

## Willkommen zum „IBM DB2 Newsletter“

**Liebe Leserinnen und Leser,**

Das Jahr 2012 ist nun schon wieder 3 Monate alt. Die kalte Jahreszeit gehört hoffentlich der Geschichte an, die Cebit ist vorbei und Osterferien sowie Ostern steht vor der Tür.

Damit Sie Ihre Zeit rund um Ostern optimal ausnutzen können, haben wir einen sehr interessanten und ausführlichen Beitrag für Sie. Danke an Herrn Tiefenbacher und sorry, das die Herausgabe etwas gedauert hat.

Auch DB2 wartet noch mit einer Überraschung vor Ostern auf (s.h. Artikel „[Management meets the future](#)“). Da kommt wieder viel Neues auf uns zu. Lassen Sie sich überraschen.



Und nun noch ein Aufruf in eigener Sache: Wie jedes Jahr suchen wir „SIE“, mit Ihrem Praxis-Wissen über DB2. Sie müssen keine Artikel oder Romane schreiben (natürlich nehmen wir diese auch gerne). Um Ihnen aber die Arbeit zu vereinfachen: Tipps reichen auch.

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

Ihr TechTeam

## Inhaltsverzeichnis

<b>DB2 SECTIONS EXPLAIN &amp; SECTION ACTUALS.....</b>	<b>2</b>
EXPLAIN_FROM_SECTION.....	3
EXPLAIN_FROM_ACTIVITY.....	7
WAR'S DAS? NICHT WIRKLICH – ES GEHT SOGAR NOCH MEHR!.....	9
<b>CHATS MIT DEM LABOR.....</b>	<b>10</b>
<b>SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....</b>	<b>11</b>
THINK BIG!.....	11
IBM DB2 OPTIMIERT FÜR SAP.....	11
DATA MANAGEMENT MEETS THE FUTURE.....	12
DB2 AKTUELL 2012.....	13
<b>NEWSLETTER ARCHIV.....</b>	<b>13</b>
<b>ANMELDUNG/ABMELDUNG.....</b>	<b>14</b>
<b>DIE AUTOREN DIESER AUSGABE.....</b>	<b>14</b>

## DB2 Sections Explain & Section Actuals

Explain – also die Analyse des DB2 Zugriffsweges – wird immer noch zu selten gemacht, obwohl im SQL sehr viel Performance-Potential steckt.

DB2 9.7 hat durch das explizite [Explain-Authority](#) schon einen wichtigen Schritt zu mehr Praktikabilität gemacht und bringt in Fixpack 1 und Fixpack 2 einige sehr nützliche Erweiterungen mit, die die Analyse einfacher gestalten. So ist es nun möglich die Zugriffswege, die der DB2 Optimizer gewählt hat, auch für die SQL-Abfragen aufzurufen, die bereits gelaufen sind, sich aber noch im Package Cache befinden.

Aber von vorne: Jedes SQL-Statement beschreibt ja nur „WELCHE“ Daten zur Anzeige kommen sollen, nicht aber „WIE“ man zu ihnen gelangt (also z. B. via Indexzugriff etc.). Diese Umsetzung des SQLs in einen gültigen und günstigen Zugriffspfad ist Aufgabe des DB2 Optimizers, der Bestandteil des SQL-Compilers ist. Nachdem das SQL auf syntaktische und semantische Korrektheit überprüft wurde, kann es ggf. umgeschrieben werden (re-write Mechanismus), wenn es dadurch besser optimiert werden kann. Natürlich kann dies nur insofern passieren, wie die gleiche Ergebnismenge mit einem alternativen SQL-Statement erzeugt werden kann. Zusätzlich Logik etc. kann der Optimizer hier nur bedingt ergänzen (z.B. WHERE A >= 5 AND A <= 10 ergibt WHERE A BETWEEN 5 AND 10), was zur Folge hat, dass trotz Optimierung immer noch gutes und schlechtes SQL – aus Performance-Sicht – möglich ist.

Mittels Explain kann sich ein Anwendungsentwickler oder DB2 Administrator nun das Ergebnis dieser Optimierung – also den tatsächlichen Zugriffsweg – anschauen. Während man bisher dafür das SQL extra „explained“ hat bzw. in einem Testsystem ausgeführt hat und dabei die Explain-Daten erzeugen hat lassen, ist dies nun auch für SQLs möglich, die gerade produktiv gelaufen sind.

Da diese somit unter realen Bedingungen gelaufen sind, kann man so den tatsächlich verwendeten Zugriffsweg analysieren und nicht – wie bisher – den Zugriffsweg, den DB2 unter den Testumgebungen am Testsystem ausgewählt hatte. Durch Änderungen an den Datenmengen und aktualisierten Statistik-Daten kann es sehr wohl zu Abweichungen kommen, die dann zu einem unvorhergesehenen Verhalten führen und immer nur schwer erklärbar und nicht beweisbar waren.

Section Explain wird diese neue Möglichkeit genannt, da man unter Section die ausführbaren Strukturen eines einzelnen SQLs versteht. Diese werden in einem Package oder im Package Cache gespeichert, um dort wiederverwendet zu werden.

Eine Reihe neuer Funktionen erlaubt es nun, diese Explain-Daten zu sammeln und in den Explain-Tabellen für eine weitere Analyse zur Verfügung zu stellen. Eine Darstellung mit Visual Explain ist leider nicht möglich, weil die graphischen Komponenten bei diesem Verfahren fehlen. So bleibt zur Auswertung der Explains das mächtige Befehlszeilentool [db2exfmt](#).

Die nötigen Explain-Tabellen erstellt oder migriert man am besten mit der dafür vorgesehenen Stored Procedure:

```
call SYSINSTALLOBJECTS ('EXPLAIN', 'C', '', '')
```

Wenn man statt 'C', das für „Create“ steht, mit 'M' arbeitet, werden die Tabellen migriert und so auf die für DB2 9.7 taugliche Struktur umgestellt.

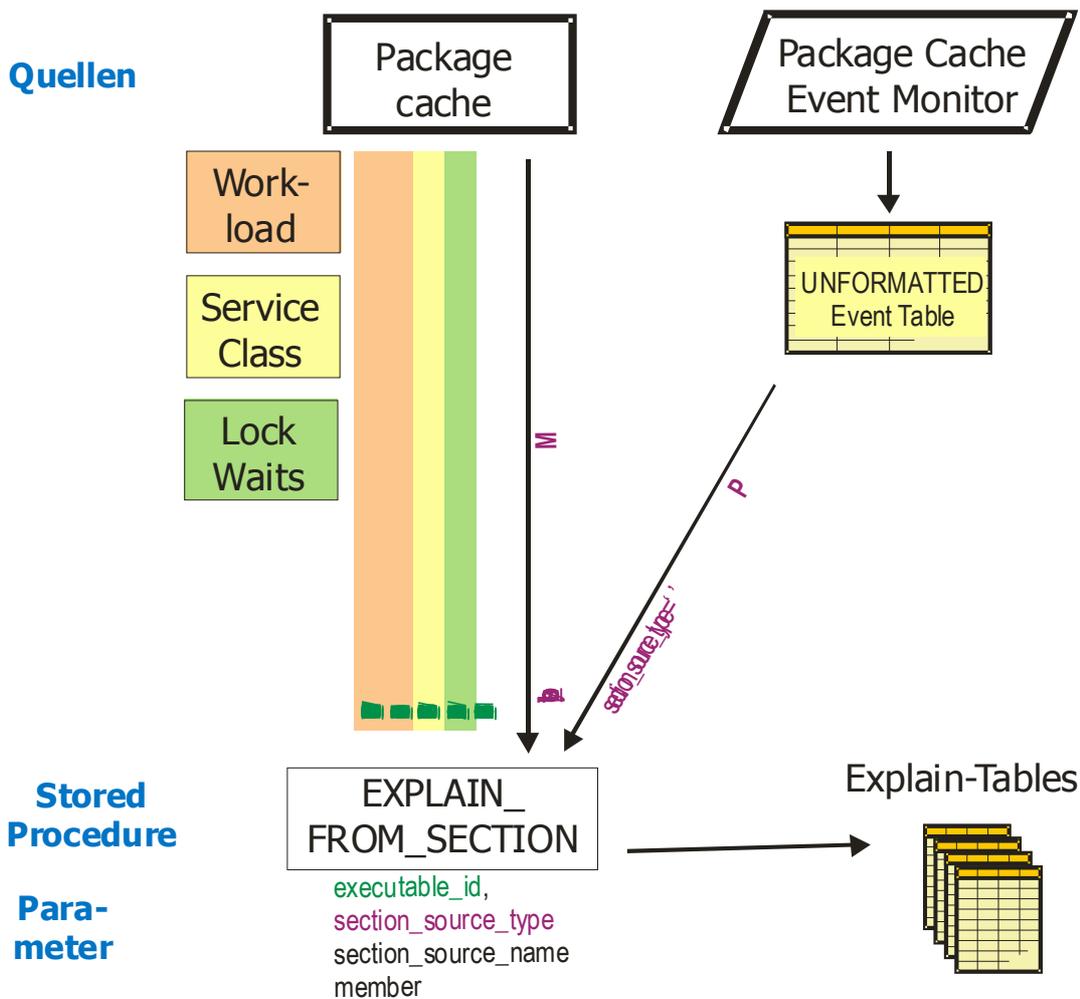
Aktuell gibt es vier Stored Procedures, um die relevanten Informationen aus unterschiedlichen Quellen zu sammeln und in den Explain-Tabellen zu speichern:

- [EXPLAIN FROM CATALOG](#)  
ist für statisches SQL gedacht, da hier die Packages aus dem System-Katalog explained werden.
- [EXPLAIN FROM SECTION](#)  
Diese Prozedur ist besonders interessant, weil Sie alternativ zwei verschiedene Quellen nutzen kann. Über einen der Parameter kann man steuern, ob die Daten aus dem Package Cache verwendet werden sollen oder aus der „unformatted“ Event-Tabelle eines Package Cache Event Monitors (dieser kam mit DB2 9.7 Fixpack 1)
- [EXPLAIN FROM DATA](#)  
Mit dieser Funktionalität lassen sich ebenfalls die Daten eines Package Cache Event Monitors analysieren, allerdings erst dann, wenn die Daten mittels [EVMON FORMAT UE TO TABLE](#) aus der „unformatted“ Event-Tabelle in ein Datenmodell überführt worden sind.
- [EXPLAIN FROM ACTIVITY](#)  
hat als Quelle einen Activity Event Monitor, den man auch ohne WLM-Feature verwenden darf, solange man keine sonstigen WLM-Objekte anlegt. Mit anderen Worten heißt dies, dass man so nur die komplette Datenbank überwachen kann und keine feineren Abstufungen nutzen kann.  
Mit Fixpack 2 kommt allerdings eine interessante Erweiterung, mit der diese Funktionalität auf Connection-Ebene – mittels [WLM SET CONN ENV](#) – aktiviert werden kann.

Besonders den Funktionen EXPLAIN\_FROM\_SECTION und EXPLAIN\_FROM\_ACTIVITY werden wir uns im Folgenden näher widmen, da diese sicherlich die höchste Relevanz und mehr Anwendungsmöglichkeiten besitzen.

### **EXPLAIN\_FROM\_SECTION**

Zur besseren Übersicht ein Bild, anhand dessen die Möglichkeiten von EXPLAIN\_FROM\_SECTION im Folgenden erörtert werden.



Beim Aufruf von EXPLAIN\_FROM\_SECTION muss eine EXECUTABLE\_ID mitgegeben werden. Diese identifiziert ein Statement und stellt ein eindeutiges binäres Token dar vom Datentyp VARCHAR(32) FOR BIT DATA. So könnte dieses beispielhaft aussehen:

```
x'010000000000000000CA05000000000000000000000000000000020020110722213236732000'
```

Somit muss man also das per Explain zu analysierende SQL-Statement erst einmal identifizieren. Dazu gibt es wieder eine Reihe von Möglichkeiten, die alle diese eindeutige EXECUTABLE\_ID liefern:

- Event Monitore
  - Activity Event Monitor
  - Package Cache Event Monitor
- WLM-Funktionalitäten
  - [WLM\\_GET\\_WORKLOAD\\_OCCURRENCE\\_ACTIVITIES\\_V97](#)
  - [MON\\_GET\\_ACTIVITY\\_DETAILS](#)
  - MON\_GET\_SERVICE\_CLASS\_AGENTS\_V97
- [MON\\_GET\\_PKG\\_CACHE\\_STMT](#) (& [MON\\_GET\\_PKG\\_CACHE\\_STMT\\_DETAILS](#))

So lässt sich, wie im folgenden Beispiel gezeigt, der aktuelle Package Cache mit der MON\_GET\_PKG\_CACHE\_STMT Tabellenfunktion auslesen.

```
select EXECUTABLE_ID, TOTAL_CPU_TIME
       from table(MON_GET_PKG_CACHE_STMT (NULL,NULL,NULL,-1)) as t
       order by TOTAL_CPU_TIME desc;
```

EXECUTABLE_ID	TOTAL_CPU_TIME
x'0100000000700F0000000100921151730568798'	446532

Selbstverständlich lassen sich hier auch weitere Kriterien und Ergebnisspalten ausgeben, dieses Beispiel ist hier nur zur Veranschaulichung gewählt.

Dies ist selbstverständlich nur solange möglich, wie das zu analysierende Statement auch im Cache enthalten ist und noch nicht verdrängt bzw. überschrieben wurde.

Wenn auch SQLs, die bereits vor einem längeren Zeitraum ausgeführt worden sind, analysiert werden sollen, kann man im Allgemeinen nicht mehr davon ausgehen, dass sich diese noch im Cache befinden. Hier hilft es einen Package Cache Monitor aufzusetzen und dann mit dem Übergabe-Parameter `section_source_type = 'P'` zu arbeiten, um so die Daten aus der unformatierten Monitoringtabelle zu holen. Voraussetzung ist natürlich, dass dieser Monitor zum Zeitpunkt der SQL-Abarbeitung aktiv war. So sind einige interessante Möglichkeiten vorstellbar – wie zum Beispiel die Überwachung der Zugriffswege und deren Veränderungen über einen längeren Zeitraum.

Sind nun die interessanten oder analyserelevanten SQL-Befehle gefunden, wird die jeweilige EXECUTABLE\_ID an die eigentliche Analyse-Stored Procedure – in diesem Falle EXPLAIN\_FROM\_SECTION – übergeben, neben allen anderen benötigten Parametern:

```
call EXPLAIN_FROM_SECTION
     (x'0100000000700F0000000100921151730568798',
      'M',NULL,0,NULL,
      ?,?, ?, ?, ?)
```

Wert der Ausgabeparameter

-----

Parametername: EXPLAIN\_SCHEMA

Parameterwert: MTUSER

Parametername: EXPLAIN\_REQUESTER

Parameterwert: MTUSER

...

Return Status = 0

Der erste Parameter ist – wie unschwer zu erkennen – die EXECUTABLE\_ID. Diese wird vom SECTION\_SOURCE\_TYPE gefolgt, der für die Auswahl der Quelle verantwortlich ist und hier per „M“ auf den Package Cache verweist – wie auch im obigen Bild gezeigt. Der dritte Parameter ist der SECTION\_SOURCE\_NAME, der zum Einsatz kommt, wenn die (unformatierte) Tabelle eines Package Cache Monitors als Quelle dient. Nach der Angabe des Members (hier 0) folgt dann noch optional das EXPLAIN\_SCHEMA, also das Schema der zu füllenden EXPLAIN-Tabellen.

Bei den fünf Fragezeichen handelt es sich um die Ausgabeparameter, die so bei all diesen Stored Procedures gleich sind. Das genügt, um alle für Explain nötigen Daten in den EXPLAIN-Tabellen zu speichern.

Jetzt fehlt „nur“ noch die Auswertung. Wie bereits angesprochen kann hier nicht auf die grafischen Hilfsmittel wie Visual Explain zurückgegriffen werden, sondern man muss die Auswertung mit Hilfe des Befehlszeilentools db2exfmt vornehmen. Dies ist allerdings auch sehr mächtig und wird von vielen Spezialisten zur Analyse bevorzugt.

Grafische Darstellungen der Ergebnis-Darstellung müssten durch eigene Tools oder am Markt verfügbare Standardtools ergänzt werden.

Will man hier nicht gleich alle Register ziehen, sondern nur schnellst möglich zum Ergebnis kommen, tut es oft folgender Aufruf:

```
db2exfmt -d sample -l -o explain_1.txt
```

Die erste Option spezifiziert die Datenbank, die letzte (-o) gibt an, in welche Datei die formatierte Textausgabe aus den Explaintabellen gespeichert werden soll. Der mittlere Parameter „-l“ ist eine Art Abkürzung und nutzt Standardwerte für viele wichtige Parameter. Er sorgt dafür, dass der aktuelle Explaineintrag verwendet wird und steht für diese Liste an Optionen:

```
-e % -n % -s % -v % -w -l -# 0
```

Das Ergebnis kann dann für einen sehr einfachen Fall so aussehen:

Explain level: **Explain from section**

Access Plan:

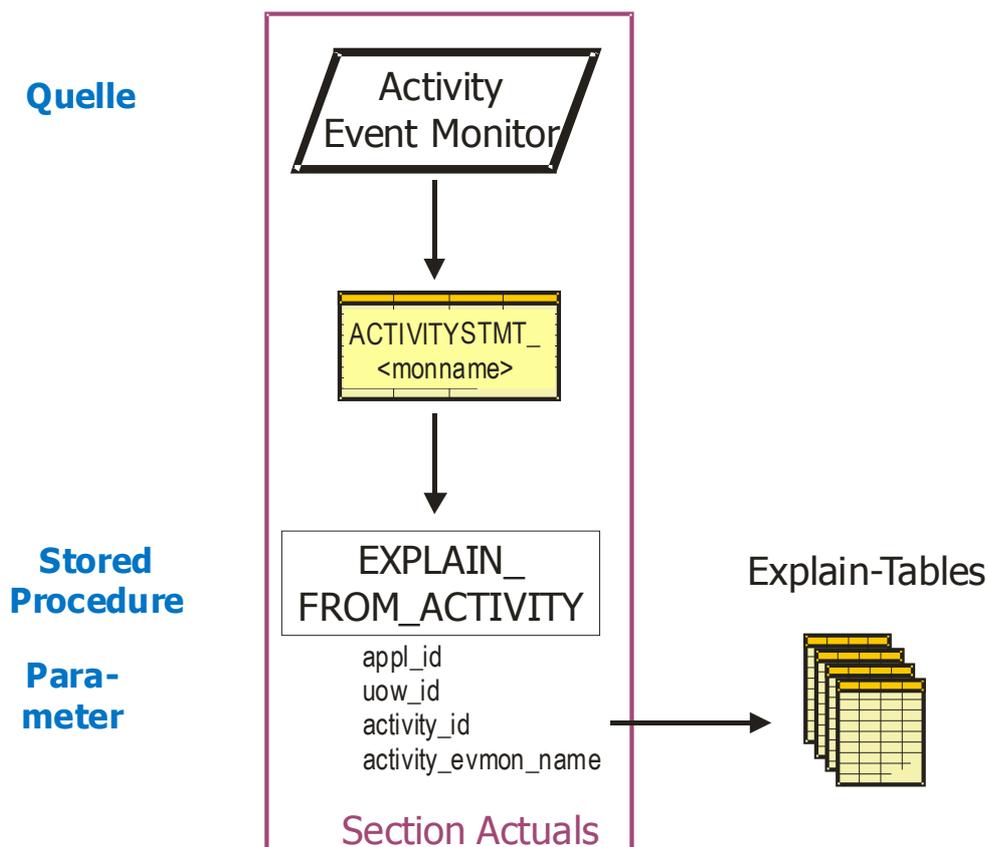
```
-----  
Total Cost:      15.1645  
Query Degree:    1  
  ROWS  
  RETURN  
  ( 1)  
  Cost  
  I/O  
  |  
  42  
  TBSCAN  
  ( 2)  
  15.1645  
  NA  
  |  
  42  
TABLE: DB2ADMIN  
EMPLOYEE  
Q1
```

In diesem gewählten Ausschnitt sieht man bereits beim „Explain level“, dass es sich um einen Section-Explain handelt. Die I/O-Kosten sind hier wie bei allen Section Explains „NA“ und somit nicht gesammelt worden. Dies ist einer der kleinen Unterschiede zu den herkömmlichen Explains, da hier nicht alle Spalten der Explaintabelle befüllt werden, was die Effizienz erhöht.

Soweit hoffe ich, stellt das keinen zu komplexen Prozess dar. Es wurde hier extra darauf verzichtet alle Optionen und Möglichkeiten aufzuzeigen, da unserer Meinung nach der Nutzen nicht in der Fülle der Möglichkeiten liegt, sondern darin, dass diese Funktionalität genutzt wird. Lieber einfach halten und so öfter anwenden – lautet das Motto.

### EXPLAIN\_FROM\_ACTIVITY

Bei der zweiten Alternative soll auch wieder ein Bild zur Orientierung dienen. Die Quelle ist diesmal ein Activity Event Monitor, der seine Daten in einer Tabelle speichert, die typischer Weise den Namen des Eventmonitors trägt mit „ACTIVITYSTMT\_“ als Prefix.



Activity Eventmonitor wurde mit der WLM (Workload Management) Funktionalität in DB2 9.5 eingeführt. Auch ohne WLM-Feature dürfen diese Monitore mit den Standard-Objekten in der Datenbank verwendet werden. Dies bedeutet, dass man zwar keine Workload und Service Classes anlegen kann,

um das Monitoring so besser zu fokussieren, aber die bestehende Workload „SYSDEFAULTUSERWORKLOAD“ und die in jeder Datenbank seit DB2 9.5 existierende Service Class „SYSDEFAULTUSERCLASS“ dürfen verwendet werden. Dies entspricht somit einem Monitoring auf Datenbankebene.

Somit wären also folgende Schritte zur Vorbereitung nötig:

```
-- Anlegen Activity Event Monitor
  create event monitor ACTMON
    for ACTIVITIES write to table;

-- Aktivieren des Monitors
  set event monitor ACTMON state 1;

-- Anpassen der Workload, um Daten zu sammeln
  alter workload SYSDEFAULTUSERWORKLOAD
    COLLECT ACTIVITY DATA on coordinator with DETAILS,SECTION;
```

Alternativ könnte auch die WLM\_SET\_CONN:ENV Prozedur verwendet werden, die in Fixpack 2 beinhaltet war. Das sähe dann folgendermaßen aus:

```
call WLM_SET_CONN_ENV(12345,
  '<collectactdata>WITH DETAILS, SECTION</collectactdata>')
```

Hier steht 12345 für den Application\_Handle und bitte beachten Sie das hier nötige Leerzeichen vor SECTION.

Bevor die EXPLAIN\_FROM\_ACTIVITY Funktion aufgerufen wird, muss zuerst das richtige SQL aus der Monitortabelle selektiert werden. „Richtig“ bedeutet in diesem Zusammenhang natürlich das Statement, das Sie analysieren wollen.

```
select APPL_ID, UOW_ID, ACTIVITY_ID, STMT_TEXT
  from ACTIVITYSTMT_ACTMON;
```

APPL_ID	UOW_ID	ACTIVITY_ID	STMT_TEXT
*N0.db2inst2.100730015738		18	1 SELECT * F

Dieses Ergebnis wird dann wiederum in die Stored Procedure eingesetzt.

```
call EXPLAIN_FROM_ACTIVITY
('*N0.db2inst2.100730016748',18,1,'ACTMON',NULL,?,?,?,?)
```

Value of output parameters

```
-----
Parameter Name : EXPLAIN_SCHEMA
Parameter Value : DB2INST2
Parameter Name : EXPLAIN_REQUESTER
Parameter Value : DB2INST2
Parameter Name : EXPLAIN_TIME
Parameter Value : 2011-09-30-04.19.25.179422
Parameter Name : SOURCE_NAME
Parameter Value : SQLC2H21
Parameter Name : SOURCE_SCHEMA
Parameter Value : NULLID
Parameter Name : SOURCE_VERSION
Parameter Value :
Return Status = 0
```

Somit wären auch in diesem Falle die Daten in den Explain-Tabellen bereitgestellt und könnten – wie oben beschrieben – per db2expln analysiert werden.

### **War's das? Nicht wirklich – es geht sogar noch mehr!**

Man kann per Section Actuals erreichen, dass das Explain mit zusätzlichen Informationen zu den tatsächlich verarbeiteten Zeilen angereichert wird, die sogenannten Laufzeitstatistiken. Die herkömmliche Explain-Ausgabe listet bisher nur die vorab geschätzte Kardinalität (Zeilenanzahl) an den jeweiligen Verarbeitungsschritten. Die Erfahrung zeigt, dass Fehlentscheidungen des Optimizers meist auf eine Fehleinschätzung der Zeilenanzahl an bestimmten Punkten zurückzuführen sind. Da per Section Explain jetzt aber der Explain im Nachhinein, also nach der eigentlichen Ausführung geschieht, kann man die echten Datenmengen erfassen und im Explain sichtbar machen.

Section Actuals können nur mittels Activity Event Monitor gesammelt werden und sind nur dann nicht möglich, wenn in der Datenbank-Konfiguration AUTO\_STATS\_PROF aktiviert ist. Allerdings ist hier zu beachten, dass bestehende Activity Monitor Tabellen neu angelegt werden müssen, falls diese vor DB2 9.7 erstellt wurden, wegen einer neuen Spalte für die Section Actuals.

Um diese Funktionalität nutzen zu können, muss noch der Datenbank-Parameter SECTION\_ACTUALS aktiviert werden. Dieser kann vom standardmäßigen NONE dynamisch auf BASE umgestellt werden. Ab der nächsten Transaktion werden dann diese Daten auf Datenbankebene erfasst.

Alternativ kann auch hier wieder die Einstellung auf Verbindungsebene erfolgen, was so weniger Overhead zufolge hat und es gezielteres Vorgehen erlaubt.

```
call WLM\_SET\_CONN\_ENV(NULL,  
'<collectactdata>WITH DETAILS, SECTION</collectactdata>  
<collectsectionactuals>BASE</collectsectionactuals>')
```

Bitte auch hier wieder das Leerzeichen vor SECTION beachten.

Da hier kein Application\_Handle angegeben wurde, werden die Einstellungen für die aktuelle Verbindung aktiv.

In der Auswertung werden die „Section Actuals“ dann unter den geschätzten Werten dargestellt:

```

Access Plan:
-----
      Total Cost:      662,213
      Query Degree:    1
      Rows
      Rows Actual
      RETURN
      (    1)
      Cost
      I/O
      |
      49547,1
      40346
      MDTQ
      (    2)
      662,213
      NA
      |
      12386,8
      NA
      TBSCAN
      (    3)
      643,007
      NA

```

Somit stehen nun noch mehr relevante Informationen zur Verfügung und diese können mit relativ wenig Zusatzaufwand eingeholt werden – also kein Grund Explain nicht umfangreicher zu nutzen wie in der Vergangenheit.

Abschließend möchte ich noch auf weiterführende Dokumentation verweisen. So finden sich alle Unterschiede vom Section Explain zum Herkömmlichen unter diesem [Link](#).

Tools die diese Verarbeitung noch einfacher machen, hat Serge Rielau in seinem [Blog](#) gepostet, den ich an dieser Stelle wärmstens empfehlen möchte.

Hier finden Sie auch die Stored Procedure PROFILE.GET\_ACTUALS, die den Aufruf nochmal vereinfacht und nur das SQL-Statement als Parameter benötigt.

So, nun viel Spaß und Erfolg bei der Analyse von SQL-Statements.

## 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.

## THINK BIG!

Können Sie sich vorstellen, das täglich weltweit 2,5 Trillionen Byte Daten erzeugt werden, das sind 18 Nullen nach der 1. Davon sind 90% des derzeitigen globalen Datenbestands in den letzten 2 Jahren entstanden. Das ist unvorstellbar. Stehen Sie auch vor der Herausforderung Unmengen an Daten zu verarbeiten und wissen nur noch nicht genau wie? Dann **Willkommen im Zeitalter von [Big Data](#)**.

***Wir laden Sie zu zwei informativen Tagen beim IBM Information Management Forum 2012 „Think Big!“ in Berlin ein. Erfahren Sie am 26. und 27. April, wie Sie mit Software-Architekturen und Tools von IBM enorme Datenmengen in Millisekunden verarbeiten können – für eine schnellere und damit effizientere Analyse Ihrer Daten und für gewinnbringende Erkenntnisse in Echtzeit.***

Hier der [Link](#) zur Agenda und Anmeldung.

## IBM DB2 optimiert für SAP

Basierend auf der "IBM DB2 optimiert für SAP Software" Initiative bietet DB2 eine Fülle von Optimierungsmöglichkeiten für SAP Kunden. Jedoch nutzen viele Kunden bisher nur einen Teil der Möglichkeiten. Von daher werden vom 21.5.-24.5.2012 vier kostenlose Workshop mit verschiedenen Schwerpunkten rund um DB2 und SAP angeboten. Hier werden neue und "alte" DB2 Technologien für SAP Software, sowie deren Einsatz und die daraus resultierenden Vorteile ausführlich dargestellt und auch die neusten Features von DB2 mit berücksichtigt.

Jeder Workshop versetzt in die Lage, den Nutzen und die technische Umsetzbarkeit der vorgestellten DB2 Technologien, für das eigene Unternehmen zu beurteilen und dessen Einsatz zu planen.

Im folgenden die Liste der kostenfreien Workshops, die jeweils im IBM SAP International Competence Center in Walldorf stattfinden.

21.5.	IBM DB2 Security Lösungen im SAP Umfeld
22.5.	DB2 Kompressionstechnologien und Performance Healthcheck für SAP Lösungen
23.5.	IBM DB2 Verfügbarkeits- und Skalierungslösungen für SAP Anwendungen
24.5.	IBM DB2 BW Themen im SAP Umfeld

Die Anmeldung kann über den beigefügten [Link](#) erfolgen.

## Data Management Meets the Future

Das Online Zeitalter nimmt Einzug bei Konferenzen



*Das rasante Datenwachstum erhöht auch den Druck auf bestehende Datenmanagement-Systeme: Sie sollen mitwachsen, noch effizienter arbeiten und die neuesten Anwendungen unterstützen – aber möglichst ohne zusätzliche Wartungskosten.*

*Höchste Zeit, dass jemand die Zukunft in die Hand nimmt – und im von immer neuen Schlagwörtern und Versprechungen geprägten Datenbankumfeld den Weg vorgibt: Mit Datenmanagement-Lösungen, die wirklich helfen, Zeit und Geld zu sparen, die Performance zu steigern und die Systeme auf die nächste Generation von Anwendungen vorzubereiten.*

***Dieser jemand heißt IBM.***

*In wenigen Tagen wird Tim Vincent, CTO IBM Information Management, gemeinsam mit dem bekannten Branchenanalyst Colin White und weiteren Vordenkern der IBM exklusiv brandaktuelle Innovationen und Trends aus den Bereichen Datenbank- und Data-Warehousing-Software bekanntgeben.*

***Sie sind herzlich eingeladen, live an Ihrem Schreibtisch dabei zu sein – bei unserer IBM Virtual Conference „The Future of Database and Data Warehousing Software“ am 4. April 2012 um 14.00 Uhr.***

***Melden Sie sich am besten [gleich](#) an!***

## DB2 Aktuell 2012

### DB2 Aktuell 2012 + Zertifizierungstag

17.-19. September  
Heidelberg



#### **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 Heidelberg.  
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 2 kostenlose technische Zertifizierungstests**

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

Das besondere Zusatzangebot:

**Der Zertifizierungstag am 17. September 2012** für nur 300 € Mehrkosten zur Vorbereitung auf die DB2 9 Fundamentals Zertifizierung , d.h Test # 730

#### **Bis 01. Juni den Frühbucher-Rabatt nutzen:**

Konferenz-Ticket DB2 Aktuell € 990,- pro Person zzgl. USt bis 01.06.

Kombi-Ticket Zertifizierungstag & Konferenz € 1.290,- pro Person zzgl. USt bis 01.06.

#### **Den 3. Teilnehmer pro Firma gibt es umsonst:**

Wenn sich 3 Mitarbeiter einer Firma anmelden, zahlen nur 2 Mitarbeiter die Teilnehmergebühr.

Eine Kombination von Rabatten ist nicht möglich. Sollten mehrere Rabattmöglichkeiten auf Ihre Buchung zutreffen, dann gilt jeweils nur die für Sie günstigste.

#### **Anmelde-Links**

Kunden: [Konferenzwebseite und Anmeldung](#)

Ansprechpartner: Jiri Stary, [stary@ch.ibm.com](mailto:stary@ch.ibm.com) Mobile 0041-79 813 34 92

## 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](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>
Michael Tiefenbacher	<a href="mailto:m.tiefenbacher@ids-system.de">m.tiefenbacher@ids-system.de</a>

### Reviewer und Ideenlieferanten:

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