

## Willkommen zum „IBM DB2 Newsletter“

### Liebe Leserinnen und Leser,

Die Sommerzeit wurde wieder zurückgestellt, die Abende/Nächte werden länger. Nicht mehr lange, dann steht Weihnachten vor der Tür und das Jahr ist vorüber.

Bis dahin hat DB2 Development noch einiges vor. So sind wichtige DB2 Releases geplant, wie

- [PureScale](#) im [Dezember](#)
- v9.7 FP1 Ende November
- v9.5 FP5 Anfang Dezember



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

Ihr TechTeam

## Inhaltsverzeichnis

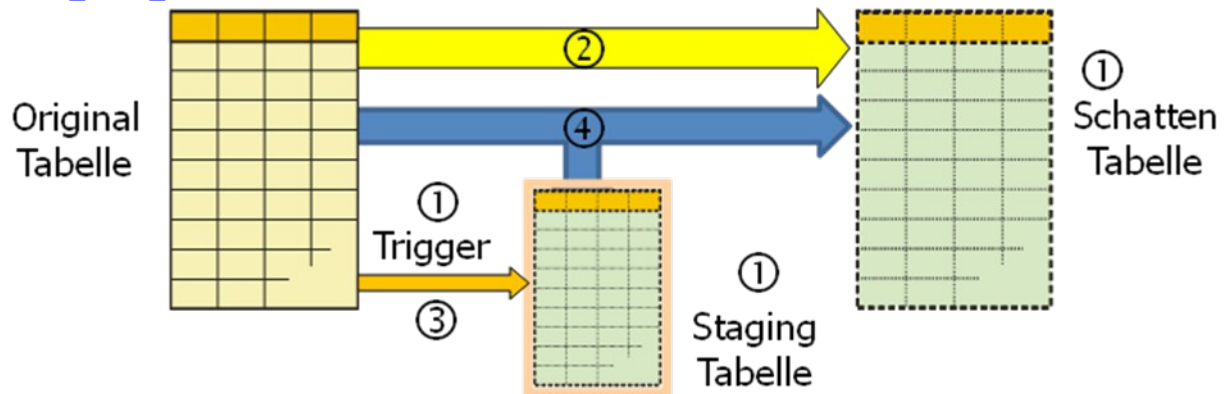
<a href="#">ARTIKELSERIE: NEUE EIGENSCHAFTEN IN DB2 V9.7.....</a>	<a href="#">2</a>
<a href="#">ONLINE TABLE MOVE – NICHT NUR FÜR DB2 9.7 .....</a>	<a href="#">2</a>
<a href="#">DB2 AUDIT IN DB2 V9.5 (TEIL 2 – AUDIT-INFORMATIONEN AUSLESEN).....</a>	<a href="#">5</a>
<a href="#">SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....</a>	<a href="#">8</a>
<a href="#">KURSE BIS ZUM JAHRESENDE.....</a>	<a href="#">8</a>
<a href="#">CHATS MIT DEM LABOR.....</a>	<a href="#">9</a>
<a href="#">NEWSLETTER ARCHIV.....</a>	<a href="#">9</a>
<a href="#">ANMELDUNG/ABMELDUNG.....</a>	<a href="#">9</a>
<a href="#">DIE AUTOREN DIESER AUSGABE:.....</a>	<a href="#">9</a>

## Artikelserie: Neue Eigenschaften in DB2 V9.7

### Online Table Move – nicht nur für DB2 9.7

DB2 9.7 hat einiges an Neuerungen zu bieten. Darunter sind auch einige von Datenbank-Administratoren lang erwarteten und herbei gewünschten Funktionen, zu denen auch die Möglichkeit zählt, Tabellen online in einen anderen Tabellenbereich zu verschieben und dabei evtl. auch noch die Tabellendefinition zu verändern.

[ADMIN\\_MOVE\\_TABLE](#) ist der Name für die in DB2 9.7 zur Verfügung gestellte Stored Procedure.



Zuerst einmal die grundlegende Funktionsweise:

Beim Aufruf der Stored Procedure werden zuerst (siehe ①) in der sogenannten INIT-Phase die notwendigen Hilfsobjekte erzeugt:

- Eine Schattentabelle – die zukünftige neue Tabelle
- Eine Stagingtabelle – mit deren Hilfe die Änderungen bis zur Fertigstellung der Tabellenübertragung zwischengespeichert werden
- Trigger – die Änderungen auf der Originaltabelle erfassen und in die Stagingtabelle übertragen

In einer weiteren Phase – der COPY-Phase (②) – werden dann die Daten von der Originaltabelle in die Schattentabelle übertragen. Dies kann sowohl über Inserts als auch per LOAD erfolgen. Alle Änderungen werden dann – wie in ③ gezeigt – über die angelegten Trigger erfasst und die Schlüssel der veränderten Datensätze werden in der Schattentabelle gespeichert. Die sogenannte SWITCH-Phase (④) stellt die Übertragung fertig indem die veränderten Datensätze erneut in die Stagingtabelle übertragen werden und – nachdem die Zieltabelle umbenannt wurde – die temporären Objekte entfernt werden (CLEANUP-Phase). Nur für die Umbenennung der Schattentabelle, die statt ihrem temporären Namen den Originaltabellennamen erhält, ist das Objekt „offline“ durch die Sperrung im Systemkatalog. Dieser Aufruf, um die Tabelle PROD.PERSONNEL in den Tabellenbereich TS\_DATA zu übertragen, könnte beispielsweise so aussehen:

```
call ADMIN_MOVE_TABLE('PROD', 'PERSONNEL',
                      'TS_DATA', 'TS_INDEX', 'TS_LOB',
                      '', '', '', '', '', 'MOVE')
```

Neben dem Schema und Tabellennamen der zu verschiebenden Tabelle, wird hier gleich dreimal der Zieltabellenbereich angegeben, so könnten verschiedene Tabellenbereiche für Daten, Indizes und Large Objects spezifiziert werden. Die vielen Optionen der Stored Procedure lassen vermuten, dass noch mehr möglich ist – und das ist auch so.

Generell gibt es zwei Varianten der Parametrisierung – die oben gezeigte, die den ganzen „Move“-Vorgang in einem Rutsch durchführen kann und eine weitere mit etwas weniger Optionen, bei der die Zieltabelle manuell vorab angelegt wird, was eine erhöhte Flexibilität bei der Tabellendefinition zulässt.

Für u.a. folgende Zwecke kann diese Stored Procedure nützlich sein:

- Tabelle online umzuwandeln in eine
  - MDC-Tabelle
  - Range-Partitionierte Tabelle
  - Database Partitioning Tabelle
  - komprimierte Tabelle
- Tabelle in einen anderen Tabellenbereich (auch mit anderer Pagesize etc.) zu verschieben

- wegen Größenlimits der Tabellenbereiche
- um neuen DB2 9.7 Tabellenbereich mit „Reclaimable“ Storage nutzen zu können

Auch einige Einschränkungen der Online Table Move Funktionalität sollten nicht unerwähnt bleiben. In folgenden Fällen ist das Verschieben einer Tabelle mit dieser Stored Procedure nicht möglich:

- für MQTs (Materialized Query Tables)
- für Tabellen mit referentieller Integrität
- für Tabellen mit generierten Spalten
- die Registry Variable DB2\_SKIPDELETED darf nicht aktiviert sein

Zur Syntax:

Methode 1 – Aufruf mit manuell angelegter Zieltabelle:

```
ADMIN\_MOVE\_TABLE(tabschema, tabname, ziel_tabname,  
option, operation)
```

Methode 2 – Aufruf mit automatisch erzeugter Zieltabelle:

```
ADMIN\_MOVE\_TABLE(tabschema, tabname,  
data_tbsp, index_tbsp, lob_tbsp,  
mdc_cols, partkey_cols, range_part, coldef,  
option, operation)
```

Die Parameter haben dabei folgende Bedeutung:

<i>tabschema</i> :	Schema der zu verschiebenden Quelltable
<i>tabname</i> :	Name der zu verschiebenden Tabelle
<i>option</i> :	Optionen die – abhängig von der Operation – gesetzt werden können
<i>operation</i> :	die Operationen erlauben es den kompletten Verschiebeprozess als Einzelaufruf abzusetzen (MOVE) oder jede einzelne Phase manuell zu starten (INIT, COPY, REPLAY, VERIFY, SWAP, CLEANUP, CANCEL)
<i>ziel_tabname</i> :	Name der manuell erzeugten Zieltabelle (wird am Ende in <i>tabname</i> umbenannt)
<i>*_tbsp</i> :	Namen der Zieltabellenbereiche für Daten, Index und Large Objects
<i>mdc_cols</i> :	Liste mit MDC-Dimensionsspalten, z. B.: 'C1, C3'
<i>partkey_cols</i> :	Liste mit Partitionierungsspalten DISTRIBUTE BY HASH(...)
<i>range_part</i> :	Range Partitionierungsklausel, z. B.: '(C1) (STARTING FROM (1) ENDING AT (1000) EVERY (100))'
<i>coldef</i> :	Liste mit Spaltendefinition der Zieltabelle

Beispiele sagen oft mehr als viele Worte.

Das folgende Beispiel zeigt einen Aufruf der Stored Procedure nach Methode 1 – bei der die Zieltabelle PROD.PERSONNEL\_T bereits vorab angelegt worden ist - in mehreren Schritten (Phasen):

```
call ADMIN\_MOVE\_TABLE('PROD', 'PERSONNEL', 'PERSONNEL_T', '', 'INIT');  
call ADMIN\_MOVE\_TABLE('PROD', 'PERSONNEL', 'PERSONNEL_T', '', 'COPY');  
call ADMIN\_MOVE\_TABLE('PROD', 'PERSONNEL', 'PERSONNEL_T', '', 'SWAP');
```

Somit lassen sich die einzelnen Phasen zeitlich trennen und auch während der Verarbeitung noch weitere Optionen spezifizieren. Dazu gibt es noch eine spezielle Zusatz-Prozedur [ADMIN\\_MOVE\\_TABLE\\_UTIL](#). So könnte man im obigen Beispiel nach dem ersten Schritt (INIT-Phase) eine Commit-Frequenz spezifizieren, um Logging-Probleme zu vermeiden:

```
call ADMIN\_MOVE\_TABLE\_UTIL('PROD', 'PERSONNEL', 'UPSERT',  
'COMMIT_AFTER_N_ROWS', '1000');
```

Folgendes Beispiel verschiebt die Tabelle PROD.PERSONNEL in einem Schritt in den Zieltabellenbereich „MOVEZIEL3“ und wandelt diese dabei noch in eine MDC-Tabelle mit nur einer Dimension (JOB\_FTN) um und partitioniert sie gleichzeitig nach EMPNO-Ranges. Damit nicht genug – zusätzlich wird die Datenübertragung per (unrecoverable) LOAD durchgeführt und die Zieltabelle noch reorganisiert. Die KEEP-Option stellt sicher, dass die ursprüngliche Tabelle am Ende nicht gelöscht wird, sondern, mit einem temporären Namen versehen, erhalten bleibt.

Für dies alles genügt ein einzelner Aufruf:

```
call ADMIN\_MOVE\_TABLE('PROD', 'PERSONNEL',  
'MOVEZIEL3', 'MOVEZIEL3', 'MOVEZIEL3',  
'JOB_FTN',  
'',  
'(EMPNO) (STARTING FROM (10) ENDING AT (220) EVERY (10))',
```

```

',
'KEEP, REORG, COPY_USE_LOAD, FORCE',
'MOVE');
    
```

Wie war doch noch gleich der Titel des Artikels: „Online Table Move – nicht nur für DB2 9.7“? [ADMIN\\_MOVE\\_TABLE](#) ist zwar als neue Funktion in DB2 9.7 aufgeführt, aber so ganz neu ist sie doch nicht. Es handelt sich nämlich um eine Funktionalität, die für DB2 mit SAP schon länger existiert – unter dem Namen `ONLINE_TABLE_MOVE`. Naja - Sie werden denken, dass die Anschaffung von SAP vielleicht ein wenig übertrieben ist, um diese Stored Procedure zu bekommen, aber da kann ich Sie beruhigen – die Funktion kann auch ohne SAP im DB2 aktiviert werden – wenn man weiß wie!

Ab DB2 9.1.6 – also ab Fix 6 – ist die `ONLINE_TABLE_MOVE` in jedem DB2 aktivierbar – per:

```

db2 connect to sample
db2 invoke db2sap
    
```

Hier am Beispiel der SAMPLE Datenbank gezeigt, taucht nach diesem „invoke“ Befehl die Stored Procedure in der Funktionsliste der Datenbank auf und kann ähnlich wie die oben gezeigt [ADMIN\\_MOVE\\_TABLE](#) Procedure genutzt werden.

```

ONLINE_TABLE_MOVE('tabschema', 'tabname',
                  'Daten-TS', 'Index-TS', 'Lob-TS',
                  'MDC-Columns', 'Partkey-Cols',
                  'Operation')
    
```

Da diese Prozedur noch keinen separaten „Option“-Parameter hat, muss man die Optionen komma-separiert in der Operation mitgeben, die ansonsten identisch zu der DB2 9.7 Ausprägung ist.

Auch hier ein paar Beispiele:

- MOVE: führt wieder die komplette Verschiebung der Tabelle durch
- COPY,COPY\_USE\_LOAD: Kopiert die Daten mit Load statt Inserts (BACKUP nötig!)
- REORG,SWAP: führt offline REORG vor SWAP durch
- SWAP,KEEP: Originaltabelle bleibt bestehen (bekommt aber temporären Namen)

Die Überwachung des Verschiebe-Vorgangs erfolgt über die von der Stored Procedure erzeugten Hilfstabelle `SAPTOOLS.ONLINE_TABLE_MOVE`, in der die diversen Status und Zeitstempel Auskunft über Dauer und die jeweilige Phase geben. In der DB2 9.7 Variante heißt diese `SYSTOOLS.ADMIN_MOVE_TABLE` – nicht wirklich schwer zu merken – oder? Und so kann diese aussehen, wenn die COPY-Phase gerade abgeschlossen ist:

TABSCHEMA	TABNAME	KEY	VALUE
DB2ADMIN	CUSTOMER	COPY_START	2009-05-13-21.40.11.578000
DB2ADMIN	CUSTOMER	COPY_OPTS	ARRAY_INSERT
DB2ADMIN	CUSTOMER	COPY_TOTAL_ROWS	300000
DB2ADMIN	CUSTOMER	COPY_END	2009-05-13-21.40.28.843000
DB2ADMIN	CUSTOMER	REPLAY_START	2009-05-13-21.40.28.843000
DB2ADMIN	CUSTOMER	REPLAY_TOTAL_ROWS	0
DB2ADMIN	CUSTOMER	REPLAY_TOTAL_TIME	0
DB2ADMIN	CUSTOMER	STATUS	REPLAY
DB2ADMIN	CUSTOMER	VERSION	09.05.0003
DB2ADMIN	CUSTOMER	AUTHID	DB2ADMIN
DB2ADMIN	CUSTOMER	INIT_START	2009-05-13-17.14.37.078000
DB2ADMIN	CUSTOMER	INDEXSCHEMA	
DB2ADMIN	CUSTOMER	INDEXNAME	
DB2ADMIN	CUSTOMER	STAGING	CUSTOMERAKIUUis
DB2ADMIN	CUSTOMER	TARGET	CUSTOMERAKIUUit
DB2ADMIN	CUSTOMER	INIT_END	2009-05-13-17.14.37.609000

Der Status „REPLAY“ zeigt, dass die REPLAY-Phase, die auch Bestandteil der SWAP-Phase ist, als nächste Aktion erwartet wird. Die COPY-Phase hat 300000 Datensätze in die Zieltabelle mit dem temporären Namen „CUSTOMERAKIUUit“ übertragen und war damit am 13.05.2009 um 21:40 Uhr fertig. Der Benutzer DB2ADMIN hatte diesen Verschiebe-Vorgang am selben Tag um 17:14 Uhr angestoßen.

Nach der Fertigstellung mittels „SWAP“ Operation könnte die Tabelle so aussehen:

TABSCHEMA	TABNAME	KEY	VALUE
DB2ADMIN	CUSTOMER	COPY_START	2009-05-13-21.40.11.578000
DB2ADMIN	CUSTOMER	COPY_OPTS	ARRAY_INSERT
DB2ADMIN	CUSTOMER	COPY_TOTAL_ROWS	300000
DB2ADMIN	CUSTOMER	COPY_END	2009-05-13-21.40.28.843000
DB2ADMIN	CUSTOMER	REPLAY_START	2009-05-13-21.40.28.843000
DB2ADMIN	CUSTOMER	REPLAY_TOTAL_ROWS	4
DB2ADMIN	CUSTOMER	REPLAY_TOTAL_TIME	1
DB2ADMIN	CUSTOMER	REPLAY_END	2009-05-13-22.02.19.437000
DB2ADMIN	CUSTOMER	SWAP_START	2009-05-13-22.02.19.546000
DB2ADMIN	CUSTOMER	SWAP_END	2009-05-13-22.02.19.796000
DB2ADMIN	CUSTOMER	SWAP_RETRIES	0
DB2ADMIN	CUSTOMER	CLEANUP_START	2009-05-13-22.02.19.812000
DB2ADMIN	CUSTOMER	CLEANUP_END	2009-05-13-22.02.19.953000
DB2ADMIN	CUSTOMER	STATUS	COMPLETE
DB2ADMIN	CUSTOMER	VERSION	09.05.0003
DB2ADMIN	CUSTOMER	AUTHID	DB2ADMIN
DB2ADMIN	CUSTOMER	INIT_START	2009-05-13-17.14.37.078000
DB2ADMIN	CUSTOMER	INDEXSCHEMA	
DB2ADMIN	CUSTOMER	INDEXNAME	
DB2ADMIN	CUSTOMER	INIT_END	2009-05-13-17.14.37.609000

Der Status „COMPLETE“ zeigt die erfolgreiche Fertigstellung um 22:02 Uhr (SWAP-END). Während der REPLAY-Phase, die Bestandteil der SWAP-Operation ist, wurden vier Zeilen erneut in die Zieltabelle übertragen, weil diese in der Originaltabelle verändert worden waren. Ich hoffe, dieser Artikel hat Sie neugierig gemacht und Ihnen diese neue bzw. versteckte Funktionalität näher gebracht. Viel Spaß beim selber testen wünscht Ihnen der [Autor](#).

## DB2 Audit in DB2 V9.5 (Teil 2 – Audit-Informationen auslesen)

Im ersten Teil des Artikels zum Auditing in DB2 (DB2 NL Ausgabe 04/2009) wurde die Konfiguration des Audit, sowie die Einrichtung und Auswahl der zu überwachenden Aktionen behandelt. In diesem zweiten Teil wird erklärt wie die erstellten Audit-Informationen ausgelesen und ausgewertet werden können.

Die Audit-Files (sowohl für die Instanz, als auch für die Datenbank) können nur dann ausgelesen werden, wenn diese „geschlossen“ und „archiviert“ wurden.

In DB2 Version 8 war diese Auswertung ohne Archivierung möglich, was zu Problemen der Audit-Überwachung in der Archivierungszeit führte.

Die Archivierung beendet das aktuelle Audit-File, fügt eine TIMESTAMP an und startet die Protokollierung in einem neuen File.

Das Ausschreiben der Audit-Buffer in das Audit-File wird aufgerufen durch:

```
db2audit flush
```

Dieser Befehl sollte vor einem Archivieren der Audit-Informationen unbedingt als erste Aktion ausgeführt werden um einen aktuellen Stand zuzusichern.

Das Archivieren der Audit-Files erfolgt danach mittels der Aufrufe:

```
db2audit archive                                (für das Instanz-Audit)
db2audit archive database <dbname>            (für das Datenbank Audit)
```

Die Audit-Protokolle (Instanz und Datenbank) liegen, falls dies nicht anders konfiguriert wird, unter

```
<INSTANCE_HOME>/sqlllib/security/auditdata
```

Die Audit-Files der Instanz haben den Namen:

```
db2audit.instance.log.<node_number>
```

Die Audit-Files der Datenbank sind zu finden unter:

```
db2audit.db.<db_name>.log.<node_number>
```

Das Auslesen der Audit-Informationen kann entweder in Files erfolgen, oder die Audit-Informationen in Unloadfiles ablegen, die anschließend für die detaillierte Auswertung in eine Datenbank geladen werden können.

Der Aufruf für das Auslesen der Informationen aus den Audit-Files der Instanz (ohne Filter) in ein File erfolgt mittels:

```
db2audit extract file 'xx.audit.out'  
from files db2audit.instance.log.0.20091011164801
```

Der Aufruf für das Auslesen der Informationen aus den Audit-Files der Datenbanken (ohne Filter) in ein File erfolgt mittels (Beispieldatenbank IBM):

```
db2audit extract file 'yy.audit.out'  
from files db2audit.db.IBM.log.0.20091011164807
```

Im Beispiel eines Inserts hat die Ausgabe die Form:

```
timestamp=2009-10-11-15.39.34.639967;  
category=EXECUTE;  
audit event=STATEMENT;  
event correlator=4;  
event status=0;  
database=IBM;  
userid=kalu;  
authid=KALU;  
session authid=KALU;  
application id=*LOCAL.db2inst2.091011143933;  
application name=db2bp;  
package schema=DB2INST2;  
package name=SQLC2G13;  
package section=203;  
local transaction id=0x59af000000000000;  
global transaction id=0x0000000000000000000000000000000000000000000000000000000;  
uow id=2;  
activity id=1;  
statement invocation id=0;  
statement nesting level=0;  
activity type=WRITE_DML;  
statement text=insert into aud_test values ('x');  
statement isolation level=CS;  
Compilation Environment Description  
  isolation: CS  
  query optimization: 5  
  min dec div 3: NO  
  degree: 1  
  SQL rules: DB2  
  refresh age: +00000000000000.000000  
  resolution timestamp: 2009-10-11-15.39.34.000000  
  federated asynchrony: 0  
  schema: KALU  
  maintained table type: SYSTEM;  
rows modified=1;  
rows returned=0;  
  
timestamp=2009-10-11-15.39.34.641944;  
category=EXECUTE;  
audit event=COMMIT;  
event correlator=4;  
event status=0;  
database=IBM;  
userid=kalu;  
authid=KALU;  
session authid=KALU;  
application id=*LOCAL.db2inst2.091011143933;  
application name=db2bp;  
package schema=NULLID;
```

```
package name=SQLC2G13;  
package section=0;  
local transaction id=0x5aaf000000000000;  
global transaction id=0x000000000000000000000000000000000000000000000000;  
activity type=OTHER;
```

Um die Audit-Informationen als unload Format zu erhalten sieht der Aufruf folgendermassen aus:

```
db2audit extract DELASC  
to /home/audit  
from files db2audit.db.IBM.log.0.20091011164807
```

Die Daten werden in folgende Dateien entladen:

- audit.del
- checking.del
- objmaint.del
- secmaint.del
- sysadmin.del
- validate.del
- context.del

Die DDL-Struktur der Tabellen um diese Informationen wieder einladen zu können, sind in den Online-Handbüchern zum Thema Auditing zu finden.

Die Beschreibung der [DDL-Struktur](#) für die notwendigen Tabellen ist in den Online-Handbüchern zum Thema Auditing zu finden.

Die Tabellen werden unter dem Schema AUDIT erstellt. Daher muss zuerst dieses Schema erstellt werden mit:

```
CREATE SCHEMA AUDIT  
SET CURRENT SCHEMA = 'AUDIT'
```

Danach können die Tabellen durch das bei DB2 enthaltene Skript angelegt werden:

```
db2 +o -tf sqllib/misc/db2audit.ddl
```

Anschließend können die Tabellen für die verschiedenen Kategorien befüllt werden.

Beispiel OBJMAINT:

```
LOAD FROM objmaint.del OF DEL  
MODIFIED BY DELPRIORITYCHAR LOBSINFILE  
INSERT INTO AUDIT.OBJMAINT
```

### Filtern der Audit Daten:

Beim Auslesen können die Audit-Informationen gefiltert werden. Dabei besteht die Möglichkeit auf Kategorien einzuschränken (z.B. nur OBJMAINT und VALIDATE) und dabei nur die erfolgreichen, nur die erfolglosen oder beide Ausführungen zu entladen.

Die Filterung kann sowohl beim Auslesen in ein File, als auch beim Erstellen der Unloads für die Audit-Protokoll Tabellen genutzt werden.

Beispiel: Nur Statements mit Errorcode:

```
db2audit extract file 'yy.audit.out'  
STATUS FAILURE  
from files db2audit.db.IBM.log.0.20091011164807
```

Beispiel: Statements von Typ OBJMAINT mit Fehlercode und erfolgreiche AUDIT-Aktivitäten:

```
db2audit extract file 'yy.audit.out'
```

CATEGORY OBJMAINT STATUS FAILURE  
CATEGORY AUDIT STATUS SUCCESS  
from files db2audit.db.IBM.log.0.20091011164807

## Schulungen / Tagungen / Informationsveranstaltung

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

### Kurse bis zum Jahresende

#### DB2 Familie:

-----

CE03D1DE: DB2 Familie - Grundlagen, 5.11., München  
CE03Z1DE: Certification Class - DB2 Family Fundamentals, 24.11., Ludwigsburg  
CE12D2DE: SQL - Die Sprache, 23.11., Hamburg  
CE13D2DE: SQL - Erweiterung, 25.11., Hamburg

#### DB2 für LUW:

-----

CFH1D2DE: DB2 für LUW - SQL Performance, 2.11., Frankfurt/Main  
CI49D1DE: DB2 for Linux, UNIX, Windows - Adv. Recovery and High Availability, 3.11., München  
CL25D1DE: DB2 für LUW - Multi-Partition DPF Environment für Single-Partition DBAs, 4.11., Stuttgart  
CI411DDE: DB2 für LUW - Performance and Tuning für Single-Partition Datenbanken, 9.11., Stuttgart  
CL62D1DE: DB2 für LUW - SAP Performance and Tuning, 9.11., Frankfurt/Main  
CI65D2DE: DB2 für LUW - Best Practices for SAP BW with DB2 V9, 11.11., Frankfurt/Main  
CL20D1DE: DB2 für Linux - Administration 1, 16.11., Frankfurt/Main  
CL21D1DE: DB2 für UNIX - Administration 1, 16.11., Frankfurt/Main  
CL23D1DE: DB2 für Windows - Administration 1, 16.11., Frankfurt/Main  
CI48D1DE: DB2 für Linux, UNIX, Windows - Administration für Umsteiger, 16.11., Stuttgart  
CL20Z1DE: Certification Class - DB2 9 für UNIX Administration, 25.11., Ludwigsburg  
CI61D1DE: DB2 für LUW - SAP Advanced Administration, 30.11., Berlin  
CL63D1DE: DB2 für LUW - DB2 Connect Problem Determination and Performance, 30.11., Frankfurt/Main  
CL10D1DE: DB2 für LUW - pureXML Daten einfügen und extrahieren, 7.12., Düsseldorf  
CL16D1DE: DB2 für LUW - pureXML Fun für Administratoren, 10.12., Düsseldorf  
CL20D1DE: DB2 für Linux - Administration 1, 14.12., Stuttgart  
CL21D1DE: DB2 für UNIX - Administration 1, 14.12., Stuttgart  
CL23D1DE: DB2 für Windows - Administration 1, 14.12., Stuttgart  
CL312DDE: DB2 9.7 for Linux, UNIX, and Windows New Features, 14.12., Stuttgart

#### DB2 für z/OS:

-----

CF92D1DE: DB2 für z/OS - für Arbeitsvorbereiter, 9.11., Frankfurt/Main  
CFDSD1DE: DB2 für z/OS - Grundlagen für Systemprogrammierer, 9.11., Frankfurt/Main  
CFS1D1DE: DB2 für z/OS - Expertenwissen - Aus der Praxis für die Praxis, 9.11., Stuttgart  
CV46D1DE: DB2 for z/OS - Data Sharing Recovery, 9.11., Berlin  
CV83D1DE: DB2 für z/OS - Datenbank Administration 1, 9.11., Düsseldorf  
CV84D1DE: DB2 für z/OS - Datenbank Administration 2, 16.11., München  
CV14D1DE: DB2 für z/OS - Anwendungsprogrammierung mit Java, 17.11., Stuttgart  
CF39D1DE: DB2 für z/OS - Data Sharing: Analyse und Optimierung, 23.11., Berlin  
CFDPD1DE: DB2 für z/OS - für Operator, 23.11., Stuttgart  
CV30D2DE: DB2 für z/OS - Neue Funktionen in Version 9, 23.11., München  
CV89D1DE: DB2 for z/OS - Recovery/Restart, 23.11., Stuttgart  
CV96D1DE: DB2 for z/OS - Application Performance and Tuning, 23.11., Stuttgart  
CV91D1DE: DB2 for z/OS - System/Disaster Recovery, 30.11., Hamburg  
WRB023DE: DB2 9 for z/OS Performance, 1.12., Frankfurt/Main (ITSO Workshop)  
CV73D2DE: DB2 für z/OS - Programmierung von Stored Procedures, 2.12., München  
CV04D1DE: DB2 für z/OS - Das ultimative Starterpaket: Erste Schritte in z/OS und DB2, 7.12., München  
CV30D2DE: DB2 für z/OS - Neue Funktionen in Version 9, 7.12., Hamburg  
CV97D2DE: DB2 für z/OS - Spotlight on zParms, 8.12., Stuttgart  
CV28D1DE: DB2 für z/OS - How to deal with Unicode, 14.12., Berlin  
CV85D1DE: DB2 für z/OS System-Administration, 14.12., Frankfurt/Main  
ES53E2DE: SAP NetWeaver on DB2 UDB for z/OS - Database Administration, 14.12., Berlin

#### Data Warehouse, Content Managment:

-----

DMS1D1DE: IBM DB2 TSM und Websphere für Content Manager, 10.11., Frankfurt/Main  
DW35D2DE: DB2 DataWarehouse Edition 9 Components, 23.11., München  
DW32D1DE: Managing Workloads for DB2 LUW and InfoSphere Warehouse, 7.12., München  
IM25D2DE: IBM DB2 Content Manager V8 API Programming 8.4, 7.12., Düsseldorf

Bei Interesse für den einen oder anderen Kurs finden Sie die Details wie gewohnt [hier](#).



## Chats mit dem Labor

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

Die Präsentationen der Chats, können angeschaut und heruntergeladen werden.

Der letzte Chat zum Thema „DB2 pureScale: Scaling Databases without Limits“ fand am 21.10 statt.

Analog zu den DB2 Chats gibt es auch noch BI Chats mit dem Labor. Die Präsentationen und Replays der Chats können [hier](#) angeschaut und heruntergeladen werden.

## Newsletter Archiv

Alte Ausgaben vom DB2-NL sind zum Nachlesen in folgenden Archiven zu finden:

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

## 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 Subjekt „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	IT-Spezialist für DB2 LUW, IBM SWG; <a href="mailto:djs@de.ibm.com">djs@de.ibm.com</a>
Gerd Kaluzinski	IT-Specialist Informix Dynamic Server und DB2 UDB IBM Software Group, Information Management <a href="mailto:gerd.kaluzinski@de.ibm.com">gerd.kaluzinski@de.ibm.com</a> <a href="#">DB2 Audit in DB2 V9.5 (Teil 2 – Audit-Informationen auslesen)</a>
Michael Tiefenbacher	LIS.TEC GmbH <a href="#">Online Table Move – nicht nur für DB2 9.7</a>

## Reviewer und Ideenlieferanten:

Nela Krawez	IBM SWG, InfoSphere Balanced Warehouse Development
Wilfried Hoge	IBM SWG, Technical Sales IM