

## Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

das Jahr 2008 neigt sich dem Ende entgegen, ein ereignisreiches und arbeitsintensives Jahr. Nun gegen Ende des Jahres ist es Zeit zu prüfen, was hatte man für Ziele und wurden diese auch erreicht, übertroffen oder liegt die Erreichbarkeit dieser Ziele in naher oder ferner Zukunft. Wir hatten uns für das Jahr 2008 eine engere Zusammenarbeit vorgenommen. Wir haben dabei Fortschritte gemacht, sind aber noch nicht am Ziel angekommen.

Es gab weitere zahlreiche Anmeldungen, Danke dafür. Aber auch hier stellen wir fest, es gibt noch Potential. Bei Kundeneinsätzen, Schulungen und Veranstaltungen treffen wir immer wieder DB2er, die noch nie was vom DB2 Newsletter gehört haben.

Auch im neuen Jahr gibt es wieder viele Herausforderungen derer wir uns stellen dürfen. Dafür benötigen wir Ihre Mitarbeit. Schicken Sie uns einfach Ihre Ideen, Anregungen, Artikel, Anfragen. So wie es können, werden wir darauf eingehen, oder es in die entsprechenden Kanäle weiterleiten.

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

Ihr TechTeam

## Inhaltsverzeichnis

<a href="#">CHATS MIT DEM LABOR.....</a>	<a href="#">2</a>
<a href="#">TECHTIPP: VALUES VS SYSIBM.SYSDUMMY1.....</a>	<a href="#">2</a>
<a href="#">TECHTIPP: BESTIMMEN DER EIGENEN APPLICATION ID.....</a>	<a href="#">3</a>
<a href="#">TECHTIPP: TYP-KONVERTIERUNG ALPHANUMERISCH NACH NUMERISCH.....</a>	<a href="#">3</a>
<a href="#">ERSTELLEN EINER TEST TABELLE UND LADEN DER TESTDATEN.....</a>	<a href="#">3</a>
<a href="#">NOTWENDIGE EIGENSCHAFTEN ZUR TYP-KONVERTIERUNG.....</a>	<a href="#">4</a>
<a href="#">VERHALTEN DER NORMALEN TYP-KONVERTIERUNG.....</a>	<a href="#">5</a>
<a href="#">PROBLEMLÖSUNG: ALTERNATIVE TYP-KONVERTIERUNG.....</a>	<a href="#">5</a>
<a href="#">BEISPIELE, WIE DIE TYP-KONVERTIERUNG EINGESETZT WERDEN KANN.....</a>	<a href="#">7</a>
<a href="#">TECHNOTES.....</a>	<a href="#">9</a>
<a href="#">INHALTSZUSAMMENFASSUNG FÜR DEN DB2 NL IN 2008.....</a>	<a href="#">10</a>
<a href="#">REDBOOKS/WHITEPAPERS U.A.....</a>	<a href="#">10</a>
<a href="#">SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....</a>	<a href="#">10</a>
<a href="#">NEWSLETTER ARCHIV.....</a>	<a href="#">11</a>
<a href="#">ANMELDUNG/ABMELDUNG.....</a>	<a href="#">11</a>
<a href="#">DIE AUTOREN DIESER AUSGABE:.....</a>	<a href="#">11</a>
<a href="#">IBM SWG IM SERVICES - IHR DIREKTER DRAHT ZUR PRODUKTENTWICKLUNG.....</a>	<a href="#">11</a>

## Chats mit dem Labor

Eine Liste der bereits durchgeführten Chats ist zu finden unter:

<http://www-306.ibm.com/software/data/db2/9/labchats.html>

Die Präsentationen der Chats, können als pdf angeschaut und heruntergeladen werden.

## TechTipp: values vs sysibm.sysdummy1

Es gibt immer wieder Momente wo man Informationen aus der Datenbank erhalten will, zu denen explizit keine Tabelle gibt (z.B. um Inhalte aus den DB2 Spezialregistern ermitteln oder einfach Test einer DB2 Funktion).

Dies kann auf verschiedenen Wegen vorgenommen werden, z.B. :

- values
- sysibm.sysdummy1

In diesem Artikel wollen wir kurz Unterschiede und Gemeinsamkeiten der beiden Methoden aufzeigen:

	values()	sysibm.sysdummy1
	Ist eine Form eines Statements.	SYSIBM.SYSDUMMY1 ist ein View, mit einem Datensatz und wird u.a. dort benötigt, wo Kompatibilität mit DB2 z/OS gefordert ist.
Abfrage	values(<...>)	select <...> from sysibm.sysdummy1
Einschränkungen	<p>Mit values kann kein Spaltennamen vergeben werden:</p> <pre>values(int('1234')) as intSpalte; SQL0104N Auf "values(INT('1234'))" folgte das unerwartete Token "as". Zu den möglichen Token gehören: "..". SQLSTATE=42601</pre> <p>values(int('1234') as intSpalte); SQL0104N Auf "values(INT('1234') " folgte das unerwartete Token "as". Zu den möglichen Token gehören: ", ". SQLSTATE=42601</p>	<p>Im Select-Statement können Spaltennamen vergeben werden:</p> <pre>select int('1234') as intSpalte from sysibm.sysdummy1;</pre> <pre>INTSPALTE -----       1234  1 Satz/Sätze ausgewählt.</pre>
Anzeige aktuelles Datum	<pre>values(current date)  1 ----- 2008-12-18  1 Satz/Sätze ausgewählt.</pre>	<pre>select current date from sysibm.sysdummy1  1 ----- 2008-12-18  1 Satz/Sätze ausgewählt.</pre>
Konvertierungsfunktion	<pre>values(to_date('20001218','YYYYM MDD'))  1 ----- 2000-12-18-00.00.00.000000  1 Satz/Sätze ausgewählt.</pre>	<pre>select to_date('20001218','YYYYMMDD') from sysibm.sysdummy1  1 ----- 2000-12-18-00.00.00.000000  1 Satz/Sätze ausgewählt.</pre>

Für welche der Methode man sich letztendlich entscheidet, bleibt einem selbst überlassen.

Eine Liste der Special Register (wie CURRENT\_DATE) finden sie ebenfalls im Online-Handbuch:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0008404.html>

## TechTipp: Bestimmen der eigenen Application ID

Es gibt immer wieder das Problem aus der Liste der vielen Anwendungen herauszufinden, welche die eigene Anwendung ist. Beim Erstellen des TechTipp zu values und sysdummy1 bin ich dann über folgende Funktion gestolpert: APPLICATION\_ID:

```
$ db2 "SELECT APPLICATION_ID() AS APPL_ID FROM SYSIBM.SYSDUMMY1"
```

```
APPL_ID
```

```
-----  
*LOCAL.DB2.081218084459
```

```
1 Satz/Sätze ausgewählt.
```

Natürlich kann diese Funktion auch im Rahmen der values() Funktion verwendet werden:

```
$ db2 "values (APPLICATION_ID())"
```

```
1
```

```
-----  
*LOCAL.DB2.081218084532
```

```
1 Satz/Sätze ausgewählt.
```

Link zum Online-Handbuch und Beschreibung:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.sql.rtn.doc/doc/r0011856.html>

Administrative SQL Routinen and Views (wie APPLICATION\_ID()) finden sie unter:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.sql.rtn.doc/doc/r0023485.html>

## TechTipp: Typ-Konvertierung alphanumerisch nach numerisch

Bei der Umwandlung eines alphanumerischen in einen numerischen Wert bricht das zugehörige SQL-Statement ab, wenn das alphanumerische Feld nicht nur aus Ziffern besteht. Folgende Fehlermeldung wird dabei ausgegeben:

```
select int('ABCDE') from sysibm.sysdummy1  
1
```

```
-----  
SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein  
ungültiges Zeichen gefunden. SQLSTATE=22018
```

Während bei der normalen Typ-Umwandlung zum Beispiel folgendes Ergebnis zurückgegeben wird:

```
select int('123') from sysibm.sysdummy1  
1
```

```
-----  
123
```

```
1 Satz/Sätze ausgewählt.
```

In diesem Artikel wollen wir nun eine Methode vorstellen, die relativ einfach und übersichtlich das Problem löst, indem fehlerhafte Sätze übersprungen werden.

## Erstellen einer Test Tabelle und Laden der Testdaten

Um zu demonstrieren, wie DB2 in den einzelnen Situationen reagiert wird folgende Test-Tabelle definiert und mit verschiedenen Daten befüllt:

```
create table test.tabelle1 (i1 integer, c2 char(20));
insert into test.tabelle1 values (1,'1'), (2,'2a'), (3,'3'), (4,'3405'),
(5,'ABCDEFGF'), (6,'ACD'), (7,' 123'), (8,'123 '), (9,'12 3 '), (10,' 1 2 3 '),
(11,'');
```

```
select t.* , length(rtrim(C2)) length from test.tabelle1 t
```

I1	C2	LENGTH_C2
1	1	1
2	2a	2
3	3	1
4	3405	4
5	ABCDEFGF	7
6	ACD	3
7	123	4
8	123	3
9	12 3	4
10	1 2 3	6
11		0

11 Satz/Sätze ausgewählt.

## Notwendige Eigenschaften zur Typ-Konvertierung

Folgende Eigenschaften müssen auf den Feldinhalt zutreffen, damit eine Typ-Konvertierung von alphanumerisch nach integer funktioniert (in Klammern sind die Datenzeilen enthalten, die eine Abweichung entsprechend beinhalten):

- beliebige Länge (aber Wert gültig für Integer)

- Feld enthält keine Buchstaben (Zeilen 2, 5, 6)

```
select i1,int(c2) from test.tabelle1 where i1=5
I1      2
-----
```

SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018

```
select i1,int(c2) from test.tabelle1 where i1=2
I1      2
-----
```

SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018

- keine Leerzeichen innerhalb der Ziffernfolge (Zeile 10)

```
select i1,int(c2) from test.tabelle1 where i1=10
I1      2
-----
```

SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018

- leeres Feld (Zeile 11)

```
select i1,int(c2) from test.tabelle1 where i1=11
I1      2
-----
```

SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018

Vermessen Sie in dieser Aufzählung führenden und nachfolgende Leerzeichen (wenn auch in den Testdaten enthalten), so ist festzustellen, dass die Konvertierungsroutine int damit problemlos umgeht.

```
select int(' 123 ') from sysibm.sysdummy1
1
-----
      123
```

1 Satz/Sätze ausgewählt.

## Verhalten der normalen Typ-Konvertierung

Wird auf eine alphanumerische Spalte mit nicht-numerischen Inhalt (Bedingungen im Abschnitt vorher) mittels Typ-Konvertierung zugegriffen, wird an der Zeile abgebrochen, wo der Fehler auftritt. In unserem Beispiel ist es die Zeile 2 (I1=2). Die Zeile mit I1=1 wird dabei ausgegeben.

```
select i1,int(c2) from test.tabelle1
I1      2
-----
1      1
SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges
Zeichen gefunden.  SQLSTATE=22018
```

## Problemlösung: Alternative Typ-Konvertierung

Damit eine Abfrage mit Typ-Konvertierung nicht abbricht, kann das SQL-Statement wie folgt erweitert werden:

- **translate(ltrim(rtrim(<colname>)), 'X', ' 0123456789') = ''**

Mit dieser Erweiterung werden vorhandene Ziffern durch Blanks ersetzt, so dass im Falle des rein-numerischen Inhaltes das Ergebnis der Translate-Funktion leer ist (Blank-String).

Sind nicht numerische Zeichen enthalten, ist das Ergebnis der translate-Funktion nicht leer.

Blanks innerhalb der Zeichenfolge werden durch 'X' ersetzt und daher als nicht gültige Zeichenfolge erkannt und auch nicht konvertiert.

Hier ein Beispiel, was bei der translate-Funktion für unsere Testdaten herauskommt.

```
select i1,c2,translate(ltrim(rtrim(c2)), 'X', '
0123456789'),length(rtrim(c2)) from test.tabelle1
I1      C2      3      4
-----
1 1      1
2 2a      a      2
3 3      1
4 3405    4
5 ABCDEFG ABCDEFG  7
6 ACD     ACD     3
7 123     4
8 123     3
9 12 3    X      4
10 1 2 3  X X    6
11      0
12 1234   6
13 1234   4
```

13 Satz/Sätze ausgewählt.

Setzt man nun das SQL-Statement mit der translate-Funktionserweiterung ab, wird man jedoch feststellen, dass dies alleine noch nicht ausreicht. Es werden zwar schon mehr Datenzeilen ausgegeben, führt aber trotzdem noch zu einem Abbruch.

```
select i1,int(c2) from test.tabelle1 where translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' '
```

```
I1          2
-----
          1          1
          3          3
          4         3405
          7         123
          8         123
```

SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018

Was war die Ursache?

Sicherlich liegt es zum einen an unseren Testdaten (in einigen Fällen läuft das Statement ohne Fehler durch und es wird das komplette Ergebnis geliefert). Mit unseren Testdaten haben wir aber versucht alle bekannten Sonderfälle abzudecken. Also woran lag es nun, dass das Statement nach Zeile 8 abgebrochen wurde?

Bei der Analyse der Daten, die nach der Zeile 8 kommen, stellen wir fest, dass in Zeile 11 die Spalte C2 leer ist (length(c2)=0). Leere Strings können durch die Typ-Konvertierung nicht umgewandelt werden:

```
select int(' ') from sysibm.sysdummy1
```

```
1
-----
SQL0420N In einem Zeichenfolgeargument der Funktion "INTEGER" wurde ein ungültiges Zeichen gefunden. SQLSTATE=22018
```

Daher ist das SQL Statement noch um folgende Abfrage zu erweitern:

- **and length(rtrim(<colname>)) > 0**

Dies bedeutet, dass keine Leerstrings und keine Blanks enthalten sein dürfen.

Erweitern wir also das Statement um die Längenabfrage:

```
select i1,c2,int(c2) from test.tabelle1 where translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' ' and length(rtrim(c2)) > 0;
```

```
I1          C2          3
-----
          1 1          1
          3 3          3
          4 3405         3405
          7 123         123
          8 123         123
         12 1234         1234
         13 1234         1234
```

7 Satz/Sätze ausgewählt.

Nun werden alle Datensätze, die einen numerischen Wert in der alphanumerischen Spalte C2 enthalten, ausgegeben.

Zusammenfassend stellen wir fest, dass mit den beiden Erweiterungen folgendes passiert:

- Ein beliebig langes, nur mit Ziffern gefülltes Feld

- mit führenden und nachfolgenden Blanks
- aber ohne Blanks innerhalb der Ziffernfolge
- wird in einen numerischen Wert umgewandelt

Randbemerkungen:

- Wenn keine führenden und oder nachfolgenden Blanks erlaubt sein sollen, einfach ltrim und oder rtrim weglassen
- Für die Ausgabe der "fehlerhaft" gefüllten alphanumerischen Felder ist obige Bedingung zu negieren ( ... <> ' ' or ... = 0)

## Beispiele, wie die Typ-Konvertierung eingesetzt werden kann

### Ausgabe von alphanumerischen Werten, die nur aus zusammenhängenden Ziffernfolgen bestehen

```
select i1,c2 from test.tabelle1 where translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' ' and length(rtrim(c2)) > 0
```

```
I1          C2
-----
1          1
3          3
4         3405
7         123
8         123
```

5 Satz/Sätze ausgewählt.

Im obigen Beispiel werden die Dateninhalte der Spalte C2 als Character ausgegeben, wie zu erkennen ist am führenden Blank bei der Zeichenfolge ' 123'.

Im nachfolgenden Beispiel werden dann die Ergebnisse noch nach integer konvertiert.

```
select i1,int(c2) from test.tabelle1 where translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' ' and length(rtrim(c2)) > 0
```

```
I1          2
-----
1             1
3             3
4            3405
7            123
8            123
```

5 Satz/Sätze ausgewählt.

### Klassifizierung der Daten nach numerisch (N) und nicht-numerisch (C)

In diesem Beispiel wird der Inhalt der Spalte C2 wie folgt klassifiziert:

- Ausgabe aller Spalten mit Kennung "N" für numerisch, "C" für nicht-numerisch
- Umwandlung des Inhalts in DECIMAL(10, 0) mit -999999999 als Dummy-Wert für Inhalte, die nicht konvertiert werden können (hier: ohne Betrachtung der maximal möglichen Länge der Ziffernfolge):

```
select i1,c2, case when translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' ' and length(rtrim(c2)) > 0 then 'N' else 'C' end, decimal(case when translate(ltrim(rtrim(c2)), 'X', '0123456789') = ' ' and length(rtrim(c2)) > 0
```

```
then bigint(c2) else -999999999 end, 10, 0) from test.tabelle1
```

```
I1          C2          3 4
-----
1 1          N          1,
2 2a         C -999999999,
3 3          N          3,
4 3405       N          3405,
5 ABCDEFG    C -999999999,
6 ACD        C -999999999,
7 123        N          123,
8 123        N          123,
9 12 3       C -999999999,
10 1 2 3     C -999999999,
11          C -999999999,
```

11 Satz/Sätze ausgewählt.

### Umwandeln von alphanumerischen Inhalten nach DECIMAL(3, 0)

In diesem Beispiel werden nur die Datensätze Ausgabe nur da, wo es möglich ist:

```
select i1,cast(c2 as decimal(3, 0)) from test.tabelle1 where
translate(ltrim(rtrim(c2)), 'X', ' 0123456789') = ' ' and length(rtrim(c2))
between 1 and 3
```

```
I1          2
-----
1          1,
3          3,
8 123,
```

3 Satz/Sätze ausgewählt.

### Ausgabe der Sätze, bei denen die Umwandlung nach DECIMAL(3,0) nicht möglich ist

In diesem Beispiel werden auch Datensätze ausgegeben, bei denen die Länge der Daten der C2 Spalte länger als 3 ist (length >3):

```
select i1,c2 from test.tabelle1 where translate(ltrim(rtrim(c2)), 'X', '
0123456789') <> ' ' or length(ltrim(rtrim(c2))) > 3 or length(rtrim(c2)) = 0
```

```
I1          C2
-----
2 2a
4 3405
5 ABCDEFG
6 ACD
9 12 3
10 1 2 3
11
```

7 Satz/Sätze ausgewählt.

### Ausgabe aller Inhalte, die linksbündig nur aus einer 3-stelligen Ziffernfolge bestehen

```
select i1,c2 from test.tabelle1 where length(rtrim(c2)) = 3 and
translate(rtrim(c2), 'X', ' 0123456789') = ' '
```

```
I1          C2
-----
8 123
```

1 Satz/Sätze ausgewählt.



## Technotes

Anbei eine Liste hilfreiche DB2 LUW Troubleshooting Technotes:

- Technote #1322385: Collecting data for DB2 hang on LINUX  
<http://www-01.ibm.com/support/docview.wss?uid=swg21322385>
- Technote #1324849: Collecting data for DB2 hang on AIX  
<http://www-01.ibm.com/support/docview.wss?uid=swg21324849>
- Technote #1324850: Collecting data for DB2 hang on SUN SOLARIS  
<http://www-01.ibm.com/support/docview.wss?uid=swg21324850>
- Technote #1324853: Collecting data for DB2 hang on HP-UX  
<http://www-01.ibm.com/support/docview.wss?uid=swg21324853>
- Technote #1324857: Collecting data for DB2 hang on Windows  
<http://www-01.ibm.com/support/docview.wss?uid=swg21324857>
- Technote #1326400: Collecting data for DB2 Abend (crash)  
<http://www-01.ibm.com/support/docview.wss?uid=swg21326400>
- Technote #1273299: Collecting data for Fixpak installation failures on DB2 UDB V8  
<http://www-01.ibm.com/support/docview.wss?uid=swg21273299>
- Technote #1273313: Collecting data for fix pack installation failures for DB2 V9  
<http://www-01.ibm.com/support/docview.wss?uid=swg21273313>
- Technote #1286651: MustGather: DB2 Client Connectivity  
<http://www-01.ibm.com/support/docview.wss?uid=swg21286651>
- Technote #1284894: MustGather: CLI Applications  
<http://www-01.ibm.com/support/docview.wss?uid=swg21284894>
- Technote #1106550: MustGather: Collecting Diagnostics for DB2 Compiler Issues  
<http://www-01.ibm.com/support/docview.wss?uid=swg21106550>
- Technote #1244563: Collecting data for index corruption and/or index or data inconsistency issues  
<http://www-01.ibm.com/support/docview.wss?uid=swg21244563>
- Technote #1322155: MustGather: DB2 data corruption  
<http://www-01.ibm.com/support/docview.wss?uid=swg21322155>
- Technote #1321787: MustGather: DB2 Deadlocks  
<http://www-01.ibm.com/support/docview.wss?uid=swg21321787>
- Technote #1322231: MustGather: DB2 lock timeouts  
<http://www-01.ibm.com/support/docview.wss?uid=swg21322231>
- Technote #1243808: Submitting diagnostic information to IBM Technical Support for problem determination  
<http://www-01.ibm.com/support/docview.wss?uid=swg21243808>

## Inhaltszusammenfassung für den DB2 NL in 2008

Folgende Artikel wurden in diesem Jahr geliefert:

200801	DWC/SQW MIGRATION	Andrea Ott
200801	NEUERUNGEN ZU STATISTIKEN IN DB2 9 UND 9.5	Wilfried Hoge
200801	PERFTIPP: LOCKS	Doreen Stein
200801	TECHTIPP: LOCK-INFORMATION	Doreen Stein
200802	ARTIKELSERIE: DB2 TOOLS NEWS	
200802	ARTIKELSERIE: DWC/SQW MIGRATION - TEIL 2	
200802	ARTIKELSERIE: NEUERUNGEN IN V9.5	
200802	DB2 PERFORMANCE EXPERT FOR LUW UND WORKLOAD MANAGEMENT	
200802	TECHTIPP: HADR – WELCHE MÖGLICHKEITEN GIBT ES, WENN EIN TABLESPACE AUF EINEN	
200802	HADR STANDBY SERVER "OFFLINE" GESETZT WURDE?	Volker Fraenkle
200802	ZERTIFIZIERUNG	
200806	IBMS NEUE DATA STUDIO ADMINISTRATION CONSOLE	Holger Kam
200806	NEUES VON DER IOD: „SOLIDDB CACHE FOR DB2“	Jürgen Buck
200806	TECHTIPP: ERMITTELN VON SPALTEN	Matthias Rawohl
200808	ANWENDUNG DER DATENKOMPRIMIERUNG IN DB2 LUW	Diana Gfroerer
200809	ABLAGE DER DB2 V9.5 DIAGNOSE LOGS IN WINDOWS	Peter Schurr
200809	NEUE FUNKTIONEN VON DB2 V9.5 IN KOMBINATION MIT SAP	Stefan Ulrich
200809	SEIT ENDE AUGUST IST DAS FP 2 FÜR DIE DB2 V9.5 RAUS	
200810	AUTORUNSTATS	Stefan Ulrich
200810	DB2 LUW REMOTE LOAD FROM CURSOR OHNE FEDERATION	Volker Fraenkle
200810	FIXPAK FP2A FÜR DIE DB2 VERSION V9.5	
200810	JDBC TREIBER	Peter Schurr
200810	MONITORING MIT DB2TOP	Wilfried Hoge
200810	NEUE FUNKTIONEN VON DB2 V9.5 IN KOMBINATION MIT SAP	Stefan Ulrich
200811	ROLLEN IN DB2	Volker Fraenkle
200811	RUNSTATS	
200811	WIE GROSS IST EINE LARGE TABELLE	
200811	ZUSAMMENFASSUNG DB2 AKTUELL 2008	Jens Bobe
200812	BESTIMMEN DER APPLICATION ID	
200812	INHALTSZUSAMMENFASSUNG FÜR DEN DB2 NL 2008	
200812	TECHNOTES	
200812	TYPKONVERTIERUNG ALPHANUMERISCH NACH INTEGER	
200812	VALUES VS SYSIBM.SYSDUMMY1	

## Redbooks/Whitepapers u.a.

- **InfoSphere™ Warehouse: Cubing Services and Client Access Interfaces**

Dieses Redbook wurde am 12. Dezember veröffentlicht.

InfoSphere™ Warehouse ermöglicht Zugriff sowohl auf strukturierte, als auch unstrukturierte Daten, als auch Transaktions- oder Operationale Daten. In diesem Redbook werden die Möglichkeiten der Komponente Cubing Service näher beleuchtet. Zugriff zum Redbook erfolgt über:

<http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247582.html?Open>

## Schulungen / Tagungen / Informationsveranstaltung

Die Basis Admin Kurse (CL20/21/23) (ehemals CF20/21/23) wurden jetzt auf die aktuelle DB2 Version V9.5 umgestellt.

Hier noch ein Hinweis auf einen neuen z/OS Kurs:

- Using Omegamon XE for DB2 Performance Expert on z/OS - Instructor Led Online (3B640DE)  
[http://www-05.ibm.com/services/learning/de/ta-iris.nsf/\(ExtCourseNr\)/3B640DE](http://www-05.ibm.com/services/learning/de/ta-iris.nsf/(ExtCourseNr)/3B640DE)

## Newsletter Archiv

Alte Ausgaben vom DB2-NL sind nun zum Nachlesen

- im Archiv von BYTEC zu finden:  
[https://www.bytec.de/de/software/ibm\\_software/newsletter/db2newsletter/](https://www.bytec.de/de/software/ibm_software/newsletter/db2newsletter/)
- Seit einigen Tagen finden Sie bei DRAP Solutions, wo seit langer Zeit ein Archiv des INFORMIX Newsletters zu finden ist, auch das Archiv des DB2 Newsletters. Der Link <http://www.drap.de/link/db2> führt Sie in das Archiv.
- Weiterhin haben wir nun auch ein DB2 NL Archiv bei der Cursor Software AG und kann durch folgenden Link erreicht werden:  
<http://www.cursor-distribution.de/index.php/aktuelles/db2-newsletter>

## 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 UDB, IBM SWG; <a href="mailto:djs@de.ibm.com">djs@de.ibm.com</a>
Dirk Fechner	IT Specialist, IBM SWG <a href="mailto:fechner@de.ibm.com">fechner@de.ibm.com</a> Artikel: <a href="#">Technotes</a>

## Reviewer und Ideenlieferanten:

Nela Krawez	IBM SWG, IT- Spezialist
Wilfried Hoge	IBM SWG, Technical Sales
Volker Fränkle	IBM SWG, IT-Spezialist
Jürgen Wittliff	Dresdner Bank Hat die Basis für den Artikel: <a href="#">Typ-Konvertierung alphanumerisch nach numerisch</a> geliefert, den wir umgeschrieben und die enthaltenen Beispiele mit nachvollziehbaren Beispielen ersetzt haben.

## IBM SWG IM Services - Ihr direkter Draht zur Produktentwicklung

Der Ihnen vorliegende Newsletter wird durch "IBM Software Group Information Management Services" herausgegeben. Die Artikel werden in Zusammenarbeit mit der Produktentwicklung und dem Support erstellt und publiziert. All dies dürfte Ihnen sicherlich bekannt sein.

Aber wissen Sie, dass das Team des IBM SWG IM Services tief gehende Expertise zu folgenden Spezialthemen hat?

- Hochverfügbarkeit
- Replikation
- Performance
- Security

Diese Expertisen werde auf Tagesbasis abgerufen. Informationen zu dieser Expertise erhalten Sie über Herrn Jens Krumbiegel ([krumbieg@de.ibm.com](mailto:krumbieg@de.ibm.com)). Informieren Sie sich auch über unsere exklusiven Business-Partner Expertisen.