Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles

Liebe Leserinnen und Leser,

die Events wie das INFORMIX Bootcamp in Düsseldorf und die INFORMIX Infobahn Roadshow haben viele Informationen über die neue INFORMIX Version gebracht. Die Aussage „INFORMIX is everywhere“ trifft dabei auf den Alltag zu. Mit großer Wahrscheinlichkeit werden die Blumen zum Muttertag über eine INFORMIX Datenbank verschickt. Die Pralinen aus dem Supermarkt werden über ein System mit INFORMIX bestellt und über ein Logistik Unternehmen, das INFORMIX einsetzt, transportiert. Machen Sie den Planeten ein wenig smarter und nutzen Sie INFORMIX um Ihren Lieben eine Freude zu machen.

Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt.

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: TimeSeries Datablade (Teil 4: Funktionen)

Um einen Eindruck der Vielfalt der zur Verfügung stehenden Funktionen zu vermitteln, haben wir hier einmal die am meisten genutzten Funktionen aufgelistet:

Informationen über die Timeseries:
GetOrigin, GetInterval, GetCalendar, GetCalendarName, GetContainerName, GetMetaData, GetMetaTypeName, IsRegular, InstanceId

Konvertierungen zwischen Offset und Timestamp:
GetIndex , GetStamp

Anzahl der Elemente:
GetNelems, ClipGetCount

Werte auslesen:
GetElem, GetLastValid, GetNextValid, GetPreviousValid, GetNthElem, GetFirstElem, GetLastElem , GetLastNonNull, GetNextNonNull

Ändern des Inhalts:
PutElem, PutElemNoDups, PutNthElem, PutSet, DelElem, DelClip, DelTrim, InsElem, InsSet, UpdElem , UpdSet, PutTimeSeries

Metadaten ändern:
UpdMetaData

Sichtbarkeit beim Auslesen beeinflussen:
HideElem, HideRange, RevealElem, RevealRange, ElemIsHidden, ElemIsNull, Apply, Clip, ClipCount , WithinC, WithinR

Kalender ändern:
ApplyCalendar

Laden von Werten:
BulkLoad

Timeseries erstellen:
TSCreate, TSCreateIrr

Schnittmenge und Vereinigungsmenge:
Intersect, Union

Tabellen und Listen erstellen:
Transpose, TSSetToList, TSColNameToList, TSColNumToList , TSRowToList, TSRowNameToList, TSRowNumToList

Werte für Statistische Berechnungen (im Apply):
Sum, TSAddPrevious, TSDecay, TSRunningAvg, TsRunningCor, TsRunningMed, TSRunningSum, TSRunningVar , TSCmp, TSPrevious

Operationen auf mehreren TimeSeries:
Plus, Minus, Times, Divide, Pow, Abs, Exp, Logn, Mod, Negate, Positive, Round, Sqrt, Cos, Sin, Tan, Acos, Asin, Atan , Atan2, ApplyBinaryTsOp, ApplyUnaryTsOp, ApplyOpToTsSet, AggregateBy, AggregateRange

Weitere Funktionen:
Lag, SetOrigin

Funktionen auf Containern:
TSContainerCreate, TSContainerDestroy, SetContainerName

Da wir im Newsletter ausser dem TimeSeries Datablade dieses Jahr noch weitere Themen behandeln wollen, haben wir eine kleine Auswahl an Funktionen getroffen, die wir Ihnen jeweils an einem Beispiel vorstellen wollen.

TechTipp: TimeSeries Datablade (Teil 5: Virtuelle Tabellen)

Unter den vielen Funktionen des TimeSeries Datablades findet sich eine Möglichkeit, die Daten der TimeSeries über Virtuelle Tabellen bereitzustellen, auf die wie gewohnt mit SQL zugegriffen werden kann.

Hinweis: Die Beispiele beziehen sich auf die Datenbank, deren Aufbau und Erstellung in der Ausgabe März dieses Newsletters erstellt wurden.

Im Beispiel kann dies so aussehen:
Zuerst werden die Daten in eine Virtuelle Tabelle geschrieben:

```
EXECUTE PROCEDURE TSCreateVirtualTab('v_wetterdaten', 'tageswetter',
            'calendar(Wetter Tag), origin(2011-03-14 00:00:00.00000),
            container(Wetter_Day_Cont)', 0, 'wetterdaten' );
```

Als Argumente sind erforderlich: der Name der neuen, virtuellen Tabelle (v_wetterdaten), der Name der ursprünglichen Tabelle (tageswetter), der Kalender der Timeseries (Wetter Tag), das Ausgangsdatum der Daten (2011-03-14 00:00:00.00000), sowie der Name des Containers (Wetter_Day_Cont), der TSVTMode (*) und des ROW-Type (wetterdaten).

(*) Der TSVTMode bietet die Optionen:

```
0 und 1     Default. Nur belegte Werte ausgeben.
2           „Null“ Elemente ebenfalls in der Virtuellen Tabelle ausgegeben.
512         „Hidden elements“ erscheinen in der Virtuellen Tabelle.
```

Danach können die Daten mittels SQL ausgelesen werden:

```
select location, extend(t_stamp, year to hour) as wann, temperatur, druck, wind
from v_wetterdaten order by 1,2
```

Ergebnis:

```
location     wann           temperatur       druck        wind
Lindau       2011-03-14 00        6.60        1024          13
Lindau       2011-03-14 01        7.60        1026          11
Lindau       2011-03-14 02        8.60        1026           9
Lindau       2011-03-14 03        9.60        1026           4
...
Lindau       2011-03-14 10       14.30        1024           4
Lindau       2011-03-14 11       14.60        1024           7
Lindau       2011-03-14 12       14.90        1024           7
Lindau       2011-03-14 13       15.20        1024           7
Lindau       2011-03-14 14       15.70        1025          11
Lindau       2011-03-14 17       16.30        1022           5
Ravensburg   2011-03-14 00        3.60        1024          13
Ravensburg   2011-03-14 01        4.60        1025           3
```

Der Umweg über die Virtuellen Tabellen vereinfacht den Zugriff über SQL, bedingt aber, dass die effiziente Speicherung der Daten als TimeSeries und der schnelle Zugriff über die internen Funktionen des Datablades dadurch umgangen werden.

TechTipp: TimeSeries Datablade (Teil 6: Umformungen)

Die als TimeSeries gespeicherten Daten können in Row-Types transformiert werden, bei denen in jedem Record die expandierte Zeitangabe zu finden ist.

Der Aufruf der Funktion

```
execute function Transpose((select wetterdaten
   from tageswetter where kfz = 'LI'));

(expression)  ROW('2011-03-14 00:00:00.00000', 6.60, 0, 1024, 34.00, 13, 142)
(expression)  ROW('2011-03-14 01:00:00.00000', 7.60, 0, 1026, 32.00, 11, 141)
(expression)  ROW('2011-03-14 02:00:00.00000', 8.60, 0, 1026, 31.00, 9, 141)
(expression)  ROW('2011-03-14 03:00:00.00000', 9.60, 0, 1026, 31.00, 4, 140)
(expression)  ROW('2011-03-14 04:00:00.00000', 10.50, 0, 1025, 31.00, 4, 149)
...
```

Die Ausgabe der umgewandelten TimeSeries als Row-Type kann auch als Liste erstellt werden:

```
SELECT TSsettolist(wetterdaten)::list(r_wetter_time not null)
FROM tageswetter
WHERE kfz = 'LI'

(expression)  LIST{ROW('2011-03-14 00:00:00.00000',6.60    ,0          ,1024
                  ,34.00   ,13         ,142        ),ROW('2011-03-14 01:00:00.0
                  0000',7.60    ,0          ,1026       ,32.00   ,11         ,141
                        ),ROW('2011-03-14 02:00:00.00000',8.60    ,0          ,1026
                         ,31.00   ,9          ,141        ),ROW('2011-03-14 03:00:0
                  0.00000',9.60    ,0          ,1026       ,31.00   ,4          ,14
                  0        ),ROW('2011-03-14 04:00:00.00000',10.50   ,0          ,1
                  025       ,31.00   ,4          ,149        ),ROW('2011-03-14 05:0
                  0:00.00000',11.80   ,0          ,1025       ,31.00   ,4
                  ,143        ),ROW('2011-03-14 06:00:00.00000',11.90   ,0
                   ,1025       ,31.00   ,4          ,143        ),ROW('2011-03-14 0
                  7:00:00.00000',12.60   ,0          ,1025       ,31.00   ,4
                     ,143        ),ROW('2011-03-14 08:00:00.00000',13.30   ,0
                      ,1024       ,31.00   ,4          ,143        ),ROW('2011-03-1
                  4 09:00:00.00000',14.10   ,0          ,1024       ,31.00   ,4
                        ,148        ),ROW('2011-03-14 10:00:00.00000',14.30   ,0
                         ,1024       ,32.00   ,4          ,140        ),ROW('2011-0
                  3-14 11:00:00.00000',14.60   ,0          ,1024       ,33.00   ,7
                           ,147        ),ROW('2011-03-14 12:00:00.00000',14.90   ,0
                            ,1024       ,33.00   ,7          ,147        ),ROW('201
                  1-03-14 13:00:00.00000',15.20   ,0          ,1024       ,33.00
                  ,7          ,141        ),ROW('2011-03-14 14:00:00.00000',15.70
                   ,0          ,1025       ,33.00   ,11         ,141        ),ROW('
                  2011-03-14 17:00:00.00000',16.30   ,0          ,1022       ,30.00
                     ,5          ,137        )}
```

Die so erstellte Liste kann als „derived Table“ im SQL eingesetzt werden:

```
SELECT * FROM TABLE (
            (SELECT TSsettolist(wetterdaten)::list(r_wetter_time not null)
             FROM tageswetter
             WHERE kfz = 'LI')
);

t_stamp       2011-03-14 00:00:00.00000
temperatur    6.60
regen         0
druck         1024
hygro         34.00
wind          13
windrichtung  142

t_stamp       2011-03-14 01:00:00.00000
temperatur    7.60
regen         0
druck         1026
hygro         32.00
wind          11
windrichtung  141
```

Eine Ausgabe ausgewählter Werte als errechnete Tabelle ist ebenfalls möglich:

```
create row type r_result (
        t_stamp datetime year to fraction(5),
        temp decimal(6,2)
);

SELECT * FROM TABLE (
(SELECT TSColNumToList((Apply('$temperatur',
        wetterdaten)::TimeSeries(r_result)),1)::list (decimal(6,2) not null)
 FROM tageswetter
 WHERE kfz = 'LI')
);

unnamed_col_1

         6.60
         7.60
         8.60
...
        14.30
        14.60
        14.90
        15.20
        15.70
        16.30
```

Statt dem Parameter $temperatur könnte auch $0.1 angegeben werden, was für die Ausgabe des ersten Arguments nach der TimeSeries steht.

TechTipp: TimeSeries Datablade (Teil 7: Speichervergleich)

Das in den vorhergehenden Artikeln zu TimeSeries genutzte Beispiel hatte einen sehr geringen Anteil an fixen Informationen in den Messdaten. Stationsname und Kfz wurden mit 15 Byte erfasst.
Speichert man mit diesem „Overhead“ 2000 Messerte, so benötigen diese in den TimeSeries **34 Datenpages**. Als herkömmliche Tabelle würden **55 Datenpages** benötigt, was eine Einsparung von rund **40%** ergibt.
Geht man von der realistischen Annahme aus, dass der „Header“ 200 Byte umfasst, so ändert sich das Bild gewaltig. Die Speicherung in TimeSeries kann dies weiterhin in **34 Pages** bewerkstelligen, da die zusätzlichen Informationen noch auf der ersten Page Platz haben. Die herkömmliche Tabelle hingegen benötigt nunmehr **251 Pages** für die Daten und damit ist dies mehr als **Faktor 7**.
Je mehr Daten erfasst werden müssen und je größer der „Overhead“ der Stationsdaten gegenüber den Nutzdaten ist, umso mehr kommt der Vorteil der TimeSeries zur Geltung. Da wenige Pages für eine Vielzahl an Informationen ausreichen, ist entsprechend die Performance deutlich besser, da einerseits viel weniger I/O-Vorgänge notwendig sind um die Daten auszulesen, andererseits auch viel mehr Daten im Cache gehalten werden können, was die Anzahl der kostspieligen Zugriffe auf Platte minimiert.

TechTipp: SQL - „IF NOT EXISTS“

Das bedingte Erstellen oder Löschen von Datenbankobjekten wurde in Version 11.70 in vielen Bereichen ermöglicht. Als Syntax wurde einheitlich der Ausdruck „IF NOT EXISTS“ in die Befehle aufgenommen.

Beispiele:

```
create table if not exists …
create temp table if not exists …
create index if not exists …
create procedure if not exists …
create distinct type if not exists
create implicit cast if not exists
```

Beim Drop lautet die Syntax „IF EXISTS“:

```
drop table if exists …
drop index if exists …
drop procedure if exists
drop type if exists
drop cast if exists
```

TechTipp: Growth Edition – was ist NICHT erlaubt ?

Teilweise sind die Einschränkungen der Growth Edition bekannt. Besonders die Einschränkungen im Bereich CPUs (max. 4 Sockets, max 16 Cores) und Memory (max. 16 GB) werden immer zuerst genannt. Das Feature „Parallel Data Query“ PDQ und die Fragmentierung bzw. Partitionierung gehören auch noch zu den bekannteren Einschränkungen, da diese direkt von der Workgroup Edition übernommen wurden.
Aus diesem Grund wollen wir nochmals die Administratoren auf die eher unbekannten Einschränkungen hinweisen:

Folgende Funktionen bzw. Tools dürfen NICHT eingesetzt werden:

- Direct I/O (Aktivierung in ONCONFIG: DIRECT_IO)
- Recovery Time Objective (Aktivierung in ONCONFIG: RTO_SERVER_RESTART)
- Memory Cache für CPUs (Aktivierung in ONCONFIG: VP_MEMORY_CACHE_KB)
- Paralleler Backup/Restore mit On-Bar (Aktivierung in ONCONFIG: BAR_MAX_BACKUP)
- Parallel Database Query (Environment PDQPRIRITY oder SQL)
- Nutzung des High Performance Loaders (HPL, onpladm, onpload, ipload)
- Partitionierung und Fragmentierung von Tabellen und Indizes
- Verschlüsselung von Daten auf Spaltenebene (SQL: encrypt_aes(), …)
- Data Compression

Der Einsatz einiger der Features wird in der Tabelle sysmaster:syslicenseinfo protokolliert, die nicht änderbar ist. Daher sollten Tests mit Features, die nicht in der erworbenen Version enthalten sind, auf einem Testserver mit der „Developer Edition“ durchgeführt werden.

TechTipp: ONCONFIG - VP_MEMORY_CACHE_KB

Dieser neue Parameter in der ONCONFIG ermöglicht die Zuordnung von Memory Cache an die CPU-VPs der Instanz. Dieses private Memory wird genutzt wenn ein Thread bei der Verarbeitung Memoryblöcke benötigt. Gelesene Seiten von Platte, sowie Seiten des Bufferpools werden nicht in den privaten Memory Cache eines CPU-VPs geschrieben.
Hat ein CPU-VP keinen privaten Memory Cache, dann wird im „Virtuellen Shared Memory“ nach aufeinander folgenden freien Blöcken gesucht. Diese Suche kann nicht parallel erfolgen und kann daher auf Systemen mit viel „Virtuellem Memory“ und vielen CPU-VPs zu Engpässen sorgen.
Wurde genug Platz als „private Cache“ dem CPU-VP zugeordnet, kann diese Suche entfallen, da der CPU-VP den privaten Cache selbst verwaltet.

Der Parameter kann Werte zwischen 800k und 40% des SHMTOTAL des Systems annehmen.
Besonders auf großen Systemen kann mit diesem Parameter eine deutliche Verbesserung der Performance der CPU-VPs erzielt werden, da die konkurrierende Suche im „Virtuellen Shared Memory“ entfällt. Der Speicher wird in Blöcken zu 4096 Byte genutzt.

Der Wert kann Dynamisch mittels „onmode -wm bzw. -wf“ gesetzt werden.
Beispiel:

```
onmode -wf VP_MEMORY_CACHE_KB=2048
Value of VP_MEMORY_CACHE_KB has been changed to 2048.
```

Der Wert für VP_MEMORY_CHACHE_KB sollte nicht zu hoch gewählt werden, da dies zu Lasten des verfügbaren Speichers und damit der Bufferpools geht.
Der optimale Wert hängt von der jeweiligen Applikation ab und sollte anhand der Nutzung dieses privaten Memorys dynamisch angepasst werden.
Die Analyse kann mittels „onstat -g vpcache“ erfolgen (siehe nächster Artikel).

TechTipp: onstat -g vpcache

Die Ausgabe von „onstat -g vpcache“ zeigt die Nutzung des privaten Memorys je CPU-VP an.
Für eine Feinabstimmung sind hier besonders die Informationen über „Free cache“ und „Hit percentage“ zu beachten. Das Verhältnis der Werte in „miss“ zu „alloc“ zeigt wie oft Platz in der angefragten Größe vergeben werden konnte und wie oft kein Platz in ausreichender Größe zur Verfügung stand.

Beispiel:

```
CPU VP memory block cache statistics - 4096 byte blocks

Number of 4096 byte memory blocks requested for each CPU VP:256
vpid    pid         Blocks held  Hit percentage   Free cache
1       5924        242           81.6 %            93.6 %

Current VP total allocations from cache:     4102, Total frees:     4505
   size  cur blks   alloc      miss       free      drain
    1     59         2036       413        2294      199
    2     20         805        334        853       38
...
    31    0          0          0          0         0
    32    0          0          0          0         0

size: size of memory blocks in 4096 byte blocks
cur blks:current number of 4096 blocks - multiple of size field
alloc:number of times we gave a requestor a block of this size
miss:number of times block was requested but none were available
free:number of times we placed a memory block into the cache
drain:number of time we forced an aged block out to make room
```

Termin: INFORMIX 11.70 – New Features (IX3020DE)

Unsere Schulungsabteilung hat einen neuen 3-tägigen Kurs ins Angebot aufgenommen: Informix 11.7 New Features
Der Kurs beinhaltet praktische Übungen zu vielen der neuen Features und ist daher die ideale Gelegenheit sich mit diesen vertraut zu machen.

Die bisher geplanten Termine sind:

04.-06. Juli 2011	Stuttgart-Degerloch	(ZMCE)
19.-21. September 2011	München	(ZMCG)

Termin: IBM Informix auf dem IBM Information Management Forum

Das IBM Information Management Forum ist eine neue Veranstaltungsreihe für ITler von ITlern und findet am **23. und 24. Mai im Maritim Hotel, Darmstadt** statt.
Interessierte Teilnehmer können hier alles über Trends, Entwicklungen und innovative Lösungen rund um das Thema Information Management erfahren.
Sie profitieren dabei von den Erfahrungsberichten anderer Unternehmen, den Einschätzungen unabhängiger Analysten und externer Sprecher und dem Know How von IBM Experten.

Am 2. Tag wird es unter anderem zwei Vorträge zu Informix geben. Alexander Körner (IBM) stellt in seinem Vortrag:
„Geschäftsinnovationen mit Zeitreihendaten: Smart Meter-, Sensor- und Finanz-Daten erschließen“
eine IBM Datenbanktechnologie vor, die in ihrer Art technologisch führend ist und es erlaubt, große Mengen von zeitreihenbasierten Daten sehr effizient zu verwalten und auszuwerten.

Am Nachmittag betrachten dann Dr. Andreas Weininger (IBM) und Sabine Geisler (IBM) in
"In-Memory Datenbanken –Hype oder Zukunft"
was für Auswirkungen der Trend zu In-Memory Datenbanken auf OLTP- und Data-Warehousing-Anwendungen hat und stellt revolutionäre Neuentwicklungen der IBM in diesen Bereichen vor.

Weitere Informationen und Anmeldung unter: http://www.ibm.com/de/events/im-forum/

Versionsinfo: 11.50.xC8W3 ist verfügbar

Seit einigen Tagen ist die Version 11.50.xC8W3 für alle unterstützten Plattformen und Editionen verfügbar.
W-Release enthalten eine Zusammenfassung von Verbesserungen,
X-Relases sind Einzelkorrekturen.

Anmeldung / Abmeldung / Anmerkung

Der Newsletter wird ausschließlich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie eine Email mit dem Betreff „**ANMELDUNG**“ an **ifmxnews@de.ibm.com** senden.
Im Falle einer Abmeldung senden Sie „ABMELDUNG“ an diese Adresse.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149
http://www.informix-zone.com/informix-german-newsletter
http://www.drap.de/link/informix
http://www.nsi.de/informix/newsletter
http://www.bytec.de/de/software/ibm_software/newsletter/
http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter
http://www.listec.de/Informix_Newsletter/
http://www.bereos.eu/software/informix/newsletter/

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist Informix Dynamic Server und DB2 UDB
IBM Software Group, Information Management
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
IBM Software Group, Information Management
martinfu@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

Die Versionsinfo stammt aus dem Versions-Newsletter der CURSOR Software AG
http://www.cursor-distribution.de/download/informix-vinfo

Fotonachweis: Gerd Kaluzinski (Testbestellung schon vor dem Muttertag)