt an, ob der JDBC-Wrapper Protokolldateien für die Fehlertraceverarbeitung erstellt. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei werden keine Protokolldateien erstellt. Falls diese Serveroption auf Y gesetzt ist, schreibt der JDBC-Wrapper JDBC-Protokolldateien in die Datei <code>jdbc_wrapper_produk-t-id.log</code>. Hierbei steht <i>produkt-id</i> für die Produkt-ID. Die Protokolldatei wird in dem Verzeichnis gespeichert, das im Konfigurationsparameter <code>DIAGPATH</code> des DB2-Datenbankmanagers angegeben ist. Auf UNIX-Systemen heißt das Standardverzeichnis <code>inst_home/sql1lib/db2dump</code>. Empfehlung: Wenn Sie diese Serveroption auf Y setzen, wird das Leistungsverhalten beeinträchtigt. Die Protokollierung sollte daher auf Produktionssystemen nicht aktiviert werden.</p>

Tabelle 47. Serveroptionen für JDBC (Forts.)

Name	Beschreibung
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N. Der Standardwert ist Y. Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.
TIMEFORMAT	Gibt das Zeitformat an, das von der Datenquelle verwendet wird. Verwenden Sie 'hh12', 'hh24', 'mm', 'ss', 'AM' und 'A.M.', um das Zeitformat anzugeben. Das Format 'hh12:mm:ss AM' gibt z. B. eine Zeit wie 8:00:00 AM an. Der Wert kann Nullwerte enthalten.
TIMESTAMPFORMAT	Gibt das Zeitmarkenformat an, das von der Datenquelle verwendet wird. Gültige Werte weisen das Format auf, das in der Option DATEFORMAT und in der Option TIMEFORMAT verwendet wird. Geben Sie 'n' für eine Zehntelsekunde, 'nn' für eine Hundertstelsekunde, 'nnn' für Millisekunden usw. bis hin zu 'nnnnnn' für Mikrosekunden an. Das Format 'JJJJ-MM-TT-hh24:mm:ss.nnnnnn' gibt beispielsweise eine Zeitmarke wie 1994-01-01-24:00:00,000000 an. Der Wert kann Nullwerte enthalten.

Tabelle 47. Serveroptionen für JDBC (Forts.)

Name	Beschreibung
URL	<p>Gibt die JDBC-Verbindungszeichenfolge des fernen Servers an.</p> <p>Die JDBC-Verbindungszeichenfolge besteht aus drei Teilen, die jeweils durch einen Doppelpunkt voneinander abgegrenzt sind:</p> <ul style="list-style-type: none"> • Datenbankprotokoll • Datenbanktypname oder Konnektivitätstreibername • Datenbankidentität durch Aliasname oder Subname <p>Beispiel</p> <p>Im folgenden Beispiel lautet die JDBC-Verbindungszeichenfolge <code>jdbc:db2://cn.ibm.com:50471/testdb:</code></p> <p>URL <code>'jdbc:db2://cn.ibm.com:50471/testdb'</code></p> <p>Sie können den Server für mehrere JDBC-Datenquellen registrieren, falls der JDBC-Treiber der Version 3.0 oder höher der JDBC-Spezifikation entspricht. Angaben zu den JDBC-Spezifikationen sowie Informationen zum Festlegen der Serveroption URL finden Sie in der Dokumentation des JDBC-Treibers.</p> <p>Wichtig: Falls Sie diese Option angeben, müssen Sie auch die Serveroption DRIVER_CLASS angeben.</p>
VARCHAR_NO_TRAILING_BLANKS	<p>Gibt an, ob die Datenquelle VARCHAR-Spalten mit mindestens einem abschließenden Leerzeichen enthält. Der Standardwert ist N. Hierbei enthalten die VARCHAR-Spalten mindestens ein abschließendes Leerzeichen.</p>

Benutzerzuordnungsoptionen

Tabelle 48. Benutzerzuordnungsoptionen für JDBC

Name	Beschreibung
REMOTE_AUTHID	<p>Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.</p>
REMOTE_PASSWORD	<p>Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.</p>

Spaltenoptionen

Tabelle 49. Spaltenoptionen für JDBC

Name	Beschreibung
NUMERIC_STRING	Gibt an, ob die Spalte Zeichenfolgen mit numerischen Zeichen enthält, die Leerzeichen einschließen. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei enthält die Spalte keine numerischen Zeichenfolgen mit Leerzeichen. Wenn die Spalte nur numerische Zeichenfolgen enthält, auf die abschließende Leerzeichen folgen, geben Sie nicht Y an. Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.
VARCHAR_NO_TRAILING_BLANKS	Gibt an, ob sich mindestens ein abschließendes Leerzeichen in der VARCHAR-Spalte befindet.

Referenz für Microsoft SQL Server-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 50. Wrapperoptionen für Microsoft SQL Server

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.

Tabelle 50. Wrapperoptionen für Microsoft SQL Server (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 51. Serveroptionen für Microsoft SQL Server

Name	Beschreibung
CODEPAGE	Gibt die Codepage an, die dem codierten Zeichensatz der Clientkonfiguration für die Datenquelle entspricht. Auf UNIX- und Microsoft Windows-Systemen, die eine föderierte Nicht-Unicode-Datenbank verwenden, ist der Standardwert die Codepage der föderierten Datenbank. Auf UNIX-Systemen, die eine föderierte Unicode-Datenbank verwenden, ist der Standardwert 1208. Auf Windows-Systemen, die eine föderierte Unicode-Datenbank verwenden, ist der Standardwert 1202.
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.

Tabelle 51. Serveroptionen für Microsoft SQL Server (Forts.)

Name	Beschreibung
COMM_RATE	<p>Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.</p>
CPU_RATIO	<p>Gibt an, wie viel schneller oder langsamer die Systemeinheit der Datenquelle im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4.</p>
DBNAME	<p>Erforderlich. Gibt den Aliasnamen für die Datenbank an, auf die Sie zugreifen wollen. Bei dem Wert muss die Groß-/Kleinschreibung beachtet werden.</p>
DB2_MAXIMAL_PUSHDOWN	<p>Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet. Wenn mehrere Zugriffspläne dieses Kriterium erfüllen, wird der Plan mit dem geringsten Aufwand ausgewählt.</p>
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	<p>Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.</p>

Tabelle 51. Serveroptionen für Microsoft SQL Server (Forts.)

Name	Beschreibung
DB2_TWO_PHASE_COMMIT	<p>Gibt an, ob der Server mit föderierten Datenbanken über ein Protokoll für zweiphasige Commits oder über ein Protokoll für einphasige Commits eine Verbindung zur Datenquelle herstellt. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stellt der Server mit föderierten Datenbanken die Verbindung über ein Protokoll für einphasige Commits her. Y gibt an, dass der Server mit föderierten Datenbanken die Verbindung über das Protokoll für zweiphasige Commits herstellt.</p> <p>Wichtig: Wenn Sie diese Option auf Y setzen, müssen Sie auch <code>XA_OPEN_STRING_OPTION</code> angeben.</p>
DB2_UM_PLUGIN	<p>Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.</p>
DB2_UM_PLUGIN_LANG	<p>Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C. Der Standardwert ist Java.</p>
FOLD_ID	<p>Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>
FOLD_PW	<p>Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>

Tabelle 51. Serveroptionen für Microsoft SQL Server (Forts.)

Name	Beschreibung
IO_RATIO	<p>Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.</p>
NODE	<p>Erforderlich. Wenn der Server mit föderierten Datenbanken Microsoft Windows verwendet, ist der Wert für NODE der System-DSN-Name, den Sie für Microsoft SQL Server angeben. Wenn der Server mit föderierten Datenbanken UNIX oder Linux verwendet, wird der Wert für NODE in der Datei <code>odbc.ini</code> definiert. Bei dem Wert muss die Groß-/Kleinschreibung beachtet werden.</p>
OLD_NAME_GEN	<p>Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.</p>
PASSWORD	<p>Gibt an, ob Kennwörter an eine Datenquelle gesendet werden oder nicht. Der Standardwert ist Y.</p>

Tabelle 51. Serveroptionen für Microsoft SQL Server (Forts.)

Name	Beschreibung
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N. Der Standardwert ist Y. Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.
XA_OPEN_STRING_OPTIONS	Erforderlich, wenn DB2_TWO_PHASE_COMMIT auf Y gesetzt ist. Gibt die Ressourcenmanager-ID der Registrierdatenbank von Microsoft SQL Server an.

Benutzerzuordnungsoptionen

Tabelle 52. Benutzerzuordnungsoptionen für Microsoft SQL Server

Name	Beschreibung
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Spaltenoptionen

Tabelle 53. Spaltenoptionen für Microsoft SQL Server

Name	Beschreibung
NUMERIC_STRING	Gibt an, wie numerische Zeichenfolgen behandelt werden. Der Standardwert ist N . Wenn die Zeichenfolgespalte für Datenquellen nur numerische Zeichenfolgen und keine anderen Zeichen, einschließlich Leerzeichen, enthält, setzen Sie die Option NUMERIC_STRING auf Y . Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.

Referenz für ODBC-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 54. Wrapperoptionen für ODBC

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt. Wichtig: Wenn Sie diese Option für ein UNIX-System auf Y setzen, müssen Sie auch die Wrapperoption DB2_SOURCE_CLIENT_MODE festlegen.

Tabelle 54. Wrapperoptionen für ODBC (Forts.)

Name	Beschreibung
DB2_SOURCE_CLIENT_MODE	Gibt an, dass der Client für die Datenquelle ein 32-Bit-Client ist und dass die Datenbankinstanz auf dem Server mit föderierten Datenbanken eine 64-Bit-Instanz ist. Der einzige gültige Wert lautet 32-bit . Diese Option ist nur für UNIX-Systeme gültig. Wichtig: Wenn Sie diese Option setzen, müssen Sie auch die Wrapperoption DB2_FENCED auf Y setzen.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
MODULE	Erforderlich für Server mit föderierten Datenbanken, die auf einem UNIX-System ausgeführt werden. Gibt den vollständigen Pfad der Bibliothek an, die die ODBC-Treibermanagerimplementierung oder die SQL/CLI-Implementierung enthält. Für UNIX gibt es keinen Standardwert. Auf Microsoft Windows-Systemen lautet der Standardwert odbcc32.dll.

Serveroptionen

Tabelle 55. Serveroptionen für ODBC

Name	Beschreibung
CODEPAGE	Gibt die Codepage an, die dem codierten Zeichensatz der Clientkonfiguration für die Datenquelle entspricht. Auf UNIX- und Windows-Systemen, die eine föderierte Nicht-Unicode-Datenbank verwenden, ist der Standardwert die von der föderierten Datenbank verwendete Codepage. Auf UNIX-Systemen, die eine föderierte Unicode-Datenbank verwenden, ist der Standardwert 1208. Auf Windows-Systemen, die eine föderierte Unicode-Datenbank verwenden, ist der Standardwert 1202.

Tabelle 55. Serveroptionen für ODBC (Forts.)

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/ Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.
DATEFORMAT	Gibt das Datumsformat an, das von der Datenquelle verwendet wird. Verwenden Sie 'TT', 'MM' und 'JJ' bzw. 'JJJJ', um das Datumsformat anzugeben. Sie können ein Begrenzungszeichen angeben, wie z. B. ein Leerzeichen, einen Bindestrich oder ein Komma. Das Format 'JJJJ-MM-TT' gibt beispielsweise ein Datum wie 1958-10-01 an. Der Wert kann Nullwerte enthalten.
DBNAME	Gibt den Namen der Datenquellendatenbank an, auf die Sie zugreifen wollen.

Tabelle 55. Serveroptionen für ODBC (Forts.)

Name	Beschreibung
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 0. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
FOLD_ID	Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.

Tabelle 55. Serveroptionen für ODBC (Forts.)

Name	Beschreibung
FOLD_PW	Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.
IO_RATIO	Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.
NODE	Erforderlich. Gibt den Namen des Knotens oder den System-DSN-Namen an, der der ODBC-Datenquelle zugeordnet wird, wenn der Datenquellename definiert wird. Bei dem Wert muss die Groß-/Kleinschreibung beachtet werden.
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.

Tabelle 55. Serveroptionen für ODBC (Forts.)

Name	Beschreibung
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N . Der Standardwert ist Y . Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.
TIMEFORMAT	Gibt das Zeitformat an, das von der Datenquelle verwendet wird. Verwenden Sie 'hh12', 'hh24', 'mm', 'ss', 'AM' und 'A.M.', um das Zeitformat anzugeben. Das Format 'hh12:mm:ss AM' gibt z. B. eine Zeit wie 8:00:00 AM an. Der Wert kann Nullwerte enthalten.
TIMESTAMPFORMAT	Gibt das Zeitmarkenformat an, das von der Datenquelle verwendet wird. Gültige Werte weisen das Format auf, das in der Option DATEFORMAT und in der Option TIMEFORMAT verwendet wird. Geben Sie 'n' für eine Zehntelsekunde, 'nn' für eine Hundertstelsekunde, 'nnn' für Millisekunden usw. bis hin zu 'nnnnnn' für Mikrosekunden an. Das Format 'JJJJ-MM-TT-hh24:mm:ss.nnnnnn' gibt beispielsweise eine Zeitmarke wie 1994-01-01-24:00:00,000000 an. Der Wert kann Nullwerte enthalten.
VARCHAR_NO_TRAILING_BLANKS	Gibt an, ob die Datenquelle VARCHAR-Spalten mit mindestens einem abschließenden Leerzeichen enthält. Der Standardwert ist N . Hierbei enthalten die VARCHAR-Spalten mindestens ein abschließendes Leerzeichen.

Benutzerzuordnungsoptionen

Tabelle 56. Benutzerzuordnungsoptionen für ODBC

Name	Beschreibung
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Spaltenoptionen

Tabelle 57. Spaltenoptionen für ODBC

Name	Beschreibung
NUMERIC_STRING	Gibt an, ob die Spalte Zeichenfolgen mit numerischen Zeichen enthält, die Leerzeichen einschließen. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei enthält die Spalte keine numerischen Zeichenfolgen mit Leerzeichen. Wenn die Spalte nur numerische Zeichenfolgen enthält, auf die abschließende Leerzeichen folgen, geben Sie nicht Y an. Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.
VARCHAR_NO_TRAILING_BLANKS	Gibt an, ob sich mindestens ein abschließendes Leerzeichen in der VARCHAR-Spalte befindet.

Referenz für Oracle-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 58. Wrapperoptionen für Oracle

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.

Tabelle 58. Wrapperoptionen für Oracle (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 59. Serveroptionen für Oracle

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.

Tabelle 59. Serveroptionen für Oracle (Forts.)

Name	Beschreibung
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplans auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_TWO_PHASE_COMMIT	Gibt an, ob der Server mit föderierten Datenbanken über ein Protokoll für zweiphasige Commits oder über ein Protokoll für einphasige Commits eine Verbindung zur Datenquelle herstellt. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stellt der Server mit föderierten Datenbanken die Verbindung über ein Protokoll für einphasige Commits her. Y gibt an, dass der Server mit föderierten Datenbanken die Verbindung über das Protokoll für zweiphasige Commits herstellt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
FED_PROXY_USER	Gibt die Berechtigungs-ID an, die verwendet werden soll, um alle abgehenden gesicherten Verbindungen herzustellen, wenn die eingehende Verbindung nicht gesichert ist. Wichtig: Der Benutzer, dessen ID in dieser Option angegeben ist, muss eine Benutzerzuordnung haben, die sowohl REMOTE_AUTHID als auch REMOTE_PASSWORD angibt.

Tabelle 59. Serveroptionen für Oracle (Forts.)

Name	Beschreibung
FOLD_ID	<p>Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>
FOLD_PW	<p>Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.</p>
IO_RATIO	<p>Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23}. Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.</p>
NODE	<p>Erforderlich. Gibt den Namen des Knotens an, auf dem sich der Oracle-Datenbankserver befindet. Sie finden den Namen des Knotens in der Datei <code>tnsnames.ora</code>.</p>

Tabelle 59. Serveroptionen für Oracle (Forts.)

Name	Beschreibung
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.
PLAN_HINTS	Gibt an, ob Planhinweise aktiviert werden sollen oder nicht. Planhinweise sind Anweisungsfragmente, die zusätzliche Informationen bereitstellen, mit denen das Optimierungsprogramm für Datenquellen die Abfrageleistung verbessern kann. Das Optimierungsprogramm für Datenquellen verwendet Planhinweise, um zu entscheiden, ob ein Index verwendet werden soll oder nicht und welcher Index oder welche Tabellenjoinsequenz verwendet werden soll. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei werden Planhinweise nicht aktiviert. Y gibt an, dass Planhinweise an der Datenquelle aktiviert werden müssen, wenn die Datenquelle Planhinweise unterstützt.
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N . Der Standardwert ist Y . Hierbei ermöglicht der Server mit föderierten Datenbanken, dass die Datenquelle Operationen auswertet. N gibt an, dass der Server mit föderierten Datenbanken Spalten von der fernen Datenquelle abrufen.
VARCHAR_NO_TRAILING_BLANKS	Gibt für einen bestimmten Server an, ob VARCHAR-Spalten abschließende Leerzeichen enthalten. Wenn Sie diese Option auf eine einzelne Spalte anwenden wollen, verwenden Sie die Spaltenoption VARCHAR_NO_TRAILING_BLANKS.
XA_OPEN_STRING_OPTIONS	Gibt zusätzliche Informationen an, die an die Zeichenfolge angehängt werden sollen. Eine solche Information kann beispielsweise das Verzeichnis für Tracedateien sein.

Benutzerzuordnungsoptionen

Table 60. Benutzerzuordnungsoptionen für Oracle

Name	Beschreibung
FED_PROXY_USER	Gibt die Berechtigungs-ID an, die verwendet werden soll, um alle abgehenden gesicherten Verbindungen herzustellen, wenn die eingehende Verbindung nicht gesichert ist. Wichtig: Der Benutzer, dessen ID in dieser Option angegeben ist, muss eine Benutzerzuordnung haben, die sowohl REMOTE_AUTHID als auch REMOTE_PASSWORD angibt. Wenn Sie die Benutzerzuordnungsoption FED_PROXY_USER angeben, müssen Sie auch die Serveroption FED_PROXY_USER angeben.
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.
USE_TRUSTED_CONTEXT	Gibt an, ob die Benutzerzuordnung gesichert ist. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei ist die Benutzerzuordnung nicht gesichert und kann nur in nicht gesicherten abgehenden Verbindungen verwendet werden. Y gibt an, dass die Benutzerzuordnung gesichert ist und sowohl in gesicherten als auch in nicht gesicherten abgehenden Verbindungen verwendet werden kann.

Spaltenoptionen

Tabelle 61. Spaltenoptionen für Oracle

Name	Beschreibung
NUMERIC_STRING	Gibt an, ob die Spalte Zeichenfolgen mit numerischen Zeichen enthält, die Leerzeichen einschließen. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei enthält die Spalte keine numerischen Zeichenfolgen mit Leerzeichen. Wenn die Spalte nur numerische Zeichenfolgen enthält, auf die abschließende Leerzeichen folgen, geben Sie nicht Y an. Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.
VARCHAR_NO_TRAILING_BLANKS	Gibt an, ob die Spalte abschließende Leerzeichen enthält.

Referenz für Scriptoptionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 62. Wrapperoptionen für Scripts

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE und SOCKS. Der Standardwert ist NONE.

Tabelle 62. Wrapperoptionen für Scripts (Forts.)

Name	Beschreibung
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.

Serveroptionen

Tabelle 63. Serveroptionen für Scripts

Name	Beschreibung
DAEMON_PORT	Gibt die Nummer des Ports an, an dem der Scriptdämon für Script-Job-Anforderungen empfangsbereit ist. Der Standardwert ist 4099. Die Portnummer muss mit der Option DAEMON_PORT in der Dämon-konfigurationsdatei identisch sein. Wenn für den Scriptdämon ein Servicename konfiguriert ist, verwenden Sie einen TCP/IP-Servicenamen.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
NODE	Erforderlich. Gibt den DNS-Hostnamen oder die IP-Adresse des Systems an, auf dem der Scriptdämon ausgeführt wird. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an.
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.

Tabelle 63. Serveroptionen für Scripts (Forts.)

Name	Beschreibung
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Service-name.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE und SOCKS . Der Standardwert ist NONE .

Benutzerzuordnungsoptionen

Tabelle 64. Benutzerzuordnungsoptionen für Scripts

Name	Beschreibung
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an. Das Kennwort wird verschlüsselt, wenn es im Katalog für föderierte Datenbanken gespeichert wird.

Kurznamenoptionen

Tabelle 65. Kurznamenoptionen für Scripts

Name	Beschreibung
DATASOURCE	Erforderlich für den Rootkurznamen. Gibt den Namen des aufzurufenden Scripts an. Das Script, das Sie als Wert für diese Option angeben, muss auch in der Konfigurationsdatei für den Scriptdämon angegeben sein. Wichtig: Diese Option ist nur für den Rootkurznamen gültig.
NAMESPACES	Gibt die Namensbereiche an, die den Namensbereichspräfixen zugeordnet sind, die in den Optionen XPATH und TEMPLATE für jede Spalte verwendet werden. Verwenden Sie folgende Syntax: <code>NAMESPACES'präfix1="tatsächlicher_namensbereich1", präfix2="tatsächlicher_namensbereich2"'</code> Trennen Sie mehrere Namensbereiche jeweils durch ein Komma. Beispiel: <code>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</code>

Tabelle 65. Kurznamenoptionen für Scripts (Forts.)

Name	Beschreibung
STREAMING	Gibt an, ob das Quelldokument für die Verarbeitung in logische Fragmente unterteilt werden soll. Die Fragmente entsprechen dem Knoten, der mit dem XPath-Ausdruck des Kurznamens übereinstimmt. Anschließend parst und verarbeitet der Wrapper Quelldaten fragmentweise. Dieser Parsing-Typ minimiert die Speicherbelegung. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei werden die Dokumente nicht geparkt. Diese Option ist nur für den Rootkurznamen gültig. Setzen Sie die Optionen STREAMING und VALIDATE nicht gemeinsam auf Y.
TIMEOUT	Gibt an, wie viele Minuten maximal auf Ergebnisse vom Dämon gewartet wird. Der Standardwert ist 60. Diese Option ist nur beim Kurznamen für das Stammelement gültig.
VALIDATE	Gibt an, ob das Quelldokument vor dem Extrahieren von Daten überprüft wird, um sicherzustellen, dass es einem XML-Schema oder einer Dokumenttypdefinition (DTD) entspricht. Der Standardwert ist N. Hierbei wird keine Überprüfung vorgenommen. Bevor Sie den Wert auf Y setzen, muss sich die Schemadatei oder die DTD-Datei an der Speicherposition befinden, die vom Quelldokument angegeben wird. Diese Option ist nur für den Rootkurznamen gültig. Setzen Sie die Optionen STREAMING und VALIDATE nicht gemeinsam auf Y.
XPATH	Gibt den XPath-Ausdruck an, mit dem die XML-Elemente angegeben werden, die einzelne Tupel darstellen. Die Kurznamenoption XPATH für einen untergeordneten Kurznamen wird im Kontext des Pfads ausgewertet, der von der Kurznamenoption XPATH des zugehörigen übergeordneten Kurznamens angegeben wird. Dieser XPath-Ausdruck wird als Kontext zur Auswertung von Spaltenwerten verwendet, die durch die vorhandene Spaltenoption XPATH angegeben werden.

Spaltenoptionen

Tabelle 66. Spaltenoptionen für Scripts

Name	Beschreibung
DEFAULT	Gibt einen Standardwert für eine Scripteingabespalte an. Wenn die SQL-Abfrage keinen Wert bereitstellt, wird dieser Standardwert verwendet.

Tabelle 66. Spaltenoptionen für Scripts (Forts.)

Name	Beschreibung
FOREIGN_KEY	<p>Gibt an, dass dieser Kurzname ein untergeordneter Kurzname ist, und gibt den Namen des entsprechenden übergeordneten Kurznamens an. Ein Kurzname kann höchstens eine Spaltenoption FOREIGN_KEY haben. Bei dem Wert für die Option muss die Groß-/Kleinschreibung beachtet werden. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join eines übergeordneten und eines untergeordneten Kurznamens verwendet werden. Eine Anweisung CREATE NICKNAME, die die Option FOREIGN_KEY enthält, schlägt fehl, wenn der übergeordnete Kurzname einen abweichenden Schemanamen hat. Wenn der Kurzname, auf den in einer Klausel FOREIGN_KEY verwiesen wird, in der Anweisung CREATE NICKNAME nicht explizit in Kleinbuchstaben oder in einer Schreibweise mit gemischter Groß-/Kleinschreibung definiert wurde, müssen Sie diesen Kurznamen in Großbuchstaben angeben, wenn Sie in der Klausel FOREIGN_KEY auf ihn verweisen. Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden.</p>
INPUT_MODE	<p>Gibt den Eingabemodus für die Spalte an. Gültige Werte sind CONFIG und FILE_INPUT. CONFIG behandelt den Wert als Eingabemodus für eine Spalte. FILE_INPUT gibt eine Datei an, die den Wert speichert. Der Wrapper übergibt den angegebenen Wert an den Scriptdämon.</p>
POSITION	<p>Gibt einen ganzzahligen Wert für positionsgebundene Parameter an. Wenn der positionsgebundene Wert auf eine ganze Zahl gesetzt ist, muss sich diese Eingabe an dieser Position in der Befehlszeile befinden. Wenn diese Option gesetzt ist, wird der Schalter an der entsprechenden Position eingefügt, wenn die Abfrage ausgeführt wird. Wenn POSITION auf -1 gesetzt ist, wird die Option als letzte Befehlszeilenoption hinzugefügt. Ein ganzzahliger Wert für POSITION kann nicht zweimal in demselben Kurznamen verwendet werden. Diese Option gilt nur für Eingabespalten.</p>

Tabelle 66. Spaltenoptionen für Scripts (Forts.)

Name	Beschreibung
PRIMARY_KEY	Erforderlich für einen übergeordneten Kurznamen, der mindestens einen untergeordneten Kurznamen hat. Gibt an, dass dieser Kurzname ein übergeordneter Kurzname ist. Der Spaltendatentyp muss VARCHAR(16) lauten. Ein Kurzname kann nur eine einzige Spaltenoption PRIMARY_KEY haben. Yes ist der einzige gültige Wert. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join von übergeordneten und untergeordneten Kurznamen verwendet werden. Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden.
SWITCH	Gibt eine Markierung für das Script in der Befehlszeile an. Der Wert dieser Option geht dem Spaltenwert voraus, der von WSSCRIPT.ARGS oder vom Standardwert, falls verfügbar, bereitgestellt wird. Wenn Sie keinen Wert für diese Option angeben und ein Standardwert für die Spalte vorhanden ist, wird der Standardwert ohne Schalterinformationen hinzugefügt. Diese Option ist für Eingabespalten erforderlich.
SWITCH_ONLY	Ermöglicht die Verwendung von Schaltern ohne Befehlszeilenparameter. Gültige Werte sind Y und N . Setzen Sie diese Option auf Y . Beim Eingabewert Y wird nur der Schalter der Befehlszeile hinzugefügt. Beim Eingabewert N wird der Befehlszeile kein Wert hinzugefügt.
VALID_VALUES	Gibt eine Gruppe gültiger Werte für eine Spalte an. Verwenden Sie ein Semikolon, um mehrere Werte voneinander zu trennen.
XPATH	Gibt den XPath-Ausdruck in dem XML-Dokument an, das die Daten enthält, die dieser Spalte entsprechen. Der Wrapper wertet diesen XPath-Ausdruck aus, nachdem die Anweisung CREATE NICKNAME den XPath-Ausdruck aus der Kurznamenoption XPATH angewendet hat.

Referenz für Sybase-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 67. Wrapperoptionen für Sybase

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Unter Microsoft Windows lauten die gültigen Werte Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt. Unter UNIX lautet der Standardwert, der der einzige gültige Wert ist, Y , d. h. der Wrapper muss im abgeschirmten Modus ausgeführt werden.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 68. Serveroptionen für Sybase

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.

Tabelle 68. Serveroptionen für Sybase (Forts.)

Name	Beschreibung
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.
CONV_EMPTY_STRING	Gibt an, ob der Server mit föderierten Datenbanken eine leere Zeichenfolge während der Ausführung von Replikationstasks in ein Leerzeichen konvertiert. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei konvertiert der Server mit föderierten Datenbanken leere Zeichenfolgen nicht. Setzen Sie diese Option auf Y, wenn die Datenquelle eine Zeichenspalte aufweist, die keine Nullwerte enthalten kann und in der doch eine leere Zeichenfolge gespeichert ist.
DBNAME	Erforderlich. Gibt den Namen der Datenbank an, auf die Sie zugreifen wollen. Beziehen Sie den Namen der Datenbank vom Sybase-Server.
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.

Tabelle 68. Serveroptionen für Sybase (Forts.)

Name	Beschreibung
DB2_TWO_PHASE_COMMIT	Gibt an, ob der Server mit föderierten Datenbanken über ein Protokoll für zweiphasige Commits oder über ein Protokoll für einphasige Commits eine Verbindung zur Datenquelle herstellt. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stellt der Server mit föderierten Datenbanken die Verbindung über ein Protokoll für einphasige Commits her. Y gibt an, dass der Server mit föderierten Datenbanken die Verbindung über das Protokoll für zweiphasige Commits herstellt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
FOLD_ID	Gibt die Groß-/Kleinschreibung für die Benutzer-ID an, die an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet die Benutzer-ID in Großbuchstaben. Wenn die Benutzer-ID in Großbuchstaben fehlschlägt, sendet der Server die Benutzer-ID in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.
FOLD_PW	Gibt die Groß-/Kleinschreibung für das Kennwort an, das an die Datenquelle gesendet wird. Es gibt keinen Standardwert. Der Server mit föderierten Datenbanken sendet das Kennwort in Großbuchstaben. Wenn das Kennwort in Großbuchstaben fehlschlägt, sendet der Server das Kennwort in Kleinbuchstaben. Gültige Werte sind U (Uppercase - Großschreibung), L (Lowercase - Kleinschreibung) und N (null). Vermeiden Sie es, die Einstellung N zu verwenden, da sie die Leistung beeinträchtigen kann.

Tabelle 68. Serveroptionen für Sybase (Forts.)

Name	Beschreibung
IFILE	Gibt den Pfad und den Namen der Sybase-Schnittstellendatei an, die anstelle der Standardschnittstellendatei verwendet werden soll. Der Sybase-Wrapper sucht an den folgenden Orten in der angegebenen Reihenfolge nach der Schnittstellendatei: Unter Microsoft Windows in der Serveroption IFILE, anschließend im Verzeichnis %DB2PATH%\interfaces und schließlich im Verzeichnis %SYBASE%\ini\sql.ini. Unter UNIX in der Serveroption IFILE, anschließend im Verzeichnis sqllib/interfaces und schließlich im Verzeichnis \$SYBASE/interfaces.
IO_RATIO	Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.
LOGIN_TIMEOUT	Gibt an, wie viele Sekunden der Server mit föderierten Datenbanken maximal wartet, bis er eine Anmeldeanforderung verlässt. Der Standardwert ist 0. Hierbei ist die Wartezeit des Servers mit föderierten Datenbanken uneingeschränkt.
NODE	Erforderlich. Gibt den Namen des Knotens an, auf dem sich der Sybase-Server befindet. Der Knotenname befindet sich in der Sybase-Schnittstellendatei.

Tabelle 68. Serveroptionen für Sybase (Forts.)

Name	Beschreibung
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.
PACKET_SIZE	Gibt die Paketgröße in Byte an, die die Clientbibliothek verwendet, um TDS-Pakete (Tabular Data Stream - tabellarischer Datenstrom) zu senden. Wenn der Sybase-Wrapper große Text- und Bilddatenmengen senden oder empfangen muss, müssen Sie den Wert für <code>PACKET_SIZE</code> erhöhen.
PLAN_HINTS	Gibt an, ob Planhinweise aktiviert werden sollen oder nicht. Planhinweise sind Anweisungsfragmente, die zusätzliche Informationen bereitstellen, mit denen das Optimierungsprogramm für Datenquellen die Abfrageleistung verbessern kann. Das Optimierungsprogramm für Datenquellen verwendet Planhinweise, um zu entscheiden, ob ein Index verwendet werden soll oder nicht und welcher Index oder welche Tabellenjoinsequenz verwendet werden soll. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei werden Planhinweise nicht aktiviert. Y gibt an, dass Planhinweise an der Datenquelle aktiviert werden müssen, wenn die Datenquelle Planhinweise unterstützt.
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N . Der Standardwert ist Y . Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur <code>SELECT</code> -Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. <code>WHERE=</code> , Spalten- und Skalarfunktionen, wie z. B. <code>MAX</code> und <code>MIN</code> , Sortierfunktionen, wie z. B. <code>ORDER BY</code> oder <code>GROUP BY</code> , und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.

Tabelle 68. Serveroptionen für Sybase (Forts.)

Name	Beschreibung
TIMEOUT	Gibt an, wie viele Sekunden der Server mit föderierten Datenbanken maximal auf die Reaktion des fernen Servers auf einen Befehl wartet. Der Standardwert ist 0. Hiermit wird eine uneingeschränkte Wartezeit angegeben.
XA_OPEN_STRING_OPTIONS	Gibt Öffnungszeichenfolgen für die Sybase DTM XA-Schnittstelle an. Diese Zeichenfolgen können zusätzlich zum LRM-Namen, Benutzernamen und Kennwort verwendet werden.

Benutzerzuordnungsoptionen

Tabelle 69. Benutzerzuordnungsoptionen für Sybase

Option	Beschreibung
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Spaltenoptionen

Tabelle 70. Spaltenoptionen für Sybase

Option	Beschreibung
NUMERIC_STRING	Gibt an, wie numerische Zeichenfolgen behandelt werden. Der Standardwert ist N. Wenn die Zeichenfolgespalte für Datenquellen nur numerische Zeichenfolgen und keine anderen Zeichen, einschließlich Leerzeichen, enthält, setzen Sie die Option NUMERIC_STRING auf Y. Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.

Referenz für Teradata-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 71. Wrapperoptionen für Teradata

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Unter Microsoft Windows lauten die gültigen Werte Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt. Unter UNIX lautet der Standardwert Y , d. h. der Wrapper muss im abgeschirmten Modus ausgeführt werden.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .

Serveroptionen

Tabelle 72. Serveroptionen für Teradata

Name	Beschreibung
COLLATING_SEQUENCE	Gibt an, ob die Datenquelle die gleiche Standardsortierfolge verwendet wie die föderierte Datenbank. Gültige Werte sind Y , N und I . I gibt an, dass die Groß-/Kleinschreibung nicht beachtet wird. Der Standardwert ist Y . Hierbei muss die für den Server mit föderierten Datenbanken angegebene Sortierfolge mit der Sortierfolge in der fernen Datenquelle übereinstimmen.

Tabelle 72. Serveroptionen für Teradata (Forts.)

Name	Beschreibung
COMM_RATE	Gibt die Kommunikationsgeschwindigkeit zwischen dem Server mit föderierten Datenbanken und dem Datenquellenserver in Megabyte pro Sekunde an. Gültige Werte sind ganze Zahlen, die größer als 0 und kleiner als 2.147.483.648 sind. Der Standardwert ist 2.
CPU_RATIO	Gibt an, um wie viel schneller oder langsamer die Datenquellensystemeinheit im Vergleich zur Systemeinheit des Servers mit föderierten Datenbanken ist. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass die Systemeinheit des Servers mit föderierten Datenbanken und die Systemeinheit des Datenquellenservers die gleiche Geschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Systemeinheit des Servers mit föderierten Datenbanken 50 % langsamer ist als die Datenquellensystemeinheit. Die Einstellung 2 zeigt an, dass die föderierte Systemeinheit doppelt so schnell ist wie die Datenquellensystemeinheit.
DB2_MAXIMAL_PUSHDOWN	Gibt das primäre Kriterium an, das das Abfrageoptimierungsprogramm bei der Auswahl eines Zugriffsplans verwendet. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wählt das Abfrageoptimierungsprogramm den Plan mit dem geringsten geschätzten Aufwand aus. Y gibt an, dass das Abfrageoptimierungsprogramm den Zugriffsplan auswählt, der die meisten Abfrageoperationen im Pushdown-Modus an die Datenquelle sendet.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.

Tabelle 72. Serveroptionen für Teradata (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
IO_RATIO	Gibt an, wie viel schneller oder langsamer das Ein-/Ausgabesystem der Datenquelle im Vergleich zum Ein-/Ausgabesystem des Servers mit föderierten Datenbanken ausgeführt wird. Gültige Werte sind größer als 0 und kleiner als 1×10^{23} . Der Standardwert ist 1,0. Werte können in jeder gültigen Doppelnotation angegeben werden, wie z. B. 123E10, 123 oder 1,21E4. Die Einstellung 1 zeigt an, dass der Server mit föderierten Datenbanken und der Datenquellenserver die gleiche Ein-/Ausgabegeschwindigkeit haben. Das Verhältnis ist 1:1. Die Einstellung 0,5 gibt an, dass die Geschwindigkeit des Servers mit föderierten Datenbanken halb so groß ist wie die Datenquellengeschwindigkeit. Die Einstellung 2 zeigt an, dass die Geschwindigkeit des Systems mit föderierten Datenbanken doppelt so hoch ist wie die Datenquellengeschwindigkeit.
NODE	Erforderlich. Gibt den Aliasnamen oder die IP-Adresse des Teradata-Servers an.
OLD_NAME_GEN	Gibt an, wie die Spaltennamen und Indexnamen, die sich in der Datenquelle befinden, in Spaltenkurznamen und lokale Indexnamen für den Server mit föderierten Datenbanken konvertiert werden. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei stimmen die generierten Namen mit den Namen in der Datenquelle genau überein. Y gibt an, dass die generierten Namen mit den Namen identisch sind, die in IBM WebSphere Federation Server Version 9 und früher erstellt wurden. Daher ist die Übereinstimmung zwischen den Namen und den Datenquellennamen möglicherweise nicht groß.

Tabelle 72. Serveroptionen für Teradata (Forts.)

Name	Beschreibung
PUSHDOWN	Gibt an, ob der Server mit föderierten Datenbanken es zulässt, dass die Datenquelle Operationen auswertet. Gültige Werte sind Y und N. Der Standardwert ist Y. Hierbei wertet die Datenquelle Operationen aus. N gibt an, dass der Server mit föderierten Datenbanken SQL-Anweisungen sendet, die nur SELECT-Anweisungen mit Spaltennamen enthalten. Vergleichswerte, wie z. B. WHERE=, Spalten- und Skalarfunktionen, wie z. B. MAX und MIN, Sortierfunktionen, wie z. B. ORDER BY oder GROUP BY, und Joins sind in keiner SQL-Anweisung eingeschlossen, die der Server mit föderierten Datenbanken an die Datenquelle sendet.

Benutzerzuordnungsoptionen

Tabelle 73. Benutzerzuordnungsoptionen für Teradata

Name	Beschreibung
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.

Spaltenoptionen

Tabelle 74. Spaltenoptionen für Teradata

Name	Beschreibung
NUMERIC_STRING	Gibt an, wie numerische Zeichenfolgen behandelt werden. Der Standardwert ist N . Wenn die Zeichenfolgespalte für Datenquellen nur numerische Zeichenfolgen und keine anderen Zeichen, einschließlich Leerzeichen, enthält, setzen Sie die Option NUMERIC_STRING auf Y . Wenn NUMERIC_STRING für eine Spalte auf Y gesetzt ist, erkennt das Abfrageoptimierungsprogramm, dass die Spalte keine Leerzeichen enthält, die die Sortierung der Daten in der Spalte beeinträchtigen könnten. Verwenden Sie diese Option, wenn die Sortierfolge einer Datenquelle sich von der Sortierfolge unterscheidet, die der Server mit föderierten Datenbanken verwendet. Spalten, die diese Option verwenden, werden nicht aufgrund einer abweichenden Sortierfolge von der fernen Auswertung ausgeschlossen.

Referenz für Optionen von Dateien mit Tabellenstruktur

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Wrapperoptionen

Tabelle 75. Wrapperoptionen für Dateien mit Tabellenstruktur

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt.

Serveroptionen

Tabelle 76. Serveroptionen für Dateien mit Tabellenstruktur

Name	Beschreibung
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.

Kurznamenoptionen

Tabelle 77. Kurznamenoptionen für Dateien mit Tabellenstruktur

Name	Beschreibung
COLUMN_DELIMITER	Gibt ein einzelnes Zeichen an, das als Begrenzer verwendet werden soll, um Spalten in der Datei mit Tabellenstruktur voneinander zu trennen. Der Standardwert ist ein Komma (.). Ein einfaches Anführungszeichen kann nicht als Begrenzer verwendet werden. Der Spaltenbegrenzer muss in der gesamten Datei konsistent sein. Ein Nullwert wird mit zwei direkt aufeinander folgenden Begrenzern dargestellt oder mit einem Begrenzer, auf den ein Zeilenendezeichen folgt, wenn das Nullfeld das letzte Feld in der Zeile ist. Der Spaltenbegrenzer kann nicht als gültiger Dateninhalt für eine Spalte vorhanden sein.
CODEPAGE	Gibt die Codepage der Datei an der Datenquelle an. Diese Option ist nur für föderierte Datenbanken gültig, die Unicode verwenden. Die Quelldaten werden aus der angegebenen Codepage in Unicode konvertiert.
FILE_PATH	Gibt den vollständig qualifizierten Pfad zur Datei mit Tabellenstruktur an. Schließen Sie den Dateinamen in einfache Anführungszeichen ein. Die Datendatei muss eine Standarddatei oder eine symbolische Verbindung sein, und nicht etwa eine Pipe oder ein anderer vom Standard abweichender Dateityp. Wichtig: Wenn Sie die Option FILE_PATH angeben, geben Sie keine DOCUMENT-Spalte an.

Tabelle 77. Kurznamenoptionen für Dateien mit Tabellenstruktur (Forts.)

Name	Beschreibung
KEY_COLUMN	Gibt den Namen der Spalte an, nach der die Datei sortiert ist. Eine Spalte, die die Spaltenoption DOCUMENT hat, kann keine Schlüsselspalte sein. Nur einspaltige Schlüssel werden unterstützt. Der Wert muss der Name einer Spalte sein, die in der Anweisung CREATE NICKNAME definiert ist. Die Spalte muss aufsteigend sortiert sein. Die Schlüsselspalte muss als Spalte gekennzeichnet werden, die keine Nullwerte enthalten kann. Hierfür wird der zugehörigen Definition in der Kurznamenanweisung die Option NOT NULL hinzugefügt. Bei diesem Wert muss die Groß-/Kleinschreibung beachtet werden.
SORTED	Gibt an, ob die Datei an der Datenquelle aufsteigend sortiert ist oder nicht. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei ist die Datei an der Datenquelle nicht aufsteigend sortiert. Sortierte Datenquellen müssen gemäß der Sortierfolge für die aktuelle Ländereinstellung aufsteigend sortiert sein. Diese wird durch die Einstellungen in der Kategorie für Unterstützung in der Landessprache LC_COLLATE definiert. Wenn Sie angeben, dass die Datenquelle sortiert ist, müssen Sie die Option VALIDATE_DATA_FILE auf Y setzen.
VALIDATE_DATA_FILE	Für sortierte Dateien gibt diese Option an, ob der Wrapper überprüfen soll, ob die Schlüsselspalte aufsteigend sortiert ist und ob leere Schlüssel vorhanden sind. Diese Überprüfung wird nur zu Beginn einmal durchgeführt, nämlich wenn der Kurzname erstellt wird. Der Standardwert ist N. Hierbei wird die Sortierreihenfolge nicht überprüft. Diese Option ist nur gültig, wenn die Option SORTED auf Y gesetzt und die Option DOCUMENT nicht angegeben ist.

Spaltenoptionen

Tabelle 78. Spaltenoptionen für Dateien mit Tabellenstruktur

Option	Beschreibung
DOCUMENT	Ermöglicht Ihnen, den Dateipfad während der Ausführung der Abfrage anzugeben anstatt beim Erstellen des Kurznamens. Der einzige gültige Wert lautet FILE. Mit der Option DOCUMENT kann jeweils nur eine Spalte jedes Kurznamens angegeben werden. Die Spalte, die der Option DOCUMENT zugeordnet ist, muss den Datentyp VARCHAR oder CHAR aufweisen. Wenn Sie anstelle der Kurznamenoption FILE_PATH die Kurznamenspaltenoption DOCUMENT verwenden, setzt dies voraus, dass die diesem Kurznamen entsprechende Datei während der Ausführung der Abfrage angegeben wird. Wenn die Option DOCUMENT den Wert FILE aufweist, ist der Wert, der während der Abfrageausführung angegeben wird, der vollständige Pfad der Datei, deren Schema mit der Kurznamendefinition für diesen Kurznamen übereinstimmt.

Referenz für Web-Service-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 79. Wrapperoptionen für Web-Services

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird der Wrapper im gesicherten Modus ausgeführt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.

Tabelle 79. Wrapperoptionen für Web-Services (Forts.)

Name	Beschreibung
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE , HTTP und SOCKS . Der Standardwert ist NONE .
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Servicenamen.
SSL_KEYSTORE_FILE	Gibt die Zertifikatspeicherdatei für Kommunikationsvorgänge an, die SSL oder TLS verwenden. Ein gültiger Wert ist ein vollständig qualifizierter Pfadname, auf den der Agent für föderierte Datenbanken oder ein Prozess im abgeschirmten Modus zugreifen kann. Der Standardwert ist <i>installationspfad/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Gibt das Kennwort an, das für den Zugriff auf die Datei in der Option <code>SSL_KEYSTORE_FILE</code> verwendet werden soll. Gültige Werte sind ein Kennwort, das beim Speichern im Katalog für föderierte Datenbanken verschlüsselt wird, und <i>file:dirname</i> . Dabei ist <i>dirname</i> der vollständig qualifizierte Pfad zu einer Stashdatei.
SSL_VERIFY_SERVER_CERTIFICATE	Gibt an, ob das Serverzertifikat während der SSL-Authentifizierung überprüft wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird das Zertifikat nicht überprüft.

Serveroptionen

Table 80. Serveroptionen für Web-Services

Name	Beschreibung
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
DB2_UM_PLUGIN	Gibt die Implementierung des Plug-ins für die Benutzerzuordnung an. Gibt für ein Java-Plug-in eine Zeichenfolge (String) für den Klassennamen an, der der Repositoryklasse für Benutzerzuordnungen entspricht. Bei dieser Zeichenfolge muss die Groß-/Kleinschreibung beachtet werden. Beispiel: "UserMappingRepositoryLDAP". Gibt für ein C-Plug-in einen beliebigen, gültigen C-Bibliotheksnamen an.
DB2_UM_PLUGIN_LANG	Gibt die Sprache des Plug-ins für die Benutzerzuordnung an. Gültige Werte sind Java und C . Der Standardwert ist Java .
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an.
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Service-name.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE , HTTP und SOCKS . Der Standardwert ist NONE .
SSL_CLIENT_CERTIFICATE_LABEL	Gibt das Clientzertifikat an, das während der SSL-Authentifizierung gesendet werden soll. Wenn Sie keinen Wert angeben, wird die aktuelle Berechtigungs-ID für die föderierte Datenbank verwendet, um das Zertifikat zu suchen.

Tabelle 80. Serveroptionen für Web-Services (Forts.)

Name	Beschreibung
SSL_KEYSTORE_FILE	Gibt die Zertifikatspeicherdatei für Kommunikationsvorgänge an, die SSL oder TLS verwenden. Ein gültiger Wert ist ein vollständig qualifizierter Pfadname, auf den der Agent für föderierte Datenbanken oder ein Prozess im abgeschirmten Modus zugreifen kann. Der Standardwert ist <i>installationspfad/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Gibt das Kennwort an, das für den Zugriff auf die Datei in der Option SSL_KEYSTORE_FILE verwendet werden soll. Gültige Werte sind ein Kennwort, das beim Speichern im Katalog für föderierte Datenbanken verschlüsselt wird, und <i>file:dateiname</i> . Dabei ist <i>dateiname</i> der vollständig qualifizierte Pfad zu einer Stashdatei.
SSL_VERIFY_SERVER_CERTIFICATE	Gibt an, ob das Serverzertifikat während der SSL-Authentifizierung überprüft werden soll. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird das Zertifikat nicht überprüft.

Benutzerzuordnungsoptionen

Tabelle 81. Benutzerzuordnungsoptionen für Web-Services

Name	Beschreibung
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an. Das Kennwort wird verschlüsselt, wenn es im Katalog für föderierte Datenbanken gespeichert wird.
REMOTE_AUTHID	Gibt die ferne Benutzer-ID an, der die lokale Benutzer-ID zugeordnet ist. Wenn Sie diese Option nicht angeben, wird die ID verwendet, mit der die Verbindung zur föderierten Datenbank hergestellt wird.
REMOTE_PASSWORD	Gibt das ferne Kennwort für die ferne Benutzer-ID an. Wenn Sie diese Option nicht angeben, wird das Kennwort verwendet, mit dem die Verbindung zur föderierten Datenbank hergestellt wird.
SSL_CLIENT_CERTIFICATE_LABEL	Gibt das Clientzertifikat an, das während der SSL-Authentifizierung gesendet werden soll. Wenn Sie keinen Wert angeben, wird die aktuelle Berechtigungs-ID für die föderierte Datenbank verwendet, um das Zertifikat zu suchen.

Kurznamenoptionen

Tabelle 82. Kurznamenoptionen für Web-Services

Name	Beschreibung
NAMESPACES	<p>Gibt die Namensbereiche an, die den Namensbereichspräfixen zugeordnet sind, die in den Optionen XPATH und TEMPLATE für jede Spalte verwendet werden. Verwenden Sie folgende Syntax:</p> <pre>NAMESPACES'präfix1="tatsächlicher_namensbereich1", präfix2="tatsächlicher_namensbereich2"'</pre> <p>Trennen Sie mehrere Namensbereiche jeweils durch ein Komma. Beispiel:</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
SOAPACTION	<p>Erforderlich für den Rootkurznamen. Gibt das URI-Attribut SOAPACTION aus dem WSDL-Format an (Web Services Description Language). Die URL kann eine IPv6-Adresse mit Doppelpunktrennung enthalten, wenn diese in eckige Klammern eingeschlossen ist. Beispiel: <code>http://[1080:0:0:0:8:800:200C:417A]</code></p> <p>Anmerkung: Diese Option ist für Nicht-Root-Kurznamen nicht gültig.</p>
STREAMING	<p>Gibt an, ob das Quelldokument für die Verarbeitung in logische Fragmente unterteilt werden soll. Die Fragmente entsprechen dem Knoten, der mit dem XPath-Ausdruck des Kurznamens übereinstimmt. Anschließend parst und verarbeitet der Wrapper Quelledaten fragmentweise. Dieser Parsing-Typ minimiert die Speicherbelegung. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei werden die Dokumente nicht geparkt. Diese Option ist nur für den Rootkurznamen gültig.</p>
TEMPLATE	<p>Gibt das Kurznamenschablonenfragment an, das zum Erstellen einer SOAP-Anforderung verwendet werden soll. Das Fragment muss der angegebenen Schablونسyntax entsprechen. Diese Option ist nur für den Rootkurznamen gültig.</p>
URL	<p>Erforderlich für den Rootkurznamen. Gibt die URL für den Web-Service-Endpunkt an. Unterstützte Protokolle sind HTTP und HTTPS. Die URL kann eine IPv6-Adresse mit Doppelpunktrennung enthalten, wenn diese in eckige Klammern eingeschlossen ist. Beispiel: <code>http://[1080:0:0:0:8:800:200C:417A]</code></p>

Tabelle 82. Kurznamenoptionen für Web-Services (Forts.)

Name	Beschreibung
XML_CODESET	Gibt die Codierung an, die zum Senden und Empfangen von XML-Daten verwendet werden soll. Diese Option überschreibt die interne Codierung.
XPATH	Erforderlich. Gibt den XPath-Ausdruck an, mit dem die SOAP-Antwortelemente angegeben werden, die einzelne Tupel darstellen. Der XPath-Ausdruck wird als Kontext zur Auswertung von Spaltenwerten verwendet, die durch die Spaltenoptionen für Kurznamen XPATH angegeben werden.

Spaltenoptionen

Tabelle 83. Spaltenoptionen für Web-Services

Name	Beschreibung
ESCAPE_INPUT	Gibt an, ob XML-Eingabewerte durch XML-Sonderzeichen ersetzt werden oder nicht. Verwenden Sie diese Option, um XML-Fragmente als Eingabe einzuschließen, wie z. B. XML-Fragmente mit sich wiederholenden Elementen. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei werden XML-Eingabewerte beibehalten. Der Spaltendatentyp muss VARCHAR oder CHAR lauten. Wenn ESCAPE_INPUT auf Y gesetzt ist, müssen Sie auch die Spaltenoption TEMPLATE angeben.

Tabelle 83. Spaltenoptionen für Web-Services (Forts.)

Name	Beschreibung
FOREIGN_KEY	<p>Gibt an, dass dieser Kurzname ein untergeordneter Kurzname ist, und gibt den Namen des entsprechenden übergeordneten Kurznamens an. Ein Kurzname kann höchstens eine Spaltenoption FOREIGN_KEY haben. Bei dem Wert für die Option muss die Groß-/Kleinschreibung beachtet werden. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join von übergeordneten Kurznamen und einem untergeordneten Kurznamen verwendet werden. Eine Anweisung CREATE NICKNAME, die die Option FOREIGN_KEY enthält, schlägt fehl, wenn der übergeordnete Kurzname einen abweichenden Schemanamen hat. Wenn der Kurzname, auf den in einer Klausel FOREIGN_KEY verwiesen wird, in der Anweisung CREATE NICKNAME nicht explizit in Kleinbuchstaben oder in einer Schreibweise mit gemischter Groß-/Kleinschreibung definiert wurde, müssen Sie diesen Kurznamen in Großbuchstaben angeben, wenn Sie in der Klausel FOREIGN_KEY auf ihn verweisen.</p> <p>Anmerkung:</p> <ul style="list-style-type: none"> • Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden. • Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.
PRIMARY_KEY	<p>Erforderlich für einen übergeordneten Kurznamen, der mindestens einen untergeordneten Kurznamen hat. Gibt an, dass dieser Kurzname ein übergeordneter Kurzname ist. Der Spaltentyp muss VARCHAR(16) lauten. Ein Kurzname kann nur eine einzige Spaltenoption PRIMARY_KEY haben. Yes ist der einzige gültige Wert. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join von übergeordneten und untergeordneten Kurznamen verwendet werden.</p> <p>Anmerkung:</p> <ul style="list-style-type: none"> • Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden. • Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.

Tabelle 83. Spaltenoptionen für Web-Services (Forts.)

Name	Beschreibung
SOAPACTIONCOLUMN	<p>Gibt die Spalte an, die dynamisch die SOAP-Aktion für den Web-Services-Endpunkt angibt, wenn Sie eine Abfrage ausführen. Diese Option ist nur für den Rootkurznamen gültig. Wenn der Hostname eine IPv6-Adresse (Doppelpunktrennung) ist, schließen Sie den Hostnamen in eckige Klammern ein. Beispiel: <code>'http://[1080:0:0:0:8:800:200C:417A]:99/soap'</code></p> <p>Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden.</p>
TEMPLATE	<p>Gibt das Spaltenschablonenfragment an, mit dem das XML-Eingabedokument erstellt werden soll. Das Fragment muss der angegebenen Schablonensyntax entsprechen.</p> <p>Anmerkung: Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.</p>
URLCOLUMN	<p>Gibt die Spalte an, die dynamisch die SOAP-Aktion für den Web-Services-Endpunkt angibt, wenn Sie eine Abfrage ausführen. Diese Option ist nur für den Rootkurznamen gültig. Wenn der Hostname eine IPv6-Adresse (Doppelpunktrennung) ist, schließen Sie den Hostnamen in eckige Klammern ein. Beispiel: <code>'http://[1080:0:0:0:8:800:200C:417A]:99/soap'</code></p> <p>Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden.</p>
XPATH	<p>Gibt den XPath-Ausdruck in dem XML-Dokument an, das die Daten enthält, die dieser Spalte entsprechen. Der Wrapper wertet diesen XPath-Ausdruck aus, nachdem die Anweisung CREATE NICKNAME den XPath-Ausdruck aus der Kurznamenoption XPATH angewendet hat.</p> <p>Anmerkung: Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.</p>

Referenz für XML-Optionen

Setzen und modifizieren Sie die Optionen für Wrapper, Server, Benutzerzuordnung, Kurzname und Spalte, um zu konfigurieren, wie der Server mit föderierten Datenbanken und die zugehörigen Benutzer mit einer Datenquelle interagieren.

Wrapperoptionen

In den folgenden Tabellen werden die Optionen, die für diese Datenquelle gelten, und die erforderlichen Optionen aufgelistet, die Sie angeben müssen.

Tabelle 84. Wrapperoptionen für XML

Name	Beschreibung
DB2_FENCED	Erforderlich. Gibt an, ob der Wrapper im abgeschirmten Modus oder im gesicherten Modus ausgeführt wird. Gültige Werte sind Y und N . Der Standardwert ist N . Hierbei wird der Wrapper im gesicherten Modus ausgeführt.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE , HTTP und SOCKS . Der Standardwert ist NONE .
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Service-name.
SSL_KEYSTORE_FILE	Gibt die Zertifikatspeicherdatei für Kommunikationsvorgänge an, die SSL oder TLS verwenden. Ein gültiger Wert ist ein vollständig qualifizierter Pfadname, auf den der Agent für föderierte Datenbanken oder ein Prozess im abgeschirmten Modus zugreifen kann.
SSL_KEYSTORE_PASSWORD	Gibt das Kennwort an, das für den Zugriff auf die Datei in der Option <code>SSL_KEYSTORE_FILE</code> verwendet werden soll. Gültige Werte sind ein Kennwort, das beim Speichern im Katalog für föderierte Datenbanken verschlüsselt wird, und <code>file:pathname</code> . Dabei ist <code>pathname</code> der vollständig qualifizierte Pfad zu einer Stashdatei.

Tabelle 84. Wrapperoptionen für XML (Forts.)

Name	Beschreibung
SSL_VERIFY_SERVER_CERTIFICATE	Gibt an, ob das Serverzertifikat während der SSL-Authentifizierung überprüft wird. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei wird das Zertifikat nicht überprüft.

Serveroptionen

Tabelle 85. Serveroptionen für XML

Name	Beschreibung
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Gibt die maximale Anzahl gleichzeitiger asynchroner Anforderungen von einer Abfrage an. Die gültigen Werte liegen im Bereich -1 bis 64000. Der Standardwert ist 1. -1 gibt an, dass das Optimierungsprogramm für föderierte Abfragen die Anzahl der Anforderungen ermittelt. 0 gibt an, dass die Datenquelle nicht über die Ressourcen für zusätzliche asynchrone Anforderungen verfügt.
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an.
PROXY_SERVER_NAME	Gibt den Namen oder die IP-Adresse des Proxy-Servers an. Gültige IP-Adressen haben das Format IPv4 (Trennung durch Punkt) oder IPv6 (Trennung durch Doppelpunkt). Verwenden Sie das Format IPv6 nur dann, wenn IPv6 konfiguriert ist.
PROXY_SERVER_PORT	Gibt den Port oder den Servicenamen für den Proxy-Service auf dem Proxy-Server an. Gültige Werte sind eine dezimale Portnummer von 1 bis 32760 oder ein Servicenamen.
PROXY_TYPE	Gibt den Proxytyp an, der für den Zugriff auf das Internet verwendet werden soll, wenn sich der Server mit föderierten Datenbanken hinter einer Firewall befindet. Gültige Werte sind NONE, HTTP und SOCKS. Der Standardwert ist NONE.
SOCKET_TIMEOUT	Gibt, wie viele Minuten der Server mit föderierten Datenbanken maximal auf Ergebnisse vom Proxy-Server wartet. Gültige Werte sind alle Zahlen größer-gleich 0. Der Standardwert ist 0. Hierbei ist die Wartezeit des Servers uneingeschränkt.
SSL_CLIENT_CERTIFICATE_LABEL	Gibt das Clientzertifikat an, das während der SSL-Authentifizierung gesendet werden soll. Wenn Sie keinen Wert angeben, wird die aktuelle Berechtigungs-ID für die föderierte Datenbank verwendet, um das Zertifikat zu suchen.

Tabelle 85. Serveroptionen für XML (Forts.)

Name	Beschreibung
SSL_KEYSTORE_FILE	Gibt die Zertifikatspeicherdatei für Kommunikationsvorgänge an, die SSL oder TLS verwenden. Ein gültiger Wert ist ein vollständig qualifizierter Pfadname, auf den der Agent für föderierte Datenbanken oder ein Prozess im abgeschirmten Modus zugreifen kann.
SSL_KEYSTORE_PASSWORD	Gibt das Kennwort an, das für den Zugriff auf die Datei in der Option SSL_KEYSTORE_FILE verwendet werden soll. Gültige Werte sind ein Kennwort, das beim Speichern im Katalog für föderierte Datenbanken verschlüsselt wird, und <code>file:dateiname</code> . Dabei ist <code>dateiname</code> der vollständig qualifizierte Pfad zu einer Stashdatei.
SSL_VERIFY_SERVER_CERTIFICATE	Gibt an, ob das Serverzertifikat während der SSL-Authentifizierung überprüft werden soll. Der Standardwert ist N. Hierbei wird das Zertifikat nicht überprüft.

Benutzerzuordnungsoptionen

Tabelle 86. Benutzerzuordnungsoptionen für XML

Name	Beschreibung
PROXY_AUTHID	Gibt den Benutzernamen für die Proxy-Server-Authentifizierung an.
PROXY_PASSWORD	Gibt das Kennwort für die Proxy-Server-Authentifizierung an.
SSL_CLIENT_CERTIFICATE_LABEL	Gibt das Clientzertifikat an, das während der SSL-Authentifizierung gesendet werden soll. Wenn Sie keinen Wert angeben, wird die aktuelle Berechtigungs-ID für die föderierte Datenbank verwendet, um das Zertifikat zu suchen.

Kurznamenoptionen

Tabelle 87. Kurznamenoptionen für XML

Name	Beschreibung
DIRECTORY_PATH	Gibt den Pfadnamen eines Verzeichnisses an, das mindestens eine XML-Datei enthält. Verwenden Sie diese Option, um einen einzigen Kurznamen für mehrere XML-Quelldateien zu erstellen. Der XML-Wrapper verwendet nur die Dateien mit der Erweiterung .xml, die sich in dem von Ihnen angegebenen Verzeichnis befinden. Der XML-Wrapper ignoriert alle anderen Dateien in diesem Verzeichnis. Wenn Sie diese Kurznamenoption angeben, dürfen Sie keine DOCUMENT-Spalte angeben. Diese Option ist nur für den Rootkurznamen gültig.
FILE_PATH	Gibt den Dateipfad des XML-Dokuments an. Wenn Sie FILE_PATH angeben, geben Sie keine DOCUMENT-Spalte an. Diese Option ist nur für den Rootkurznamen gültig.
INSTANCE_PARSE_TIME	Gibt in Millisekunden an, wie viel Zeit erforderlich ist, um eine Zeile des XML-Quelldokuments zu parsen. Der gültige Wert kann eine ganze Zahl oder ein Dezimalwert sein. Der Standardwert ist 7. Diese Option ist nur für Spalten des Rootkurznamens gültig. Modifizieren Sie die Optionen INSTANCE_PARSE_TIME, XPATH_EVAL_TIME und NEXT_TIME, um Abfragen großer oder komplexer XML-Quellenstrukturen zu optimieren.
NAMESPACES	Gibt die Namensbereiche an, die den Namensbereichspräfixen zugeordnet sind, die in den Optionen XPATH und TEMPLATE für jede Spalte verwendet werden. Verwenden Sie folgende Syntax: NAMESPACES 'präfix1="tatsächlicher_namensbereich1", präfix2="tatsächlicher_namensbereich2"' Trennen Sie mehrere Namensbereiche jeweils durch ein Komma. Beispiel: NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''
NEXT_TIME	Gibt in Millisekunden an, wie viel Zeit erforderlich ist, um nachfolgende Quellenelemente aus dem XPath-Ausdruck zu suchen. Der Standardwert ist 1. Diese Option ist für Rootkurznamen und Nicht-Rootkurznamen gültig. Modifizieren Sie die Optionen INSTANCE_PARSE_TIME, XPATH_EVAL_TIME und NEXT_TIME, um Abfragen großer oder komplexer XML-Quellenstrukturen zu optimieren.

Tabelle 87. Kurznamenoptionen für XML (Forts.)

Name	Beschreibung
STREAMING	Gibt an, ob das Quelldokument für die Verarbeitung in logische Fragmente unterteilt werden soll. Die Fragmente entsprechen dem Knoten, der mit dem XPath-Ausdruck des Kurznamens übereinstimmt. Anschließend parst und verarbeitet der Wrapper Quelldaten fragmentweise. Dieser Parsing-Typ minimiert die Speicherbelegung. Gültige Werte sind Y und N. Der Standardwert ist N. Hierbei werden die Dokumente nicht geparkt. Diese Option ist nur für den Rootkurznamen gültig.
VALIDATE	Gibt an, ob das Quelldokument vor dem Extrahieren von Daten überprüft wird, um sicherzustellen, dass es einem XML-Schema oder einer Dokumenttypdefinition (DTD) entspricht. Der Standardwert ist N. Hierbei wird keine Überprüfung vorgenommen. Bevor Sie den Wert auf Y setzen, muss sich die Schemadatei oder die DTD-Datei an der Speicherposition befinden, die vom Quelldokument angegeben wird. Diese Option ist nur für den Rootkurznamen gültig. Setzen Sie die Optionen STREAMING und VALIDATE nicht gemeinsam auf Y.
XPATH	Erforderlich. Gibt den XPath-Ausdruck an, mit dem die Elemente angegeben werden, die die einzelnen Tupel darstellen. Der XPath-Ausdruck wird als Kontext zur Auswertung von Spaltenwerten verwendet, die durch die Spaltenoption XPATH angegeben werden.
XPATH_EVAL_TIME	Gibt in Millisekunden an, wie viel Zeit erforderlich ist, um den XPath-Ausdruck des Kurznamens auszuwerten und das erste Element zu suchen. Der Wert kann eine ganze Zahl oder ein Dezimalwert sein. Der Standardwert ist 1. Diese Option ist für Rootkurznamen und Nicht-Rootkurznamen gültig. Modifizieren Sie die Optionen INSTANCE_PARSE_TIME, XPATH_EVAL_TIME und NEXT_TIME, um Abfragen großer oder komplexer XML-Quellenstrukturen zu optimieren.

Spaltenoptionen

Tabelle 88. Spaltenoptionen für XML

Name	Beschreibung
DOCUMENT	<p>Gibt an, dass diese Spalte eine DOCUMENT-Spalte ist. Der Wert der DOCUMENT-Spalte gibt den Typ der XML-Quellendaten an, die dem Kurznamen zur Verfügung gestellt werden, wenn die Abfrage ausgeführt wird. Diese Option ist nur für Spalten des Rootkurznamens gültig (des Kurznamens, der die Elemente auf der höchsten Ebene des XML-Dokuments angibt). Mit der Option DOCUMENT kann jeweils nur eine Spalte für jeden Kurznamen angegeben werden. Wenn Sie anstelle der Kurznamenoption FILE_PATH oder DIRECTORY_PATH die Spaltenoption DOCUMENT verwenden, wird bei Ausführung der Abfrage das Dokument zur Verfügung gestellt, das diesem Kurznamen entspricht. Gültige Werte sind FILE (Datei), DIRECTORY (Verzeichnis), URI und COLUMN (Spalte). Der Standardwert ist FILE. Die Spalte muss den Datentyp VARCHAR aufweisen.</p>

Tabelle 88. Spaltenoptionen für XML (Forts.)

Name	Beschreibung
FOREIGN_KEY	<p>Gibt an, dass dieser Kurzname ein untergeordneter Kurzname ist, und gibt den Namen des entsprechenden übergeordneten Kurznamens an. Ein Kurzname kann höchstens eine Spaltenoption FOREIGN_KEY haben. Bei dem Wert für die Option muss die Groß-/Kleinschreibung beachtet werden. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join von übergeordneten Kurznamen und einem untergeordneten Kurznamen verwendet werden. Eine Anweisung CREATE NICKNAME, die die Option FOREIGN_KEY enthält, schlägt fehl, wenn der übergeordnete Kurzname einen abweichenden Schemanamen hat. Wenn der Kurzname, auf den in einer Klausel FOREIGN_KEY verwiesen wird, in der Anweisung CREATE NICKNAME nicht explizit in Kleinbuchstaben oder in einer Schreibweise mit gemischter Groß-/Kleinschreibung definiert wurde, müssen Sie diesen Kurznamen in Großbuchstaben angeben, wenn Sie in der Klausel FOREIGN_KEY auf ihn verweisen.</p> <p>Anmerkung:</p> <ul style="list-style-type: none"> • Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden. • Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.
PRIMARY_KEY	<p>Erforderlich für einen übergeordneten Kurznamen, der mindestens einen untergeordneten Kurznamen hat. Gibt an, dass dieser Kurzname ein übergeordneter Kurzname ist. Der Spaltentyp muss VARCHAR(16) lauten. Ein Kurzname kann nur eine einzige Spaltenoption PRIMARY_KEY haben. Yes ist der einzige gültige Wert. Geben Sie für diese Spalte die Option XPATH nicht an. Die Spalte kann nur für den Join von übergeordneten und untergeordneten Kurznamen verwendet werden.</p> <p>Anmerkung:</p> <ul style="list-style-type: none"> • Wenn diese Option für eine Spalte gesetzt ist, kann für diese Spalte keine andere Option gesetzt werden. • Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.

Tabelle 88. Spaltenoptionen für XML (Forts.)

Name	Beschreibung
XPATH	<p>Gibt den XPath-Ausdruck in dem XML-Dokument an, das die Daten enthält, die dieser Spalte entsprechen. Der Wrapper wertet diesen XPath-Ausdruck aus, nachdem die Anweisung CREATE NICKNAME den XPath-Ausdruck aus der Kurznamenoption XPATH angewendet hat.</p> <p>Anmerkung: Wenn Sie diese Spaltenoption festlegen, können Sie später nicht die Anweisung ALTER NICKNAME verwenden, um die Option zu löschen. Stattdessen müssen Sie den Kurznamen löschen und ihn dann ohne diese Spaltenoption erneut erstellen.</p>

Kapitel 37. Sichten in der globalen Katalogtabelle mit Informationen föderierter Datenbanken

Die meisten Katalogsichten in einer föderierten Datenbank sind mit den Katalogsichten in allen anderen Datenbanken von DB2 für Linux, UNIX und Windows identisch.

Es gibt mehrere eindeutige Sichten, die relevante Informationen zu einem föderierten System enthalten, beispielsweise die Sicht SYSCAT.WRAPPERS.

Die SYSCAT-Sichten sind schreibgeschützt. Für eine Sicht im SYSCAT-Schema können keine Aktualisierungs- oder Einfügeoperationen abgesetzt werden. Das empfohlene Verfahren zur Aktualisierung des Systemkatalogs ist die Verwendung der SYSSTAT-Sichten. Ändern Sie Anwendungen, die auf eine SYSCAT-Sicht verweisen, so, dass sie stattdessen auf die aktualisierbare SYSSTAT-Sicht verweisen.

In der folgenden Tabelle sind die SYSCAT-Sichten aufgeführt, die Informationen zu föderierten Systemen enthalten. Diese Sichten sind schreibgeschützt.

Tabelle 89. Bei einem föderierten System üblicherweise verwendete Katalogsichten

Katalogsichten	Beschreibung
SYSCAT.CHECKS	Enthält von Ihnen definierte Informationen zur Prüfung auf Integritätsbedingungen.
SYSCAT.COLCHECKS	Enthält Spalten, auf die durch eine Prüfung auf Integritätsbedingungen verwiesen wird.
SYSCAT.COLUMNS	Enthält Spalteninformationen zu den Datenquellenobjekten (Tabellen und Sichten), für die Sie Kurznamen erstellt haben.
SYSCAT.COLOPTIONS	Enthält Informationen zu Spaltenoptionswerten, die Sie für einen Kurznamen festgelegt haben.
SYSCAT.CONSTDEP	Enthält die Abhängigkeit einer informativen Integrationsbedingung, die Sie definiert haben.
SYSCAT.DATATYPES	Enthält Datentypinformationen zu lokal integrierten und benutzerdefinierten DB2-Datentypen.
SYSCAT.DBAUTH	Enthält die Datenbankberechtigungen, die einzelne Benutzer und Gruppen besitzen.
SYSCAT.FUNCMAPOPTIONS	Enthält Informationen zu Optionswerten, die Sie für eine Funktionszuordnung festgelegt haben.
SYSCAT.FUNCMAPPINGS	Enthält die Funktionszuordnungen zwischen der föderierten Datenbank und den Datenquellenobjekten.
SYSCAT.INDEXCOLUSE	Enthält Spalten, die an einem Index beteiligt sind.
SYSCAT.INDEXES	Enthält Indexspezifikationen für Datenquellenobjekte.
SYSCAT.INDEXOPTIONS	Enthält Informationen zu Indexoptionen.

Tabelle 89. Bei einem föderierten System üblicherweise verwendete Katalogsichten (Forts.)

Katalogsichten	Beschreibung
SYSCAT.KEYCOLUSE	Enthält Spalten, die an einem Schlüssel beteiligt sind, der durch eine Integritätsbedingung über eindeutigen Schlüssel, über Primärschlüssel oder über Fremdschlüssel definiert ist.
SYSCAT.NICKNAMES	Enthält Informationen zu Kurznamen, die Sie erstellt haben.
SYSCAT.REFERENCES	Enthält Informationen zu referenziellen Integritätsbedingungen, die Sie definiert haben.
SYSCAT.ROUTINES	Enthält lokale benutzerdefinierte Funktionen oder Funktionsschablonen von DB2. Funktionsschablonen werden verwendet, um eine Zuordnung zu einer Datenquellenfunktion vorzunehmen.
SYSCAT.REVTYPEMAPPINGS	Diese Sicht wird nicht verwendet. Alle Datentypzuordnungen sind in der Sicht SYSCAT.TYPEMAPPINGS verzeichnet.
SYSCAT.ROUTINEOPTIONS	Enthält Informationen zu Optionswerten für föderierte Routinen.
SYSCAT.ROUTINEPARMOPTIONS	Enthält Informationen zu Optionswerten für Parameter föderierter Routinen.
SYSCAT.ROUTINEPARMS	Enthält einen Parameter oder das Ergebnis einer Routine, die in SYSCAT.ROUTINES definiert ist.
SYSCAT.ROUTINESFEDERATED	Enthält Informationen zu föderierten Routinen, die Sie definiert haben.
SYSCAT.SERVERS	Enthält Serverdefinitionen, die Sie für Datenquellenserver erstellen.
SYSCAT.TABCONST	Jede Zeile stellt eine Tabellen- und Kurznamenintegritätsbedingung des Typs CHECK, UNIQUE, PRIMARY KEY oder FOREIGN KEY dar.
SYSCAT.TABLES	Enthält Informationen zu allen lokalen DB2-Tabellen, föderierten Sichten und Kurznamen, die Sie erstellen.
SYSCAT.TYPEMAPPINGS	Enthält progressive Datentypzuordnungen und regressive Datentypzuordnungen. Die Zuordnung wird von den Datentypen der Datenquelle zu den lokalen DB2-Datentypen vorgenommen. Diese Zuordnungen werden verwendet, wenn Sie einen Kurznamen für ein Datenquellenobjekt erstellen.
SYSCAT.USEROPTIONS	Enthält Informationen zur Benutzerberechtigung, die Sie beim Erstellen von Benutzerzuordnungen zwischen der föderierten Datenbank und den Datenquellenservern festlegen.
SYSCAT.VIEWS	Enthält Informationen zu lokalen föderierten Sichten, die Sie erstellen.

Tabelle 89. Bei einem föderierten System üblicherweise verwendete Katalogsichten (Forts.)

Katalogsichten	Beschreibung
SYSCAT.WRAPOPTIONS	Enthält Informationen zu Optionswerten, die Sie für einen Wrapper festgelegt haben.
SYSCAT.WRAPPERS	Enthält den Namen des Wrappers und der Bibliotheksdatei für jede Datenquelle, für die Sie einen Wrapper erstellen.

In der folgenden Tabelle sind die SYSSTAT-Sichten angegeben, die föderierte Informationen enthalten. Diese Sichten mit Schreib-/Lesezugriff enthalten Statistikdaten, die Sie aktualisieren können.

Tabelle 90. Aktualisierbare Sichten des globalen Katalogs für föderierte Informationen

Katalogsichten	Beschreibung
SYSSTAT.COLUMNS	Enthält statistische Information zu allen Spalten in den Datenquellenobjekten (Tabellen und Sichten), für die Sie Kurznamen erstellt haben. Für übernommene Spalten von typisierten Tabellen werden keine Statistikdaten aufgezeichnet.
SYSSTAT.INDEXES	Enthält statistische Informationen zu allen Indexspezifikationen für Datenquellenobjekte.
SYSSTAT.ROUTINES	Enthält statistische Information zu allen benutzerdefinierten Funktionen. Integrierte Funktionen werden hierbei nicht berücksichtigt. Für übernommene Spalten von typisierten Tabellen werden keine Statistikdaten aufgezeichnet.
SYSSTAT.TABLES	Enthält Informationen zu allen Basistabellen. Informationen zu Sichten, Synonymen und Aliasnamen sind in dieser Sicht nicht enthalten. Für typisierte Tabellen wird nur die Stammtabelle einer Tabellenhierarchie in die Sicht aufgenommen. Für übernommene Spalten von typisierten Tabellen werden keine Statistikdaten aufgezeichnet.

Kapitel 38. Funktionszuordnungsoptionen für föderierte Systeme

Der Server mit föderierten Datenbanken bietet Standardzuordnungen zwischen DB2-Funktionen und Datenquellenfunktionen. Für die meisten Datenquellen sind die Standardfunktionszuordnungen in den Wrappern enthalten. Um eine vom Server mit föderierten Datenbanken nicht erkannte Datenquellenfunktion zu verwenden oder die Standardzuordnung zu ändern, erstellen Sie eine Funktionszuordnung.

Bei der Erstellung einer Funktionszuordnung geben Sie den Namen der Datenquellenfunktion an. Außerdem müssen Sie die zugeordnete Funktion aktivieren. Wenn Sie anschließend die zugeordnete Funktion verwenden, vergleicht das Abfrageoptimierungsprogramm die Kosten, die durch die Ausführung der Funktion in der Datenquelle entstehen, mit den Kosten für die Ausführung der Funktion auf dem Server mit föderierten Datenbanken.

Tabelle 91. Optionen für Funktionszuordnungen

Name	Beschreibung
DISABLE	Aktiviert bzw. inaktiviert eine Standardfunktionszuordnung. Gültige Werte sind Y und N. Der Standardwert ist N.
REMOTE_NAME	Der Name der Datenquellenfunktion. Standardwert ist der lokale Name.

Kapitel 39. Gültige Servertypen in SQL-Anweisungen

Servertypen geben den Typ der Datenquelle an, die von der Serverdefinition dargestellt wird.

Servertypen variieren je nach Hersteller, Zweck und Betriebssystem. Welche Werte unterstützt werden, ist von der Datenquelle abhängig.

Bei den meisten Datenquellen müssen Sie in der Anweisung CREATE SERVER einen gültigen Servertyp angeben.

Tabelle 92. Datenquellen und Servertypen

Datenquelle	Servertyp
BioRS	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.
Excel	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.
IBM DB2 Universal Database für Linux, UNIX und Windows	DB2/UDB
IBM DB2 Universal Database für System i und AS/400	DB2/ISERIES
IBM DB2 Universal Database für z/OS	DB2/ZOS
IBM DB2 für VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (Erforderlich für JDBC-Datenquellen, die von JDBC-Treibern der Version 3.0 und höher unterstützt werden.)
Microsoft SQL Server	MSSQLSERVER (Erforderlich für Datenquellen, die vom Treiber für DataDirect Connect ODBC 4.2 (oder höher) oder vom Treiber für Microsoft SQL Server ODBC 3.0 (oder höher) unterstützt werden.)
ODBC	ODBC (Erforderlich für ODBC-Datenquellen, die vom ODBC-Treiber der Version 3.x unterstützt werden.)
OLE DB	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.
Oracle	ORACLE (Erforderlich für Oracle-Datenquellen, die von der Oracle NET8-Client-Software unterstützt werden.)
Sybase (CTLIB)	SYBASE
Dateien mit Tabellenstruktur	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.
Teradata	TERADATA
Web-Services	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.
XML	In der Anweisung CREATE SERVER ist kein Servertyp erforderlich.

Kapitel 40. Datentypzuordnungen

Zu den Datentypzuordnungen für relationale Datenquellen gehören progressive Typzuordnungen, regressive Typzuordnungen und spezielle Typzuordnungen für Unicode. Jede der nicht relationalen Datenquellen unterstützt bestimmte Datentypen.

Standardmäßige progressive Datentypzuordnungen

Progressive Typzuordnungen und regressive Typzuordnungen sind die beiden Arten von Zuordnungen zwischen Datentypen der Datenquelle und Datentypen der föderierten Datenbank. Eine progressive Typzuordnung ist eine Zuordnung von einem fernen Typ zu einem vergleichbaren lokalen Typ.

Sie können eine Standardtypzuordnung überschreiben oder eine neue Zuordnung erstellen (Anweisung CREATE TYPE MAPPING).

Diese Zuordnungen sind - sofern nichts anderes angegeben ist - bei allen unterstützten Versionen gültig.

Bei allen standardmäßigen progressiven Datentypzuordnungen von einer Datenquelle zur föderierten Datenbank wird das föderierte Schema SYSIBM verwendet.

In den folgenden Tabellen sind die standardmäßigen progressiven Zuordnungen zwischen den Datentypen der föderierten Datenbank und den Datentypen der Datenquelle angegeben.

Standardmäßige progressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen aufgeführt.

Tabelle 93. Standardmäßige progressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows (nicht alle Spalten sind dargestellt)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	J	-	CHAR	-	0	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-

Tabelle 93. Standardmäßige progressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
DECFLOAT ²	-	-	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	J	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	p	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	J	-	VARCHAR	-	0	J
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Anmerkung:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.
2. Die Serveroption SAME_DECFLT_ROUNDING ist standardmäßig auf N gesetzt. Operationen werden nur dann im Pushdown-Modus an die ferne Datenquelle weitergeleitet, wenn SAME_DECFLT_ROUNDING auf Y gesetzt ist. Informationen zur Serveroption SAME_DECFLT_ROUNDING finden Sie unter Referenz für Optionen der DB2-Datenbank.

Standardmäßige progressive Datentypzuordnungen bei DB2 für System i-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei DB2 für System i-Datenquellen aufgeführt.

Tabelle 94. Standardmäßige progressive Datentypzuordnungen bei DB2 für System i-Datenquellen (nicht alle Spalten sind dargestellt)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	J	-	CHAR	-	0	J
CHAR	255	32672	-	-	J	-	VARCHAR	-	0	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-

Tabelle 94. Standardmäßige progressive Datentypzuordnungen bei DB2 für System i-Datenquellen (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bit-daten
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	J	-	VARCHAR	-	0	J
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Standardmäßige progressive Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen aufgeführt.

Tabelle 95. Standardmäßige progressive Datentypzuordnungen bei DB2 Server für VM und VSE (nicht alle Spalten sind dargestellt)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	J	-	CHAR	-	0	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-

Tabelle 95. Standardmäßige progressive Datentypzuordnungen bei DB2 Server für VM und VSE (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope- ra- to- ren	Föderierter Typname	Föderierte Länge	Föde- rierte Anzahl der Kom- ma- stellen	Föderierte Bitdaten
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMES- TAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	J	-	VARCHAR	-	0	J
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Standardmäßige progressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei DB2 für z/OS-Datenquellen aufgeführt.

Tabelle 96. Standardmäßige progressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen (nicht alle Spalten sind dargestellt)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope- ra- to- ren	Föderierter Typ-name	Föderierte Länge	Föde- rierte Anzahl der Kom- ma- stellen	Föderierte Bit-daten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	J	-	CHAR	-	0	J
CHAR	255	32672	-	-	J	-	VARCHAR	-	0	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
ROWID	-	-	-	-	J	-	VARCHAR	40	-	J
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-

Tabelle 96. Standardmäßige progressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typ-name	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bit-daten
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	J	-	VARCHAR	-	0	J
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Standardmäßige progressive Datentypzuordnungen bei Informix-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Informix-Datenquellen aufgeführt.

Tabelle 97. Standardmäßige progressive Datentypzuordnungen bei Informix (nicht alle Spalten sind dargestellt)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	4	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DATETIME ²	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP(6)	10	6	-
DATETIME	6	10	11	15	-	-	TIMESTAMP(6)	10	6	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	130	-	-	-	-	DOUBLE	8	-	-
DECIMAL	1	32	255	255	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
INTERVAL	-	-	-	-	-	-	VARCHAR	25	-	-
INT8	-	-	-	-	-	-	BIGINT	19	0	-
LVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
MONEY	1	31	0	31	-	-	DECIMAL	-	-	-
MONEY	32	32	-	-	-	-	DOUBLE	8	-	-

Tabelle 97. Standardmäßige progressive Datentypzuordnungen bei Informix (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope-ratoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
NCHAR	1	254	-	-	-	-	CHARACTER	-	-	-
NCHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
NVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

Anmerkungen:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.
2. Beim Informix-Datentyp DATETIME verwendet der föderierte Server von DB2 für UNIX und Windows das übergeordnete Informix-Qualifikationsmerkmal für REMOTE_LENGTH und das untergeordnete Informix-Qualifikationsmerkmal für REMOTE_SCALE.

Die Informix-Qualifikationsmerkmale sind die Konstanten 'TU_', die in der Datei datatime.h von Informix Client SDK definiert sind. Die Konstanten lauten folgendermaßen:

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Standardmäßige progressive Datentypzuordnungen bei JDBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei JDBC-Datenquellen aufgeführt.

Tabelle 98. Standardmäßige progressive Datentypzuordnungen bei JDBC

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope-ratoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	J
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	J
BINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
BIT	-	-	-	-	-	-	SMALLINT	2	-	-
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN-	-	-	-	-	-	-	SMALLINT	2	-	-

Tabelle 98. Standardmäßige progressive Datentypzuordnungen bei JDBC (Forts.)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Kommastellen	Ferne obere Anzahl der Kommastellen	Ferne Bitdaten	Ferne Datenoperatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Kommastellen	Föderierte Bitdaten
CHAR	-	254	-	-	-	-	CHAR	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
FLOAT	-	-	-	-	-	-	FLOAT	4	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
LONGVARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
LONGVARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
LONGVARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	J
LONGVARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
LONGNVARCHAR	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
LONGNVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCHAR ²	-	127	-	-	-	-	GRAPHIC	-	-	-
NCHAR ²	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
NCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCLOB ²	-	-	-	-	-	-	DBCLOB	-	-	-
NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
NVARCHAR ²	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
NVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
REAL							REAL	4		
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TIME	-	-	-	-	-	-	TIME	3	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
VARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	J
VARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-

Anmerkung:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.
2. Die Datentypen NCHAR, NVARCHAR, LONGVARCHAR und NCLOB werden nur durch den JDBC-Treiber der Version 4.0 unterstützt.

Die folgenden Datentypen werden durch den JDBC-Wrapper nicht unterstützt:
 DATALINK, OTHER, JAVA_OBJECT, DISTINCT, STRUCT, ARRAY und REF.

Standardmäßige progressive Datentypzuordnungen bei Microsoft SQL Server-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Microsoft SQL Server-Datenquellen aufgeführt.

Tabelle 99. Standardmäßige progressive Datentypzuordnungen bei Microsoft SQL Server

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope-rationen	Föderierter Typname	Föde-rierte Länge	Föde-rierte Anzahl der Komma-stellen	Föde-rierte Bitdaten
bigint ¹	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	J
binary	255	8000	-	-	-	-	VARCHAR	-	-	J
bit	-	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMES-TAMP(6)	10	6	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	8	-	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	-	REAL	4	-	-
image	-	-	-	-	-	-	BLOB	2147483647	-	J
int	-	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	127	-	-	-	-	CHAR	-	-	N
nchar	128	4000	-	-	-	-	VARCHAR	-	-	N
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	8	-	-
ntext	-	-	-	-	-	-	CLOB	2147483647	-	J
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	-	TIMES-TAMP(6)	10	6	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	J
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	J
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-

Tabelle 99. Standardmäßige progressive Datentypzuordnungen bei Microsoft SQL Server (Forts.)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föde-rierte Länge	Föde-rierte Anzahl der Komma-stellen	Föde-rierte Bitdaten
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_GUID	-	-	-	-	-	-	VARCHAR	-	-	J
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	J
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	6	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	J
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	254	-	-	-	-	CHARACTER	-	-	N
SQL_WCHAR	255	8800	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	J
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	J	-	VARCHAR	16	-	J
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	J
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

Anmerkung:

1. Diese Typzuordnung ist nur bei Microsoft SQL Server Version 2000 gültig.

Standardmäßige progressive Datentypzuordnungen bei ODBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei ODBC-Datenquellen aufgeführt.

Tabelle 100. Standardmäßige progressive Datentypzuordnungen bei ODBC (nicht alle Spalten sind dargestellt)

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bit-daten
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	J
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	J
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	8	-	-	-	-	FLOAT	8	-	-
SQL_FLOAT	-	4	-	-	-	-	FLOAT	4	-	-
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	2147483647	-	J
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	4	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	J
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N
SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N

Anmerkung:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.

Standardmäßige progressive Datentypzuordnungen bei Oracle NET8-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Oracle NET8-Datenquellen aufgeführt.

Tabelle 101. Standardmäßige progressive Datentypzuordnungen bei Oracle NET8

Ferner Typname	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Kommastellen	Ferne obere Anzahl der Kommastellen	Ferne Bitdaten	Ferne Datenoperatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Kommastellen	Föderierte Bitdaten
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	J
CHAR	1	254	0	0	-	\0	CHAR	0	0	N
CHAR	255	2000	0	0	-	\0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	\0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	\0	TIMESTAMP(6)	0	0	N
FLOAT	1	126	0	0	-	\0	DOUBLE	0	0	N
LONG	0	0	0	0	-	\0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	\0	BLOB	2147483647	0	J
NUMBER	10	18	0	0	-	\0	BIGINT	0	0	N
NUMBER	1	38	-84	127	-	\0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	4	0	0	-	\0	SMALLINT	0	0	N
NUMBER	5	9	0	0	-	\0	INTEGER	0	0	N
NUMBER	-	10	0	0	-	\0	DECIMAL	0	0	N
RAW	1	2000	0	0	-	\0	VARCHAR	0	0	J
ROWID	0	0	0	NULL	-	\0	CHAR	18	0	N
TIMESTAMP(p) ¹	-	-	-	-	-	\0	TIMESTAMP(6)	10	6	N
VARCHAR2	1	4000	0	0	-	\0	VARCHAR	0	0	N

Anmerkung:

1.

- `TIMESTAMP(p)` stellt eine Zeitmarke mit einer variablen Anzahl der Kommastellen von 0-9 dar. Die Anzahl der Kommastellen der Oracle-Zeitmarke wird standardmäßig `TIMESTAMP(6)` zugeordnet. Sie können diese Standardtypzuordnung ändern und den Oracle-Typ `TIMESTAMP` zu einem föderierten Typ `TIMESTAMP` mit derselben Anzahl der Kommastellen zuordnen. Hierzu verwenden Sie eine benutzerdefinierte Typzuordnung.
- Diese Typzuordnung ist nur bei Client- und Serverkonfigurationen von Oracle 9i (oder höher) gültig.

Standardmäßige progressive Datentypzuordnungen bei Sybase-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Sybase-Datenquellen aufgeführt.

Tabelle 102. Standardmäßige progressive Datentypzuordnungen bei Sybase CTLIB

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typ-name	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bit-daten
binary	1	254	-	-	-	-	CHAR	-	-	J
binary	255	32672	-	-	-	-	VARCHAR	-	-	J
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (siehe varchar)										
date	-	-	-	-	-	-	DATE	-	-	-
date	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
datetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
datetimn	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	32672	-	-	-	-	VARCHAR	-	-	N
nchar null (siehe nvarchar)										
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
nvarchar	1	32672	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	-	-	-	-	-	-	VARCHAR	30	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
Zeit	-	-	-	-	-	-	TIME	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	J

Tabelle 102. Standardmäßige progressive Datentypzuordnungen bei Sybase CTLIB (Forts.)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typ-name	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bit-daten
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N
unichar null (siehe univarchar)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	J
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

Anmerkung:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.
2. Bei föderierten Datenbanken gültig, die nicht Unicode verwenden.

Standardmäßige progressive Datentypzuordnungen bei Teradata-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Teradata-Datenquellen aufgeführt.

Tabelle 103. Standardmäßige progressive Datentypzuordnungen bei Teradata (nicht alle Spalten sind dargestellt)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-operatoren	Föderierter Typname	Föderierte Länge	Föderierte Anzahl der Komma-stellen	Föderierte Bitdaten
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	J
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	J
BYTE	32673	64000	-	-	-	-	BLOB	-	-	-
BYTEINT	-	-	-	-	-	-	SMALLINT	-	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	64000	-	-	-	-	CLOB	-	-	-
CLOB	1	2097088000 (lateinischer Zeichensatz)	-	-	-	-	CLOB	-	-	-
CLOB	1	1048544000 (Unicode)	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DOUBLE PRECISION	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-

Tabelle 103. Standardmäßige progressive Datentypzuordnungen bei Teradata (nicht alle Spalten sind dargestellt) (Forts.)

Ferner Typ-name	Ferne untere Länge	Ferne obere Länge	Ferne untere Anzahl der Komma-stellen	Ferne obere Anzahl der Komma-stellen	Ferne Bit-daten	Ferne Daten-ope-ratoren	Föderierter Typname	Föde-rierte Länge	Föde-rierte Anzahl der Komma-stellen	Föderierte Bitdaten
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	-	-
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
GRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
INTERVAL	-	-	-	-	-	-	CHAR	-	-	-
NUMERIC	1	18	0	18	-	-	DECIMAL	-	-	-
REAL	-	-	-	-	-	-	DOUBLE	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	0	21	0	21	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(6)	10	6	-
VARBYTE	1	32762	-	-	-	-	VARCHAR	-	-	J
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

Anmerkung:

1. Der föderierte Typ ist TIMESTAMP(0), falls der Konfigurationsparameter 'date_compat' auf ON gesetzt ist.

Beispiele für progressive Datentypzuordnungen

Durch die Verwendung der Beispiele für die progressiven Datentypzuordnungen können Sie die Unterstützung für den Datentyp TIMESTAMP mit Genauigkeit nutzen.

Bei Informix-Datenquellen werden diese Typzuordnungen für Kurznamenspalentypen, Parameter für föderierte Prozeduren, Durchgriff und Ergebnismengen für föderierte Prozeduren verwendet.

Bei anderen Datenquellen als Informix-Datenquellen betreffen diese Typzuordnungen nur die Zuordnungen für Kurznamenspalentypen und Parameter für föderierte Prozeduren. Auf Durchgriff und Ergebnismengen für föderierte Prozeduren wirken sich die Zuordnungen nicht aus.

Progressive Datentypzuordnungen - Beispiel für Informix

Bei der Erstellung von föderierten Objekten können Sie das für Informix bereitgestellte Beispiel für die progressive Typzuordnung verwenden.

Sie müssen die folgenden Zuordnungen erstellen, bevor Sie ein föderiertes Objekt erstellen.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,10);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,11);
```

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,12);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,13);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,14);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,15);

```

Progressive Datentypzuordnungen - Beispiel für Microsoft SQL Server

Bei der Erstellung von föderierten Objekten können Sie das für Microsoft SQL Server bereitgestellte Beispiel für die progressive Typzuordnung verwenden.

Sie müssen die folgenden Zuordnungen erstellen, bevor Sie einen Kurznamen oder eine föderierte Prozedur erstellen.

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE mssqlserver REMOTE TYPE "datetime";

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE mssqlserver REMOTE TYPE "smalldatetime";

```

Progressive Datentypzuordnungen - Beispiel für Oracle

Bei der Erstellung von föderierten Objekten können Sie das für Oracle bereitgestellte Beispiel für die progressive Typzuordnung verwenden.

Sie müssen die folgenden Zuordnungen erstellen, bevor Sie einen Kurznamen oder eine föderierte Prozedur erstellen.

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(5);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(7)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(7);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(8)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(8);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(9)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(9);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE date;

```

Progressive Datentypzuordnungen - Beispiel für Sybase

Bei der Erstellung von föderierten Objekten können Sie das für Sybase bereitgestellte Beispiel für die progressive Typzuordnung verwenden.

Sie müssen die folgenden Zuordnungen erstellen, bevor Sie einen Kurznamen oder eine föderierte Prozedur erstellen.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE sybase REMOTE TYPE datetime);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE sybase REMOTE TYPE smalldatetime);
```

Progressive Datentypzuordnungen - Beispiel für Teradata

Bei der Erstellung von föderierten Objekten können Sie das für Teradata bereitgestellte Beispiel für die progressive Typzuordnung verwenden.

Sie müssen die folgenden Zuordnungen erstellen, bevor Sie einen Kurznamen oder eine föderierte Prozedur erstellen.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(5);
```

Standardmäßige regressive Datentypzuordnungen

Für die meisten Datenquellen sind die Standardtypzuordnungen in den Wrappern enthalten.

Progressive Typzuordnungen und regressive Typzuordnungen sind die beiden Arten von Zuordnungen zwischen Datentypen der Datenquelle und Datentypen der föderierten Datenbank. Eine progressive Typzuordnung ist eine Zuordnung von einem fernen Typ zu einem vergleichbaren lokalen Typ. Der andere Zuordnungstyp - die regressive Typzuordnung - wird mit transparenter DDL zur Erstellung oder Modifizierung von fernen Tabellen verwendet.

Die Standardtypzuordnungen bei Datenquellen der DB2-Produktfamilie sind im DRDA-Wrapper enthalten. Die Standardtypzuordnungen für Informix sind im Informix-Wrapper enthalten usw.

Wenn Sie eine ferne Tabelle oder Sicht für die föderierte Datenbank definieren, umfasst die Definition eine regressive Typzuordnung. Die Zuordnung wird für jede Spalte zwischen einem lokalen Datentyp der föderierten Datenbank und dem entsprechenden fernen Datentyp vorgenommen. Es gibt beispielsweise eine standardmäßige regressive Typzuordnung, bei der der lokale Typ REAL auf den Informix-Typ SMALLFLOAT zeigt.

Föderierte Datenbanken unterstützen bei den Datentypen LONG VARCHAR und LONG VARGRAPHIC sowie bei benutzerdefinierten Typen keine Zuordnungen.

Wenn Sie mit der Anweisung CREATE TABLE eine ferne Tabelle erstellen, geben Sie die lokalen Datentypen an, die Sie in die ferne Tabelle einschließen wollen. Diese standardmäßigen regressiven Typzuordnungen ordnen diesen Spalten die entsprechenden fernen Typen zu. Beispiel: Sie verwenden die Anweisung CREATE TABLE, um eine Informix-Tabelle mit einer Spalte namens 'C2' zu definieren. Als Datentyp für die Spalte 'C2' geben Sie in der Anweisung BIGINT an. Die standardmäßige regressiv Typzuordnung für BIGINT ist von der Informix-Version abhängig, unter der Sie die Tabelle erstellen. Die Zuordnung für die Spalte 'C2' in der Informix-Tabelle ist bei Informix Version 8 der Typ DECIMAL und bei Informix Version 9 der Typ INT8.

Sie können eine standardmäßige regressiv Typzuordnung überschreiben oder eine neue regressiv Zuordnung erstellen (Anweisung CREATE TYPE MAPPING).

In den folgenden Tabellen sind die standardmäßigen regressiven Zuordnungen zwischen den lokalen Datentypen der föderierten Datenbank und den Datentypen der fernen Datenquelle angegeben.

Diese Zuordnungen sind - sofern nichts anderes angegeben ist - bei allen unterstützten Versionen gültig.

Standardmäßige regressiv Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows-Datenquellen aufgeführt.

Tabelle 104. Standardmäßige regressiv Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typ-name	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	J	-	CHAR	-	-	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	8	-	-	-	-	DECFLOAT	-	0	-
DECFLOAT ²	-	16	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-

Tabelle 104. Standardmäßige regressive Datentypzuordnungen bei DB2 Database für Linux, UNIX und Windows (nicht alle Spalten sind dargestellt) (Forts.)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Kommastellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kommastellen	Ferne Bitdaten
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	p ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	J	-	VARCHAR	-	-	J
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Anmerkung:

1. Wenn der Parameter 'date_compat' auf **OFF** gesetzt ist, wird der föderierte Datentyp DATE dem Datentyp TIMESTAMP(0) zugeordnet.
2. Die Serveroption SAME_DECFLT_ROUNDING ist standardmäßig auf **N** gesetzt. Operationen werden nur dann im Pushdown-Modus an die ferne Datenquelle weitergeleitet, wenn SAME_DECFLT_ROUNDING auf **Y** gesetzt ist. Informationen zur Serveroption SAME_DECFLT_ROUNDING finden Sie unter Referenz für Optionen der DB2-Datenbank.
3. Bei Version 9.5 oder früher hat die ferne Anzahl der Kommastellen bei TIMESTAMP den Wert 6.

Standardmäßige regressive Datentypzuordnungen bei DB2 für System i-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei DB2 für System i-Datenquellen aufgeführt.

Tabelle 105. Standardmäßige regressive Datentypzuordnungen bei DB2 für System i-Datenquellen (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Kommastellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kommastellen	Ferne Bitdaten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	J	-	CHARACTER	-	-	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-

Tabelle 105. Standardmäßige regressive Datentypzuordnungen bei DB2 für System i-Datenquellen (nicht alle Spalten sind dargestellt) (Forts.)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typ-name	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bit-daten
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)-	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	J	-	VARCHAR	-	-	J
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

Standardmäßige regressive Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei DB2 für VM und VSE-Datenquellen aufgeführt.

Tabelle 106. Standardmäßige regressive Datentypzuordnungen bei DB2 für VM und VSE (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typ-name	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	J	-	CHAR	-	-	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)-	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	J	-	VARCHAR	-	-	J
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

Standardmäßige regressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei DB2 für z/OS-Datenquellen aufgeführt.

Tabelle 107. Standardmäßige regressive Datentypzuordnungen bei DB2 für z/OS-Datenquellen (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Kommastellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kommastellen	Ferne Bitdaten
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	J	-	CHAR	-	-	J
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	J	-	VARCHAR	-	-	J
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Standardmäßige regressive Datentypzuordnungen bei Informix-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Informix-Datenquellen aufgeführt.

Tabelle 108. Standardmäßige regressive Datentypzuordnungen bei Informix

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Kommastellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kommastellen	Ferne Bitdaten
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	J	-	BYTE	-	-	-
CLOB	1	2147483647	-	-	-	-	TEXT	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-

Tabelle 108. Standardmäßige regressive Datentypzuordnungen bei Informix (Forts.)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	DATETIME	6	10	-
TIMES-TAMP	-	10	-	-	-	-	DATETIME	0	15	-
VARCHAR	1	254	-	-	N	-	VARCHAR	-	-	-
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	J	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

Anmerkung:

1. Diese Typzuordnung ist nur bei Informix Server Version 8 (oder früher) gültig.
2. Diese Typzuordnung ist nur bei Informix Server Version 9 (oder höher) gültig.

Beim Informix-Datentyp DATETIME verwendet der föderierte Server das übergeordnete Informix-Qualifikationsmerkmal für REMOTE_LENGTH und das untergeordnete Informix-Qualifikationsmerkmal für REMOTE_SCALE.

Die Informix-Qualifikationsmerkmale sind die Konstanten 'TU_', die in der Datei `datatime.h` von Informix Client SDK definiert sind. Die Konstanten lauten folgendermaßen:

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Standardmäßige regressive Datentypzuordnungen bei JDBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei JDBC-Datenquellen aufgeführt.

Tabelle 109. Standardmäßige regressive Datentypzuordnungen bei JDBC

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	J	-	BINARY	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	NCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-

Tabelle 109. Standardmäßige regressiv Datentypzuordnungen bei JDBC (Forts.)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typ-name	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	NCHAR	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	min(9,p)	-
VARCHAR	-	-	-	-	J	-	VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	-	-	NVARCHAR	-	-	-

Standardmäßige regressiv Datentypzuordnungen bei Microsoft SQL Server-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Microsoft SQL Server-Datenquellen aufgeführt.

Tabelle 110. Standardmäßige regressiv Datentypzuordnungen bei Microsoft SQL Server (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	J	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-
VARCHAR	1	8000	-	-	J	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	J	-	image	-	-	-

Tabelle 110. Standardmäßige regressive Datentypzuordnungen bei Microsoft SQL Server (nicht alle Spalten sind dargestellt) (Forts.)

Föderierter Typname	Föde- rierte untere Länge	Föderierte obere Länge	Föde- rierte untere Anzahl der Komma- stellen	Föde- rierte obere Anzahl der Komma- stellen	Föde- rierte Bitdaten	Föderierte Daten- operatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma- stellen	Ferne Bit- daten
------------------------	------------------------------------	------------------------------	---	--	-----------------------------	------------------------------------	-------------------	----------------	---	---------------------

Anmerkung:

1. Diese Typzuordnung ist nur bei Microsoft SQL Server Version 2000 gültig.

Standardmäßige regressive Datentypzuordnungen bei ODBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei ODBC-Datenquellen aufgeführt.

Tabelle 111. Standardmäßige regressive Datentypzuordnungen bei ODBC (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föde- rierte untere Länge	Föde- rierte obere Länge	Föde- rierte untere Anzahl der Komma- stellen	Föde- rierte obere Anzahl der Komma- stellen	Föde- rierte Bitdaten	Föde- rierte Daten- operatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kom- ma- stellen	Ferne Bit- daten
BIGINT	-	-	-	-	-	-	SQL_BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_LONGVARBINARY	-	-	-
CHAR	-	-	-	-	J	-	SQL_BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_CHAR	-	-	-
CLOB	-	-	-	-	-	-	SQL_LONGVARCHAR	-	-	-
DATE	-	4	-	-	-	-	SQL_TYPE_DATE	-	-	-
DBCLOB	-	-	-	-	-	-	SQL_WLONGVARCHAR	-	-	-
DECIMAL	-	-	-	-	-	-	SQL_DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	SQL_DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	SQL_FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	SQL_WCHAR	-	-	-
INTEGER	-	-	-	-	-	-	SQL_INTEGER	-	-	-
REAL	-	4	-	-	-	-	SQL_REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SQL_SMALLINT	-	-	-
TIME	-	3	-	-	-	-	SQL_TYPE_TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	SQL_TYPE_TIMESTAMP(p)	-	-	-
VARCHAR	-	-	-	-	J	-	SQL_VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	J	-	SQL_WVARCHAR	-	-	-

Standardmäßige regressive Datentypzuordnungen bei Oracle NET8-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Oracle NET8-Datenquellen aufgeführt.

Tabelle 112. Standardmäßige regressive Datentypzuordnungen bei Oracle NET8

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Kommastellen	Föderierte obere Anzahl der Kommastellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Kommastellen	Ferne Bitdaten
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	J	\0	BLOB	0	0	J
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	J	\0	RAW	0	0	J
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N
DATE ¹	0	4	0	0	N	\0	DATE	0	0	N
DECIMAL	0	0	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	8	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	16	0	0	N	\0	NUMBER	0	0	N
DOUBLE	0	8	0	0	N	\0	FLOAT	126	0	N
FLOAT	0	8	0	0	N	\0	FLOAT	126	0	N
INTEGER	0	4	0	0	N	\0	NUMBER	10	0	N
REAL	0	4	0	0	N	\0	FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0	NUMBER	5	0	N
TIME	0	3	0	0	N	\0	DATE	0	0	N
TIMESTAMP ²	0	10	0	0	N	\0	DATE	0	0	N
TIMESTAMP(p) ³	-	-	-	-	N	\0	TIMESTAMP(p)	-	-	N
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	J	\0	RAW	0	0	J

Anmerkung:

1. Wenn der Parameter 'date_compat' auf **OFF** gesetzt ist, wird der föderierte Datentyp DATE dem Oracle-Datum zugeordnet. Wenn der Parameter 'date_compat' auf **ON** gesetzt ist, wird der föderierte Datentyp DATE (äquivalent zu TIMESTAMP(0)) dem Oracle-Datentyp TIMESTAMP(0) zugeordnet.
2. Diese Typzuordnung ist nur bei Oracle Version 8 gültig.
3.
 - TIMESTAMP(p) stellt eine Zeitmarke mit einer variablen Anzahl der Kommastellen von 0-9 bei Oracle und von 0-12 bei Federation dar. Bei einer Anzahl der Kommastellen von 0-9 hat der ferne Oracle-Datentyp TIMESTAMP dieselbe Anzahl der Kommastellen wie der föderierte Datentyp TIMESTAMP. Ist die Anzahl der Kommastellen des föderierten Datentyps TIMESTAMP größer als 9, ist die entsprechende Anzahl der Kommastellen des Oracle-Datentyps TIMESTAMP 9, also die größte Anzahl der Kommastellen bei Oracle.
 - Diese Typzuordnung ist nur bei Oracle Version 9, 10 und 11 gültig.

Standardmäßige regressive Datentypzuordnungen bei Sybase-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Sybase-Datenquellen aufgeführt.

Tabelle 113. Standardmäßige regressive Datentypzuordnungen bei Sybase CTLIB

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	J	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR ¹	1	255	-	-	N	-	varchar	-	-	-
VARCHAR ¹	256	32672	-	-	N	-	text	-	-	-
VARCHAR ²	1	16384	-	-	N	-	varchar	-	-	-
VARCHAR ²	16385	32672	-	-	N	-	text	-	-	-
VARCHAR ¹	1	255	-	-	J	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	J	-	image	-	-	-
VARCHAR ²	1	16384	-	-	J	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	J	-	image	-	-	-

Anmerkung:

1. Diese Typzuordnung ist nur bei CTLIB mit Sybase Server Version 12.0 (oder früher) gültig.
2. Diese Typzuordnung ist nur bei CTLIB mit Sybase Server Version 12.5 (oder höher) gültig.

Standardmäßige regressive Datentypzuordnungen bei Teradata-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Teradata-Datenquellen aufgeführt.

Tabelle 114. Standardmäßige regressive Datentypzuordnungen bei Teradata (nicht alle Spalten sind dargestellt)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Datenoperatoren	Ferner Typname	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	J	-	BYTE	-	-	-
CLOB	1	2097088000	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-

Tabelle 114. Standardmäßige regressive Datentypzuordnungen bei Teradata (nicht alle Spalten sind dargestellt) (Forts.)

Föderierter Typname	Föderierte untere Länge	Föderierte obere Länge	Föderierte untere Anzahl der Komma-stellen	Föderierte obere Anzahl der Komma-stellen	Föderierte Bitdaten	Föderierte Daten-operatoren	Ferner Typ-name	Ferne Länge	Ferne Anzahl der Komma-stellen	Ferne Bitdaten
DBCLOB ¹	1	64000	-	-	-	-	VARGRAPHIC	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DECIMAL	19	31	0	31	-	-	FLOAT	8	-	-
DOUBLE	-	-	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	FLOAT	8	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	J	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Anmerkung:

1. Der Teradata-Datentyp VARGRAPHIC kann nur die angegebene Länge (1 bis 32000) eines Datentyps DBCLOB enthalten.

Standardmäßige regressive Datentypzuordnungen für Unicode

Bestimmte Datenquellen unterstützen progressive und regressive Datentypzuordnungen für Unicode-Datenbanken.

Standardmäßige progressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei JDBC-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 115. Standardmäßige progressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen

UTF-8	JDBC	
Datentyp	Datentyp	Länge
CHAR	CHAR	1 bis 254 Byte
VARCHAR	VARCHAR	1 bis 32672 Byte
CLOB	CLOB	-
GRAPHIC	NCHAR	1 bis 127 Zeichen
VARGRAPHIC	NVARCHAR	1 bis 16336 Zeichen
DBCLOB	NCLOB	-

Standardmäßige regressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei JDBC-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 116. Standardmäßige regressive Datentypzuordnungen für Unicode bei JDBC-Datenquellen

UTF-8		JDBC
Datentyp	Länge	Datentyp
CHAR	1 bis 254 Byte	CHAR
VARCHAR	1 bis 32672 Byte	VARCHAR
CLOB	1 bis 2147483647 Byte	CLOB
GRAPHIC	1 bis 127 Zeichen	NCHAR
VARGRAPHIC	1 bis 16336 Zeichen	NVARCHAR
DBCLOB	1 bis 1073741823 Zeichen	NCLOB

Standardmäßige progressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Microsoft SQL Server-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 117. Standardmäßige progressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen

UTF-8	Microsoft SQL Server	
Datentyp	Datentyp	Länge
CHAR	CHAR	1 bis 254 Byte
VARCHAR	CHAR	255 bis 8000 Byte
	VARCHAR	1 bis 8000 Byte
CLOB	TEXT	-
GRAPHIC	NCHAR	1 bis 127 Zeichen
VARGRAPHIC	NCHAR	128 bis 16336 Zeichen
	NVARCHAR	1 bis 16336 Zeichen
DBCLOB	NTEXT	-

Standardmäßige regressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Microsoft SQL Server-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 118. Standardmäßige regressive Datentypzuordnungen für Unicode bei Microsoft SQL Server-Datenquellen

UTF-8		Microsoft SQL Server
Datentyp	Länge	Datentyp
CHAR	1 bis 254 Byte	CHAR
VARCHAR	1 bis 32672 Byte	VARCHAR
CLOB	1 bis 2147483647 Byte	TEXT
GRAPHIC	1 bis 127 Zeichen	NCHAR
VARGRAPHIC	1 bis 16336 Zeichen	NVARCHAR
DBCLOB	1 bis 1073741823 Zeichen	NTEXT

Standardmäßige progressive Datentypzuordnungen für Unicode bei NET8-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei NET8-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 119. Standardmäßige progressive Datentypzuordnungen für Unicode bei NET8-Datenquellen

UTF-8		Oracle
Datentyp	Datentyp	Länge
CHAR	CHAR	1 bis 254 Byte
VARCHAR	CHAR	255 bis 2000 Byte
	VARCHAR2	1 bis 4000 Byte
DBCLOB	NCLOB	
GRAPHIC	NCHAR	1 bis 127 Zeichen
VARGRAPHIC	NCHAR	128 bis 1000 Zeichen
	NVARCHAR2	1 bis 2000 Zeichen

Standardmäßige regressive Datentypzuordnungen für Unicode bei NET8-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei NET8-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 120. Standardmäßige regressive Datentypzuordnungen für Unicode bei NET8-Datenquellen

UTF-8		Oracle
Datentyp	Länge	Datentyp
CHAR	1 bis 254 Byte	CHAR
VARCHAR	1 bis 4000 Byte	VARCHAR2
CLOB	1 bis 2147483647 Byte	CLOB
GRAPHIC	1 bis 127 Zeichen	NCHAR
VARGRAPHIC	1 bis 2000 Zeichen	NVARCHAR2
DBCLOB	1 bis 1073741823 Zeichen	NCLOB

Standardmäßige progressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei ODBC-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 121. Standardmäßige progressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen

UTF-8	ODBC	
Datentyp	Datentyp	Länge
CHAR	SQL_CHAR	1 bis 254 Byte
VARCHAR	SQL_CHAR	255 bis 32672 Byte
	SQL_VARCHAR	1 bis 32672 Byte
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	1 bis 127 Zeichen
VARGRAPHIC	SQL_WCHAR	128 bis 16336 Zeichen
	SQL_WVARCHAR	1 bis 16336 Zeichen
DBCLOB	SQL_WLONGVARCHAR	-

Standardmäßige regressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei ODBC-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 122. Standardmäßige regressive Datentypzuordnungen für Unicode bei ODBC-Datenquellen

UTF-8	ODBC	
Datentyp	Länge	Datentyp
CHAR	1 bis 254 Byte	SQL_CHAR
VARCHAR	1 bis 32672 Byte	SQL_VARCHAR
CLOB	1 bis 2147483647 Byte	SQL_LONGVARCHAR
GRAPHIC	1 bis 127 Zeichen	SQL_WCHAR
VARGRAPHIC	1 bis 16336 Zeichen	SQL_WVARCHAR
DBCLOB	1 bis 1073741823 Zeichen	SQL_WLONGVARCHAR

Standardmäßige progressive Datentypzuordnungen für Unicode bei Sybase-Datenquellen

In der folgenden Tabelle sind die standardmäßigen progressiven Datentypzuordnungen bei Sybase-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 123. Standardmäßige progressive Datentypzuordnungen für Unicode bei Sybase CTLIB-Datenquellen

UTF-8	Sybase	
Datentyp	Datentyp	Länge
CHAR	char	1 bis 254 Byte
	nchar	1 bis 127 Zeichen
VARCHAR	char	255 bis 32672 Byte
	varchar	1 bis 32672 Byte
	nchar	128 bis 16336 Zeichen
	nvarchar	1 bis 16336 Zeichen
CLOB	text	
GRAPHIC	unichar	1 bis 127 Zeichen
VARGRAPHIC	unichar	128 bis 16336 Zeichen
	univarchar	1 bis 16336 Zeichen

Standardmäßige regressive Datentypzuordnungen für Unicode bei Sybase-Datenquellen

In der folgenden Tabelle sind die standardmäßigen regressiven Datentypzuordnungen bei Sybase-Datenquellen aufgeführt, die gültig sind, wenn es sich bei der föderierten Datenbank um eine Unicode-Datenbank handelt.

Tabelle 124. Standardmäßige regressive Datentypzuordnungen für Unicode bei Sybase CTLIB-Datenquellen

UTF-8	Sybase	
Datentyp	Länge	Datentyp
CHAR	1 bis 254 Byte	char
VARCHAR	1 bis 32672 Byte	varchar
CLOB	1 bis 2147483647 Byte	text
GRAPHIC	1 bis 127 Zeichen	unichar
VARGRAPHIC	1 bis 16336 Zeichen	univarchar

Datentypen, die für nicht relationale Datenquellen unterstützt werden

Bei den meisten nicht relationalen Datenquellen müssen Sie bei der Erstellung der Kurznamen für den Zugriff auf die Datenquelle die Spalteninformationen einschließlich des Datentyps angeben.

Einige der nicht relationalen Wrapper erstellen alle Spalten, die für den Zugriff auf eine Datenquelle erforderlich sind. Diese werden als *feste Spalten* bezeichnet. Bei anderen Wrappern können Sie einige oder alle Datentypen für die Spalten in der Anweisung CREATE NICKNAME angeben.

In den folgenden Abschnitten sind die Wrapper, bei denen Sie die Datentypen angeben können, sowie die vom jeweiligen Wrapper unterstützten Datentypen aufgeführt.

Vom BioRS-Wrapper unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom BioRS-Wrapper unterstützt werden.

Tabelle 125. BioRS-Datentypen mit Zuordnung zu DB2-Datentypen

BioRS-Datentypen	DB2-Datentyp
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

Die maximal zulässige Länge für den Datentyp CLOB beträgt 5 Megabyte.

Vom Excel-Wrapper unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom Excel-Wrapper unterstützt werden.

Tabelle 126. Excel-Datentypen mit Zuordnung zu DB2-Datentypen

Excel-Datentypen	DB2-Datentyp
character	CHARACTER
date	DATE
number	DOUBLE
number	FLOAT
integer	INTEGER
character	VARCHAR

Vom Script-Wrapper unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom Script-Wrapper unterstützt werden.

Tabelle 127. Script-Datentypen mit Zuordnung zu DB2-Datentypen

XML-Datentypen	DB2-Datentyp
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (maximale Länge = 5 Megabyte)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT

Tabelle 127. Script-Datentypen mit Zuordnung zu DB2-Datentypen (Forts.)

XML-Datentypen	DB2-Datentyp
character	VARCHAR
character	VARCHAR FOR BIT DATA

Vom Wrapper für Dateien mit Tabellenstruktur unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom Wrapper für Dateien mit Tabellenstruktur unterstützt werden.

Tabelle 128. Datentypen von Dateien mit Tabellenstruktur mit Zuordnung zu DB2-Datentypen

Datentypen von Dateien mit Tabellenstruktur	DB2-Datentyp
character	CHARACTER
character	CLOB (maximale Länge = 5 Megabyte)
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR

Vom Web-Services-Wrapper unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom Web-Services-Wrapper unterstützt werden. Der Web-Services-Wrapper verwendet XML-Datentypen.

Tabelle 129. XML-Datentypen für den Web-Services-Wrapper mit Zuordnung zu DB2-Datentypen

XML-Datentypen	DB2-Datentyp
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (maximale Länge = 5 Megabyte)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA

Vom XML-Wrapper unterstützte Datentypen

Die folgende Tabelle enthält die DB2-Datentypen, die vom XML-Wrapper unterstützt werden.

Tabelle 130. XML-Datentypen für den XML-Wrapper mit Zuordnung zu DB2-Datentypen

XML-Datentypen	DB2-Datentyp
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (maximale Länge = 5 Megabyte)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA
XML-Dokument	XML

Produktdokumentation

Die Produktdokumentation steht in einer Reihe unterschiedlicher Formate zur Verfügung und kann über verschiedene Zugriffsmöglichkeiten abgerufen werden, zum Beispiel in Hilfetexten, die direkt über die Produktschnittstelle geöffnet werden, in einem die gesamte Suite umfassenden Information Center und in PDF-Büchern.

Sie können IBM Veröffentlichungen auch in Hardcopyformat online oder über den zuständigen IBM Ansprechpartner bestellen.

Wenn Sie Veröffentlichungen online bestellen möchten, rufen Sie das IBM Publications Center unter www.ibm.com/shop/publications/order auf.

Kommentare zur Dokumentation können Sie uns wie folgt zukommen lassen:

- Über das Onlineformular: www.ibm.com/software/data/rcf/.
- Per E-Mail: comments@us.ibm.com.

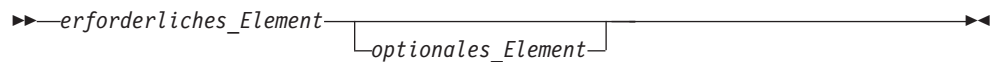
Syntaxdiagramme lesen

Die folgenden Regeln gelten für die Syntaxdiagramme, die in den vorliegenden Informationen verwendet werden:

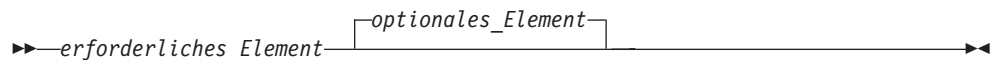
- Lesen Sie die Syntaxdiagramme von links nach rechts und von oben nach unten und folgen Sie dabei der Linie. Die folgenden Konventionen werden verwendet:
 - Das Symbol >>--- kennzeichnet den Anfang eines Syntaxdiagramms.
 - Das Symbol ---> gibt an, dass ein Syntaxdiagramm in der nächsten Zeile fortgesetzt wird.
 - Das Symbol >--- gibt an, dass ein Syntaxdiagramm von der vorhergehenden Zeile fortgesetzt wird.
 - Das Symbol --->< kennzeichnet das Ende eines Syntaxdiagramms.
- Erforderliche Elemente werden auf der horizontalen Linie (dem Hauptpfad) angezeigt.



- Optionale Elemente werden unterhalb des Hauptpfads angezeigt.

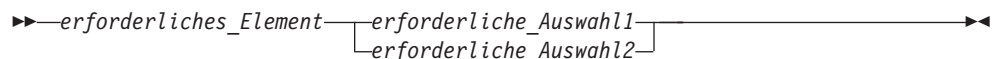


Wenn ein optionales Element oberhalb des Hauptpfads angezeigt wird, hat es keine Auswirkungen auf die Ausführung des Syntaxelements und wird nur zur besseren Lesbarkeit verwendet.

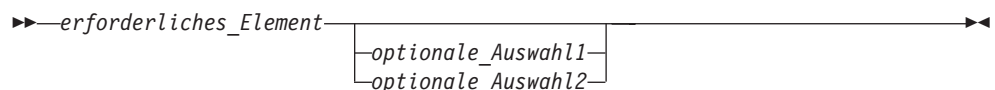


- Wenn Sie aus zwei oder mehr Elementen auswählen können, werden diese vertikal in einem Stapel angezeigt.

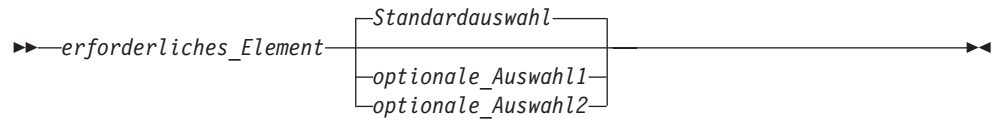
Wenn eines dieser Elemente ausgewählt werden muss, steht ein Element des Stapels auf dem Hauptpfad.



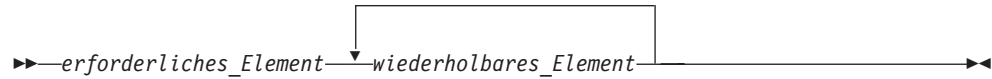
Wenn die Auswahl eines der Elemente optional ist, wird der gesamte Stapel unterhalb des Hauptpfads angezeigt.



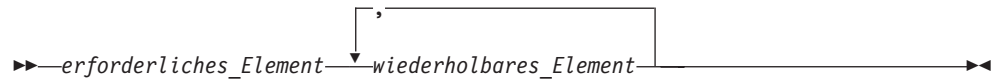
Wenn es sich bei einem der Elemente um den Standardwert handelt, wird es oberhalb des Hauptpfads angezeigt, während die übrigen Auswahlmöglichkeiten unterhalb des Hauptpfads stehen.



- Ein Pfeil zurück nach links, der oberhalb des Hauptpfads angezeigt wird, gibt ein wiederholbares Element an.

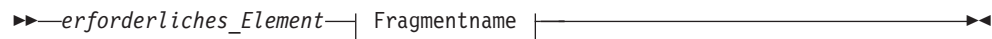


Enthält der Wiederholungspfeil ein Komma, müssen wiederholte Elemente durch ein Komma getrennt werden.

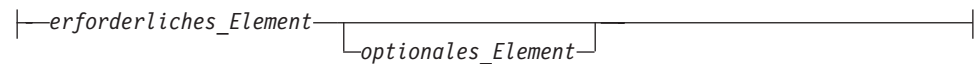


Ein Wiederholungspfeil oberhalb eines Stapels gibt an, dass die Elemente des Stapels wiederholt werden können.

- In manchen Fällen muss ein Diagramm in Fragmente aufgeteilt werden. Das Syntaxfragment wird separat vom eigentlichen Syntaxdiagramm angezeigt, der Inhalt des Fragments muss jedoch so gelesen werden, als sei er Teil des Hauptpfads des Diagramms.



Fragmentname:



- Schlüsselwörter und die zugehörigen Mindestabkürzungen, falls zutreffend, werden in Großbuchstaben angezeigt. Sie müssen exakt in der angezeigten Schreibweise eingegeben werden.
- Variablen werden kursiv in gemischter Groß-/Kleinschreibung dargestellt (zum Beispiel Name_der_Spalte). Sie stellen Namen oder Werte dar, die der Benutzer angibt.
- Schlüsselwörter und Parameter müssen durch mindestens ein Leerzeichen voneinander getrennt werden, falls im Diagramm keine entsprechende Interpunktion als Trennzeichen angegeben ist.
- Geben Sie Interpunktionszeichen, runde Klammern, arithmetische Operatoren und andere Symbole genau so ein wie im Diagramm dargestellt.
- Fußnoten werden durch eine Zahl in runden Klammern angegeben, zum Beispiel (1).

Dokumentation im behindertengerechten Format

Zur Darstellung der Dokumentation wird das Format XHTML verwendet, das mit den meisten Web-Browsern geöffnet werden kann.

XHTML ermöglicht das Aufrufen der Dokumentation mit den Anzeigeeinstellungen, die Sie in Ihrem Browser definiert haben. Darüber hinaus ist der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen möglich.

Syntaxdiagramme stehen in der Schreibweise mit Trennzeichen zur Verfügung. Dieses Format ist nur dann verfügbar, wenn Sie mithilfe eines Sprachausgabeprogramms auf die Onlinedokumentation zugreifen.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in dieser Dokumentation beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Die Musterprogramme werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung zur Verfügung gestellt. IBM haftet nicht für Schäden, die durch Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corporation abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Marken

IBM Marken und bestimmte Nicht-IBM Marken sind bei ihrem ersten Vorkommen in diesen Informationen mit dem entsprechenden Symbol gekennzeichnet.

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corp. in den USA und/oder anderen Ländern. Weitere Unternehmens-, Produkt- oder Servicenamen können Marken von IBM oder anderer Hersteller sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Die folgenden Namen sind Marken oder eingetragene Marken anderer Unternehmen:

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

IT Infrastructure Library ist eine eingetragene Marke der Central Computer and Telecommunications Agency. Die Central Computer and Telecommunications Agency ist nunmehr in das Office of Government Commerce eingegliedert worden.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

ITIL ist als eingetragene Marke und eingetragene Gemeinschaftsmarke des Office of Government Commerce beim US Patent and Trademark Office registriert.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Cell Broadband Engine wird unter Lizenz verwendet und ist eine Marke der Sony Computer Entertainment, Inc. in den USA und/oder anderen Ländern.

Java und alle auf Java basierenden Marken sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicennamen können Marken anderer Hersteller sein.

Index

A

Abfragen
asynchrone Verarbeitung 267
Cachetabellen, weiterleiten 299
Fragmente 11
weiterleiten, Cachetabellen 299
Abfrageoptimierung
Beschreibung 11
Aktivieren
föderiertes zweiphasiges Com-
mit 129
Aktualisierung auf mehreren Systemen,
föderiert
siehe föderiertes zweiphasiges Com-
mit 117
Aktualisierungen
an großen Objekten (LOBs) 145
Berechtigungen 143
Beschreibung 114
Einschränkungen 144
fern 114
lokal 114
referenzielle Integrität 144
Aktualisierungsfunktion für Kurznamen-
statistiken
unterstützte Datenquellen 359
ALTER NICKNAME, Anweisung
Beispiel
lokaler Datentyp 56
ALTER WRAPPER, Anweisung 24
Ändern 24
LONG-Datentypen 59
Anforderungen 131
Anforderungen an Datenquellen für föde-
riertes zweiphasiges Commit 131
Anweisung CREATE PROCEDURE
(Quellenfunktion)
föderierte Prozeduren 90
Anwendungen
Kurznamen in 209
Anwendungsprogramme 63
Anwendungssicherungspunkte 152
unterstützte Datenquellen 359
Architektur
föderiertes zweiphasiges Commit 118
Asynchrone Verarbeitung
aktivieren 274
Beispiele 267
Beschreibung 267
Einschränkungen 276
Optimierung 276
Asynchronitätsoptimierung
unterstützte Datenquellen 359
Atomizität 125
in Anweisungen beibehalten 145

B

Befehlszeilenprozessor (CLP)
Funktionen föderierter Datenban-
ken 7
Befehlszentrale
Einsatz für föderierte Systeme 7
Beispiele 150
Befehl IMPORT 156
föderiertes zweiphasiges Com-
mit 121
Benutzerdefinierte Funktionen (User-Defi-
ned Functions, UDFs) 18
in Anwendungen in föderierten Systeme-
men 63
Transaktionsunterstützung für 114
Benutzerdefinierte Typen (User-Defined
Types, UDTs)
nicht unterstützte Datentypen 17
Benutzerzuordnungen
Beispiel-Plug-ins 327
Beschreibung 14
externe Repositories 327
speichern 14, 327
Benutzerzuordnungsoptionen
BioRS 361
DB2-Datenbank 365
Informix 374
JDBC 380
Microsoft SQL Server 387
ODBC 393
Oracle 399
Script 405
Sybase 410
Teradata 417
Web-Services 424
XML 432
Benutzerzuordnungsrepository
unterstützte Datenquellen 359
Beschreibung 357
BioRS
Benutzerzuordnungsoptionen 361
Kurznamenoptionen 361
Serveroptionen 361
Spaltenoptionen 361
unterstützte Datentypen 478
unterstützte föderierte Funktio-
nen 359
Wrapperoptionen 361

C

Cachetabelle, Assistent 296
Assistenten 296
Cachetabellen 293
aktivieren 300
ändern 298
Anzeigen von Einstellungen 298
Archivprotokollierung 296
Beschreibung 295
Cachetabelle, Assistent 296

Cachetabellen (*Forts.*)
Datenquellen 296
erstellen 296
inaktivieren 300
löschen 301
Materialized Query Tables
(MQTs) 298
modifizieren 298
Replikation 295
unterstützte Datenquellen 359
Voraussetzungen 296
Weiterleiten von Abfragen 299
Zeitpläne, Replikation 295
CLP (Befehlszeilenprozessor)
Funktionen föderierter Datenban-
ken 7
Codepages
Beschreibung 19
COLLATING_SEQUENCE, Serveroption
Auswirkungen auf globale Optimie-
rung 279
Auswirkungen auf Pushdown-Mög-
lichkeiten 237
Beispiel 19
COMM_RATE, Serveroption
Auswirkungen auf globale Optimie-
rung 279
CPU_RATIO, Serveroption
Auswirkungen auf globale Optimie-
rung 279
CREATE FUNCTION, Anweisung
(Quellenfunktion oder Schablone) 61,
68
CREATE FUNCTION MAPPING, Anwei-
sung 61, 64
CREATE INDEX, Anweisung 18, 71
CREATE NICKNAME, Anweisung 49
CREATE SERVER, Anweisung 5
CREATE TYPE MAPPING, Anwei-
sung 48, 51, 52, 53
CURRENT FEDERATED ASYNCHRONY,
Sonderregister 274
CURRENT IMPLICIT XMLPARSE
OPTION, Sonderregister 195
D
DATALINK, Datentyp
nicht unterstützt 17
Dateien mit Tabellenstruktur
gültige Objekte für Kurznamen 16
Kurznamenoptionen 421
Serveroptionen 421
Spaltenoptionen 421
unterstützte Datentypen 478
unterstützte föderierte Funktio-
nen 359
unterstützte Versionen 2
Wrapperoptionen 421
Datenimport
unterstützte Datenquellen 359

- Datenquelle
 - Semantik für VARCHAR2 238
- Datenquellen 6, 11
 - Anforderungen für föderiertes zweiphasiges Commit 131
 - Beschreibung 5
 - E/A-Geschwindigkeit und Leistung 279
 - ferne Planhinweise und Leistung 279
 - gültige Servertypen 447
 - Kurznamen erstellen 165
 - Optionen 361
 - Prozessorgeschwindigkeit und Leistung 279
 - Sortierfolge und Leistung 279
 - Übertragungsgeschwindigkeit und Leistung 279
 - unterstützte Funktionen 359
- Datenquellenobjekte 164
 - Beschreibung 15
 - gültige Objekttypen 16
- Datentypen 106
 - Auswirkungen auf Pushdown-Analyse 244
 - für nicht relationale Datenquellen 478
 - nicht unterstützt 17
 - Semantik für VARCHAR2 238
 - TIMESTAMP 55
- Datentypen für große Objekte (LOB)
 - Aktualisierungsoperationen 145
 - Einschränkungen 222
 - Leistungsaspekte 223
 - Querverweise 222
- Datentypzuordnungen
 - Auswirkungen auf Pushdown-Analyse 237
 - Beschreibung 17
 - erstellen 47
 - für ein bestimmtes Datenquellenobjekt 56
 - für einen bestimmten Datenquellenotyp 51
 - für einen bestimmten Server 53
 - für einen bestimmten Servertyp und eine bestimmte Serverversion 52
 - in einem föderierten System 47
 - nicht relationale 49
 - nicht unterstützte Datentypen 47
 - progressiv 449
 - Beschreibung 49
 - regressiv 464
 - Beschreibung 49
 - Situationen, die neue Zuordnungen erfordern 48
 - Syntax 49
 - Umsetzung 54
- Datenzugriff
 - föderierte Sichten 167
- Datenzugriff mit Kurznamen 160
- DB2
 - Toolskatalog 184
- DB2 Database für Linux, UNIX und Windows-Datenquellen
 - standardmäßige progressive Datentypzuordnungen 449
- DB2 Database für Linux, UNIX und Windows-Datenquellen (*Forts.*)
 - standardmäßige regressive Datentypzuordnungen 465
- DB2-Datenbank
 - Benutzerzuordnungsoptionen 365
 - Kurznamenoptionen 365
 - Serveroptionen 365
 - Spaltenoptionen 365
 - unterstützte föderierte Funktionen 359
 - Wrapperoptionen 365
- DB2-Datenquellen
 - föderierte Prozeduren 81
- DB2 für Linux, UNIX und Windows
 - föderierte LOB-Unterstützung 221
 - gültige Objekte für Kurznamen 16
 - standardmäßige progressive Typzuordnungen 449
 - standardmäßige regressive Typzuordnungen 464
 - unterstützte Versionen 2
- DB2 für System i
 - föderierte LOB-Unterstützung 221
 - gültige Objekte für Kurznamen 16
 - standardmäßige progressive Typzuordnungen 449
 - standardmäßige regressive Typzuordnungen 464
 - unterstützte Versionen 2
- DB2 für System i-Datenquellen
 - standardmäßige progressive Datentypzuordnungen 450
 - standardmäßige regressive Datentypzuordnungen 466
- DB2 für VM und VSE
 - föderierte LOB-Unterstützung 221
 - gültige Objekte für Kurznamen 16
 - standardmäßige progressive Typzuordnungen 449
 - standardmäßige regressive Typzuordnungen 464
 - unterstützte Versionen 2
- DB2 für VM und VSE-Datenquellen
 - standardmäßige progressive Datentypzuordnungen 451
 - standardmäßige regressive Datentypzuordnungen 467
- DB2 für z/OS
 - föderierte LOB-Unterstützung 221
 - gültige Objekte für Kurznamen 16
 - unterstützte Versionen 2
- DB2 für z/OS-Datenquellen
 - standardmäßige progressive Datentypzuordnungen 452
 - standardmäßige regressive Datentypzuordnungen 468
- DB2 für z/OS und OS/390
 - standardmäßige progressive Typzuordnungen 449
 - standardmäßige regressive Typzuordnungen 464
- DB2_MAX_ASYNC_REQUESTS_PER_QUERY, Serveroption 274
- DB2_MAXIMAL_PUSHDOWN, Serveroption
 - Auswirkungen auf Pushdown-Möglichkeiten 237
 - Entscheidungen für Pushdown-Analyse 246
- DB2-Steuerzentrale 8
- DB2_UM_PLUGIN, Option 339
- DB2_UM_PLUGIN_LANG, Option 339
- db2exfmt, Tool
 - Zugriffspläne anzeigen 246, 285
- db2expln, Tool
 - Zugriffspläne anzeigen 246, 285
- DB2FEDGENTF, Befehl
 - Beispiele 98
 - Syntax 98
- DDL
 - föderiertes zweiphasiges Commit 125
- DELETE, Anweisung 144
 - Entscheidungen für die Auswertung von Zugriffsplänen 248
- Diagnoseanzeiger von föderierten Systemen
 - unterstützte Datenquellen 359
- Diagnosezentrale
 - Diagnoseanzeiger 201
 - Diagnosemomentaufnahme 203
- DISABLE, Funktionszuordnungsoption
 - gültige Einstellungen 445
- Distributed Relational Database Architecture (DRDA)
 - für föderiertes zweiphasiges Commit konfigurieren 131
- Dokumentation
 - Eingabehilfen 483, 487
- DUOW
 - siehe verteilte Arbeitseinheit 138
- Durchgriffsfunktion
 - Beschreibung 13
 - Einschränkungen 13
 - LOB-Unterstützung 222
 - Transaktionsunterstützung für 114
- Durchgriffssitzungen 167
 - unterstützte Datenquellen 359
- dynexpln, Tool
 - Zugriffspläne anzeigen 246, 285

E

- Eingabehilfen 483, 487
- Einphasiges Commit
 - Definition 113
- Excel
 - Kurznamenoptionen 373
 - Serveroptionen 373
 - unterstützte föderierte Funktionen 359
 - Wrapperoptionen 373
- Excel-Dateien
 - gültige Objekte für Kurznamen 16
 - unterstützte Datentypen 478
 - unterstützte Versionen 2
- EXPLAIN-Funktion 254
- EXPLAIN-Tools 264

EXPORT, Befehl
Verwendung mit Kurznamen 155,
157

F

FEDERATED_ASYNC, Konfigurationsparameter 274
FEDERATED_ASYNC, Bindeoption 274
Fehlerbehebung
föderiertes zweiphasiges Commit 136, 140
Fehlerbehebung bei Problemen mit föderiertem zweiphasigem Commit 140
Fehlertoleranz
aktivieren 198
Beispiel 199
Beschreibung 197
Datenquellenunterstützung 200
Einschränkungen 200
unterstützte Datenquellen 359
Ferne Aktualisierungen 114
Ferne Kataloginformationen 10
Ferne Objekte, Kurznamen 159
Föderierte Datenbank
lokale Objekte 159
Wrapper 6
Wrappermodule 6
Föderierte Datenbanken
Beschreibung 6
Systemkatalog 10
Föderierte Prozeduren 19
Anweisung CREATE PROCEDURE (Quellenfunktion) 90
aufrufen 93
Datenquellenunterstützung 81
DB2 81
erstellen 90
Fehlerbehebung 100
Katalogsichten 93
löschen 94
Microsoft SQL Server 83
Oracle 84
Parameter 93
Sybase 88
Übersicht 79
unterstützte Datenquellen 359
Verknüpfen von Ergebnismengen 95
Zugriffsrechte, erteilen 91
Föderierte Sichten
Datenzugriff 167
erstellen 169
Föderierte Statistikdaten
aktualisieren 178
Föderierte Systeme
Isolation auf Anweisungsebene 218
Isolation auf Verbindungsebene 219
Isolationsstufen 217
Übersicht 1
Föderierter gesicherter Kontext
Oracle-Proxyauthentifizierung 357
unterstützte Datenquellen 359
Föderiertes zweiphasiges Commit
aktivieren 129
Anforderungen an Datenquellen 131
Architektur 118

Föderiertes zweiphasiges Commit (*Forts.*)
Atomizität 125
Beispiele 121
DDL 125
Fehlerbehebung 136
Konfigurationen 125
konfigurieren 129
Leistung 141
Leistung verbessern 142
Planung 117
transparente DDL 125
föderiertes zweiphasiges Commit 125
Übersicht 117
zulässige Operationen 125
Funktionen
Plug-ins für Benutzerzuordnungen 327
Funktionsschablonen
Beschreibung 68
Vergleichselemente 251
Funktionszuordnungen
Auswirkungen auf Pushdown-Analyse 237
Beschreibung 18, 61
erstellen 64
Optionen 445
Standardzuordnungen 61
Zuordnung zu benutzerdefinierten Funktionen (UDFs) 64

G

Gemischte Parallelität
Datenquellen in föderierten Datenbanken
Datenverarbeitung 263
Übersicht 253
Zugriffsplan 264
Gesicherte Verbindungen föderierter Systeme
Anforderungen an Benutzerzuordnungen 310, 314
Anwendungsdesign 309
APIs 309, 313
Beispielanwendung 313
Benutzer-ID-Anforderungen 310, 314, 318
Benutzerzuordnungen 318, 322
Beschreibung 305, 307
End-to-End-Verbindungen 307, 310, 314
erstellen
gesicherter Kontext 310, 314
Gesicherter Kontext 318
explizit 305
gesicherte, abgehende Verbindungen 307, 318
gesicherte Verbindungen föderierter Systeme
eingehende Verbindungen 310
Proxybenutzer 318
Gesicherte Verbindungen föderierter Systeme
abgehende Verbindungen 310

Gesicherte Verbindungen föderierter Systeme (*Forts.*)
implizit
Gesicherte Verbindungen föderierter Systeme 305
Kennwortanforderungen 310, 314, 318
konfigurieren 310, 314, 318
Proxybenutzer des föderierten Systems
Beschreibung 318
Serverdefinitionen 318
Szenarios 310, 313, 314, 318
Typen 307
Vorteile 306
Gesicherter Kontext
Beschreibung 305
Gespeicherte Prozeduren
Statistikdaten zu Kurznamen 186
Globale Optimierung 251
Auswirkungen von Servermerkmalen 279
Beschreibung 279
Globaler Katalog 47
Beschreibung 10
Sichten mit Informationen föderierter Datenbanken 441
Statistikdaten aktualisieren 251
GROUP BY, Operator
Entscheidungen für die Auswertung von Zugriffsplänen 248
Entscheidungen für die Optimierung von Zugriffsplänen 286
Gruppenoperatoren
Entscheidungen für die Auswertung von Zugriffsplänen 248

H

Heuristische Operationen
unbestätigte Transaktion auflösen föderierte Systeme 137
HIGH2KEY-Statistik 184
HTTP-Proxy
unterstützte Datenquellen 359

I

IMPORT, Befehl
Beispiele mit Kurznamen 156
Einschränkungen bei Kurznamen 155
Verwendung mit Kurznamen 155
Indexspezifikationen
Beschreibung 18
in föderierten Systemen 71
Informative Integrationsbedingungen
Kurznamen 173, 175
informative Integrationsbedingungen für Kurznamen 174
Informix
Benutzerzuordnungsoptionen 374
föderierte LOB-Unterstützung 221
für föderiertes zweiphasiges Commit konfigurieren 133
gültige Objekte für Kurznamen 16

Informix (*Forts.*)
 Serveroptionen 374
 Spaltenoptionen 374
 standardmäßige progressive Typzuordnungen 449
 standardmäßige regressive Typzuordnungen 464
 unterstützte föderierte Funktionen 359
 unterstützte Versionen 2
 Wrapperoptionen 374
 Informix-Datenquellen
 standardmäßige progressive Datentypzuordnungen 453, 468
 InfoSphere Data Architect 9
 INSERT, Anweisung 143, 144
 Entscheidungen für die Auswertung von Zugriffsplänen 248
 Integrierte Funktionen 18
 IO_RATIO, Serveroption
 Auswirkungen auf globale Optimierung 279
 Isolation auf Anweisungsebene
 föderierte Systeme 218
 unterstützte Datenquellen 217, 359
 Isolation auf Verbindungsebene
 föderierte Systeme 219
 unterstützte Datenquellen 359
 Isolationsstufen
 föderierte Systeme 217
 IUD_APP_SVPT_ENFORCE, Serveroption
 Beispiele 145

J

JDBC
 Benutzerzuordnungsoptionen 380
 föderierte LOB-Unterstützung 221
 gültige Objekte für Kurznamen 16
 Serveroptionen 380
 Spaltenoptionen 380
 unterstützte föderierte Funktionen 359
 unterstützte Versionen 2
 Wrapperoptionen 380
 JDBC-Datenquellen
 standardmäßige progressive Datentypzuordnung für Unicode 474
 standardmäßige progressive Datentypzuordnungen 454
 standardmäßige regressive Datentypzuordnungen 469
 standardmäßige regressive Datentypzuordnungen für Unicode 475
 Joins
 Entscheidungen für die Optimierung von Zugriffsplänen 286

K

Katalog
 siehe 'Globaler Katalog' 441
 Katalog, Tools 184
 Kennsatzbasierte Zugriffssteuerung (LBAC)
 Federation 293

Kennsatzbasierte Zugriffssteuerung (LBAC) (*Forts.*)
 föderiert 325
 MQT (Materialized Query Table) 293
 unterstützte Datenquellen 359
 Kompensation, Beschreibung 12
 Konfiguration
 föderiertes zweiphasiges Commit 129
 Kurzname
 Abrufmethoden 179
 Indexnamen 166
 Spaltennamen 166
 Kurznamen
 ändern
 Beispiel für lokalen Datentyp 56
 Beispiele für IMPORT-Befehl 156
 Beschreibung 15
 Cachetabellen 295
 Datenzugriff 160, 164
 Einschränkungen bei EXPORT-Befehl 157
 Einschränkungen bei IMPORT-Befehl 155
 erstellen 165
 gültige Datenquellenobjekte 16
 informative Integrationsbedingungen 173, 174
 Informative Integrationsbedingungen für Kurznamen 173
 Integritätsbedingungen 144
 Kurzname für Kurzname 169
 lokale und ferne Objekte 159
 Oracle Label Security 178
 SQL-Anweisungen 161
 Statistikdaten 180, 183
 automatische Erfassung 188
 einzelner Kurzname 182
 mehrere Kurznamen 181
 Statistikdaten aktualisieren 185
 Statusaktualisierungen
 DB2-Steuerzentrale 185
 Trigger 160
 Verwendung bei EXPORT-Befehl 155
 Verwendung bei IMPORT-Befehl 155
 WITH HOLD-Syntax 159
 XML-Daten 191
 XQuery-Sprache 191
 Zugriff auf Datenquellen 209
 zwischenspeichern 325
 Kurznamenoptionen
 BioRS 361
 Dateien mit Tabellenstruktur 421
 DB2-Datenbank 365
 Script 405
 Web-Services 424
 XML 432
 Kurznamenoptionen
 Beschreibung 16

L

LBAC
 Federation 293
 MQT (Materialized Query Table) 293
 LDAP-Server
 Benutzerzuordnungen 327

Leistung 173, 236, 246, 261, 263
 asynchrone Abfrageverarbeitung 267
 CPU-Geschwindigkeit 279
 E/A-Geschwindigkeit 279
 ferne Planhinweise 279
 föderiert 175, 178, 183, 186, 233
 föderiertes zweiphasiges Commit 141
 verbessern 142
 siehe auch Optimierung 234
 Sortierfolge 279
 Sortierfolgen 237
 SQL-Unterschiede 237
 Übertragungsgeschwindigkeit 279
 LOB
 unterstützte Datenquellen, schreibgeschützt 359
 unterstützte Datenquellen für Schreib-/Lesezugriff 359
 LOB-Datentypen (Large Object)
 Aktualisierungsoperationen 145
 LOB-Datentypen (LOB = Large Object)
 Einschränkungen 222
 Querverweise 222
 Lokale Aktualisierungen 114
 Lokale Objekte 159
 Lokaler Katalog
 siehe 'Globaler Katalog' 10
 LONG-Datentypen 59
 LOW2KEY-Statistik 184

M

Managementsystem für verteilte Datenbanken 1
 Marken 491
 Materialized Query Tables (MQTs)
 Cachetabellen 295
 hinzufügen, zu Cachetabellen 298
 löschen, aus Cachetabellen 301
 Microsoft Excel
 siehe Excel-Dateien 2
 Microsoft SQL Server
 Benutzerzuordnungsoptionen 387
 föderierte LOB-Unterstützung 221
 für föderiertes zweiphasiges Commit konfigurieren 134
 gültige Objekte für Kurznamen 16
 Serveroptionen 387
 Spaltenoptionen 387
 standardmäßige progressive Datentypzuordnung für Unicode 475
 standardmäßige progressive Typzuordnungen 449
 standardmäßige regressive Typzuordnungen 464
 unterstützte föderierte Funktionen 359
 unterstützte Versionen 2
 Wrapperoptionen 387
 Microsoft SQL Server-Datenquellen
 föderierte Prozeduren 83
 standardmäßige progressive Datentypzuordnungen 456
 standardmäßige regressive Datentypzuordnungen 470

Microsoft SQL Server-Datenquellen
(*Forts.*)
standardmäßige regressive Datentyp-
zuordnungen für Unicode 476

Monitorschalter
föderiert 289

MQT (Materialized Query Table) 211,
292
Einschränkungen für Kurz-
namen 293
föderiert
Übersicht 291
für Kurznamen 244

MQTs (gespeicherte Abfragetabellen)
Einschränkungen für Kurz-
namen 293
Oracle Label Security (OLS) 293
föderiert
Übersicht 291
für Kurznamen 244

MQTs (Materialized Query Tables)
unterstützte Datenquellen 359

N

NET8-Datenquellen
standardmäßige progressive Datentyp-
zuordnungen für Unicode 476
standardmäßige regressive Datentyp-
zuordnungen für Unicode 476

Nicht relationale Datenquellen
Datentypzuordnungen angeben 17
unterstützte Datentypen 478

NUMERIC_STRING, Spaltenoption
Auswirkungen auf Pushdown-Mög-
lichkeiten 244

O

ODBC
Benutzerzuordnungsoptionen 393
föderierte LOB-Unterstützung 221
gültige Objekte für Kurznamen 16
Serveroptionen 393
Spaltenoptionen 393
standardmäßige progressive Typzu-
ordnungen 449
unterstützte föderierte Funktio-
nen 359
unterstützte Versionen 2
Wrapperoptionen 393

ODBC-Datenquellen
standardmäßige progressive Datentyp-
zuordnung für Unicode 477
standardmäßige progressive Datentyp-
zuordnungen 458
standardmäßige regressive Datentyp-
zuordnungen 471
standardmäßige regressive Datentyp-
zuordnungen für Unicode 477

OLE DB
unterstützte Versionen 2

OLS (Oracle Label Security)
Einschränkungen für Kurznamen
MQT (Materialized Query
Table) 293

Optim Data Studio 9

Optimierung
Abfrageverarbeitung 234
asynchrone Abfragen 275
Auswirkungen von Server-
merkmalen 279
föderiertes zweiphasiges Com-
mit 141
Leistung verbessern 142
Indexspezifikationen 279
Katalogstatistik 279
Kurznamenspaltenoptionen 244
MQTs (Materialized Query Tab-
les) 244
Serveroptionen 237
siehe auch Leistung 234
Sortierfolgen 237

Optimierungsprogramm
Beschreibung 11

Oracle 357
Benutzerzuordnungsoptionen 399
Fehlerbehebung bei Problemen mit
föderiertem zweiphasigem Com-
mit 140
föderierte LOB-Unterstützung 221
für föderiertes zweiphasiges Commit
konfigurieren 132
gültige Objekte für Kurznamen 16
leere Zeichenfolgen 151
Serveroptionen 399
Spaltenoptionen 399
standardmäßige progressive Typzu-
ordnungen 449
standardmäßige regressive Typzu-
ordnungen 464
unterstützte föderierte Funktio-
nen 359
Wrapperoptionen 399

Oracle-Datenquellen
föderierte Prozeduren 84
Sicherheit 357
überladene Prozeduren 86

Oracle Label Security
Beschreibung 357

Oracle Label Security (OLS)
Einschränkungen für Kurznamen
MQT (Materialized Query
Table) 293

Oracle NET8-Datenquellen
standardmäßige progressive Datentyp-
zuordnungen 459
standardmäßige regressive Datentyp-
zuordnungen 472

Oracle-Proxyauthentifizierung 357

ORDER BY, Operator
Entscheidungen für die Auswertung
von Zugriffsplänen 248

P

Parallelität 253, 261, 263
föderiert 253, 255

Partitionsinterne Parallelität 253
föderiert 253
Zugriffspläne für föderierte Syste-
me 254

Partitionsübergreifende Parallelität 253

Partitionsübergreifende Parallelität
(*Forts.*)
föderiert 255, 259, 261, 263

PLAN_HINTS, Serveroption
Auswirkungen auf globale Optimie-
rung 279

Planung
föderiertes zweiphasiges Commit 117

Plug-ins für Benutzerzuordnungen
DB2_UM_PLUGIN, Option 339
DB2_UM_PLUGIN_LANG, Opti-
on 339

Programmiersprache C
aktualisieren 340
Beispiel-Plug-in 340
Benennung 339
Build 338
Einschränkungen 329
entwickeln 334
Fehlerbehandlung 336
Funktion 'FSUMconnect' 333
Funktion 'FSUMdisconnect' 334
Funktion 'FSUMfetchUM' 333
Funktion 'FSUMPluginInit' 332
Funktion 'FSUMPluginTerm' 334
Funktionen 327
Funktionen deklarieren 335
Funktionsdeklarationen 335
Funktionszeiger 336
implementieren 339
Kopfdatendatei 330
testen 338
Übersicht 327
unterstützte Plattformen 329

Programmiersprache Java
Architektur 341
Beispieldateien 347, 350, 351
Dateien kompilieren 352
entwickeln 348, 349
implementieren 354
Klassen 343, 344, 345, 346
Konfigurationsdatei 353
testen 353
Zugriff konfigurieren 355
Zugriff auf Server mit föderierten
Datenbanken 339

Progressive Typzuordnungen
Beschreibung 49
Standardzuordnungen 449
Unicode
JDBC-Datenquellen 474
Microsoft SQL Server 475
NET8-Datenquellen 476
ODBC-Datenquellen 477
Sybase-Datenquellen 478

Proxy-Server
Beschreibung 303

Prozeduren
föderiert 79
föderiert, Ergebnismengen verknüp-
fen 95
föderiert, erstellen 90
föderiert, Fehlerbehebung 100
föderiert, löschen 94

Pushdown-Analyse
Auswirkungen von Abfrage-
merkmalen 246

Pushdown-Analyse (*Forts.*)
Auswirkungen von Kurznamen-
merkmalen 244
Auswirkungen von Server-
merkmalen 237
Beschreibung 11, 236
Vergleichselemente mit Funktions-
schablonen 251

R

Rechenpartitionsgruppen 261
Rechtliche Bemerkungen 489
Referenzielle Integrität 144
Regeln
Zuordnungssemantik in föderierten
Systemen 148
Regressive Typzuordnung
Unicode
NET8-Datenquellen 476
Regressive Typzuordnungen
Beschreibung 49
Standardzuordnungen 464
Unicode
JDBC-Datenquellen 475
Microsoft SQL Server-Daten-
quellen 476
ODBC-Datenquellen 477
Sybase-Datenquellen 478
REMOTE_NAME, Funktionszuordnungs-
option
gültige Einstellungen 445
RETURN DATA UNTIL, Klausel 198

S

Schreiboperationen
siehe Aktualisierungen 114
Script
Benutzerzuordnungsoptionen 405
Kurznamenoptionen 405
Serveroptionen 405
Spaltenoptionen 405
unterstützte föderierte Funktio-
nen 359
unterstützte Versionen 2
Wrapperoptionen 405
Secure Socket Layer (SSL)
unterstützte Datenquellen 359
Server mit föderierten Datenbanken 5
Beschreibung 5
Semantik für VARCHAR2 238
Serverdefinitionen
Beschreibung 13
Serveroptionen
Asynchronität 274
Auswirkungen auf globale Optimie-
rung 279
Auswirkungen auf Pushdown-Analy-
se 237
Beschreibung 13
BioRS 361
Dateien mit Tabellenstruktur 421
DB2-Datenbank 365
Excel 373
Informix 374

Serveroptionen (*Forts.*)

JDBC 380
Microsoft SQL Server 387
ODBC 393
Oracle 399
Script 405
Sybase 410
temporäre 13
Teradata 417
Web-Services 424
XML 432

Servertypen
gültige föderierte Typen 447
SET SERVER OPTION, Anweisung
Option temporär festlegen 13
Sicherheit 357
Oracle 357

Sicherungspunkte
Datenquellen-APIs 145
in Federation 152
Snapshot Monitor 204
Fragmente von Abfragen für föde-
rierte Datenbanken 206
Kurznamen und Server 203
Kurznamen und Server föderierter
Datenbanken 201

SOCKS-Proxy
unterstützte Datenquellen 359

Sortierfolgen
Beschreibung 19
Planung 19
Übersicht 237

Sortierung 19

Spaltenoptionen
Auswirkungen auf Pushdown-Analy-
se 244
Beschreibung 16
BioRS 361
Dateien mit Tabellenstruktur 421
DB2-Datenbank 365
Informix 374
JDBC 380
Microsoft SQL Server 387
ODBC 393
Oracle 399
Script 405
Sybase 410
Teradata 417
Web-Services 424
XML 432

Sperrenanforderungsklausel 218

Sprachausgabeprogramme 483, 487

SQL-Anweisungen

Kurznamen 161

SQL-Compiler

Flussdiagramm einer Abfrage-
analyse 234
in einem föderierten System 11

SQL Explain

Zugriffspläne anzeigen 246, 285

SQL-Programmversion 167

Auswirkungen auf Pushdown-Analy-
se 237

Beschreibung 12

SQL/XML-Funktionen 191

Standardmäßige progressive Datentypzu-
ordnungen

Beispiel für Informix 462

Beispiel für Microsoft SQL Ser-
ver 463

Beispiel für Oracle 463

Beispiel für Sybase 464

Beispiel für Teradata 464

Beispiele 462

DB2 Database für Linux, UNIX und
Windows-Datenquellen 449

DB2 für System i-Datenquellen 450

DB2 für VM und VSE-Daten-
quellen 451

DB2 für z/OS-Datenquellen 452

Informix-Datenquellen 453, 468

JDBC-Datenquellen 454

Microsoft SQL Server-Daten-
quellen 456

ODBC-Datenquellen 458

Oracle NET8-Datenquellen 459

Sybase-Datenquellen 460

Teradata-Datenquellen 461

Standardmäßige regressive Datentypzu-
ordnungen

DB2 Database für Linux, UNIX und
Windows-Datenquellen 465

DB2 für System i-Datenquellen 466

DB2 für VM und VSE-Daten-
quellen 467

DB2 für z/OS-Datenquellen 468

JDBC-Datenquellen 469

Microsoft SQL Server-Daten-
quellen 470

ODBC-Datenquellen 471

Oracle NET8-Datenquellen 472

Sybase-Datenquellen 473

Teradata-Datenquellen 473

Statistikdaten

Abrufmethoden 179

automatische Erfassung 188

einzelner Kurzname 182

HIGH2KEY 184

Kurzname 178, 180, 183

LOW2KEY 184

mehrere Kurznamen 181

Statusaktualisierungen

Kurznamen 185

Steuerzentrale

Schnittstelle für föderierte Systeme 7

Stundenfeld 151

Subskriptionsgruppeneintrag

Cachetabellen 300

Sybase 140

Benutzerzuordnungsoptionen 410

föderierte LOB-Unterstützung 221

für föderiertes zweiphasiges Commit
konfigurieren 135

gültige Objekte für Kurznamen 16

Serveroptionen 410

Spaltenoptionen 410

standardmäßige progressive Typzu-
ordnungen 449

standardmäßige regressive Typzu-
ordnungen 464

unterstützte föderierte Funktio-
nen 359

Sybase (*Forts.*)
 unterstützte Versionen 2
 Wrapperoptionen 410
 Sybase-Datenquellen
 föderierte Prozeduren 88
 standardmäßige progressive Datentypzuordnung für Unicode 478
 standardmäßige progressive Datentypzuordnungen 460
 standardmäßige regressive Datentypzuordnungen 473
 standardmäßige regressive Datentypzuordnungen für Unicode 478
 SYSCAT, Katalogsichten 64, 441
 SYSPROC.FED_STATS, Tabelle 185
 SYSPROC.NNSTAT, gespeicherte Prozedur 186
 SYSSTAT, Katalogsichten 441
 Systemmonitorschalter
 föderiert 289

T

Teradata
 Benutzerzuordnungsoptionen 417
 föderierte LOB-Unterstützung 221
 gültige Objekte für Kurznamen 16
 Serveroptionen 417
 Spaltenoptionen 417
 standardmäßige progressive Typzuordnungen 449
 standardmäßige regressive Typzuordnungen 464
 unterstützte föderierte Funktionen 359
 Wrapperoptionen 417
 Teradata-Datenquellen
 standardmäßige progressive Datentypzuordnungen 461
 standardmäßige regressive Datentypzuordnungen 473
 TIMESTAMP 55
 Toolskatalog, DB2 184
 Transaktionen
 Aktualisierungen 114
 Übersicht 113
 Transparente DDL
 Längen von LOB-Spalten 106
 Transaktionsunterstützung für 114
 Trigger 160

U

überladene Prozeduren
 Föderierte Prozeduren 86
 Übersicht
 föderiertes zweiphasiges Commit 117
 Umsetzung 54
 Unbestätigte Transaktionen
 auflösen
 föderierte Systeme 137
 beheben 136
 DUOWs verfolgen 138
 resynchronisieren in föderierten Systemen 136

Unicode
 unterstützte Datenquellen 359
 Unstrukturierte Dateien
 siehe auch Dateien mit Tabellenstruktur 2
 UPDATE, Anweisung 144
 Entscheidungen für die Auswertung von Zugriffsplänen 248

V

VARCHAR_NO_TRAILING_BLANKS, Serveroption
 Auswirkungen auf Pushdown-Möglichkeiten 237
 VARCHAR_NO_TRAILING_BLANKS, Spaltenoption
 Auswirkungen auf Pushdown-Möglichkeiten 244
 VARCHAR2
 Semantik 238
 VARCHAR2-Kompatibilität 238
 Vergleichselemente
 Entscheidungen für die Auswertung von Zugriffsplänen 248
 mit Funktionsschablonen 251
 Verknüpfen von Ergebnismengen
 Befehl DB2FEDGENTE, Beispiele 98
 Befehlssyntax von
 DB2FEDGENTF 98
 Verschachtelte Tabellenausdrücke
 Fehlertoleranz 197
 Verschlüsselung
 Beschreibung 303
 Verteilte Arbeitseinheit
 in Datenquellen verfolgen 138
 Visual Explain
 Zugriffspläne anzeigen 246, 285

W

Web-Services
 Benutzerzuordnungsoptionen 424
 Kurznamenoptionen 424
 Serveroptionen 424
 Spaltenoptionen 424
 unterstützte Datentypen 478
 unterstützte föderierte Funktionen 359
 Wrapperoptionen 424
 WITH HOLD, Option 159
 WITH HOLD, Syntax 159
 Wrapper
 Beschreibung 6
 Wrapperoptionen
 BioRS 361
 Dateien mit Tabellenstruktur 421
 DB2-Datenbank 365
 Excel 373
 Informix 374
 JDBC 380
 Microsoft SQL Server 387
 ODBC 393
 Oracle 399
 Script 405
 Sybase 410

Wrapperoptionen (*Forts.*)
 Teradata 417
 Web-Services 424
 XML 432

X

XML
 Benutzerzuordnungsoptionen 432
 Datentyp
 Einschränkungen 196
 Unterstützung 191
 Dokumente
 prüfen 193
 zerlegen 194
 gültige Objekte für Kurznamen 16
 Kurznamenoptionen 432
 Schemarepository 194
 Schemata registrieren 192
 Serveroptionen 432
 Spaltenoptionen 432
 unterstützte Datentypen 478
 unterstützte Versionen 2
 Wrapperoptionen 432
 XML-Wrapper 233
 XMLVALIDATE, Funktion 193

Z

Zeichenfolgen
 leer, mit Oracle verwendet 151
 Sortierfolgen 19
 Zeichensätze
 Beschreibung 19
 Zeitangaben, mit 24 im Stundenfeld 151
 Zeitmarken
 mit 24 im Stundenfeld 151
 Zeitmarkenmonitorschalter 289
 Zugriffsplan 264
 Zugriffspläne 259
 anzeigen 246, 285
 Asynchronitätsoptimierung 269
 Beschreibung 11
 Entscheidungen für Auswertung 248
 Entscheidungen für Optimierung 286
 Leistung 286
 Zuordnungen
 föderiert 148
 Zuordnungssemantik in föderierten Systemen
 Beispiele 150
 Zweiphasiges Commit
 Operationen 113
 unterstützte Datenquellen 359
 Zweiphasiges Commit für föderierte Transaktionen
 siehe föderiertes zweiphasiges Commit 117
 Zwischenspeichern 211, 291, 292
 Kurznamen 325



SC12-3759-02



Spine information:

IBM InfoSphere Federation Server

Version 9.7

Föderierte Systeme - Verwaltung

