

Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

der Jahreswechsel ist schon wieder Geschichte. Wer den Schnee, das Hochwasser und die Eiseskälte überstanden hat, wartet sicherlich wie wir auch endlich auf den Frühling. Damit die Wartezeit nicht zu lang wird, hier die 1te Ausgabe des 2011 Newsletters.

Wie immer wünschen wir viel Spaß beim Lesen.



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

Ihr TechTeam

Inhaltsverzeichnis

| | |
|--|-------------------|
| END OF SUPPORT 9.1..... | 1 |
| CHATS MIT DEM LABOR..... | 1 |
| PRAXISBEISPIEL DB2TOP: ANALYSE VON LOCK WAITING IM INTERACTIVE MODE..... | 2 |
| V9.7 : DB AUTHORITIES..... | 5 |
| NEU IN V9.7: LOCK ROUTINEN..... | 7 |
| SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG..... | 7 |
| DB2 AKTUELL 2011..... | 7 |
| NEWSLETTER ARCHIV..... | 8 |
| ANMELDUNG/ABMELDUNG..... | 8 |
| DIE AUTOREN DIESER AUSGABE..... | 8 |

End Of Support 9.1

Der End of support für die DB2 Version 9.1 ist nun offiziell auf den 30.04.2012 gesetzt wurden. Weitere Informationen zu diesem Thema finden Sie [hier](#).

Chats mit dem Labor

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

Das letzte Webinar mit them Labor fand am 28.02.2011 zum Thema [DB2 Update - What's New since V9.7 GA](#) statt. An dem Link hängt jetzt gleich das pdf zum Download dran.

Der nächste Chat mit dem Labor findet am 31.03.2011 um 18:30 zum Thema

DB2 and Oracle - An Architectural Comparison. Anmelden können Sie sich über [diesen Link](#). In diesem Chat sollen DB2 und ORACLE auf den Ebenen Architektur (sowohl Server, Memory), Parameter, Umgebung und Register Variablen, Datenbankspeicherung und Grundlagen der Administration verglichen werden. Ebenso werden Fragen zum ORACLE Kompatibilitätsmodus in DB2 beantwortet.

Praxisbeispiel db2top: Analyse von Lock Waiting im Interactive Mode

In der Februarausgabe 2009 wurde die grundsätzliche Funktionsweise von db2top vorgestellt. Mit dem Tool db2top stellt IBM eine Möglichkeit bereit, DB2 Umgebungen zu überwachen und Probleme schnell und effizient zu lokalisieren.

In diesem Artikel soll nun beispielhaft dargestellt werden, wie man db2top dazu nutzen kann Probleme wie Lock Waits zu analysieren und zu beheben.

In diesem Szenario läuft eine hohe Anzahl an Transaktionen im Hintergrund. Ein Simulationsprogramm versucht Tabellenzeilen zu löschen, was zur Folge hat, dass andere Sessions im „Lock Waiting“ Status gefangen sind.

Beim Analysieren des Bottleneck Bildschirms in db2top (Figur 1), kann man eine sehr hohe Zahl an Lock Waits erkennen:

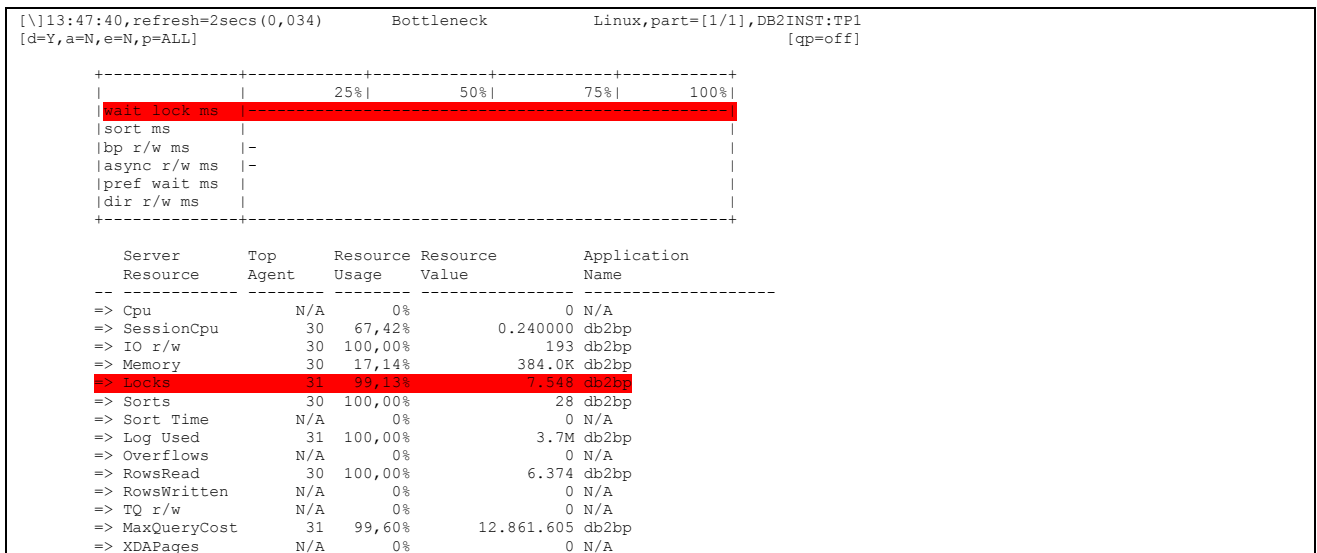


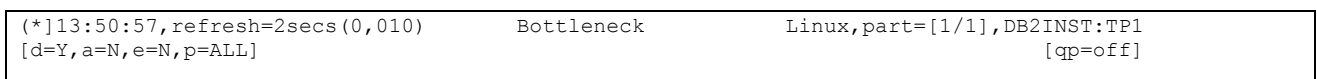
Abbildung 1: Lock Waits

Betrachtet man die Box im oberen Teil des Bildschirms lässt sich leicht erkennen, dass der Eintrag „wait lock ms“ im Vergleich zu den anderen Operationen die meiste Zeit in Anspruch nahm. Der Screenshot zeigt, dass eine oder mehr Applikationen im „lock waiting mode“ festsitzen und darauf warten, dass Sperren freigegeben werden.

Eine bewährte Vorgehensweise ist nun herauszufinden, welche Applikation in diesem Szenario die meisten Sperren hält. Figur 1 zeigt an, dass die Anwendung mit der Application ID (appid) 31 unter den „Top Agents“ ist. Die Spalte „Resource Usage“ zeigt an, dass 99,13% aller Sperren der Datenbank von dieser Applikation gehalten werden.

Um genauer zu sehen, was die Applikation tut, kann der Applikation Bildschirm verwendet werden (durch Eingabe von „a“). Den Session Bildschirm kann man zu Rate ziehen, um zu sehen welche Applikationen auf Sperren warten (durch Eingabe von „l“).

Gibt man „a“ im Bottleneck Bildschirm ein, erscheint ein Prompt um die agent id einzugeben. In unserem Fall ist dies die 31 und führt zu folgendem Bildschirm



```
*LOCAL.db2inst.110214124648, UOW Waiting in the application

select * from ACCT where acctid > 210
```

Abbildung 2: Lock haltende Applikation

Es ist notwendig zu verifizieren, dass dies in der Tat die SQL Anweisung ist, die andere Applikationen blockiert. Manchmal ist es möglich, dass ein „lock waiting“ Status auftritt, wenn eine Applikation auf Table Locks anstelle von Row Locks wartet, welche von einer Applikation mit sehr wenigen Sperrern gehalten werden.

Durch die Eingabe von „r“ gelangt man wieder zurück zum Bottleneck Bildschirm. Durch die Eingabe von „l“ kommt man in die Ansicht der Sessions. Hier sieht man u.U. sehr viele Sessions.

```
[\]14:56:09,refresh=2secs(0,006) Sessions Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL] [qp=off]
```

| Application Handle(Stat) | Cpu% Total | IO% Total | Mem% Total | Application Status | Application Name | Delta RowsRead/s | Delta RowsWritten/s |
|--------------------------|------------|-----------|------------|--------------------------------|------------------|------------------|---------------------|
| 30(i) | 0,00% | 0,00% | 10,53% | UOW Waiting in the application | db2bp | 0 | 0 |
| 31(i) | 0,00% | 0,00% | 8,77% | UOW Waiting in the application | db2bp | 0 | 0 |
| 155(i) | 0,00% | 0,00% | 8,77% | UOW Waiting in the application | sqltpltst | 0 | 0 |
| 156(i) | 0,00% | 0,00% | 12,28% | UOW Waiting in the application | sqltpltst | 0 | 0 |
| 157(i) | 0,00% | 0,00% | 8,77% | UOW Waiting in the application | sqltpltst | 0 | 0 |
| 158(i) | 0,00% | 0,00% | 7,02% | UOW Waiting in the application | sqltpltst | 0 | 0 |

Abbildung 3: Sessions ("l")

Da bei vielen Anwendungen diese Anzeige nicht hilfreich ist und man eigentlich nur die Lock-Informationen sehen möchte, geht man durch Eingeben von "U" direkt zum Locks Bildschirm:

```
[\]14:56:10,refresh=2secs(0,068) Locks Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL] [qp=off]
```

| Agent Id(State) | Application Name | Application Status | Object Name | Lock Mode |
|-----------------|------------------|--------------------------------|--------------------|-----------|
| 17(*) | db2fw0 | UOW Executing | DB2INST.LOCKMON | IX |
| 31(l) | db2bp | Lock Waiting | DB2INST.ACCT | X |
| 31(l) | db2bp | Lock Waiting | DB2INST.ACCT[0] | X [IS] |
| 31(l) | db2bp | Lock Waiting | Internal Plan | S |
| 31(l) | db2bp | Lock Waiting | Internal Variation | S |
| 155(l) | sqltpltst | Lock Waiting | Internal Plan | S |
| 155(l) | sqltpltst | Lock Waiting | DB2INST.ACCT[0] | IX [IX] |
| 156(l) | sqltpltst | Lock Waiting | Internal Plan | S |
| 156(l) | sqltpltst | Lock Waiting | DB2INST.ACCT[0] | IX [IX] |
| 157(l) | sqltpltst | Lock Waiting | Internal Plan | S |
| 157(l) | sqltpltst | Lock Waiting | DB2INST.ACCT[0] | IX [IX] |
| 158(l) | sqltpltst | Lock Waiting | Internal Plan | S |
| 158(l) | sqltpltst | Lock Waiting | DB2INST.ACCT[0] | IX [IX] |
| 30(i) | db2bp | UOW Waiting in the application | DB2INST.ACCT | IS |
| 30(i) | db2bp | UOW Waiting in the application | Internal Plan | S |
| 30(i) | db2bp | UOW Waiting in the application | Internal Variation | S |

Abbildung 4: Locks ("U")

Die Application mit id 30 zeigt den "UOW Waiting" Status an, während Application mit id 31 sich im „Lock Waiting“ Status befindet. Durch Drücken der linken Pfeiltaste kann man in dem Locks Screen scrollen, z.B. zur nachfolgenden Ansicht:

```
[/]14:57:09,refresh=2secs(0,059) Locks Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL] [qp=off]
```

| | |
|---------------------|---------------|
| Locks held.....: | 7.555 [0,37%] |
| Agents waiting...: | 5 |
| Appls Connected...: | 6 |

| Agent Id(State) | Object Name | Lock Mode | Object Type | Lock Status | Lock Count | Is Blocker | Locked By |
|-----------------|--------------------|-----------|-------------|-------------|------------|------------|-----------|
| 17 (*) | DB2INST.LOCKMON | IX | Table | Granted | 1 | No | - |
| 31 (1) | DB2INST.ACCT | X | Row | Granted | 7544 | Yes | 30 |
| 31 (1) | DB2INST.ACCT[0] | X [IS] | Table | - | - | Yes | 30 |
| 31 (1) | Internal Plan | S | Plan | Granted | 1 | Yes | 30 |
| 31 (1) | Internal Variation | S | Variation | Granted | 1 | Yes | 30 |
| 155 (1) | Internal Plan | S | Plan | Granted | 1 | No | 31 |
| 155 (1) | DB2INST.ACCT[0] | IX [IX] | Table | - | - | No | 31 |
| 156 (1) | Internal Plan | S | Plan | Granted | 1 | No | 31 |
| 156 (1) | DB2INST.ACCT[0] | IX [IX] | Table | - | - | No | 31 |
| 157 (1) | Internal Plan | S | Plan | Granted | 1 | No | 31 |
| 157 (1) | DB2INST.ACCT[0] | IX [IX] | Table | - | - | No | 31 |
| 158 (1) | Internal Plan | S | Plan | Granted | 1 | No | 31 |
| 158 (1) | DB2INST.ACCT[0] | IX [IX] | Table | - | - | No | 31 |
| 30 (i) | DB2INST.ACCT | IS | Table | Granted | 1 | Yes | - |
| 30 (i) | Internal Plan | S | Plan | Granted | 2 | Yes | - |
| 30 (i) | Internal Variation | S | Variation | Granted | 1 | Yes | - |

Abbildung 5: Locks ("←")

Sollten Sie nicht alle Spalten wie im o.g. Beispiel, sehen, verbreitern Sie ihren Bildschirm.

Die interessanten Spalten aus dieser Ansicht sind der Lock Mode, Lock Status und Locked By.

Hier sieht man, dass die Application mit id 31 mehr als 7547 exclusive (Lock Mode = X) Row Locks (Object Type = Row) auf die Tabelle ACCT (Object Name) hält und diese Row Locks in einen Table Lock umwandeln möchte (Lock Mode = „X [IS]“), dies aber nicht kann, da die Tabelle durch Application mit id 30 (Locked By) mit Intent Share (IS) blockiert wird. Dies ist daran zu erkennen, dass der Lock Status "-" zeigt.

Weiterhin sieht man, dass die Application mit id 30 einen Intent Share (Lock Mode = IS) Table Lock (Object Type = Table) auf die Tabelle ACCT hält.

Weitere Informationen zu den Lock Modes sind [hier](#) zu finden.

Werden zu viele Locks angezeigt, kann man auch mittel „L“ sich die Lock Kette anzeigen lassen.

```
[*]14:56:18,refresh=2secs(0,023)                               Locks                               Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL]                                           [qp=off]
-----
|
|   Blocked/Blocking Agent Chain
|   -----
|   30->31->155
|   31->155
|
|
|   Press any key to resume...
|
+-----
```

Abbildung 6: Lock Chain ("L")

Nachdem nun der Lock-Holder identifiziert wurde, möchte man auch wissen, welches SQL diese Anwendung ausführt. Dies erfolgt durch "a" und dann Eingabe der Agent ID (in unserem Fall 31).

```
(*]14:48:26,refresh=1secs(0,003)                               Locks                               Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL]                                           [qp=off]
*LOCAL.db2inst.110214124722, Lock Waiting, blocked by 30 on DB2INST .ACCT

delete from acct where acct > 50
```

Abbildung 7: Lock Waiting Anwendung ("L")

Diese Abbildung zeigt, dass die Anwendung mit id 31 ein delete auf die Tabelle ACCT ausführt. Weiterhin sieht man auch, wer die Tabelle blockiert (blocked by 30).

Nachdem die SQL für beide Anwendungen bekannt sind, kann entschieden werden, welche der beiden gelöscht werden soll (Warten auf ein SQL0911 ist nicht möglich, da LOCKTIMEOUT für diesen Test auf -1 gesetzt wurde und Voraussetzungen für den Deadlock auch nicht gegeben sind). In der „SQL“ Ansicht kann mittels „f“ die Anwendung beendet werden.

```

15:15:16,refresh=1secs(0,001)                               Locks                               Linux,part=[1/1],DB2INST:TP1
[d=Y,a=N,e=N,p=ALL]                                         [qp=off]

Are you sure you want to force application 30 (N/y) ?
    
```

Abbildung 8: force Application ("f")

Möchte man jedoch das Select-Statement löschen, geht man mit „r“ zurück zur Locks-Ansicht und dann mit a und der Agent ID (30) in die SQL-Ansicht der Anwendung mit dem select Statement und kann dort mit „f“ die Anwendung beendet.

Weiterführende Informationen sind im Artikel ["DB2 problem determination using db2top utility"](#) zu finden.

V9.7 : DB Authorities

Bei der Recherche zu den Event Monitoren bin ich über eine Authority SQLADM gestolpert, die neu in DB2 V9.7 eingeführt wurde. Zusätzlich zu dieser Authority wurden noch weitere neue DB Authorities eingeführt, die ebenso mittels:

```
db2 grant|revoke <AUTHORITY> to|from user|group <USER/GROUP>
```

gesetzt oder entfernt werden können.

Hier eine Kurze zusammenstellung der neuen Authorities und deren Verwendung:

| DB Authority | Beschreibung | Granted durch Authority |
|--------------|--|-------------------------|
| SECADM | Security Authority, erlaubt Anlegen und Verwalten security-relevanter Objekte (wie Rollen, LBAC, Audit policies, ..) und zur Vergabe der Rechte auf Datenbank Objekte. Mit dieser Authority kann nur auf System-Katalog Daten, jedoch nicht auf die Daten in den User-Tabellen zugegriffen werden. SECADM wurde bereits in V9.5 eingeführt, in V9.7 wurden die Privilegien wesentlich erweitert. Der Instanz-Owner hat nicht von vornherein das SECADM-Recht und kann sich dieses Recht auch nicht selbst zuordnen (SQL0554 An authorization ID cannot grant a privilege or authority to itself). | SECADM |
| SQLADM | Ermöglicht Monitoring und Tuning von SQL. Wird benötigt, um mit Event Monitoren zu arbeiten (Anlegen, Starten, Stoppen und Löschen). Weiterhin ermöglicht diese Authority das Durchführen von Runstats und Reorg, sowie das Absetzen von SQL Abfragen auf die WLM (Workload management) Objekte. Ist eine Untermenge der DBADM Authority. | ACCESSCTRL, SECADM |
| EXPLAIN | Wird benötigt, um EXPLAIN Pläne zu erstellen, ohne jedoch auf die Dateninhalte zuzugreifen. Ist eine Untermenge der SQLADM Authority. | ACCESSCTRL, SECADM |
| ACCESSCTRL | Mit dieser Authority können Rechte auf User-Objekte gesetzt oder entfernt werden. Diese Rechte sind jedoch nicht vererbbar. | SECADM |
| DATAACCESS | Zugriff auf die Daten der User-Objekte. Dieses Recht ist nicht für PUBLIC möglich. DATAACCESS beinhaltet LOAD, SELECT, INSERT, UPDATE, DELETE auf die Tabellen, Views, Indizes, etc. DATAACCESS beinhaltet zusätzlich auch noch die Rechte zum EXECUTE auf Packages und Functions. | SECADM |

| DB Authority | Beschreibung | Granted durch Authority |
|--------------|---|-------------------------|
| WLMADM | Diese Authority ist notwendig, um WLM zu managen, d.h. zum Anlegen, Löschen, Verändern, Verwalten der Zugriffsrechte auf diese Objekte. | ACCESSCTRL, SECADM |

Entsprechend der neu eingeführten Authorities wurde auch der View `SYSCAT.DBAUTH` erweitert:

```
>db2 describe table syscat.dbauth
```

```
Data type Column
Column name      schema      Data type name  Length Scale Nulls
-----
```

| Column name | schema | Data type name | Length | Scale | Nulls |
|---------------------|--------|----------------|--------|-------|-------|
| GRANTOR | SYSIBM | VARCHAR | 128 | 0 | No |
| GRANTORTYPE | SYSIBM | CHARACTER | 1 | 0 | No |
| GRANTEE | SYSIBM | VARCHAR | 128 | 0 | No |
| GRANTEETYPE | SYSIBM | CHARACTER | 1 | 0 | No |
| BINDADDAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| CONNECTAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| CREATETABAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| DBADMAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| EXTERNALROUTINEAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| IMPLSCHEMAAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| LOADAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| NOFENCEAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| QUIESCECONNECTAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| LIBRARYADMAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| SECURITYADMAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| SQLADMAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| WLMADMAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| EXPLAINAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| DATAACCESSAUTH | SYSIBM | CHARACTER | 1 | 0 | No |
| ACCESSCTRLAUTH | SYSIBM | CHARACTER | 1 | 0 | No |

20 record(s) selected.

Auch wenn EXPLAIN eine Untermenge von SQLADM ist, wird durch setzen von SQLADM nicht automatisch EXPLAIN gesetzt.

```
>db2 grant sqladm on database to user db2admin
DB20000I The SQL command completed successfully.
```

```
>db2 "select * from syscat.dbauth where grantee='DB2ADMIN'"
```

```
GRANTOR
GRANTORTYPE GRANTEE
GRANTEETYPE BINDADDAUTH CONNECTAUTH CREATETABAUTH DBADMAUTH EXTERNALROUTINEAUTH IMPLSCHEMA
AUTH LOADAUTH NOFENCEAUTH QUIESCECONNECTAUTH LIBRARYADMAUTH SECURITYADMAUTH SQLADMAUTH WLMADMAUTH EXPLAINAUTH DATAACCESS
AUTH ACCESSCTRLAUTH
-----
-----
-----
-----
-----
-----
DB2USER
U DB2ADMIN
U N N N N N N
N N N N N Y N N N
N
```

1 record(s) selected.

Mit dem SQLADM hat man keine Rechte auf den Inhalt von Tabellen.

```
>db2 "select * from db2user.employee"
SQL0551N "DB2ADMIN" does not have the required authorization or privilege to
perform operation "SELECT" on object "DB2USER.EMPLOYEE". SQLSTATE=42501
```

Nach dem jedoch die Explain Tabellen angelegt wurden, ist die Erstellung des Zugriffsplanes möglich:

```
>db2 "explain plan for select * from db2user.employee"
DB20000I The SQL command completed successfully.
```

```
>db2exfmt -1 -d sample -o db2user.out
DB2 Universal Database Version 9.7, 5622-044 (c) Copyright IBM Corp. 1991, 2009
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool
Connecting to the Database.
Connect to Database Successful.
Binding package - Bind was Successful
Using only explain schema found: DB2ADMIN.
Output is in db2user.out.
Executing Connect Reset -- Connect Reset was Successful.
```

Eine interessante Aufstellung und Übersicht der Authorities und deren Zusammenspiel ist [hier](#) zu finden. Weitere Informationen sind in der [DB2 V9.7 Online Hilfe](#) zu finden.

Neu in V9.7: LOCK Routinen

Beginnend mit V9.7 FP1 stehen die neuen Monitor Interfaces `MON_GET_APPL_LOCKWAITS`, `MON_GET_LOCKS` und `MON_FORMAT_LOCK_NAME` zur Verfügung, um LOCK bezogene Situationen zu erkennen und zu analysieren. Diese neuen Monitore sind wesentlich effizienter und erzeugen weniger Overhead.

Folgende Routinen stehen zur Verfügung:

- [MON_GET_APPL_LOCKWAITS](#)
Listet alle Locks und deren Anwendungen auf, die im LOCKWAIT Mode stehen (bezogen auf die verbundene Datenbank).
- [MON_GET_LOCKS](#)
Zeigt alle LOCKS einer Datenbank an.
- [MON_FORMAT_LOCK_NAME](#)
Formatiert die internen Lock Namen und zeigt Details des jeweiligen LOCKs an. Die Ausgabe erfolgt zeilenbasiert.

Darüber hinaus gibt es den Administrations-View [MON_LOCKWAITS](#) (Schema: SYSIBMADM), die Informationen zu den DB2 Agents anzeigt, die auf LOCKs warten.

Interessant ist die neue Routine `MON_FORMAT_LOCK_NAME`, die eine wesentlich bessere Darstellung eines internen Locknamens erlaubt. Hier ein Beispiel:

```
SELECT SUBSTR(NAME,1,20) AS NAME, SUBSTR(VALUE,1,50) AS VALUE  
FROM TABLE( MON_FORMAT_LOCK_NAME('00000000E00000000000B00C152')) as LOCK
```

Ergebnis:

| NAME | VALUE |
|------------------|-------------|
| LOCK_OBJECT_TYPE | ROW |
| TBSP_NAME | SYSCATSPACE |
| TABSHEMA | SYSIBM |
| TABNAME | SYSPLANDEP |
| ROWID | 000000B00C1 |

Den binären Locknamen findet man z.B. im `db2diag.log` oder als Wert des Monitorelements `lock_name`.

Diese neuen Monitor-Schnittstellen ersetzen die folgenden, abgekündigten Administratorviews:

- SNAPLOCK
- SNAPLOCKWAIT
- LOCKS_HELD
- LOCKWAITS

und die Tabellen-Funktion `SNAPGETLOCK`.

Weiterführende Informationen sind in der [DB2 V9.7 Online Hilfe](#) zu finden.

Schulungen / Tagungen / Informationsveranstaltung

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

DB2 Aktuell 2011

**Für DB2 Systemprogrammierer, Administratoren und Anwendungsentwickler:
Zwei Tage Wissenstransfer und Erfahrungsaustausch, die sich lohnen!**

Nutzen Ihre Kunden bereits das volle Wertschöpfungspotential ihrer DB2 Systeme?
Wie können Ihre Kunden von DB2-Neuerungen bestmöglich profitieren?

Die DB2 Aktuell bietet Ihren Kunden Antworten auf diese Fragen und:

Zwei Tage konzentrierten Wissenstransfer mit hochkarätigen Referenten.

Viele DB2-Größen kommen aus den Silicon Valley und Toronto Labs der IBM nach Bremen. Die vorgestellten Referenzprojekte bieten konkrete Praxiseinblicke.

Eine hervorragende Plattform für aktiven Erfahrungsaustausch und Kontaktpflege.

Die Teilnehmer können ihre Fragen, Ideen und Probleme im direkten Austausch mit den DB2-Experten der IBM im Detail diskutieren - auch außerhalb der Agenda. Der Austausch mit anderen DB2-Anwendern bietet neue Impulse für weitere DB2 Projekte.

Bis zu 3 kostenlose technische Zertifizierungstests

können im Gegenwert von jeweils 200 US\$ kostenfrei vor Ort abgelegt werden.

Das besondere Zusatzangebot:

Der Zertifizierungstag am 19. September 2011 für nur 300 € Mehrkosten zur Vorbereitung auf die DB2 9 Fundamentals Zertifizierung , d.h Test # 730

Bis 31. Mai den Frühbucher-Rabatt nutzen:

Konferenz-Ticket DB2 Aktuell € 990,- pro Person zzgl. USt bis 31.05.

Kombi-Ticket Zertifizierungstag & Konferenz € 1.290,- pro Person zzgl. USt bis 31.05.

Den 3. Teilnehmer pro Firma gibt es umsonst: Die 3-für-2-Option

Wenn sich 3 Mitarbeiter einer Firma anmelden, zahlen nur 2 Mitarbeiter die Teilnehmergebühr Anmelde-Links

Kunden: [Konferenzwebseite und externe Anmeldung](#)

IBM-intern: [Learning@IBM-Link](#)

Ansprechpartner: [Sabine Kaschta](#) sabine_kaschta@de.ibm.com

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 | IT-Spezialist für DB2 LUW, IBM SWG Chief-Editor DB2NL djs@de.ibm.com |
| Gerhard Müller | Senior Certified IT-Specialist, IBM SWG Artikel: Neu in V9.7: LOCK Routinen |
| Annetta Fourkiotis | IT-Specialist, IBM SWG Artikelvorlage: Praxisbeispiel db2top: Analyse von Lock Waiting im Interactive Mode |

Reviewer und Ideenlieferanten:

| | |
|-----------------|---------|
| Dirk Fechner | IBM SWG |
| Frank Berghofer | IBM SWG |
| Wilfried Hoge | IBM SWG |