

## Willkommen zum „IBM DB2 Newsletter“

### Liebe Leserinnen und Leser,

als ich letzte Woche die Einleitung für diese Ausgabe fertigstellte, stimmte der nachfolgende Text noch. „Der Herbst ist in eine Phase übergegangen, in der die Tage grauer und kürzer werden. Der Winter ist noch nicht ganz da, der mit seinem Schnee dann wieder alles weiß macht.“ Aber inzwischen kam der erste Advent und mit ihm zeitgleich der Schnee. Zeit für Weihnachtsmarkt und Glühwein, sowie besinnliche Vorweihnachtszeit. Wobei mit Besinnlichkeit haben die 4 Wochen Weihnachtsstress nicht wirklich mehr was zu tun.



Weihnachtsfeier hier, Einkäufe da, Jahresend-Aktivitäten in den Firmen. Um diese trübe/kalte/hektische Jahreszeit zu verkürzen, haben wir wieder interessante Themen für Sie aufgegriffen. Viel Spaß beim Lesen und Ausprobieren.

Für Fragen und Anregungen unsere Kontaktadresse: [db2news@de.ibm.com](mailto:db2news@de.ibm.com).

Ihr TechTeam

## Inhaltsverzeichnis

<b>TECHTIPP: ÄNDERN NULL-WERTIGKEIT .....</b>	<b>2</b>
<b>TECHTIPP: DB2 SYSTEM ODBC DATASOURCES AUF X64.....</b>	<b>4</b>
PROBLEM.....	4
PROBLEMBESCHREIBUNG.....	4
LÖSUNG (WORKAROUND).....	5
<b>TECHTIPP: DB2 TABELLEN STATUS.....</b>	<b>5</b>
MOTIVATION.....	5
IMPLEMENTIERUNG.....	7
<b>TECHTIPP: ITDS UND LDAP_MAXCARD.....</b>	<b>8</b>
PROBLEM.....	8
URSACHE.....	8
LÖSUNG.....	9
<b>CHATS MIT DEM LABOR.....</b>	<b>9</b>
<b>SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....</b>	<b>9</b>
FRIENDLY REMINDER: DB2 LUW WORKING GROUP .....	9
<b>NEWSLETTER ARCHIV.....</b>	<b>10</b>
<b>ANMELDUNG/ABMELDUNG.....</b>	<b>10</b>
<b>DIE AUTOREN DIESER AUSGABE.....</b>	<b>10</b>

## TechTipp: Ändern NULL-Wertigkeit

Ich hatte für mehrere Tabellen des Schemas MYSCHEMA folgende Aufgabe:

- QUELLE\_VON\_TS sollte als NOT NULLABLE
- DWH\_VON\_TS sollte als NOT NULLABLE
- MODULE\_RUN\_ID sollte als NULLABLE

gesetzt werden und

- die Alternate-Keys (in Form von Unique Indexes) sollten nicht die Spalte QUELLE\_BIS\_TS beinhalten.

Nachfolgend ein Beispiel für das Layout einer dieser Tabellen:

```
CREATE TABLE "MYSCHEMA"."MYTABLE" (
  "MYCOLUMN_ID" INTEGER NOT NULL ,
  "MYNAT1" CHAR(2) NOT NULL ,
  "MYNAT2" CHAR(5) NOT NULL ,
  "QUELLE_VON_TS" TIMESTAMP WITH DEFAULT CURRENT TIMESTAMP ,
  "QUELLE_BIS_TS" TIMESTAMP NOT NULL WITH DEFAULT '9999-12-31-23.00.00.000000' ,
  "MYCOL1" VARCHAR(20) ,
  "MODULE_RUN_ID" INTEGER NOT NULL WITH DEFAULT -1 ,
  "DWH_VON_TS" TIMESTAMP WITH DEFAULT CURRENT TIMESTAMP ,
  "DWH_BIS_TS" TIMESTAMP NOT NULL WITH DEFAULT '9999-12-31-23.00.00.000000' )
IN "USERSPACE1" ;

- DDL Statements for primary key on Table "MYSCHEMA"."MYTABLE"

ALTER TABLE "MYSCHEMA"."MYTABLE"
  ADD PRIMARY KEY
    ("MYCOLUMN_ID");

- DDL Statements for unique constraints on Table "MYSCHEMA"."MYTABLE"

ALTER TABLE "MYSCHEMA"."MYTABLE"
  ADD CONSTRAINT "XAKMYTABLE" UNIQUE
    ("MYNAT1",
     "MYNAT2",
     "QUELLE_BIS_TS",
     "DWH_BIS_TS");
```

Da es hier nicht um 1 oder 2 Tabellen ging, sondern bei den ersten 3 Punkten um knapp 200 und bei der letzten Aufgabe um ca. 60, war eine manuelle Änderung jeder einzelnen Tabelle für mich undenkbar. Auch ein drop und create der Tabellen war keine Lösung.

Daher dachte ich darüber nach, wie ich mit SQL und Shell-Mitteln diese Aufgabe lösen konnte.

### 1. Ermitteln der Tabellen, die von den NULL-/NON-NULL Problematik betroffen sind:

```
select tabschema,tabname,colname,nulls
  from syscat.columns
 where colname in ('QUELLE_VON_TS','DWH_BIS_TS','MODULE_RUN_ID')
    and tabschema='MYSCHEMA' with ur "
```

```
$ db2 "select tabschema,tabname,colname,nulls
      from syscat.columns
      where colname in ('QUELLE_VON_TS','DWH_VON_TS','MODULE_RUN_ID')
        and tabschema='MYSCHEMA' with ur " |tr -s " "
```

```
TABSCHEMA TABNAME COLNAME NULLS
-----
MYSCHEMA MYTABLE DWH_VON_TS Y
MYSCHEMA MYTABLE MODULE_RUN_ID N
MYSCHEMA MYTABLE QUELLE_VON_TS Y
```

3 record(s) selected.

### 2. Erstellen der Kommandos zum Umsetzen der NULLability

#### a) NOT NULL

Spalten werden auf NOT NULL umgestellt, durch alter table alter column set not null  
alter table <SCHEMA>.<TABLE> alter column <COLUMNNAME> set not null

Angewendet auf die obige Aufgabe bedeutet dies z.B.:

```
db2 -x "select tabschema,tabname,colname,nulls from syscat.columns where colname in
('QUELLE_VON_TS','DWH_VON_TS') and tabschema='MYSCHEMA' and NULLS='Y' with ur " |tr -s " " |
while read schema table column junk; do echo "alter table $schema.$table alter column $column
SET not null;"; done
```

#### b.) NULL

Spalten werden auf NULLable umgestellt durch alter table alter column drop null.

```
db2 -x "select tabschema,tabname,colname,nulls from syscat.columns where colname in
('MODULE_RUN_ID') and tabschema='MYSCHEMA' and NULLS='N' with ur " |tr -s " " | while read
schema table column junk; do echo "alter table $schema.$table alter column $column drop not
null;"; done
```

Die DDL speichern, ausführen und danach ist ein Reorg notwendig, da sonst SQL0668 geliefert wird, mit ReasonCode 7.

### 3. Erstellen der DDL zum ändern der Alternate Keys

In diesem Fall, war es einfach, aufgrund der Namenskonventionen der Alternate-Key als XAK... zu identifizieren.

```
> db2 describe indexes for table MYSCHEMA.MYTABLE show detail | tr -s " "
Index Index Unique Number of Index Index Index Data Max Varchar Xml Column
schema name rule columns type partitioning ID type Hashed Length pattern names
-----
MYSCHEMA XPKMYTABLE P 1 RELATIONAL DATA - 1 - - - +MYCOLUMN_ID
MYSCHEMA XAKMYTABLE U 4 RELATIONAL DATA - 2 - - - +MYNAT1+MYNAT2+QUELLE_BIS_TS+DWH_BIS_TS

  2 record(s) selected.

db2 "select indname,tabschema,tabname from syscat.indexes where indname like 'XAK%' and colnames like '%QUELLE_BIS_TS%' | tr
-s "
-----
INDNAME TABSCHEMA TABNAME
-----
XAK1MYTABLE MYSCHEMA MYTABLE
```

Das Problem war nun, wie bekomme ich aus dieser Information der Indizes, die Änderungsstatements heraus. Ein Alter index drop column gibt es ja leider nicht, d.h. erst muss der Constraint entfernt und dann wieder angelegt werden.

Ein gemeinsames Ausführen des drop und recreate des Constraints in einer Anweisung führt zum Fehler SQL0668 mir ReasonCode 7. Daher müssen die Statements getrennt werden.

```
db2 -x "select indname,tabschema,tabname,colnames from syscat.indexes where indname like 'XAK%' and colnames like
'%QUELLE_BIS_TS%' | tr -s " " | while read indname schema table colnames; do echo $colnames | sed -e "s####" -e
"s#QUELLE_BIS_TS#g" -e "s####+g" -e "s#+#, 'g" | read colnames; echo "alter table $schema.$table drop unique $indname add
constraint $indname unique ('$colnames');"; done

alter table MYSCHEMA.MYTABLE drop unique XAK1MYTABLE add constraint XAKMYTABLE unique ('MYNAT1','MYNAT2','DWH_BIS_TS');

db2 "alter table MYSCHEMA.MYTABLE drop unique XAK1MYTABLE add constraint XAKMYTABLE unique (MYNAT1,MYNAT2,DWH_BIS_TS)"
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL
processing it returned:
SQL0668N Operation not allowed for reason code "7" on table "MYSCHEMA.MYTABLE". SQLSTATE=57016
```

Nach dem Ändern der Constraints ist in den meisten Fällen ebenfalls ein Reorg notwendig, was natürlich auch zu einem nachfolgenden Runstats führt.

Daher könnte das Erstellen der Kommandos zum recreate der Constraints wie folgt aussehen:

```
db2 -x "select indname,tabschema,tabname,colnames from syscat.indexes where indname like 'XAK%' and colnames like
'%QUELLE_BIS_TS%' | while read indname schema table colnames; do colnames=`echo $colnames | sed -e "s####" -e
"s#QUELLE_BIS_TS#g" -e "s####+g" -e "s#+#, 'g"; echo "alter table $schema.$table drop unique $indname;" ; echo "reorg table
$schema.$table with distribution and detailed indexes all;"; done
alter table MYSCHEMA.MYTABLE drop unique XAK1MYTABLE;
reorg table MYSCHEMA.MYTABLE;
alter table MYSCHEMA.MYTABLE add constraint XAK1MYTABLE unique (MYNAT1,MYNAT2,DWH_BIS_TS);
runstats on table MYSCHEMA.MYTABLE with distribution and detailed indexes all;
```

Nach den o.g. Schritten, sieht die Tabelle dann wie folgt aus:

```
CREATE TABLE "MYSCHEMA"."MYTABLE" (
  "MYCOLUMN_ID" INTEGER NOT NULL ,
  "MYNAT1" CHAR(2) NOT NULL ,
  "MYNAT2" CHAR(5) NOT NULL ,
  "QUELLE_VON_TS" TIMESTAMP NOT NULL WITH DEFAULT CURRENT TIMESTAMP ,
  "QUELLE_BIS_TS" TIMESTAMP NOT NULL WITH DEFAULT '9999-12-31-23.00.00.000000' ,
  "MYCOL1" VARCHAR(20) ,
  "MODULE_RUN_ID" INTEGER WITH DEFAULT -1 ,
  "DWH_VON_TS" TIMESTAMP NOT NULL WITH DEFAULT '9999-12-31-23.00.00.000000' ,
  "DWH_BIS_TS" TIMESTAMP NOT NULL WITH DEFAULT '9999-12-31-23.00.00.000000' )
IN "USERSPACE1" ;

-- DDL Statements for primary key on Table "MYSCHEMA"."MYTABLE"

ALTER TABLE "MYSCHEMA"."MYTABLE"
  ADD PRIMARY KEY
    ("MYCOLUMN_ID");

-- DDL Statements for unique constraints on Table "MYSCHEMA"."MYTABLE"
```

```
ALTER TABLE "MYSCHEMA"."MYTABLE"  
  ADD CONSTRAINT "XAKMYTABLE" UNIQUE  
    ("MYNAT1",  
     "MYNAT2",  
     "DWH_BIS_TS");
```

#### Hinweis:

Wenn Sie eine Spalte auf NOT NULL setzen, muss auch einen DEFAULT-Wert gesetzt werden, da sonst der Fehler [SQL0407N](#) auftritt.

```
$ db2 "alter table MYSCHEMA.MYTABLE alter column MYCOL1 set not null "  
DB21034E The command was processed as an SQL statement because it was not a  
valid Command Line Processor command. During SQL processing it returned:  
SQL0407N Assignment of a NULL value to a NOT NULL column "MYCOL1" is not  
allowed. SQLSTATE=23502
```

## TechTipp: DB2 System ODBC Datasources auf X64

### Problem

System ODBC Datasources auf X64 (64-bit) werden nicht als 32-bit ODBC Datasource Inhalt gelistet.

### Problembeschreibung

Wenn eine DB2 **System** ODBC Datasource (also nicht als User ODBC Datasource) auf einem Windows **64-bit** System (X64) katalogisiert wird, wird der System DSN Eintrag im 32-bit Administrations-Tool nicht angezeigt. Die Datasource ist somit auch nicht in den 32-bit ODBC Anwendungen (wie MS Excel, MS Access) verfügbar.

Beispiel eines Katalog Kommandos:

```
db2 catalog system odbc data source Z05XAT07
```

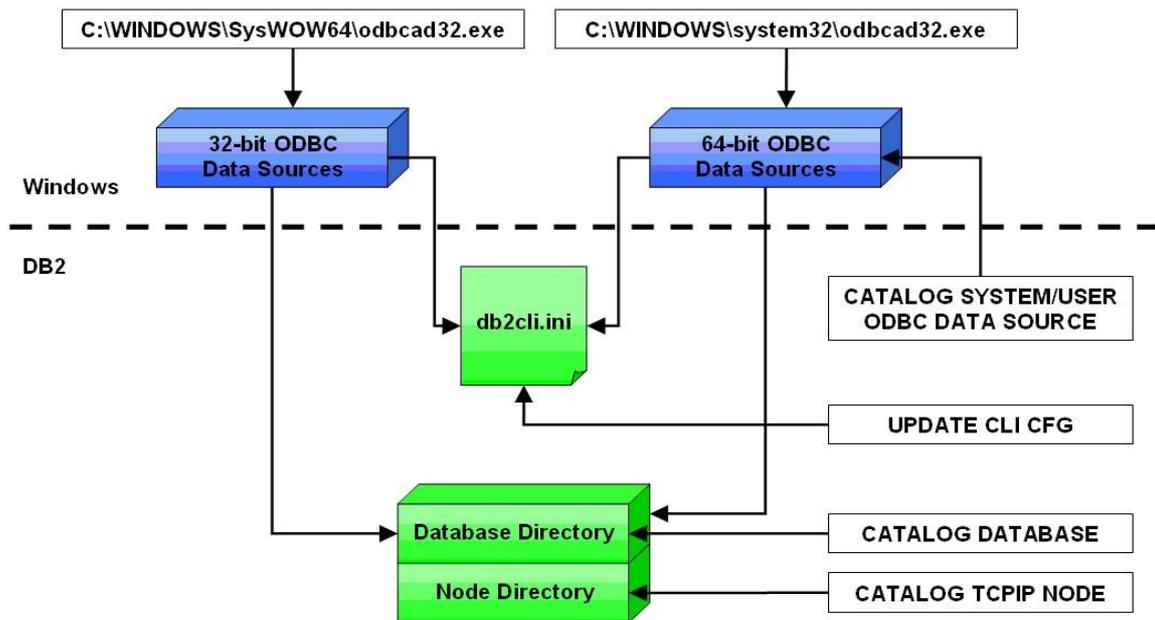
Ergebnisse:

1. 32 bit ODBC administrator tool:  
C:\windows\Syswow64\odbcad32.exe

Der System DSN Eintrag Z05XAT07 wird nicht angezeigt.

2. 64 bit ODBC administrator tool:  
C:\windows\system32\odbcad32.exe

Der System DSN Eintrag Z05XAT07 wird korrekt angezeigt.



## Lösung (Workaround)

- Verwendung des 32 bit ODBC administrator tools „C:\windows\Syswow64\odbcad32.exe“. Hier kann der System DSN Eintrag direkt erstellt werden, es wird auf das Katalog Kommando verzichtet.
- Verwendung des Windows Tools „odbcconf.exe“, welches es ermöglicht System DSN Einträge einzufügen. Hier können die Details auf dem Katalog Befehl weiterverwendet werden.
  - Aufruf:
 

```
C:\Windows\Syswow64\odbcconf.exe /A {CONFIGSYSDSN "IBM DB2 ODBC DRIVER" "DSN=dsname|DBALIAS=dbalias"}
```
  - Beispiel:
 

```
C:\Windows\Syswow64\odbcconf.exe /A {CONFIGSYSDSN "IBM DB2 ODBC DRIVER" "DSN=Z04SYS|DBALIAS=Z05XAT04"}
```
- Verwendung des DB2 CLI Utilities "db2cli32.exe" (verfügbar ab DB2 9.7 FP4), das ebenso wie "odbcconf.exe" ermöglicht, einen DSN-Eintrag zu erzeugen:

```
db2cli32 registerdsn -add dsname -system
```

## TechTipp: DB2 Tabellen Status

### Motivation

Wird an den DB2 Betrieb von Entwicklern oder Anwendern die Frage gestellt, weshalb eine Tabelle nicht verfügbar ist, dann sind mehrere Spalten im DB2 Catalog zu beachten, um den exakten Status einer Tabelle zu ermitteln. Gemeinsam mit einem Kunden wurde dafür eine SQL Abfrage erstellt, welche alle vier potentiellen Sonderstati (Load Pending, Reorg Pending, Check Pending und Rollforward Pending) einer Tabelle berücksichtigt. Die Stati werden einzeln angezeigt, zudem wird ein Gesamtstatus ermittelt (Tabelle ist verfügbar oder nicht verfügbar). Die einzelnen Stati können teilweise auch kombiniert auftreten (bspw. kann Check Pending und Rollforward Pending parallel auftreten).

In Table 1 werden die potentiellen Stati beschrieben, die also nicht einem normalen Status entsprechen. Die zweite Spalte zeigt die Catalog View und Spalte an, in welcher der Status gespeichert wird. Zudem werden die Ausprägungen beschrieben. In der dritten Spalte ist die Fehlermeldung notiert, die auftritt, wenn auf die Tabelle zugegriffen wird. Die Spalte „Lösung“ listet in Kürze die erforderliche Operation, um den unnormalen Status aufzulösen und die Tabelle in den Status „Normal“ zu überführen.

Status	Tabelle+Spalte	Fehlermeldung	Lösung
Load Pending	SYSIBMADM.ADMINTABINFO LOAD_STATUS: 'PENDING' , 'IN_PROGRESS'	SQL0668N, reason code "3"	Load Restart/Termination
Reorg Pending	SYSIBMADM.ADMINTABINFO REORG_PENDING: 'Y', 'N'	SQL20054N, Reason code="23".	Reorg
Check Pending oder Inoperativ	SYSCAT.TABLES STATUS 'N','C','X'	SQL0668N, reason code "1"	Set Integrity ...
Rollforward Pending	SYSIBMADM.ADMINTABINFO AVAILABLE 'Y','N'	SQL1477N	Restore einer früheren Version

**Table 1: Tabellen Stati**

In Table 2 werden Beispiele beschrieben, wie die Stati aufgelöst werden können.

Status	Beispiel
Load Pending	LOAD FROM t4.del OF DEL TERMINATE INTO state.t4 ;
Reorg Pending	REORG TABLE state.t3 ;
Check Pending oder Inoperativ	SET INTEGRITY FOR state.t2 IMMEDIATE CHECKED FOR EXCEPTION IN state.t2 USE state.t2_ex ;
Rollforward Pending	RESTORE DB sample WITHOUT ROLLING FORWARD WITHOUT PROMPTING

**Table 2: Beispiele zur Lösung der Stati**

Beispielhaft werden in Table 3 Aktionen gelistet, durch die eine Tabelle in den entsprechenden Status übergeführt wird. Dies sind also auch potentielle Ursachen für diese Stati (Liste ist aber nicht vollständig).

Status	Beispiel für Provokation (, bzw. Ursachen der Stati)
Load Pending	<ul style="list-style-type: none"> <li>• Erstelle Tablespace für Daten TSP1D</li> <li>• Erstelle Tablespace für Index TSP1I mit zu kleiner Anzahl Pages</li> <li>• Erstelle Tabelle T1 in diesen Tablespace TSP1D und TSP1I</li> <li>• LOAD in die Tabelle T1</li> </ul> => Nach Füllung des Tablespace TSP1I kommt es zu Fehler SQL0289N und die Tabelle wird in Status „Load Pending“ überführt
Reorg Pending	<ul style="list-style-type: none"> <li>• Erstelle Tabelle T2</li> <li>• Führe vier aufeinanderfolgende ALTER TABLE ADD COLUMN Statements auf der Tabelle T2 durch</li> </ul> => Tabelle ist in „Reorg Pending“ Status
Check Pending oder Inoperativ	<ul style="list-style-type: none"> <li>• Erstelle Tabelle T3P mit Primärschlüssel</li> <li>• Erstelle Tabelle T3C mit Fremdschlüssel auf erste Tabelle T3P</li> <li>• LOAD von Daten in zweite Tabelle T3C , welche ungültige Referenz auf erste Tabelle T3P enthält</li> </ul> => Tabelle kann wegen Check Pending nicht selektiert werden.
Rollforward Pending	<ul style="list-style-type: none"> <li>• Erstelle Tabelle T4 in wiederherstellbarer Datenbank in separatem Tablespace TSP4</li> <li>• Erstelle Tablespace Backup von TSP4</li> <li>• LOAD in die Tabelle T4 im NONRECOVERABLE Mode</li> <li>• RESTORE des Tablespace TSP4 und ROLLFORWARD des Tablespace TSP4</li> <li>• anschließend Backup des Tablespace TSP4 (erforderlich)</li> </ul> => Tabelle ist nicht verfügbar, der Tablespace TSP4 jedoch wieder im Status „normal“.

Table 3: Beispiele zur Provokation der Stati

## Implementierung

Hier die Abfrage, die auf die beiden Catalog Views zugreift.

Der Gesamtstatus „Verfügbar“ wird als 0 notiert, „Nicht Verfügbar“ wird als 1 notiert.

In der CTE (Common Table Expression) „T\_MYSTATE“ wird für die beiden Gesamtstati ein Text hinterlegt. Dieser kann je nach Belieben angepasst werden, das Beispiel verwendet als Text „Good“ und „Bad“. Die Abfrage zeigt nur die Tabellen mit Gesamtstatus 1 an, es kann dies natürlich angepasst werden und stattdessen beispielsweise auf einen Schemanamen gefiltert werden.

Die vier Einzelstati werden durch passende Beschreibungen angezeigt, der Gesamtstatus über CASE Konstrukte ermittelt.

```

WITH t_mytstate ( id , text ) AS ( VALUES (0, 'GOOD'), (1, 'BAD') ),
res (schema, table, tspace, TBSPACEID, IN_VALIDATE_TIME, AVAILABLE, REORG_PENDING, LOAD_STATUS, status, state ) AS
( SELECT
  substr(t.TABSCHEMA,1,20) as SCHEMA,
  substr(t.TABNAME,1,20) as TABLE,
  substr(t.TBSPACE,1,20) as TBSPACE,
  TBSPACEID,
  t.INVALIDATE_TIME as IN_VALIDATE_TIME,
  CASE a.AVAILABLE WHEN 'Y' THEN '-' ELSE 'UNAVAILABLE' END AS AVAILABLE ,
  CASE a.REORG_PENDING WHEN 'Y' THEN 'REORG_PENDING' ELSE '-' END AS REORG_PENDING,
  CASE a.LOAD_STATUS WHEN 'PENDING' THEN 'PENDING' WHEN 'IN_PROGRESS' THEN 'IN_PROGRESS' ELSE '-' END AS
LOAD_STATUS ,
  CASE t.STATUS WHEN 'C' THEN 'CHECK_PENDING' WHEN 'X' THEN 'INOPERATIVE' ELSE '-' END AS STATUS,
  CASE t.STATUS
    WHEN 'N' THEN
      CASE a.AVAILABLE
        WHEN 'Y' THEN
          CASE a.REORG_PENDING
            WHEN 'N' THEN
              CASE a.LOAD_STATUS WHEN 'PENDING' THEN 1 ELSE 0 END
            WHEN 'Y' THEN 1
          END
        WHEN 'N' THEN 1
      END
    WHEN 'C' THEN 1
    WHEN 'X' THEN 1
  END AS state

```

## DB2 Newsletter – Ausgabe 03/2012

```
FROM      SYSCAT.TABLES t ,
          SYSIBMADM.ADMINTABINFO a
WHERE     a.tabschema = t.tabschema
AND       a.tabname = t.tabname
AND t.tabschema NOT LIKE 'SYS%'
)
SELECT   schema, table,
         tbspace, TBSPACEID,
         IN_VALIDATE_TIME, AVAILABLE, REORG_PENDING, LOAD_STATUS, status, t.text AS myState
FROM res r, t_mytstate t
WHERE r.state = t.id
AND state = 1
ORDER BY SCHEMA, TABLE
;
```

### Abfragebeispiel:

In diesem Beispiel sind fünf Tabellen im Gesamtstatus 1 (Nicht Verfügbar), es sind die potentiellen Einzelstati sowie eine Kombination (Tabelle STATE.T6) gelistet.

SCHEMA	TABLE	TBSPACE	TBSPACEID	IN_VALIDATE_TIME	AVAILABLE	REORG_PENDING	LOAD_STATUS	STATUS	MYSTATE
STATE	T2	USERSPACE1	2	2012-09-24-19.58.50.005000	-	-	-	CHECK_PENDING	BAD
STATE	T3	TSP3	55	2012-09-24-19.58.51.674006	-	REORG_PENDING	-	-	BAD
STATE	T4	TSP4	56	2012-09-24-19.58.49.241004	-	-	PENDING	-	BAD
STATE	T5	TSP5	58	2012-09-24-19.58.54.186001	UNAVAILABLE	-	-	-	BAD
STATE	T6	TSP5	58	2012-09-24-19.59.22.172000	UNAVAILABLE	-	-	CHECK_PENDING	BAD

5 record(s) selected.

## TechTipp: ITDS und LDAP\_MAXCARD

### Problem

Ein Aufruf des UNIX Kommandos „/usr/sbin/luser -R LDAP ALL“, welcher alle LDAP User rekursiv listet, dauert auf der Konsole mehrere Minuten, seitdem Benutzer-Daten durch ITDS (IBM Tivoli Directory Server) in DB2 Tabellen gespeichert werden. Bisher dauerte das Kommando nur wenige Sekunden.

### Ursache

Ein SQL Snapshot auf der verwendeten Datenbank zeigt dieses Statement, welches ITDS Tabellen verbindet und ausliest:

```
Number of executions           = 68896
Number of compilations         = 0
Worst preparation time (ms)    = 6
Best preparation time (ms)     = 4
Internal rows deleted          = 0
Internal rows inserted        = 0
Rows read                      = 749726275
Internal rows updated          = 0
Rows written                   = 374863136
Statement sorts                = 137792
Statement sort overflows       = 68896
Total sort time                = 212672
Buffer pool data logical reads = 10265507
Buffer pool data physical reads = 0
Buffer pool temporary data logical reads = 23149056
Buffer pool temporary data physical reads = 0
Buffer pool index logical reads = 1127759992
Buffer pool index physical reads = 0
Buffer pool temporary index logical reads = 0
Buffer pool temporary index physical reads = 0
Buffer pool xda logical reads  = 0
Buffer pool xda physical reads = 0
```

```
Buffer pool temporary xda logical reads      = 0
Buffer pool temporary xda physical reads    = 0
Total execution time (sec.microsec)= 1563.935992
Total user cpu time (sec.microsec) = 970.538885
Total system cpu time (sec.microsec)= 4.184714
Total statistic fabrication time (milliseconds) = 0
Total synchronous runstats time (milliseconds) = 0
Statement text
= SELECT SULDAP62.OBJECTCLASS.EID , ALIASEDOBJECT FROM
SULDAP62.OBJECTCLASS, SULDAP62.ALIASEDOBJECT, SULDAP62.LDAP_DESC WHERE SULDAP62.LDAP_DESC.AEID=? AND
SULDAP62.LDAP_DESC.DEID = SULDAP62.OBJECTCLASS.EID AND SULDAP62.LDAP_DESC.DEID <>
SULDAP62.LDAP_DESC.AEID AND ( SULDAP62.OBJECTCLASS.OBJECTCLASS=? OR
SULDAP62.OBJECTCLASS.OBJECTCLASS=?) AND SULDAP62.OBJECTCLASS.EID = SULDAP62.ALIASEDOBJECT.EID FOR
FETCH ONLY
```

Dieses Statement dauert im Schnitt zwar nur 0,02 s, in Summe durch die häufigen Aufrufe aber ca. 26 Minuten.

Der Explain dazu (siehe Anhang) zeigt immens hohe Kosten (6.38678e+09), der Indexbaum wird wiederholt gelesen.

Auffallend ist, dass die Kardinalität der Tabelle SULDAP62.LDAP\_DESC verfälscht wurde:

```
Number of rows:          90000000000000000000
```

Das Zurücksetzen durch einen Standard Runstats löst das Problem nicht, der Wert wird wieder überschrieben, auch wenn Automatic Maintenance auf der Datenbank deaktiviert wird.

## Lösung

Im ITDS muss im Profile von root **LDAP\_MAXCARD=NO** und export **IBMSLDAP\_USE\_SELECTIVITY=YES** gesetzt werden.

Durch LDAP\_MAXCARD=NO wird das periodische Überschreiben der Kardinalität vermieden (bei ONCE erfolgt dies einmalig bei jedem Start des ITDS, bei YES periodisch – das ist leider der Default!).

Ein einmaliger Runstats (db2 RUNSTATS ON TABLE SULDAP62.LDAP\_DESC WITH DISTRIBUTION AND DETAILED INDEXES ALL ) nach dem Setzen der Werte und Neustart des ITDS löst das Problem. Die Statistikwerte werden nicht mehr überschrieben, die Laufzeit beträgt wieder nur wenige Sekunden.

Es wird auch ein anderer Zugriffsplan gewählt, die Kosten liegen nun nur noch bei ca. 200.

## Links:

- 1) [LDAP\\_MAXCARD behavior in ITDS versions 5.x, 6.0, 6.1, 6.2 and 6.3.](#)
- 2) [DB2 SELECTIVITY](#)

## Chats mit dem Labor

Eine Liste der bereits durchgeführten Chats ist [hier](#) zu finden.

Die Präsentationen der Chats können dort angeschaut und heruntergeladen werden.

## Schulungen / Tagungen / Informationsveranstaltung

Eine Liste der anstehenden Konferenzen ist [hier](#) zu finden.

### Friendly Reminder: DB2 LUW Working Group

Das diesjährige Arbeitstreffen der DB2 LUW Working Group findet am 13./14.12.2012 in Heidelberg statt. Also nicht vergessen.

## Newsletter Archiv

Alte Ausgaben vom DB2-NL sind nun zum Nachlesen in den Archiven zu finden von:

- [Lis.Tec](#)
- [Bytec](#)
- [Drap](#)
- [Cursor Software AG](#)
- [ids-System GmbH](#)

## Anmeldung/Abmeldung

Sie erhalten diesen Newsletter bis zur 3ten Ausgabe ohne Anmeldung. Wenn Sie weiterhin diesen Newsletter empfangen wollen, schicken Sie Ihre Anmeldung mit dem Subject „ANMELDUNG“ an [db2news@de.ibm.com](mailto:db2news@de.ibm.com).

## Die Autoren dieser Ausgabe

Sollten Sie Anfragen zu den Artikeln haben, können Sie sich entweder direkt an den jeweiligen Autor wenden oder stellen Ihre Frage über den DB2 NL, denn vielleicht interessiert ja die Antwort auch die anderen DB2 NL Leser.

Doreen Stein	Certified IT-Spezialist für DB2 LUW, IBM SWG; Chief-Editor DB2NL, <a href="mailto:djs@de.ibm.com">djs@de.ibm.com</a>
Peter Schurr	IT-Spezialist für DB2 LUW, IBM SWG; Artikel: TechTipp: DB2 System ODBC Datasources auf X64 und TechTipp: DB2 Tabellen Status, sowie

### Reviewer und Ideenlieferanten:

Peter Schurr	IBM SWG
Wilfried Hoge	IBM SWG
Frank Berghofer	IBM SWG
Dirk Fechner	IBM SWG