

## Willkommen zum „IBM Informix Newsletter“

### Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: SERIAL / SERIAL8 / BIGSERIAL.....	2
TechTipp: SEQUENCE.....	4
TechTipp: ONCONFIG - SEQ_CACHE_SIZE.....	5
TechTipp: DRINTERVAL bei RSS.....	6
Hinweis: Ultimate- Growth- Innovator-C- Developer Edition.....	6
Anmeldung / Abmeldung / Anmerkung.....	7
Die Autoren dieser Ausgabe.....	7

### Aktuelles

Liebe Leserinnen und Leser,

rechtzeitig zur Urlaubssaison zeigt sich der Sommer von seiner schönsten Seite. Trotz einer Flut von „Out-Of-Office“-Antworten, die nach der Verteilung der Newsletter Juli und August auf uns einregnen, werden wir Sie auch im Sommer mit aktuellen Tipps und Tricks begleiten. Damit die Kosten für das Laminieren des Newsletter für den Gebrauch im Freibad nicht zu hoch sind, haben wir den Umfang für diese beiden Ausgaben reduziert.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

## TechTipp: SERIAL / SERIAL8 / BIGSERIAL

Je Tabelle kann eine Spalte vom Datentyp SERIAL und eine Spalte von Typ SERIAL8 bzw. BIGSERIAL genutzt werden, in der Werte beim Einfügen in die Tabelle automatisch vergeben werden. Die automatische Vergabe der Serialwerte erfolgt, wenn bei einem Insert in der Serial-Spalte eine 0 eingefüllt wird, oder die Spalte beim Insert nicht gefüllt wird.

Beim Erzeugen dieser speziellen Spalte kann optional ein Startwert mitgegeben werden. Wird kein Startwert angegeben, so startet der Serialwert bei 1. Wird bei einem ALTER TABLE ein Startwert angegeben, der kleiner ist als der größte Wert der Spalte (wenn diese z.B. zuvor als INTEGER angelegt war), so wird beim Erstellen der höchste Wert der Spalte ermittelt und um 1 erhöht.

Der Wertebereich bei Serial entspricht dem positiven Bereich von INTEGER, er reicht daher von 1 bis 2147483647.

Erreicht ein Serial das Maximum, so fängt der Serial wieder bei 1 zu zählen an. Sollte dieser Wertebereich nicht ausreichen, so kann statt SERIAL der Datentyp SERIAL8 / BIGSERIAL verwendet werden, dessen Wertebereich bis  $(2^{exp63})-1$ , also 9,223,372,036,854,775,807 reicht.

Beispiel zu Serial:

```
create table ser_test (  
f1      int,  
f2      char(20)  
);  
insert into ser_test values (42,"Die Antwort");  
alter table ser_test modify f1 serial(38);  
insert into ser_test values (0,"Limit nach 38");  
insert into ser_test values (2147483646,"Ganz viel");  
insert into ser_test values (0,"Limit nach maxint");  
insert into ser_test values (0,"Limit noch mehr");  
insert into ser_test values (0,"Limit und noch mehr");  
select * from ser_test
```

Ergebnis:

```
 f1 f2  
42 Die Antwort  
43 Limit nach 38  
2147483646 Ganz viel  
2147483647 Limit nach maxint  
 1 Limit noch mehr  
 2 Limit und noch mehr
```

**ACHTUNG:** Der Datentyp Serial bzw. Serial8 bedeutet keine Eindeutigkeit. Explizite Werte können jederzeit in die Spalte eingetragen werden. Doppelte Einträge werden nur verhindert, wenn zum Serial ein Unique Constraint auf der Spalte erstellt wird.

Beispiel:

```
create table ser_test (
f1      serial(42),
f2      char(18)
);
insert into ser_test values (42,"Die Antwort");
insert into ser_test values (42,"Nochmals 42");
insert into ser_test values (42,"Wieder 42");
insert into ser_test values (42,"Diesmal 23");
select * from ser_test
```

Ergebnis:

```
f1 f2
42 Die Antwort
42 Nochmals 42
42 Wieder 42
42 Diesmal 23
```

Da je Tabelle sowohl eine Spalte vom Datentyp SERIAL als auch eine Spalte vom Datentyp BIGSERIAL erlaubt ist, besteht die Möglichkeit zwei Zähler in einer Tabelle zu definieren:

Beispiel für eine Tabelle mit einem Serial und einem Bigserial:

```
create table sertest2 (
f1      serial (42),
f2      bigserial(23),
f3      char(12)
);
insert into sertest2(f3) values ("TEST");
insert into sertest2(f1) values (100);
insert into sertest2(f2) values (1000);

select * from sertest2
```

```
f1          f2 f3
42          23 TEST
100         24
101         1000
```

Beim Versuch zwei Spalten vom Datentyp SERIAL oder BIGSERIAL zu erstellen bekommt man die Fehlermeldung:

```
-362 Can have only one column of serial/(serial8 or bigserial) type.
```

## TechTipp: SEQUENCE

Sequence ist die Möglichkeit eine automatische Zahlenvergabe in der Datenbank zu hinterlegen. Im Gegensatz zu Serials, die an eine Tabelle gebunden sind, und nur ein Increment um 1 erstellen können, kann eine Sequenz frei definierte Intervalle nutzen. Zudem ist es möglich bei einem Insert in eine Tabelle mehr als eine Sequenz zu nutzen.

Beispiel einer Sequence:

```
create sequence kalu_seq
    increment by 3
    start with 42
    maxvalue 100
    minvalue 13
    cycle
    cache 8
    order;
```

Als Zähler für „**increment**“ kann jede positive oder negative ganze Zahl genutzt werden. Wird diese nicht angegeben, so wird immer um 1 erhöht.

Mit „**start with**“ kann ein Startwert angegeben werden.

Der Wert „**maxvalue**“ begrenzt die Werte nach oben, der Wert „**minvalue**“ gibt eine untere Grenze an. Der Default ist „**nomaxvalue**“ bzw. „**nominvalue**“.

Mit „**cycle**“ wird angegeben, dass die Sequence beim Erreichen des „**maxvalue**“ wieder beim „**minvalue**“ (oder bei absteigender Sequence umgekehrt) weiter machen darf. Wird hingegen „**nocycle**“ angegeben, so ist nach Erreichen des „**maxvalue**“ (minvalue) Schluss und es kommt die Fehlermeldung

„**8313: Sequence (kalu.kalu\_seq) exceeds its MAXVALUE**“.

Der Default ist „**nocycle**“.

Mit „**cache**“ wird angegeben wie viele Werte der Sequenz vorab im Speicher bereitgestellt werden sollen. Der Vorteil hierbei ist, dass die Performance bei vielen konkurrierenden Zugriffen auf die Sequenz deutlich verbessert wird. Der Nachteil ist, dass beim Shutdown der Instanz die Werte, die bereits in den Speicher geladen wurden, verloren gehen und damit eine „Lücke“ in der von der Sequenz vergebenen Zahlenreihe entsteht.

Es kann nur eine bestimmte Anzahl an Sequenzen mit Cache in einer INFORMIX Instanz definiert werden. Mehr hierzu im folgenden TechTipp.

Der Wert „**order**“ oder „**noorder**“ hat keine Auswirkung, da Sequenzen immer sortiert, also mit „**order**“ angelegt werden. Dieser Parameter wurde aus Gründen der Kompatibilität zu anderen SQL-Datenbanken hinzugefügt.

Die Werte einer Sequenz werden mit **nextval** angefordert. Der Wert **currval** gibt den zuletzt vergebenen Wert der Sequenz aus.

Beispiele:

```
select kalu_seq.nextval from systables where tabid;
select kalu_seq.currval from systables where tabid;
```

Eine Sequenz wird in der Datenbank in den Tabellen „systables“ und „syssequences“ eingetragen. In „systables“ steht der Name der Sequenz und der Eigentümer, sowie das

Erstellungsdatum. Der „tabtype“ in systables ist mit „Q“ angegeben.

```
tabname      kalu_seq
owner        kalu
partnum      3146347
tabid        281
rowsize      10
...
created      28.06.2010
version      18415617
tabtype      Q
```

In der Tabelle „syssequences“ stehen die Werte der Sequenz. Zu unserem obigen Beispiel sind dies:

```
select * from syssequences where tabid = 281;
seqid        8
tabid        281
start_val    42
inc_val      3
min_val      13
max_val      100
cycle        1
restart_val
cache        8
order        1
```

### TechTipp: ONCONFIG - SEQ\_CACHE\_SIZE

Per Default wird in einer Instanz nur für 10 Objekte vom Typ Sequenz der Cache bereitgestellt. Reicht dies nicht aus, so kann in der ONCONFIG der Parameter SEQ\_CACHE\_SIZE hinzugefügt und mit dem notwendigen Wert definiert werden. Der Parameter ist in der onconfig.std der Version 11.50.xC7 (und früher) nicht eingetragen. Der Parameter kann am Ende der ONCONFIG hinzugefügt werden. Nach einem Restart der Instanz wird der erweiterte Platz für den Cache der Sequenzen eingerichtet.

Der aktuelle Wert kann mittels SQL abgefragt werden:

```
select * from sysmaster:syscftab where cf_name = "SEQ_CACHE_SIZE"
```

## TechTipp: DRINTERVAL bei RSS

Beim Aufbau einer HDR Replikation ist der Onconfig-Parameter DRINTERVAL entscheidend, ob die Replikation synchron oder spätestens nach einem bestimmten Intervall die Daten an den Secondary überträgt. DRINTERVAL gibt dabei die maximale Zeit (in Sekunden) an, nach der der Log Buffer in die Replications Buffer übertragen und an den Secondary Server gesendet wird. Der Parameter DRINTERVAL muss dabei auf dem Primary und dem Secondary Server gleich gesetzt sein.

Beim Aufbau einer Replikation, die statt einem HDR Secondary Server einen RSS Server (Remote Standalone Secondary Server) nutzt, schlägt die Initialisierung der Replikation fehl, wenn das DRINTERVAL nicht auf dem RSS identisch wie auf dem Primary Server gesetzt ist. Auch in diesem Fall bestimmt das DRINTERVAL den Zeitpunkt, an dem die Loginformationen aus dem LogBuffer in den Replikationsbuffer übertragen und gesendet werden. Da ein RSS Server nicht mit synchroner Replikation betrieben werden kann und es zudem die Möglichkeit gibt, mittels DELAY\_APPLY die Replikation zeitversetzt abarbeiten zu lassen, scheint dieser Parameter auf den ersten Blick nicht zu dieser Form der Replikation zu passen.

Bedenkt man jedoch, dass aus dem RSS Server ein HDR Secondary Server oder sogar ein Primary Server werden kann, leuchtet es ein, dass der Parameter DRINTERVAL sowie die anderen DR-Parameter der ONCONFIG auf allen an der Replikation beteiligten Instanzen identisch gesetzt sein müssen.

## Hinweis: Ultimate- Growth- Innovator-C- Developer Edition

Im letzten Newsletter hatten wir darauf verwiesen, dass in dieser Ausgabe ausführliche Informationen über die neuen Editions des INFORMIX Servers erklärt werden. Da es derzeit noch letzte Abstimmungen über den Umfang der Bundles gibt, ist der Beitrag bisher nicht komplett und wir müssen Sie daher auf eine der nächsten Ausgaben vertrösten. Der Kollege, der den Beitrag erstellt, versucht die neuesten Informationen einzupflegen. Er kann aber derzeit noch keine endgültige Version liefern, die veröffentlicht werden kann, da weitere Änderungen diskutiert werden.

Für den 20.Juli ist eine ausführliche Vorstellung der neuen Bundles und Preismodelle angekündigt, so dass wir danach auch hier im Newsletter darüber berichten können.

## 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](mailto: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.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/index.php?option=com\\_content&task=view&id=36&Itemid=87](http://www.nsi.de/index.php?option=com_content&task=view&id=36&Itemid=87)

[http://www.bytec.de/de/software/ibm\\_software/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.listec.de/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](mailto:gerd.kaluzinski@de.ibm.com)                      +49-175-228-1983

Martin Fuerderer                      IBM Informix Entwicklung, München  
   IBM Software Group, Information Management  
[martinfu@de.ibm.com](mailto:martinfu@de.ibm.com)

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Redaktion (Carmen Kaluzinski) – Redaktionsgarten