

Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

Der Sommer, naja, soweit man diese extremen Wetterlagen als Sommer bezeichnen kann, ist vorbei. Der Herbst wartete bereits mit sehr schönen aber inzwischen auch wieder A-kalten Tagen auf.

Sehr schöne Momente konnten auch die Teilnehmer der DB2 Aktuell erleben. Die dort erworbenen Kenntnisse wurden in DB2 Security umgesetzt, damit Sie sicher durch den Winter kommen.

Wie immer wünschen wir viel Spaß beim Lesen.



Für Fragen und Anregungen unsere Kontaktadresse: db2news@de.ibm.com.

Ihr TechTeam

Inhaltsverzeichnis

| | |
|---|---|
| CHATS MIT DEM LABOR..... | 1 |
| WHAT'S NEW IN DB2 V9.7 FP4..... | 2 |
| DB2 AKTUELL RÜCKBLICK..... | 2 |
| TECHTIPP: VERKETTEN VON AUSGABEN..... | 2 |
| DIE BERECHTIGUNGEN FÜR EINEN USER ERMITTELN UND ENTFERNEN..... | 4 |
| TECHTIPP: DIE AUTHORITIES FÜR EINEN USER ERMITTELN UND ENTFERNEN..... | 6 |
| SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG..... | 7 |
| NEWSLETTER ARCHIV..... | 7 |
| ANMELDUNG/ABMELDUNG..... | 7 |
| DIE AUTOREN DIESER AUSGABE..... | 8 |

Chats mit dem Labor

Eine Liste der bereits durchgeführten Chats ist [hier](#) zu finden.
Die Präsentationen der Chats können dort angeschaut und heruntergeladen werden.

Sie hatten auch keine Chance am Chat mit dem Labor zum Thema **DB2 and Oracle - An Automatic Computing Comparison** teilzunehmen? Dann finden Sie hier die [Presentation Slides \(.pdf, 2.4MB\)](#) zum Download.

What's new in DB2 V9.7 FP4

Sind Sie auch schon über Dinge gestolpert, die es vorher nicht gab bzw. vorher nicht aufgefallen sind?

Bei der Auswertung der Instanz und Datenbank Konfiguration bin ich auf ALT_DIAGPATH gestoßen. Der Name ist eigentlich schon recht aussagekräftig. Aber es kam schon die Frage rein, wann dieser Parameter reinkam. Der Titel des Artikels lässt natürlich ahnen, im FP4.

Andere [Punkte des FP4](#) sind:

- Der [DB2 Health Advisor](#) ist deprecated, das alternative Tool ist IBM Data Studio Health Monitor
- [Überprüfung der archivierten Logfiles](#) vor dem rollforward, mittels [db2cklog](#)
- Verbesserung [Service](#) für große Datenbanken
- neue Funktionen (siehe nachfolgenden Beitrag)
- Windows: Der Pfad zur [db2diag.log](#) hat sich verändert
- Erweiterte Funktionalität für Tools
 - db2look
 - db2pd -recovery
-

Sie sehen also, es kommen neue Features mit den Fix Packs, daher halten Sie die Fix Pack News im Auge.

DB2 Aktuell Rückblick

Die DB2 Aktuell, die Mitte September, diesmal in Bremen, stattfand, ist vorüber. Für die Teilnehmer wurden interessante Beiträge geliefert, angereichert mit Highlights, wie Stadtrundgang mit anschließendem Sekt-Empfang im neuen Rathaus. Der Höhepunkt des Abends war dann das große Buffet im historischen Saal des alten Rathauses. Der historische Saal bietet ein wunderschönes Ambiente, um sich näher kennenzulernen und Informationen und Erfahrungen rund um das Thema DB2 auszutauschen.

Einiges, was ich dort erfahren bzw. schon mal gewusst, wieder vergessen und daher wieder aufgefrischt habe, habe ich bereits für diese Ausgabe ausprobiert.

Themen im LUW Bereich waren:

- | | |
|----------------------------|----------------------------------|
| • DB2 HA mit Q-Replication | Christian Lenke und Olaf Stephan |
| • DB2 LUW Page Cleaning | Matt Huras |
| • DB2 pureScale | |
| • Always up 2 date | |
| • The latest from the Lab | Matt Huras |
| • Explain mal schnell | Michael Tiefenbacher |
| • DB2 LUW Internals | Matt Huras |
| • SQL on Fire III | Serge Rielau |
| • DB2 und SAS | Matthias Nicola |
| • XML Table function | Matthias Nicola |

Hier nun der erste Beitrag, basierend auf der Session SQL on Fire von Serge Rielau.

TECHTIPP: Verketteten von Ausgaben

Serge Rielau hat auf der DB2 Aktuell 2011 einen Beitrag zum Thema SQL on Fire, Part III gebracht.

Dabei ist mir folgende Funktion besonders ins Auge gefallen, die schon seit V9.5 existiert, mir jedoch bisher entgangen war, und deren Funktionalität ich bisher mittels Shell-Skripting abgedeckt hatte.

Sollte es bei Ihnen auch so sein, hier ein paar Beispiele, wie man diese Funktion einsetzen kann.

[LISTAGG](#)

Mögliche Varianten des Aufrufs sind:

| | |
|---|---|
| <code>listagg(<Spalte>)</code> | Ausgabe einer einfacher blank separierter List. |
| <code>listagg(<Spalte>, '<Seperator>')</code> | Ausgabe einer Seperator-serparierten Liste. |
| <code>listagg(<Spalte>, '<Seperator>') WITHIN GROUP (ORDER by <Spalte> ASC DESC)</code> | |

Ist die Menge der Daten, die zusammengefügt werden soll, größer als 4000 Bytes, kommt folgende Fehlermeldung:

```
db2 -x "select objecttype,listagg(privilege,',') within group (order by privilege) from
sysibmadm.privileges group by objecttype order by objecttype"
SQL0137N The length resulting from "LISTAGG" is greater than "4000".
SQLSTATE=54006
```

Nun ein paar Beispiele, wie man diese Funktion einsetzen kann:

Möchte man eine Komma-separierte Liste der Schemata, erhalten, kann dies erfolgen mit:

```
db2 -x "select listagg(schename,',') from syscat.schemata"
DOREEN ,NULLID ,SQLJ ,SYSCAT ,SYSFUN ,SYSIBM ,SYSIBMADM,SYSIBMINTERNAL,SYSIBMTS,SYSPROC
,SYSPUBLIC,SYSSTAT ,SYSTOOLS
```

Um die vielleicht störenden Leerzeichen in der Aufzählung zu entfernen, wird das SQL wie folgt erweitert :

```
db2 -x "select listagg(rtrim(schename),',') from syscat.schemata"
DOREEN,NULLID,SQLJ,SYSCAT,SYSFUN,SYSIBM,SYSIBMADM,SYSIBMINTERNAL,SYSIBMTS,SYSPROC,SYSPUBLIC,SYSSTAT,SY
STOOLS
```

Wenn man beispielsweise wissen möchte, welche Rechte auf welchen Objekttypen vergeben wurden, kann dies erfolgen mit:

```
select objecttype,listagg(privilege,',') within group (order by privilege)
from (select distinct objecttype, privilege
from sysibmadm.privileges) as a
group by objecttype order by objecttype
```

| OBJECTTYPE | 2 |
|-------------|--|
| DB2 PACKAGE | BIND, CONTROL, EXECUTE |
| FUNCTION | EXECUTE |
| INDEX | CONTROL |
| MODULE | EXECUTE |
| PROCEDURE | EXECUTE |
| SCHEMA | CREATEIN, DROPIN |
| TABLE | ALTER, CONTROL, DELETE, INDEX, INSERT, REFERENCE, SELECT, UPDATE |
| TABLESPACE | USE |
| VIEW | SELECT, UPDATE |
| WORKLOAD | USAGE |
| XML OBJECT | USAGE |

Verändert man dieses Beispiel nun um AUTHID und OBJECTNAME, dann könnte das Ergebnis wie folgt aussehen:

```
select AUTHID, OBJECTNAME, listagg(PRIVILEGE, ',') within group (order by privilege)
from sysibmadm.privileges
group by authid, objectname
fetch first 1 row only
```

| AUTHID | OBJECTNAME | 1 |
|--------|------------|------------------------|
| DOREEN | AOTMH00 | BIND, CONTROL, EXECUTE |

Mit dieser Funktion kann man sich die Umwandlung der Ausgabe mit Verskriptung auf Command Line Ebene ersparen.

Serge ist in seiner Präsentation auch auf das Gegenteil der LISTAGG Funktion eingegangen. Momentan gibt es dafür noch keine vordefinierte Funktion. Daher hier die Funktion, die Serge dazu anbietet:

```
CREATE FUNCTION SPLIT(text VARCHAR(32000), split VARCHAR(10))
RETURNS TABLE(element VARCHAR(60))
RETURN WITH rec(rn, column_value, pos) AS
  (VALUES (1, VARCHAR(SUBSTR(text, 1,
    DECODE(INSTR(text, split, 1),0,LENGTH(text),INSTR(text, split, 1) - 1)),
    255),
    INSTR(text, split, 1) + LENGTH(split))
  UNION ALL
  SELECT rn + 1, VARCHAR(SUBSTR(text, pos,DECODE(INSTR(text, split, pos),0,
    LENGTH(text) - pos + 1,
    INSTR(text, split, pos) - pos)),255),
    INSTR(text, split, pos) + LENGTH(split)
  FROM rec WHERE rn < 30000 AND pos > LENGTH(split))

  SELECT column_value FROM rec
@
```

Erzeugen der Funktion: db2 -td@ -f split.ddl

```
SELECT phone FROM emp JOIN TABLE(split('John,Jack,Jill', ','))ON emp.name = split.element@
PHONE
-----
4165554321
9055551234
-
3 record(s) selected.
```

Hier sind die Links zu

- Blog Beitrag zum „[aggregating strings](#)“
- Blog Eintrag zum Thema „[How to split a string into a set of rows \(anti-LISTAGG\)](#)“
- [Serges Blog](#)

Die Berechtigungen für einen User ermitteln und entfernen

In früheren Versionen hat man jede einzelne AUTH-Views/Tabellen abfragen müssen, um alle Rechte für einen User oder Gruppe zu ermitteln.

```
db2 „select * from SYSCAT.TABAUTH where GRANTEE = 'DOREEN'“
db2 „select * from SYSCAT.TBSPACEAUTH where GRANTEE = 'DOREEN'“
...

```

Um diese Abfragen zu vereinfachen gibt seit DB2 V9.5 den View [SYSIBMADM.PRIVILEGES](#).

```
db2 "describe table sysibmadm.privileges"
Column name          Data type          Data type name     Column
ulls                 schema            Data type name     Length   Scale N
-----
AUTHID               SYSIBM            VARCHAR             128      0 No
AUTHIDTYPE           SYSIBM            CHARACTER            1        0 No
PRIVILEGE            SYSIBM            VARCHAR              11        0 No
GRANTABLE            SYSIBM            VARCHAR              1         0 Yes
OBJECTNAME           SYSIBM            VARCHAR             128      0 Yes
OBJECTSCHEMA         SYSIBM            VARCHAR             128      0 No
OBJECTTYPE           SYSIBM            VARCHAR              24        0 Yes

7 record(s) selected.
```

Mit diesem View können alle Rechte für Views, Tabellen, Schemas und Tablespaces, Packages, ... angezeigt werden, da die Inhalte der AUTH-Objekte über UNION ALL zusammengefügt werden. Dabei ergibt jedes Recht auf ein Objekt einen eigenen Datensatz. Nicht enthalten sind jedoch die Authorities auf Datenbank-Ebene, wie CONNECT, EXPLAIN, DBADM, LOAD, usw.

Möchte man nun wissen, welche Rechte User DOREEN hat, kann dies erfolgen mit:

```
db2 "select * from sysibmadm.privileges where AUTHID='DOREEN' fetch first 5 rows only"
AUTHID AUTHIDTYPE PRIVILEGE GRANTABLE OBJECTNAME OBJECTSCHEMA OBJECTTYPE
```

```
-----
DOREEN U      EXECUTE  Y      SQLE0H00    NULLID      DB2 PACKAGE
DOREEN U      EXECUTE  Y      SQLE2H0V    NULLID      DB2 PACKAGE
DOREEN U      EXECUTE  Y      P7260389    SYSIBMADM   DB2 PACKAGE
DOREEN U      EXECUTE  Y      P7260503    SYSIBMADM   DB2 PACKAGE
DOREEN U      EXECUTE  Y      P7260510    SYSIBMADM   DB2 PACKAGE
```

5 record(s) selected.

Möchte man wissen, welche Rechte für welche Objekttypen existieren, kann dies erfolgen mit:

```
db2 "select distinct objecttype,privilege from sysibmadm.privileges order by objecttype"
```

```
OBJECTTYPE      PRIVILEGE
-----
DB2 PACKAGE     BIND
DB2 PACKAGE     CONTROL
DB2 PACKAGE     EXECUTE
FUNCTION         EXECUTE
MODULE           EXECUTE
PROCEDURE        EXECUTE
```

6 record(s) selected.

Sollen nun die Rechte pro Objekt in einer Zeile dargestellt werden, kann dies unter Verwendung der Funktion LISTAGG (s.h. Beitrag LISTAGG) erfolgen:

```
db2 "select authid, rtrim(objectschema)||'.'||objectname, substr(listagg(PRIVILEGE,',')
within group (order by privilege),1,120) from sysibmadm.privileges where AUTHID='DOREEN' group by au-
thid, rtrim(objectschema)||'.'||objectname order by 2 fetch first 5 rows only"
```

```
AUTHID          2                               3
-----
DOREEN          NULLID.AOTMH00                          BIND,CONTROL,EXECUTE
DOREEN          NULLID.ATSH04                          BIND,CONTROL,EXECUTE
DOREEN          NULLID.DB2XDBMI                        BIND,CONTROL,EXECUTE
DOREEN          NULLID.PRINTSG                          BIND,CONTROL,EXECUTE
DOREEN          NULLID.REVALH03                        BIND,CONTROL,EXECUTE
```

5 record(s) selected.

Möchte man nun die Rechte für einen bestimmten User (im Beispiel User DOREEN) entfernen, kann dies basierend auf der Spalte PRIVILEGE entsprechend des Objekttyps erfolgen und könnte wie folgt aussehen (nur ein Auszug)

```
db2 "select objecttype,privilege,objschema, objectname from sysibmadm.privileges where
AUTHID='DOREEN'" | while read objtype priv objschema objname rest; do
    case objtype in
        "TABLE") db2 revoke $priv on $objschem.$objname from USER 'DOREEN';;
        "VIEW")  ...
    esac
done
```

Wer aber nicht der Fan von Shell-Skripting ist und dies lieber mittels SQL-Kommando machen möchte, kann dies mit folgender Abfrage durchführen:

```
VALUES '--- Usage: db2 -txf revoke_privileges.sql -z revoke_privileges.log ' ;

WITH orderdRank ( AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA,OBJECTTYPE, orderdRank) AS (
--- Ranking porder , CONTROL must be REVOKED before other PRIVILEGES , preventing SQL0558N
SELECT
AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE,
CASE
    privilege
    WHEN 'EXECUTE' THEN 1
    WHEN 'BIND' THEN 1
    WHEN 'CONTROL' THEN 2
ELSE 0
END AS orderdRank
FROM sysibmadm.privileges
--where AUTHID='DOREEN'-- diese Abfrage kann mit rein, muss aber nicht, da sonst an 2 Stellen korri-
giert
)
select 'REVOKE ' ||
case a.OBJECTTYPE
WHEN 'GLOBAL VARIABLE' THEN TRIM(a.privilege) || ' ON VARIABLE '
```

```

WHEN 'INDEX' THEN TRIM(a.privilege) || ' ON INDEX '
WHEN 'DB2 PACKAGE' THEN TRIM(a.privilege) || ' ON PACKAGE '
WHEN 'FUNCTION' THEN TRIM(a.privilege) || ' ON SPECIFIC FUNCTION '
WHEN 'PROCEDURE' THEN TRIM(a.privilege) || ' ON SPECIFIC PROCEDURE '
WHEN 'MODULE' THEN TRIM(a.privilege) || ' ON MODULE '
WHEN 'SCHEMA' THEN TRIM(a.privilege) || ' ON SCHEMA '
WHEN 'SEQUENCE' THEN TRIM(a.privilege) || ' ON SEQUENCE '
WHEN 'TABLE' THEN TRIM(a.privilege) || ' ON '
WHEN 'SERVER' THEN TRIM(a.privilege) || ' ON SERVER '
WHEN 'SETSESSIONUSER' THEN TRIM(a.privilege) || ' ON '
WHEN 'VIEW' THEN TRIM(a.privilege) || ' ON '
WHEN 'NICKNAME' THEN TRIM(a.privilege) || ' ON '
WHEN 'TABLESPACE' THEN TRIM(a.privilege) || ' OF TABLESPACE '
WHEN 'WORKLOAD' THEN TRIM(a.privilege) || ' ON WORKLOAD '
ELSE '@' || a.OBJECTTYPE
END
--- tablespace, schema, workload uses different objectnames
|| case a.OBJECTTYPE
WHEN 'TABLESPACE' THEN TRIM(a.OBJECTNAME)
WHEN 'WORKLOAD' THEN TRIM(a.OBJECTNAME)
WHEN 'SCHEMA' THEN TRIM(a.OBJECTSCHEMA)
ELSE TRIM(a.OBJECTSCHEMA) || '.' || TRIM(a.OBJECTNAME)
END
|| ' FROM USER ' || a.AUTHID
--- functions and procedures uses keyword RESTRICT
|| case a.OBJECTTYPE
WHEN 'FUNCTION' THEN ' RESTRICT ;'
WHEN 'PROCEDURE' THEN ' RESTRICT ;'
ELSE ' ; '
END
from sysibadm.privileges a
JOIN orderdRank b
ON
a. AUTHID = b.AUTHID
AND a.PRIVILEGE = b.PRIVILEGE
AND a.OBJECTNAME = b.OBJECTNAME
AND a.OBJECTSCHEMA = b.OBJECTSCHEMA
AND a.OBJECTTYPE = b.OBJECTTYPE
where a.AUTHID='DOREEN'
ORDER BY orderdRank DESC
;

```

Die Ausgabe sieht dann ungefähr wie folgt aus:

```

>db2 -txf gen_revoke_privileges.sql
--- Usage: db2 -txf revoke_privileges.sql -z revoke_privileges.log

REVOKE CONTROL ON PACKAGE SYSIBMADM.P7260389 FROM USER DOREEN ;
REVOKE CONTROL ON PACKAGE NULLID.SQLE0H00 FROM USER DOREEN ;
REVOKE CONTROL ON PACKAGE NULLID.SQLE2H0V FROM USER DOREEN ;
REVOKE CONTROL ON PACKAGE SYSIBMADM.P7261462 FROM USER DOREEN ;
REVOKE CONTROL ON PACKAGE SYSIBMADM.P7260503 FROM USER DOREEN ;
REVOKE CONTROL ON PACKAGE SYSIBMADM.P7260510 FROM USER DOREEN ;

```

Bei der Entfernung der Rechte eines Objekt-Owners, muss sichergestellt werden, dass adäquate Rechte für andere Rollen/User/Usergruppen vorhanden sind. Zum Beispiel ist es kritisch EXECUTE Rechte für PUBLIC von Packages und Funktions zu entfernen.

TechTipp: Die Authorities für einen User ermitteln und entfernen

In dem vorhergehenden Artikel habe ich beschrieben, wie die Rechte an Objekten für einen User ermittelt werden kann, mittels SYSIBMADM.PRIVILEGES.

In diesem Artikel habe ich darauf hingewiesen, das PRIVILEGES nicht die Datenbank Authorities beinhaltet.

Darauf hin habe ich mich hingewetzt und mittels unpivoting eine Liste zu erstellen, die ähnlich der Ausgabe des Views PRIVILEGES aussieht. Hier nun die entsprechende SQL dazu:

```

select grantee authid, granteetype authidtype, authority, case wert when 'G' then 'G' else NULL end
from syscat.dbauth as d,
lateral (values ('BINDADD',d.bindaddauth),
('CONNECT',d.connectauth),
('CREATETAB',d.createtabauth),

```

```
        ('DBADM',d.dbadmauth),
        ('EXTERNALROUTINE',d.externalroutineauth),
        ('IMPLSCHEMA',d.implschemaauth),
        ('LOAD',d.loadauth),
        ('NOFENCE',d.nofenceauth),
        ('QUIESCECONNECT',d.quiesceconnectauth),
        ('LIBRARYADM',d.libraryadmauth),
        ('SECURITYADM',d.securityadmauth),
        ('SQLADM',d.sqladmauth),
        ('WLMADM',d.wlmadmauth),
        ('EXPLAIN',d.explainauth),
        ('DATAACCESS',d.dataaccessauth),
        ('ACCESSCTRL',d.accessctrlauth) as T (AUTHORITY, WERT)

where wert!= 'N'
;
```

Die Ausgabe sieht dann wie folgt aus:

```
>db2 -tf gen_auth_privileges.sql
```

| AUTHID | AUTHIDTYPE | AUTHORITY | 4 |
|--------|------------|-------------|---|
| DOREEN | U | DBADM | - |
| DOREEN | U | SECURITYADM | - |
| DOREEN | U | DATAACCESS | - |
| DOREEN | U | ACCESSCTRL | - |
| PUBLIC | G | BINDADD | - |
| PUBLIC | G | CONNECT | - |
| PUBLIC | G | CREATETAB | - |
| PUBLIC | G | IMPLSCHEMA | - |

8 record(s) selected.

Ich habe einige Zeit gebraucht das LATERAL in der DB2 Online Dokumentation zu finden. Aber am Ende hat es doch geklappt und dieses LATERAL gibt es schon mindestens seit DB2 V9.1.
nested-table-expression

```
|-----+----- (fullselect)-----|
| (2) |
|'-LATERAL-----+-----|
|'-| continue-handler |--WITHIN-'
```

Denn dort wurde ich als erste fündig. Hier nun der [Link](#) zu der Information basierend auf V9.7.

Hier noch der Link zum [PIVOT/UNPIVOT](#) Artikel von Serge Rielau, der mich überhaupt erst auf diese Idee brachte.

Schulungen / Tagungen / Informationsveranstaltung

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

Newsletter Archiv

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

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

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.

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, djs@de.ibm.com |
| Peter Schurr | IT-Spezialist für DB2 LUW, IBM SWG REVOKE-IN SQL aus Artikel: Die Berechtigungen für einen User ermitteln und entfernen |
| | |

Reviewer und Ideenlieferanten:

| | |
|---------------|--------------------------------------|
| Dirk Fechner | IBM SWG |
| Nela Krawez | IBM SWG |
| Wilfried Hoge | IBM SWG |
| Peter Schurr | IBM SWG |
| Serge Rielau | STSM, SQL Architect, IBM Toronto Lab |