

IBM

IBM i 7.1 Overview – DB2



Power your planet.

© 2010 IBM Corporation

IBM

SQL Data Description and Data Manipulation Language

- XML support
- Support for the MERGE statement
- Global variables
- Support for arrays in procedures
- Result set support in embedded SQL
- Encryption enhancements (FIELDPROC)
- Removal of identity column and constraint restrictions on partition tables
- MQ Series integration functions
- Parameter marker enhancements
- Expressions in the CALL statement
- Three-part names in statements and aliases
- Currently committed concurrent access resolution
- REPLACE option on CREATE commands
- BIT scalar functions
- Encoded vector indexes INCLUDE of aggregate functions
- Inlining of some SQL scalar function

Power your planet.

© 2010 IBM Corporation

IBM

XML Data Type

- Comparable to a Large Object (LOB) – maximum size of 2 GB – but you cannot define a maximum length
- Can be used in:
 - CAST a parameter marker, XML, or NULL to XML
 - XMLCAST a parameter marker, XML, or NULL to XML
 - IS NULL predicate
 - COUNT and COUNT_BIG aggregate functions
 - COALESCE, IFNULL, HEX, LENGTH, CONTAINS, and SCORE scalar functions
 - XML scalar functions
 - A SELECT list without DISTINCT
 - INSERT VALUES clause, UPDATE SET clause, and MERGE
 - SET and VALUES INTO
 - Procedure parameters
 - User-defined function arguments and result
 - Trigger correlation variables
 - Parameter marker values for a dynamically prepared statement

Power your planet.

© 2010 IBM Corporation

IBM

XML Data Type

- Can not be used in:
 - A SELECT list containing the DISTINCT keyword
 - A GROUP BY clause
 - An ORDER BY clause
 - A subselect of a fullselect that is not UNION ALL
 - A basic, quantified, BETWEEN, DISTINCT, IN, or LIKE predicate
 - An aggregate function with the DISTINCT keyword
 - A primary, unique, or foreign key
 - A check constraint
 - An index column
- Any EBCDIC single byte, mixed CCSID or a Unicode CCSID of 1208 (UTF-8), 1200 (UTF-16), or 13488 can be used. No support for 65535 (no conversion). CCSID can be explicitly specified when defining an XML data type. If it is not explicitly specified, the CCSID will be assigned using the value of the SQL_XML_DATA_CCSID QAQQINI file parameter (1208 default)

Power your planet.

© 2010 IBM Corporation

IBM

XML Scalar functions

- xmllagg
- xmlattributes
- xmlcomment
- xmlconcat
- xmldocument
- xmlelement
- xmforest
- xmlgroup
- xmlnamespaces
- xmlparse
- xmlpi
- xmlrow
- xmlserialize
- xmltext
- xmlvalidate
- xslttransform

Power your planet.

© 2010 IBM Corporation

IBM


XML Serialization

- Converts XML data from the format in a DB2 database to the serialized string format in an application
- Implicit serialization allows the DB2 client to handle the XML data properly
- Explicit serialization requires additional handling
- Example:


```
SELECT e.empno, e.firstname, e.lastname,
       XMLSERIALIZE(XMLELEMENT(NAME "xmp:Emp",
                               XMLNAMESPACES('http://www.xmp.com' as "xmp"),
                               XMLATTRIBUTES(e.empno as "serial"),
                               e.firstname, e.lastname
                               OPTION NULL ON NULL))
       AS CLOB(1000) CCSID 1208
       INCLUDING XMLDECLARATION) AS "Result"
FROM employees e WHERE e.empno = 'A0001'
```

Power your planet.

© 2010 IBM Corporation

IBM Power Systems 


MERGE Statement

- Updates (i.e. update, delete, insert) target table or view with matching rows from a source:
 - Good potential in a Business Intelligence data load
 - Used for archiving data

```

MERGE INTO archive ar
USING (SELECT activity, description, date, last_modified
      FROM activities_groupA) ac
ON (ar.activity = ac.activity) AND ar.group = 'A'
WHEN MATCHED AND ac.date IS NULL THEN SIGNAL SQLSTATE '70001'
SET MESSAGE_TEXT = ac.activity CONCAT ' cannot be modified.
      Reason: Date is not known'
WHEN MATCHED AND ac.date < CURRENT DATE THEN DELETE
WHEN MATCHED AND ar.last_modified < ac.last_modified THEN UPDATE SET
      (description, date, last_modified) = (
        ac.description, ac.date, DEFAULT)
WHEN NOT MATCHED AND ac.date IS NULL THEN SIGNAL SQLSTATE '70002'
SET MESSAGE_TEXT = ac.activity CONCAT ' cannot be inserted.
      Reason: Date is not known'
WHEN NOT MATCHED AND ac.date >= CURRENT DATE THEN INSERT
      (group, activity, description, date)
VALUES ('A', ac.activity, ac.description, ac.date)
ELSE IGNORE
7 Power your planet. © 2010 IBM Corporation

```

IBM Power Systems 


Global Variables

- Assigns a specific variable value for a session, but is available for all processes or sessions that access the database
- Modifications to the value are not under transaction control. The value of the global variable is preserved when a transaction ends with either a COMMIT or a ROLLBACK statement
- A global variable is created as a *SRVPGM object

```

CREATE VARIABLE USER_CLASS INT DEFAULT (CLASS_FUNC(USER))
GRANT READ ON VARIABLE USER_CLASS TO PUBLIC
SELECT
      EMPNO, LASTNAME,
      CASE
        WHEN USER_CLASS = 1 THEN SALARY
        ELSE NULL
      END
FROM EMPLOYEE
WHERE WORKDEPT = 'A00'
8 Power your planet. © 2010 IBM Corporation

```

IBM Power Systems 


Support for arrays in procedures

- Example:

```

CREATE TYPE intArray AS INTEGER ARRAY[100]
CREATE TYPE stringArray AS VARCHAR(10) ARRAY[100]
CREATE TABLE persons (id INTEGER, name VARCHAR(10))
INSERT INTO persons VALUES(2, 'Tom'),
      (4, 'Gina'),
      (1, 'Kathy'),
      (3, 'John')
CREATE PROCEDURE processPersons(OUT witha stringArray)
BEGIN
DECLARE ids intArray;
DECLARE names stringArray;
SET ids = ARRAY[5,6,7];
SET names = ARRAY['Denise', 'Randy', 'Sue'];
INSERT INTO persons(id, name)
      (SELECT t.i, t.n FROM UNNEST(ids, names) AS t(i, n));
SET witha = (SELECT ARRAY_AGG(name ORDER BY id)
      FROM persons
      WHERE name LIKE '%a%');
END
9 Power your planet. © 2010 IBM Corporation

```

IBM Power Systems 


Result set support in embedded SQL

- Example (RPG):

```

D MYRS1 S SQLTYPE (RESULT_SET_LOCATOR)
D MYRS2 S SQLTYPE (RESULT_SET_LOCATOR)
...
C/EXEC SQL CALL P1(:parm1, :parm2, ...)
C/END-EXEC
...
C/EXEC SQL DESCRIBE PROCEDURE P1 USING DESCRIPTOR :MYRS2
C/END-EXEC
...
C/EXEC SQL ASSOCIATE LOCATORS (:MYRS1, :MYRS2) WITH PROCEDURE P1
C/END-EXEC
C/EXEC SQL ALLOCATE C1 CURSOR FOR RESULT SET :MYRS1
C/END-EXEC
C/EXEC SQL ALLOCATE C2 CURSOR FOR RESULT SET :MYRS2
C/END-EXEC
...
C/EXEC SQL ALLOCATE DESCRIPTOR 'SQLDES1'
C/END-EXEC
C/EXEC SQL DESCRIBE CURSOR C1 INTO SQL DESCRIPTOR 'SQLDES1'
C/END-EXEC


```

IBM Power Systems 

FIELDPROC Support

- FIELDPROC column attribute designates an external program name as an exit routine for that column (ILE program without SQL, no support for *SRVPGM, OPM *PGMs or JAVA objects)
- Transforms values in a single column
- Assigned via the CREATE TABLE and ALTER TABLE statements
- Both field encoding and field decoding are needed in the FIELDPROC routine


11 Power your planet. © 2010 IBM Corporation

IBM Power Systems 

MQ Integration


- Scalar functions:
 - MQREAD
 - MQREADCLOB
 - MQRECEIVE
 - MQRECEIVECLOB
- Table functions:
 - MQREADALL
 - MQREADALLCLOB
 - MQRECEIVEALL
 - MQRECEIVEALLCLOB


12 Power your planet. © 2010 IBM Corporation

IBM Power Systems 

Partitioned Table Support

- Allows you to partition tables that use referential integrity or identity columns
 - Allow identity columns in a partitioned table
 - Allow a referential constraint on a partitioned table
- Restrictions
 - The identity column cannot be a partitioned key
 - The parent table must either be:
 - Non-partitioned
 - Partitioned where the unique index used for the unique constraint is non-partitioned to the referential
 - Partitioned tables with RI or an identity cannot be saved to a previous release

13  © 2010 IBM Corporation


IBM Power Systems 

Parameter Marker enhancements

- Eliminates the need to specify a CAST of a parameter marker to a specific type in an SQL statement in most cases
- Prepare a statement that uses a parameter marker in the select list:


```
SET STMT1 = 'SELECT * FROM T1 WHERE C1 > ? + ? ' ;
PREPARE PREPSTMT1 FROM STMT1;
```
- Previously, this would have to be something like:


```
SET STMT1 = 'SELECT * FROM T1 WHERE C1 =
CAST(? AS DECFLOAT(34)) + CAST(? AS DECFLOAT(34)) ' ;
PREPARE PREPSTMT1 FROM STMT1;
```

14  © 2010 IBM Corporation


IBM Power Systems 

Expression in CALL

- Eliminates the need to run a query or SET statement to evaluate the expressions before the CALL
- Example: Call a procedure and pass as arguments PARAMETER1 folded to uppercase and PARAMETER2 divided by 100:


```
CALL PROC1 ( UPPER(PARAMETER1), PARAMETER2/100 )
```


15  © 2010 IBM Corporation


IBM Power Systems 

Three Part Name Support

- Bypass the explicit CONNECT or SET CONNECTION, resulting in DRDA access to a remote relational database
- Example:



```
CREATE ALIAS shkSpr.ph1 FOR w11m.shkSpr.ph1
SELECT * FROM shkSpr.ph1
```
- Restrictions:
 - All object references in a single SQL statement must reside in a single relational database
 - The alias name must be the same as the remote name but can point to another alias on the remote


16  © 2010 IBM Corporation

IBM Power Systems 

Concurrent Access Resolution


- Reduces contention when dirty data is not allowed, but data in progress of an update is not important
- Concurrent access resolution option can be:
 - Wait for outcome** (default - wait for the commit or rollback when encountering locked data that is in the process of being updated or deleted. Locked rows that are in the process of being inserted are not skipped. This option does not apply for read-only queries running under COMMIT(*NONE) or COMMIT(*CHG))
 - Use currently committed (allows to use the currently committed version of the data for read-only queries when encountering locked data in the process of being updated or deleted. Locked rows in the process of being inserted can be skipped. This option applies where possible when running under COMMIT(*CS) and is ignored otherwise) or **readers don't block writers and writers don't block readers**
 - Skip locked data** (skip rows in the case of record lock conflicts; applies only when the query is running under COMMIT(*CS) or COMMIT(*ALL))
- Be careful when using USE CURRENTLY COMMITTED and SKIP LOCKED DATA because they might affect application functionality

17  © 2010 IBM Corporation

IBM Power Systems 

Concurrent Access Resolution

- Specifying the use for concurrent access resolution:
 - With the concurrent-access-resolution clause at the statement level for a select-statement, SELECT INTO, searched UPDATE, or searched DELETE
 - By using the CONACC keyword on the CRTSQLxxx or RUNSQLSTM commands
 - With the CONACC value in the SET OPTION statement
 - In the attribute-string of a PREPARE statement
 - Using the CREATE or ALTER statement for a FUNCTION, PROCEDURE or TRIGGER
- If the concurrent access resolution option is not directly set by the application, it takes on the value of the SQL_CONCURRENT_ACCESS_RESOLUTION option in the QAQINI query options file.

18  © 2010 IBM Corporation

IBM Power Systems IBM

CREATE OR REPLACE

- Makes it easier to either create an object without having to drop when it already exists. This can be applied to the following objects:
 - ALIAS
 - FUNCTION
 - PROCEDURE
 - SEQUENCE
 - TRIGGER
 - VARIABLE
 - VIEW
- To replace an object, the user must have both *OBJEXIST rights to the object and *EXECUTE rights for the schema or library, and of course, privileges to create the object. All existing privileges on the replaced object are preserved

19 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

BIT Scalar Functions

Function	Description	A bit in the two's complement representation of the result is:
BITAND	Performs a bitwise AND operation	1 only if the corresponding bits in both arguments are 1
BITANDNOT	Clears any bit in the first argument that is in the second argument	Zero if the corresponding bit in the second argument is 1; otherwise, the result is copied from the corresponding bit in the first argument
BITOR	Performs a bitwise OR operation	1 unless the corresponding bits in both arguments are zero
BITXOR	Performs a bitwise exclusive OR operation	1 unless the corresponding bits in both arguments are the same

20 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

BIT Scalar Functions

- Return all items for which the third property bit is set
`SELECT ITEMID FROM ITEM WHERE BITAND(PROPERTIES, 4) = 4`
- Return all items for which the fourth or the sixth property bit is set
`SELECT ITEMID FROM ITEM WHERE BITAND(PROPERTIES, 40) <> 0`
- Clear the twelfth property of the item whose ID is 3412
`UPDATE ITEM SET PROPERTIES = BITANDNOT(PROPERTIES, 2048) WHERE ITEMID = 3412`
- Set the fifth property of the item whose ID is 3412.
`UPDATE ITEM SET PROPERTIES = BITOR(PROPERTIES, 16) WHERE ITEMID = 3412`
- Toggle the eleventh property of the item whose ID is 3412
`UPDATE ITEM SET PROPERTIES = BITXOR(PROPERTIES, 1024) WHERE ITEMID = 3412`
- Switch all the bits in a 16-bit value that has only the second bit on
`VALUES BITNOT(CAST(2 AS SMALLINT))`

21 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Encoded Vector Index (EVI)

- Creating an EVI allows you to use an INCLUDE statement in the index option of the command
- Specifies an aggregate function to be included in the index and return aggregate results for a query
- The aggregate function name can be:
 - AVG_COUNT
 - COUNT_BIG
 - SUM
 - STDDEV
 - STDDEV_SAMP
 - VARIANCE
 - VARIANCE_SAMP
 - Sourced function based on one of these built-in functions
- Example:
`CREATE ENCODED VECTOR INDEX GLDSTRN.RSNKRNZ_EV11 ON GLDSTRN.HMLT (JOB_TYPE, JOB_CATEGORY) INCLUDE (AVG(WORK_TIME))`

22 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Start Journaling Library (STRJRNLIB)

- Say "bye, bye" to QDFTJRN

23 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Still on Journaling - CHGRMTJRN

- Remote Journal support:
 - Specify that the operating system has to perform a recovery on it, by identifying the number of attempts and the time, expressed in seconds, between two attempts
 - Filter out journal entries that are not absolutely needed on the target system can decrease the amount of data sent across the communication line:
 - before images
 - individual objects
 - the name of the program that deposited the journal entry on the source system
 - Filtering feature available with option 42 of the IBM i Operating System, feature 5117 (HA Journal Performance)
 - Work with Journal Attributes (WRKJRNA) command allows to monitor from the target side how many seconds the target is behind in receiving journal entries from the source system, including info on the number of retransmissions occurring for a remote journal connection

24 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Even more on Journaling

- IBM i 7.1 journaling optimization now "does it" automatically – requires some small PTFs

25 © 2010 IBM Corporation

IBM Power Systems IBM

Commitment Control and IASP

- A process, running commitment control from an IASP (i.e. it has its resources registered with commitment control on that disk pool), switching to another disk pool, will fail (CPDB8EC - *The thread has an uncommitted transaction*)
- A process, switching from the system disk pool (ASP group *NONE), does not affect commitment control. The commitment definitions stay on the system disk pool. Resources, placed under commitment control in an IASP before system disk pool resources, move the commitment definition to the IASP
- If both *SYSBAS and IASP resources are registered in the same commitment definition, the system implicitly uses a two-phase commit protocol to ensure the resources are committed atomically in the event of a system failure

26 © 2010 IBM Corporation

IBM Power Systems IBM

System Managed Access Path Protection

- New F16 function on EDT/DSPRCYAP allows to view the internal threshold, used to start protecting all access paths with estimated rebuild times greater than the internal threshold

```

Display Details
ASP . . . . . : *SYSTEM
Internal threshold . . . . . : 00:52:14
Last return:
Date . . . . . : 03/09/10
Time . . . . . : 06:54:58
Last recalibrate:
Date . . . . . : 02/24/10
Time . . . . . : 08:19:44

```

27 © 2010 IBM Corporation

IBM Power Systems IBM

Query Optimization: Cost Based Optimizer Review

- Generate Several Plans and Chose the Cheapest Cost Plan
- Cost Is Time
- Fixed Costs (Per Machine)
 - Various Access Methods (Index, Scan, Hash, Sort, etc)
 - I/O Speed (Disk)
 - Memory (RAM)
 - CPU (Speed and Parallelism)
- Variable Costs (Per Query Costs)
 - Number of Rows To Be Processed
 - Hundreds of Optimizer Questions Per Query
 - Statistics Component
 - Join Order

28 © 2010 IBM Corporation

IBM Power Systems IBM

Causes of Poor Performing Queries

- Missing Indexes and/or Statistics
- Complex predicates
- Stale Statistics
- Hidden Correlations in Data
- Data Skew
- Changing Environment
 - Data
 - System

29 © 2010 IBM Corporation

IBM Power Systems IBM

Traditional Solutions to Poor Performing Queries

- Optimizer Explore More Possible Query Plans
- Innovative Statistic Techniques
- Manually Tune Queries
- Focus on Getting the Query Plan Right the First Time

30 © 2010 IBM Corporation

IBM Power Systems IBM

What Is Adaptive Query Processing (AQP)?

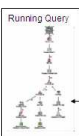
- Framework to React
 - Observe running queries
 - Stop suspicious queries
 - Reoptimize and replace plans
- Framework to Learn
 - Improve Join Orders
 - Refine Statistic Estimates
- Components
 - Current Query Inspector
 - Plan Inspector
 - Query Plan Cache
 - Global Statistics Cache (GSC)


31 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

AQP Current Query Inspector

- Wakes after two seconds
- Checks for any rows returned
 - Starvation Joins – Joins with many rows coming into the join, but few rows coming out
 - Estimated row count off from the actual row count
- Looks for
 - Initiates Reoptimization If Needed
 - Continues Polling – Checking for:
 - New Indexes
 - Manual Reoptimization
 - Rows Returned
- Minimal System Resources

Running Query


Current Query Inspector



32 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

AQP Plan Inspection

- After Query Completed
 - Checks Row Count of Key Points
 - Tops of Trees
 - Temporary Tables
 - Union Legs
- Marks for Deep Analysis if Discrepancy
 - Background Inspection
 - Checks Statistics Leading up to any Key Point Discrepancy
- Update Statistics with Actual Row Counts into GSC
- Join Order Recommendations

Global Statistics Cache
-stat 1
-stat 2
-stat 3...
-stat n



SOE Plan Cache
Query 1
Query 2
Query n

33 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Global Statistics Cache (GSC)

- System Wide Repository
 - Self Managing
 - Complex Statistics
 - Multi-Table
 - Derivations
 - All SQE Queries Leverage Information
- Observed Statistics
 - From Partially or Fully Completed Queries
 - Estimates overwritten with Actuals
 - AQP Plan Inspector may refresh with results of executing queries
- Long Running or Intensive Statistics
 - Saved for future optimizations
 - Faster Optimization Time

WHERE

$A = B \text{ and } C = D$

$X + 4 > (Y * 1.5 + 50)$

X.DATE BETWEEN
Y.DATE + 3 DAYS AND
Y.DATE + 1 MONTH + 4 DAYS

34 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Example 1: AQP Plan Inspector and Global Statistics Cache Updates

100 Million / 100 Thousand

10 Million / 10 Million

1 Million / 1 Million

Table Scan A

Table Scan B

Table Scan C

Nested Loop Join

Temporary List Probe

Temporary Sorted List

Final Select

AQP Plan Inspector

Global Statistics Cache

Join of A B C = 100 Thousand Rows

Join Multiplicative Effect of B C = 0.01 Rows

Notation: Estimate/Actual

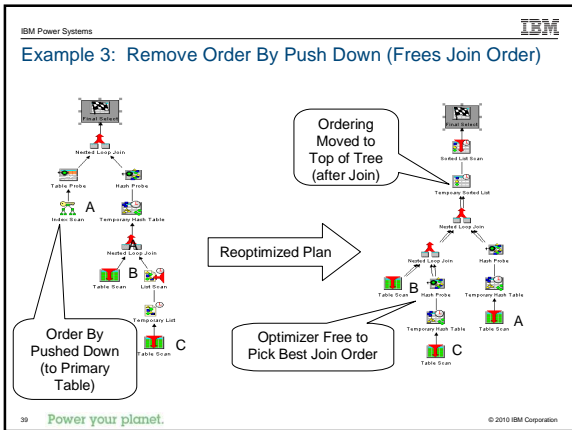
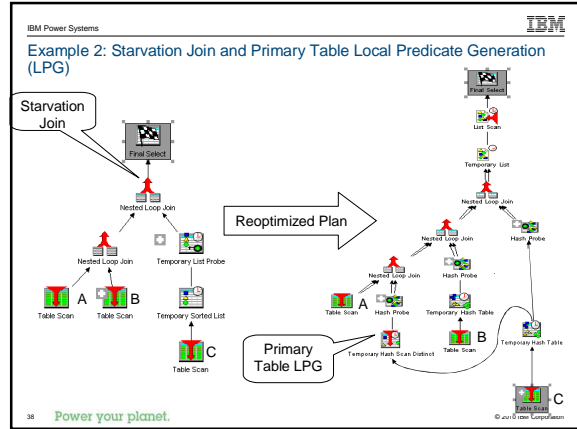
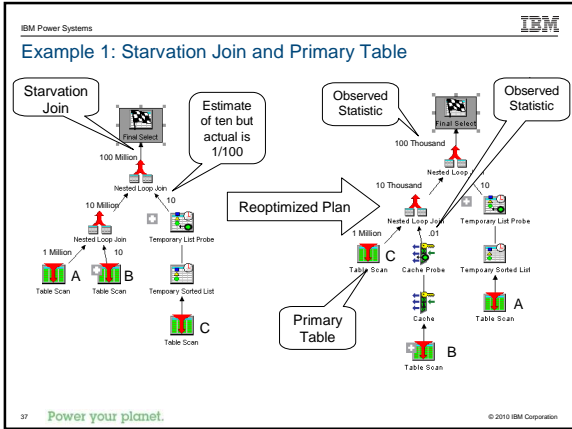
35 **Power your planet.** © 2010 IBM Corporation

IBM Power Systems IBM

Query Re-Optimizations

- In parallel to running query
- Uses any newly updated statistics
- Uses starvation join info to create new query plan:
 - Primary Table
 - Primary Table Local Predicate Generation (LPG)
 - Remove Order By Pushdown
- New Plan Guaranteed Different than Original Plan
- If Reoptimized before first row returned, then original query killed and query restarted with new plan
- Save new plan with information into plan cache

36 **Power your planet.** © 2010 IBM Corporation



- IBM Power Systems
- ### AQP - Reusable Open Data Paths (ODPs)
- Before Each New Set of Host-Variables Runs
 - Reoptimizes if new "Observed Statistics" created or updated over table(s) in query
 - Reoptimizes if new host variables estimated to select a vastly different number of rows (For example from GSC)
 - AQP Current Query Inspector monitor progress of each ODP use
 - AQP Plan Inspector verifies plan after each use of ODP
- © 2010 IBM Corporation

IBM Power Systems

AQP is Self Managing

- No Configuration or Administration Required
- Invisible to Query Users and System Administrators

Wow!

- Observed Through:
 - DataBase Monitor
 - Visual Explain

© 2010 IBM Corporation

IBM Power Systems

Visual Explain on an Active Running Query

- Refresh Visual Explain (Blue) – Draws new query tree if plan changes
- Force Reoptimization (Red) – Forces new plan if no rows yet returned

Table	Rows	Columns
Table A	1000000	10
Table B	1000000	10
Table C	1000000	10

© 2010 IBM Corporation

Summing it up ...

- DB2 on i Self Management
 - Automatic Maintained Temporary Indexes (MTIs)
 - Real time maintained index statistics
 - Automatic column statistic generation and maintenance
 - Automatic Symmetric Multi-Processing (SMP) when needed
- DB2 on i AQP Technology
 - Self Monitoring Query performance
 - Query plan mutation and improvement
 - Learning statistics engine