
We will start with a little on V8, then discuss some highlights of DB2 9 for z/OS, which became generally available March 16, 2007.

In these highlights, we'll talk briefly about the new function for pureXML and SQL. We'll note the improvements for regulatory compliance, in performance and in availability. XML work across the DB2 family is a much larger step than ever before. While V7 and V8 removed many differences from DB2 for Linux, UNIX & Windows, DB2 9 SQL takes the next big step to improved productivity and consistency. Performance improves for utilities, optimization and LOBs. Data definition on demand extends the theme of online schema evolution from V8. Additional Unicode and text enhancements continue the work from V7 and V8. Utility enhancements help with new function, more LOB and XML support, better performance and improved availability, removing the BUILD2 step from online reorg. DB2 9 enhances DB2's ability to handle new and enterprise applications. DB2 9 improves with XML, large objects, and many SQL and security improvements. DB2 9 builds upon and extends DB2 traditional strengths and the ground-breaking V8 in many areas: online schema evolution, Unicode, XML, DB2 family SQL, utilities, security and 64-bit virtual storage.

The base for most of this presentation is from Jeff Josten's IOD slides.


This is the agenda for the day, with a look at DB2 9 and beyond, then a look into how DB2 provides business value with cost reductions and increased value.

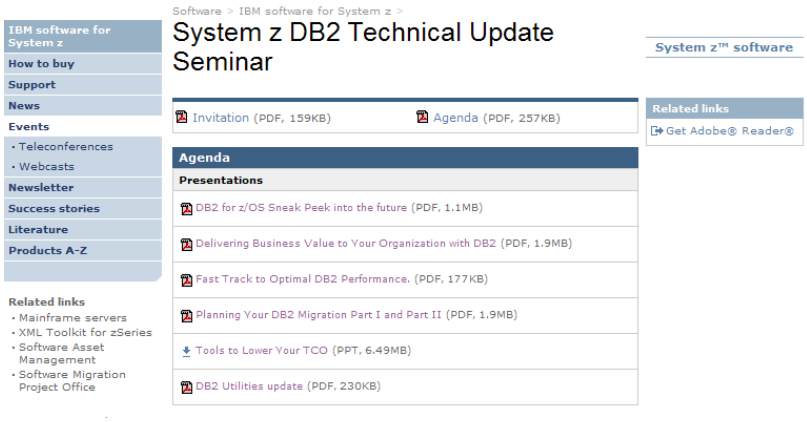
The fast track to optimal performance provides some hints and tips about improving your performance on DB2 V8 and DB2 9.

Planning your migration includes a little on DB2 V8, but is primarily about planning for DB2 9 migration.

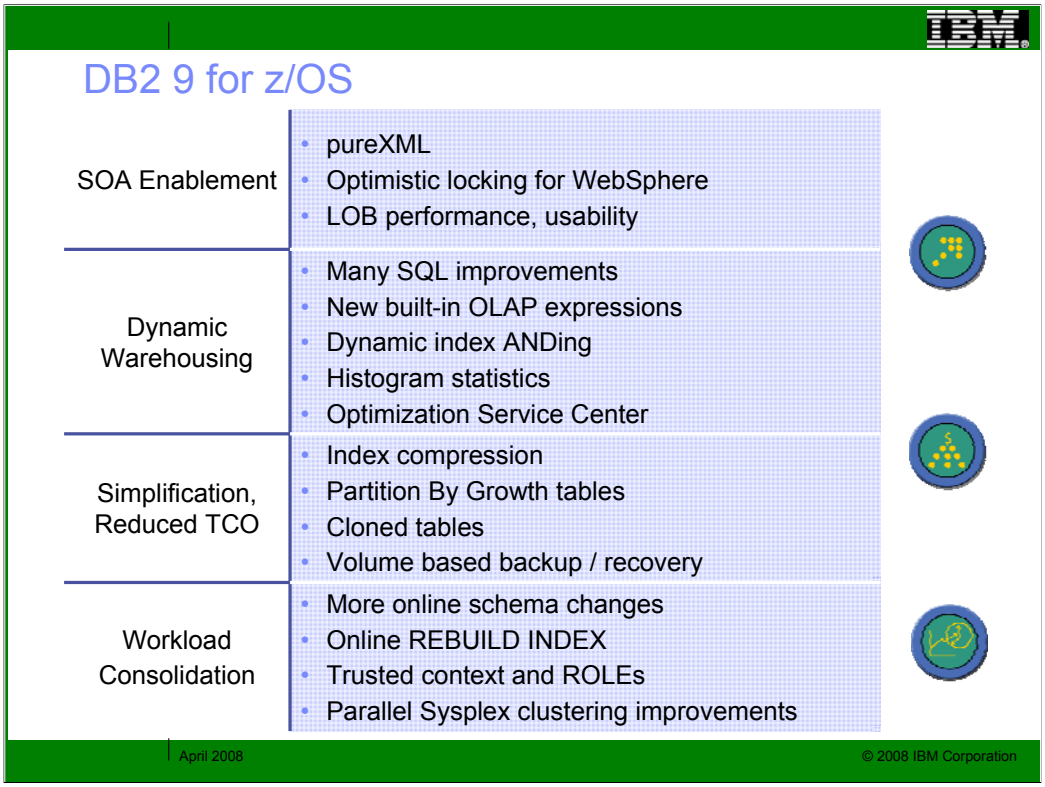
Tools to lower your TCO (and perhaps your blood pressure) is the next topic, then we'll conclude with a DB2 utility update.

We've tried to remove duplication, so that utility, performance, and migration information is primarily in those presentations.

http://www.ibm.com/software/os/systemz/seminars/db2/handouts/




Here are you online handouts. Share them with your friends and colleagues please. We do have much more information on the web, and we have a section in the migration presentation on that topic.




One of the key initiatives of V8 was online schema evolution, and that theme is expanding and changing to be data definition on demand. These are key improvements for resilience. One of the important changes is to be able to replace one table quickly with another. Another is to be able to rename a column or an index. A new type of table space combines the attributes of segmented and partitioned, without a partitioning key. Rebuild index can be run with much less disruption. Online table space reorganization for a few partitions is improved a lot, removing the BUILD2 phase for all types of secondary indexes. Table space and index logging can be altered. Changing DB2 early code does not require an IPL.

Many other improvements help with performance, with scalability and with availability. Index on an expression can be combined with caseless comparisons to improve text search. Improved insert rates can result from improved latching of the log data. Significant reductions in cpu usage are provided with new utilities.

Today's complex applications include both transactions and reporting, so performing both well is imperative. The key improvements for reporting are optimization enhancements to improve query and reporting performance and ease of use. More queries can be expressed in SQL with new SQL enhancements. Improved data is provided for the optimizer, with improved algorithms. Improved cpu and elapsed times can be achieved with the FETCH FIRST clause specified on a subquery. The INTERSECT and EXCEPT clauses make SQL easier to write.

## SQL: Productivity, DB2 family & porting

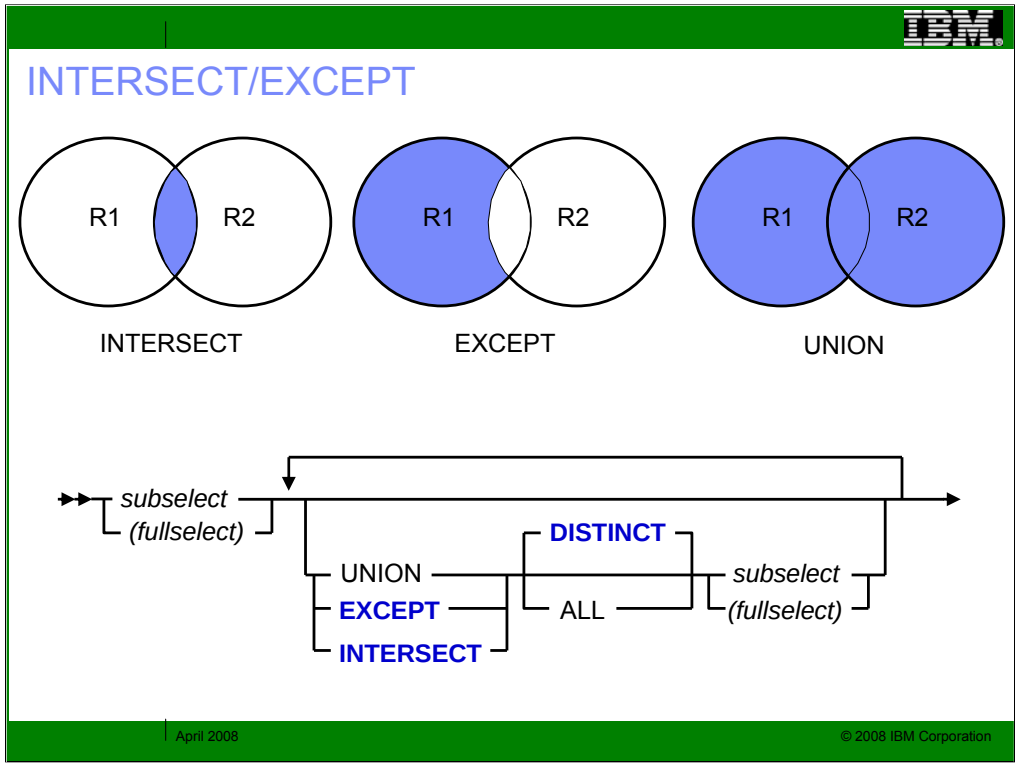

- XML
- MERGE & TRUNCATE
- SELECT FROM UPDATE, DELETE, MERGE
- INSTEAD OF TRIGGER
- BIGINT, VARBINARY, BINARY, DECIMAL FLOAT
- Native SQL Procedure Language
- Nested compound
- Optimistic locking
- LOB File reference variable & FETCH CONTINUE
- FETCH FIRST & ORDER BY in subselect and fullselect
- INTERSECT & EXCEPT
- ROLE & trusted context
- Many new built-in functions, caseless comparisons
- Index on expression
- Improved DDL consistency
- CURRENT SCHEMA

April 2008



As in Version 8, there are many improvements for SQL and for XML in DB2 9. Improvements in the SQL have made migrating from other platforms, such as Unix and Windows much easier.

DB2 9 continues the progress in SQL, with many new functions, statements and clauses. The biggest changes are in XML on the prior slide. There are new SQL data manipulation statements in MERGE and TRUNCATE. There are new data types with XML, DECIMAL FLOAT, BIGINT, BINARY and VARBINARY types. Improvements in LOBs provides more consistent handling and improved performance. Intersect and Except set operations make some SQL operations simpler to specify. Security is improved with ROLEs and network trusted context. Data definition consistency and usability are improved. DB2 9 is another big step in DB2 family consistency and in the ability to port applications to DB2 for z/OS.




We have had a UNION and a UNION ALL for a long time, but the only way to express the intersection and set difference was to code them in the WHERE predicates, rather than as a set operation. Now we have a full set of set operations.


Customers use views for read access control

Many views are not updatable, so customers have to access base tables for data changes. Triggers can be used to help control updates.

No INSERT / UPDATE / DELETE for read-only views

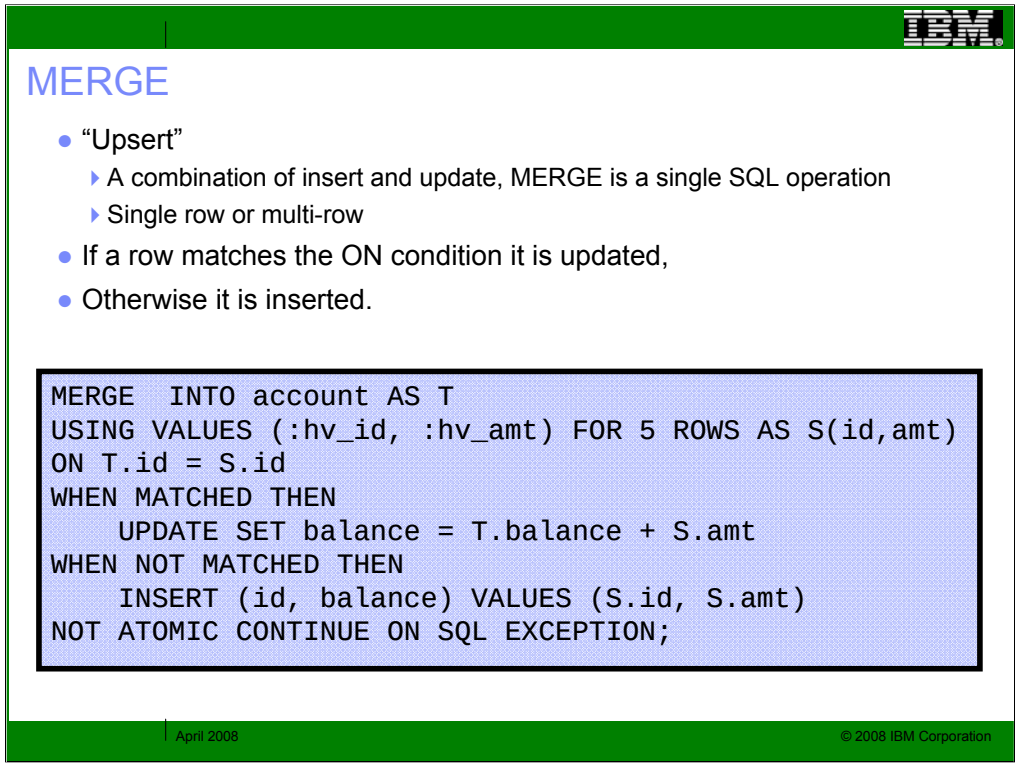
**Goal:** to provide a mechanism to unify the target for all read / write access by an application (i.e., through views) and to improve consistency with DB2 for luw

A new type of trigger (~ BEFORE, AFTER triggers)

Processed instead of the UPDATE, DELETE or INSERT statement that activated the trigger

Can only be defined on views

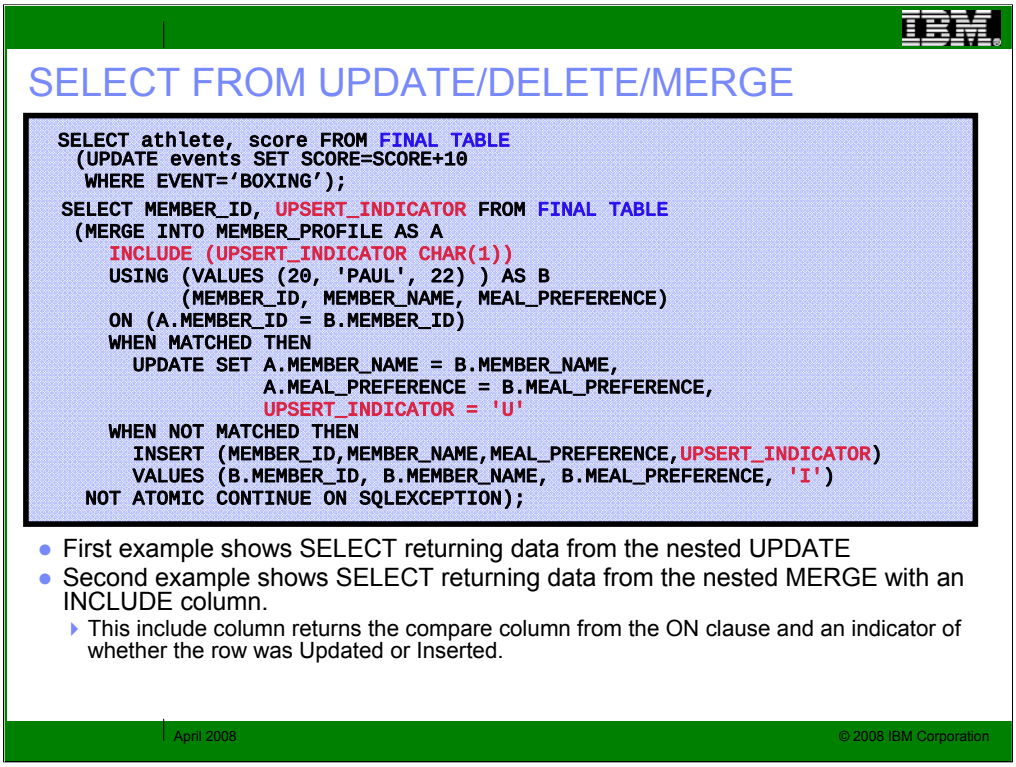
- provides an extension to the updatability of views
- requested update operation against the view gets replaced by the trigger logic
- application still believes all operations are performed against the view
- applicable even for updatable views




•MERGE – A combination UPDATE/INSERT operation. Frequently referred to an UPSERT.

•The MERGE statement updates a target (a table or view, or the underlying tables or views of a fullselect) using the specified input data. Rows in the target that match the input data are updated as specified, and rows that do not exist in the target are inserted. Updating or inserting a row in a view results in an update or insert of the row in the tables on which the view is based, if no INSTEAD OF trigger is defined for the operation on this view. If an INSTEAD OF trigger is defined, the trigger will be executed instead.

•SQL Portability and consistency




The first statement does a select from searched update. This will display an athlete and his score after it has been updated in the table.

I have decided to select the data from the FINAL TABLE in this case, but could have selected the data from OLD TABLE if I wanted to see the data before the update.

This is done in a single operation where this used to take multiple operations to produce the same result.

The second statement displays the athlete and score data that is being deleted from a table. This could be very useful for audit trail creation.

The INCLUDE column specification allows you to include additional columns, either from other tables or generated, in your nested SELECT or SELECT INTO statement

For instance, you may decide you want to include a timestamp with the data you are selecting.

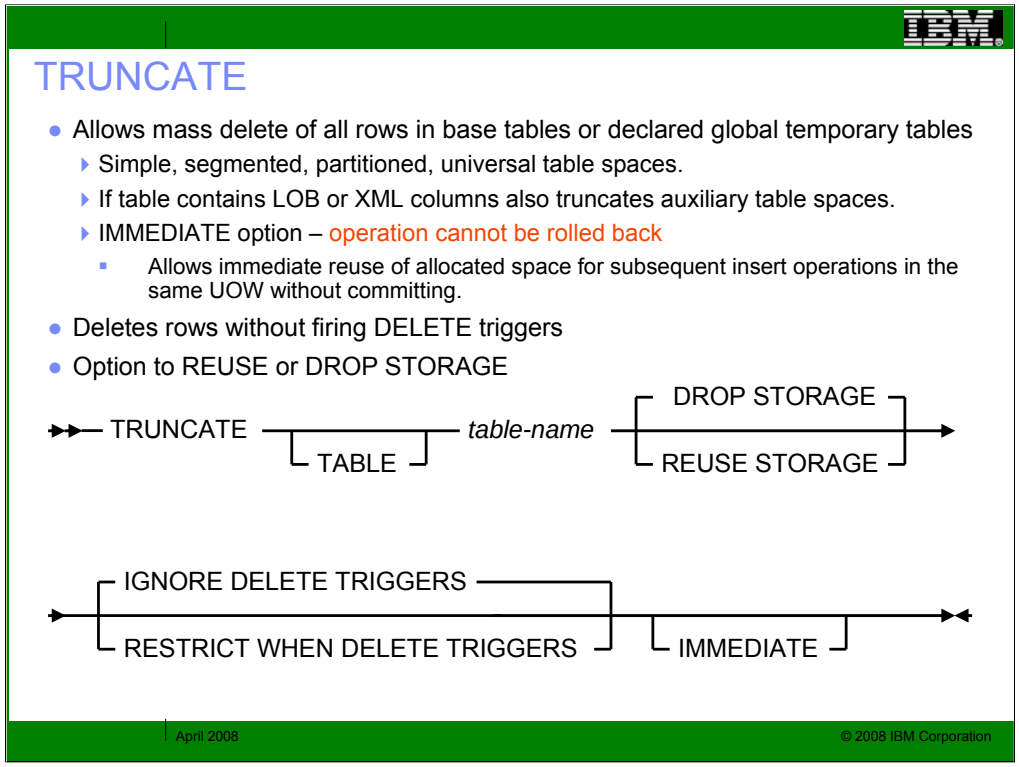
SELECT ATHLETE,EVENT,SCORE,DELETE_TS

from OLD TABLE

(DELETE from EVENTS **INCLUDE (DELETE_TS timestamp) set DELETE_ts=CURRENT TIMESTAMP** WHERE athlete='Forman')

```
ATHLETE     EVENT       SCORE       DELETE_TS
Forman      Boxing      10          2007-03-16-01.52.53.265000
  1 record(s) selected.
```

INCLUDE is used in the second example here to generate a timestamp value in conjunction with the rows being deleted.




Allows fast delete of all rows in base tables or declared global temporary tables

- Simple, segmented, partitioned, universal table spaces or tables
- If table contains LOB or XML columns the corresponding table spaces and indexes are also truncated.
- IMMEDIATE option – operation cannot be rolled back
  - Allows reusing the allocated space immediately for subsequent insert operations in the same unit of work without committing.
  - Not allowed is the table has in-flight modifications

Deletes rows without firing DELETE triggers

Option to REUSE or DROP STORAGE

- Useful for nightly refresh of summary tables, warehouses, etc.

## New Data Types: DECFLOAT and BIGINT

- DECFLOAT
  - Well suited to typical customer financial calculations
  - Similar to “calculator” mathematics
    - Eliminates rounding errors by using base 10 math
    - Has up to 34 digits of precision
  - DECFLOAT(16)
    - 10+384 to 10-383 Positive & Negative
  - DECFLOAT(32)
    - 10+6144 to 10-6143 Positive & Negative
  - Floating point convenience with fixed point precision!
- BIGINT
  - An exact numeric capable of representing 63-bit integers
  - -9223372036854775808 to 9223372036854775807

April 2008



DECFLOAT is currently supported in Java, Assembler, and REXX™ languages. The current implementation takes advantage of z9 hardware millicode

Floating-point numbers are approximations of real numbers and are considered approximate numeric types. DECFLOAT processing deals with exact numbers, not numerical

approximations of IEEE Floating Point.

New Built in functions, HDECP definitions, Rounding rules to work with DECFLOAT

## New Data Types: BINARY & VARBINARY

- BINARY fixed-length binary string
  - 1 to 255 bytes
- VARBINARY variable-length binary string
  - 1 to 32704 bytes; maximum length determined by the maximum record size associated with the table
- Compatible with BLOBs
- Not compatible with character string data types
  - Similar to FOR BIT DATA character strings
  - Can use CAST to change FOR BIT DATA character string into binary string
- Prior to DB2 9, FOR BIT DATA column use character based padding




BINARY and VARBINARY data types extend current support of binary strings (BLOB), and are compatible with BLOB data type. A binary string column is useful for storing non-character data, such as encoded or compressed data, pictures,  voice, and mixed media. Another use is to hold structured data for exploitation by distinct types, user-defined functions, and stored procedures. Note, that although binary strings and FOR BIT DATA character strings might be used for similar purposes, the two data types are not compatible.

Padded with x'00' instead of blanks as for CHAR columns. Two binary strings are equal only if the lengths are identical If two strings are equal up to the length of the shorter string length The shorter string is considered less than the longer string

To ease the migration of existing applications, altering CHAR FOR BIT DATA or VARCHAR FOR BIT DATA column data types to BINARY or VARBINARY data types is allowed (even though they are not considered to be compatible). If there is an index defined on that column, the index is placed in RBDP. Altering BINARY or VARBINARY data types to CHAR FOR BIT DATA or VARCHAR FOR BIT DATA is not allowed.

## Built-in Functions

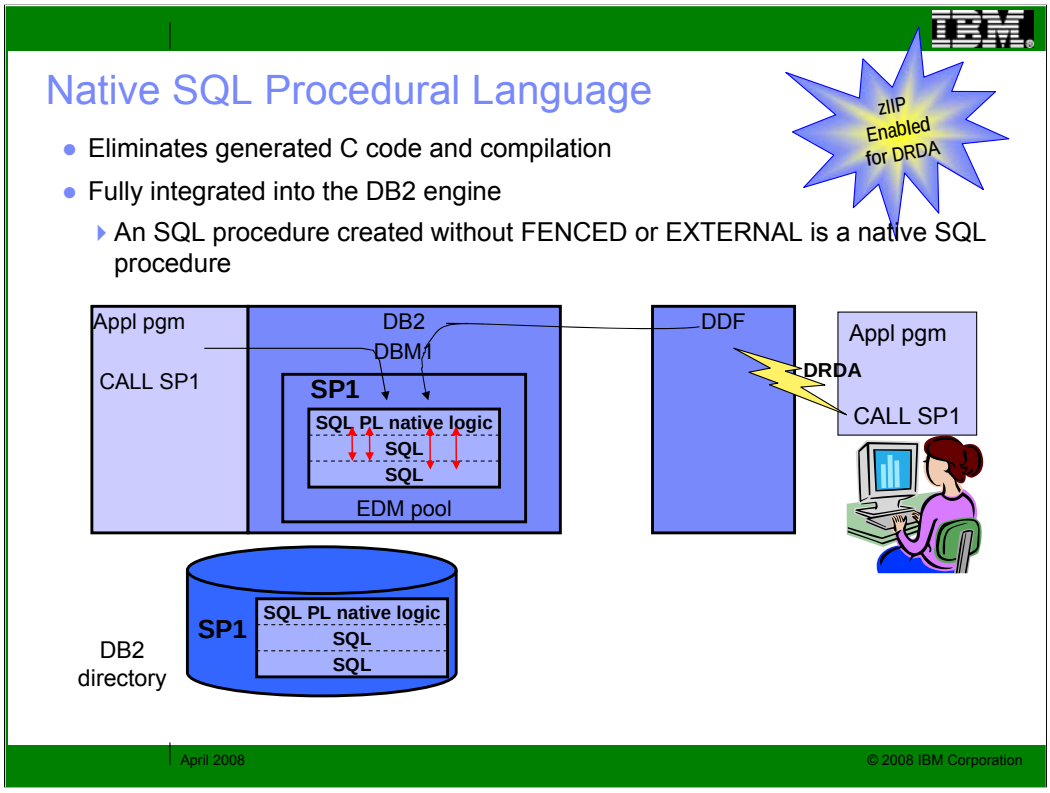

- OLAP
  - RANK
  - DENSE_RANK
  - ROW_NUMBER
- Date and Time
  - TIMESTAMPADD
  - TIMESTAMP_ISO
  - TIMESTAMP_FORMAT
  - EXTRACT
  - MONTHS_BETWEEN
- Sound representation
  - SOUNDEX
  - DIFFERENCE
- String handling
  - ASCII_STR
  - EBCDIC_STR
  - UNICODE_STR
  - ASCII_CHR
  - EBCDIC_CHR
  - COLLATION_KEY
  - RPAD, LPAD
- And More …




The EXTRACT date values function returns a portion of a date or time stamp based on its arguments.

The DIFFERENCE function returns a value from 0 to 4 that represents the difference between the sounds of two strings based on applying the SOUNDEX function to the strings.

The COLLATION_KEY function returns a varying-length binary string that represents the collation key of the string-expression in the specified collation-name, for cultural sort.




With DB2 9 new function mode, when you create a native SQL stored procedure, its procedural statements are now converted to a native representation that is stored in the DB2 catalog and directory, as it is done with other SQL statements. The parameter list and procedure options are stored in the database catalog tables as in the prior releases. When you call a native SQL procedure, DB2 loads the native representation from the catalog and the DB2 engine executes the procedure.

Extensive support for versioning: VERSION keyword on CREATE PROCEDURE
CURRENT ROUTINE VERSION special register ALTER ACTIVATE VERSION
ALTER ADD VERSION ALTER REPLACE VERSION

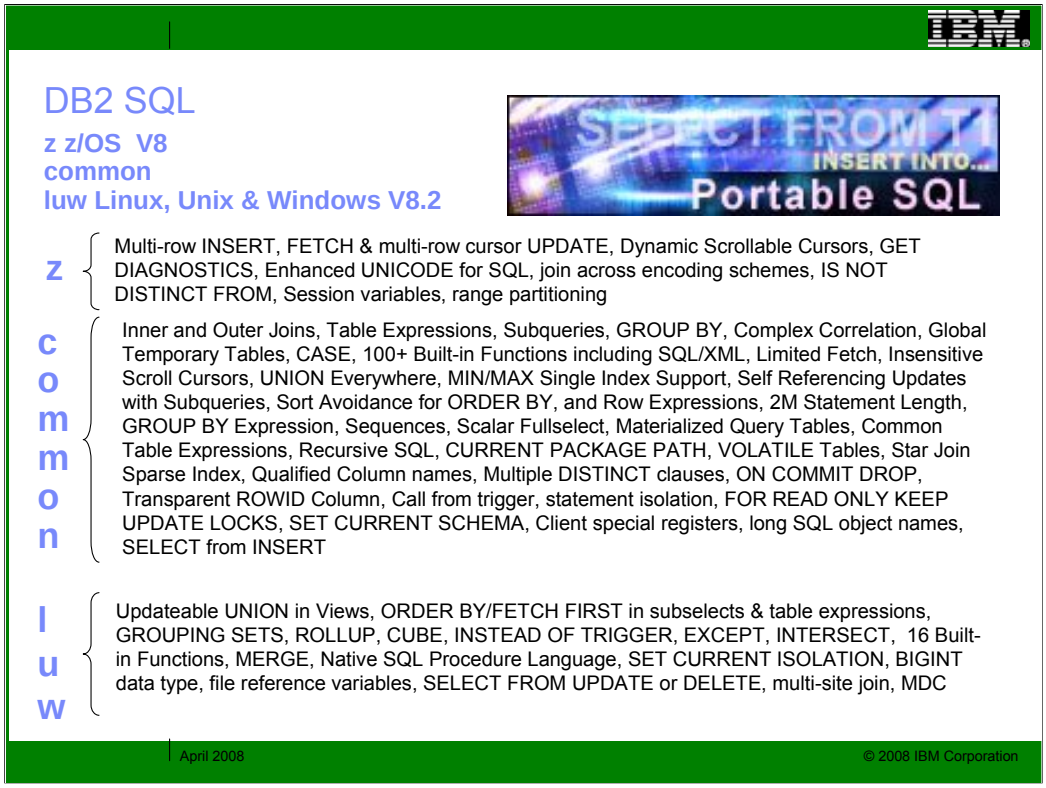
BIND PACKAGE with new DEPLOY keyword. Allow to deploy from Test to prod without a CREATE PROCEDURE statement.

**Native SQL stored procedures:** Stored procedures written in SQL procedure language enhance portability and ease of use when using DB2 for z/OS as your enterprise information source. This language is an ANSI standard language. It is similar to the proprietary stored procedure languages of several competitive databases, which assists in migrating and porting to DB2 for z/OS.

SQL stored procedures are supported by the DB2 Development Center tooling, providing an environment to code, test, and debug modules from your connected workstation. This language is currently converted to C when the CREATE PROCEDURE statement is executed. The C program is then automatically prepared, compiled, linked, and bound. The developer does not need to work with the C code.

SQL stored procedures code will be natively integrated into the DB2 engine, eliminating the conversion to C. Additionally, extensions to the bind command will allow for the promotion of the program and access paths between environments without needing to recreate the stored procedure.

•When native stored procedure requests are invoked from DRDA TCP/IP connections, the processing within the native stored procedure is eligible for zIIP specialty engine processing.




This text just shows the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS, comparing the z/OS Version 8 from March 2004 with the LUW version from October 2004.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. Sheryl Larsen provided the base for this information, but the mistakes are probably mine.

If you want to improve DB2 family consistency, then DB2 for z/OS Version 8 is a big step, changing the game from one of catch up to one of leapfrog.

DB2 SQL

z z/OS 9

common

luw Linux, Unix & Windows 9

**New cross-platform SQL book**


**z**

Multi-row INSERT, FETCH & multi-row cursor UPDATE, Dynamic Scrollable Cursors, GET DIAGNOSTICS, Enhanced UNICODE for SQL, join across encoding schemes, IS NOT DISTINCT FROM, Session variables, **TRUNCATE, DECIMAL FLOAT, VARBINARY, optimistic locking, FETCH CONTINUE, ROLE, MERGE, SELECT from MERGE**

**common**

Inner and Outer Joins, Table Expressions, Subqueries, GROUP BY, Complex Correlation, Global Temporary Tables, CASE, 100+ Built-in Functions including SQL/XML, Limited Fetch, Insensitive Scroll Cursors, UNION Everywhere, MIN/MAX Single Index Support, Self Referencing Updates with Subqueries, Sort Avoidance for ORDER BY, and Row Expressions, 2M Statement Length, GROUP BY Expression, Sequences, Scalar Fullselect, Materialized Query Tables, Common Table Expressions, Recursive SQL, CURRENT PACKAGE PATH, VOLATILE Tables, Star Join Sparse Index, Qualified Column names, Multiple DISTINCT clauses, ON COMMIT DROP, Transparent ROWID Column, Call from trigger, statement isolation, FOR READ ONLY KEEP UPDATE LOCKS, SET CURRENT SCHEMA, Client special registers, long SQL object names, SELECT from INSERT, **UPDATE or DELETE, INSTEAD OF TRIGGER, Native SQL Procedure Language, BIGINT, file reference variables, XML, FETCH FIRST & ORDER BY in subselect and fullselect, caseless comparisons, INTERSECT, EXCEPT, not logged tables**, **OmniFind, Spatial, range partitioning, compression**

**luw**

Updateable UNION in Views, GROUPING SETS, ROLLUP, CUBE, 16 Built-in Functions, SET CURRENT ISOLATION, multi-site join, MERGE, MDC, **XQuery**

April 2008



This text just shows the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 9 for z/OS. DB2 9 moves about half of the LUW unique items into the common set and adds a little more that is unique to the z platform. At about this time we'll also have a new release of DB2 9 for LUW, code named Viper. We are able to move more from the z list to the common list with Viper.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group.

## DB2 SQL

z z/OS 9
common
luw Linux, Unix & Windows **9.5**

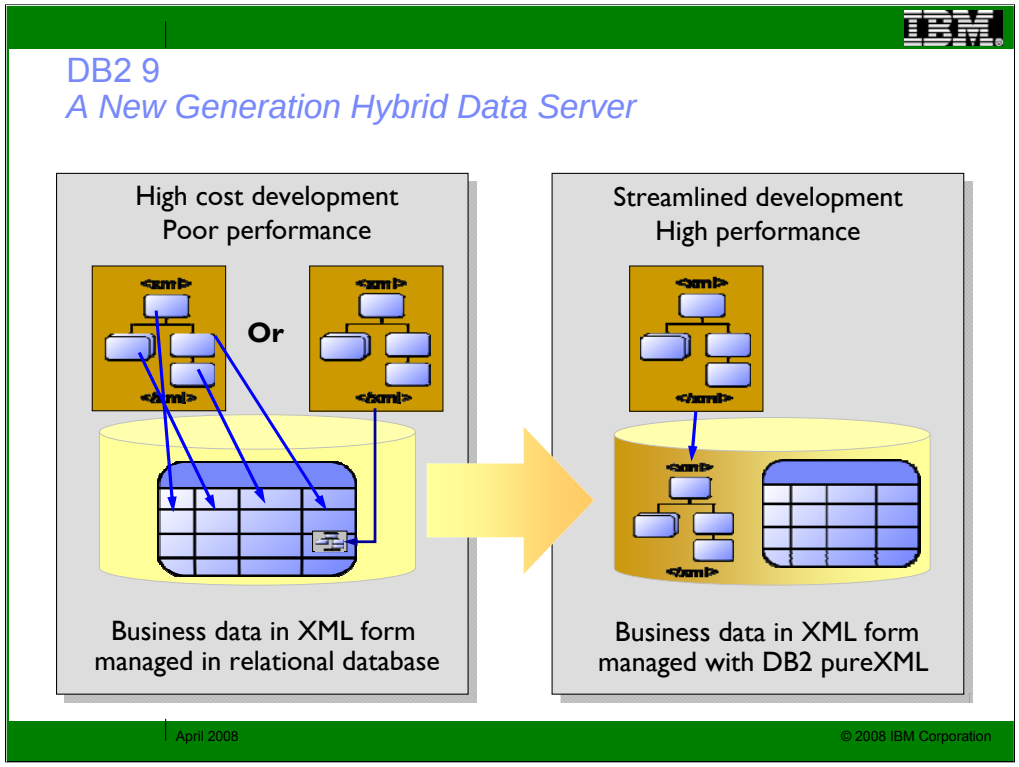

**z** — Multi-row INSERT, FETCH & multi-row cursor UPDATE, Dynamic Scrollable Cursors, GET DIAGNOSTICS, Enhanced UNICODE for SQL, join across encoding schemes, IS NOT DISTINCT FROM, TRUNCATE, VARBINARY, FETCH CONTINUE, MERGE, SELECT from MERGE, index compression

**common** — Inner and Outer Joins, Table Expressions, Subqueries, GROUP BY, Complex Correlation, Global Temporary Tables, CASE, 100+ Built-in Functions including SQL/XML, Limited Fetch, Insensitive Scroll Cursors, UNION Everywhere, MIN/MAX Single Index, Self Referencing Updates with Subqueries, Sort Avoidance for ORDER BY, and Row Expressions, 2M Statement Length, GROUP BY Expression, Sequences, Scalar Fullselect, Materialized Query Tables, Common Table Expressions, Recursive SQL, CURRENT PACKAGE PATH, VOLATILE Tables, Star Join Sparse Index, Qualified Column names, Multiple DISTINCT clauses, ON COMMIT DROP, Transparent ROWID Column, Call from trigger, statement isolation, FOR READ ONLY KEEP UPDATE LOCKS, SET CURRENT SCHEMA, Client special registers, long SQL object names, SELECT from INSERT, UPDATE or DELETE, INSTEAD OF TRIGGER, Native SQL Procedure Language, BIGINT, file reference variables, XML, FETCH FIRST & ORDER BY in subselect & fullselect, caseless comparisons, INTERSECT, EXCEPT, not logged tables, OmniFind, spatial, range partitions, data compression, **session variables, DECIMAL FLOAT, optimistic locking, ROLE**

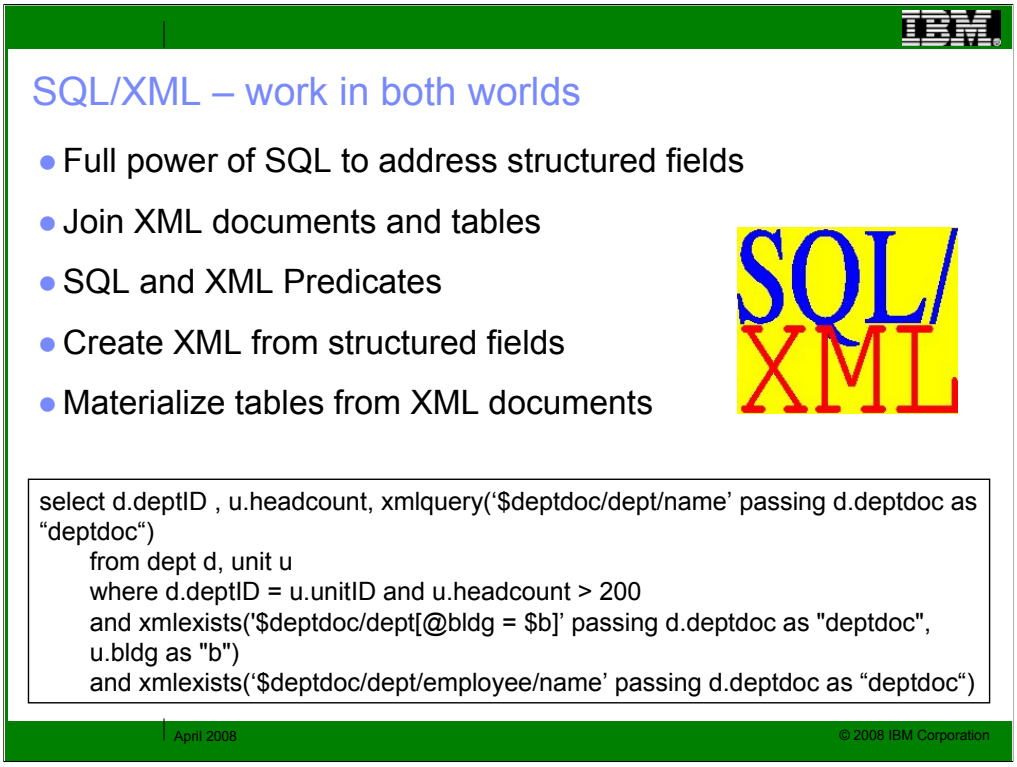
**luw** — Updateable UNION in Views, GROUPING SETS, ROLLUP, CUBE, **more** Built-in Functions, SET CURRENT ISOLATION, multi-site join, MERGE, MDC, XQuery, **XML enhancements, array data type, global variables, Oracle syntax**

April 2008 

This text just shows the newest delivered relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 9 for z/OS, (DB2 9). DB2 9 moves about half of the LUW unique items into the common set and adds a little more that is unique to the z platform. DB2 9.5 for LUW, code named Viper 2 is already generally available. We are able to move more from the unique z list to the common list with DB2 9 for luw.
There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. The changes in a specific version are not consistent. As we introduce new function, sometimes it will be on one platform first, but movement from unique lists into the common list continues to be the strongest trend.



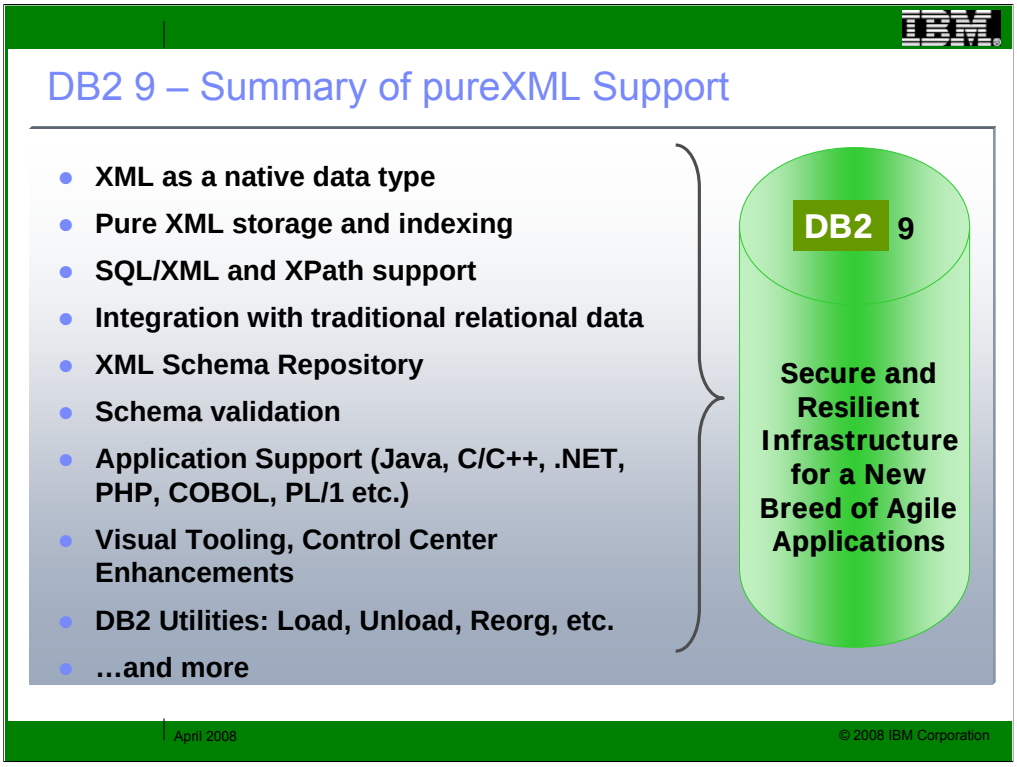

The amount of business information in XML form is already as great or greater than other forms and growing faster - failure to leverage efficiently as structured data means high cost and/or missed opportunity. DB2 9 provides the best of both worlds, pureXML[tm] for native storage and integrating XML with object-relational. Performance, integrity, protection, and scale from the proven DB2 infrastructure with the flexibility of XML/XPath and relational/SQL. This overcomes the complexity & limitations of prior models (shred, CLOB, or XML only).
The explosive growth of XML based data standards in all industries means competitive advantage for those businesses that use it most effectively and efficiently. Client, policy and claims processing in Insurance; supply chain management in Retail; financial transactions and asset management in Banking; patient care in Healthcare; citizen service in Government; implementing Service Oriented Architectures (SOA) in Computing Software and Services - and many other processes across all industries - increasingly rely on information captured and exchanged in XML form. Our clients are increasingly managing XML format text documents in a content management system for proper governance and efficient use in the business process workflow. But few are realizing the full value of all the business data they possess that are in XML format.
Early users of the pureXML feature of DB2 9 are taking advantage of the fact that data in XML format is well structured and can be queried via standard languages such as XPath and XQuery. By doing so they are bringing that data to bear in both transactional and analytic processes - with higher performance and lower development costs than previously possible with a relational database. The difference is that DB2 9 supports both relational (tabular) and XML (hierarchical) structures in the same database so that both can be easily, efficiently and securely managed, analyzed and delivered. Unlike other relational data servers - and previous versions of DB2 - pureXML eliminates the overhead of fitting the “square peg” XML tree structure into the “round hole” row and column relational structure.



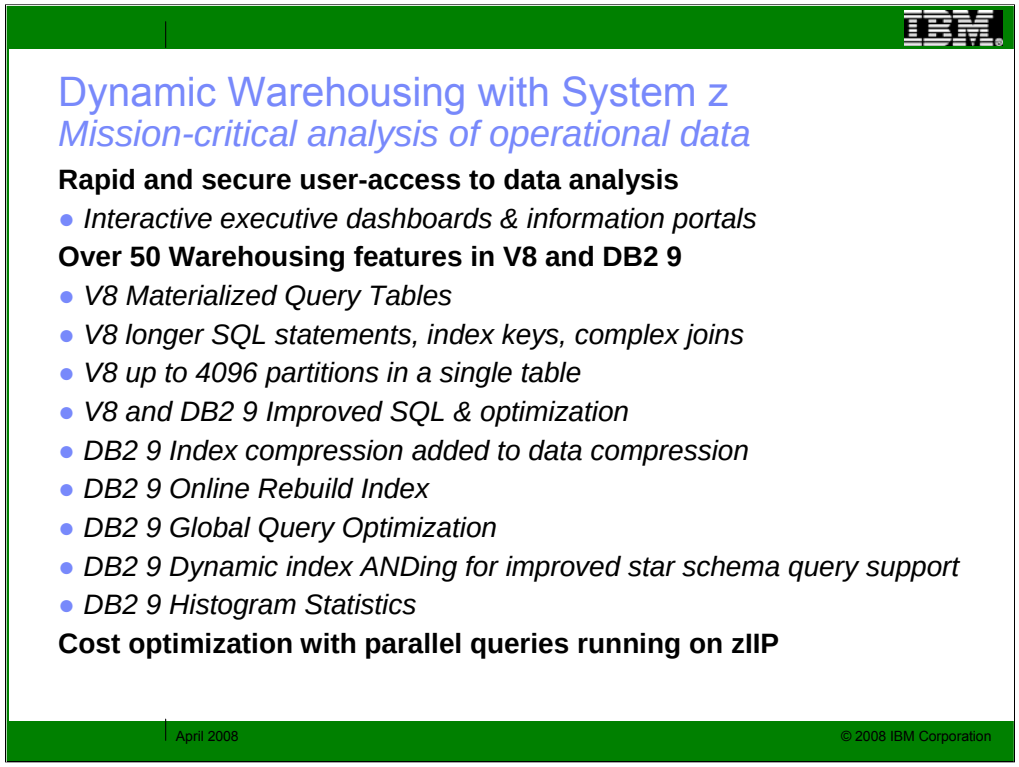

Now let's discuss the wide range of new function provided inside the DB2 for z/OS engine. This work is being done in parallel with similar changes in DB2 for Linux, Unix and Windows.

DB2 for z/OS extensibility was initially implemented via *extenders*. The XML extender provides for the storage and indexing of an XML document as a character large object (CLOB), or for the shredding of the XML document into relational columns for storage and query. DB2 Version 8 expanded on XML support by implementing several XML publishing operations as built-in DB2 functions. This allows you to perform XML document composition from relational data with improved performance without the XML Extender.

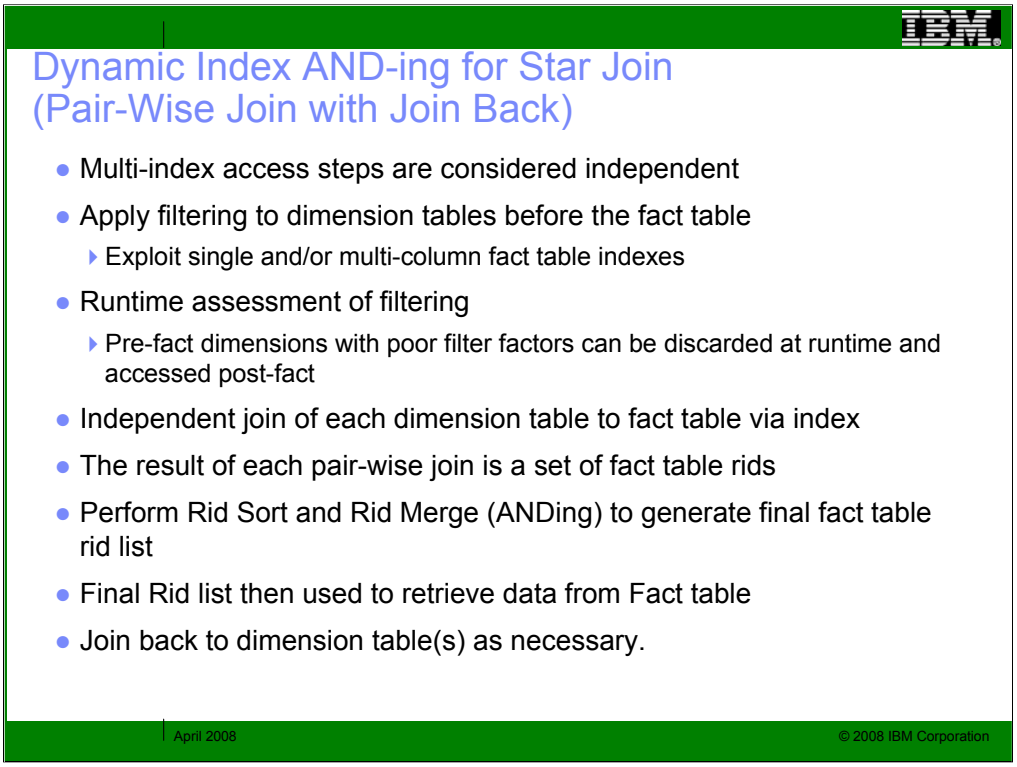
XML in DB2 9 integrates much more function into the engine. This includes an XML data type, native storage of XML documents, integration of the XPath language, and catalog extensions to support definitions of XML schemas. Utilities support creation and maintenance of XML data.




Until DB2 9, managing XML data records with a relational data servers meant decomposing the data into columns - a process known as shredding. Or by storing the entire data record in a single cell as a character large object - known as a CLOB. The CLOB approach does not cost overhead as the data records go in. But when you query these records you pay the overhead of parsing each one at runtime which can be a significant performance impact to the application. With shredding, overhead is paid up front to turn the data into a relational record that can be queried efficiently. But overhead is also paid later if the record needs to be recreated for delivery in XML format. This process also affects the fidelity of the record itself - leading to an approach that uses both shredding and CLOB methods for applications that require both performance and fidelity. This results in even more overhead to ensure the records remain in sync.
The impact of pureXML is seen by a large Banking client with a requirement to update over 500,000 XML data records per day. Attempts to use a competitors relational data server failed. Using DB2 9 with pureXML, the application was able to update more than half a million data records in less than an hour.
And a large Insurance client has seen the impact of pureXML to development time and cost with a 65% reduction in lines of code and more than 75% reduction in time required to develop services accessing XML data.



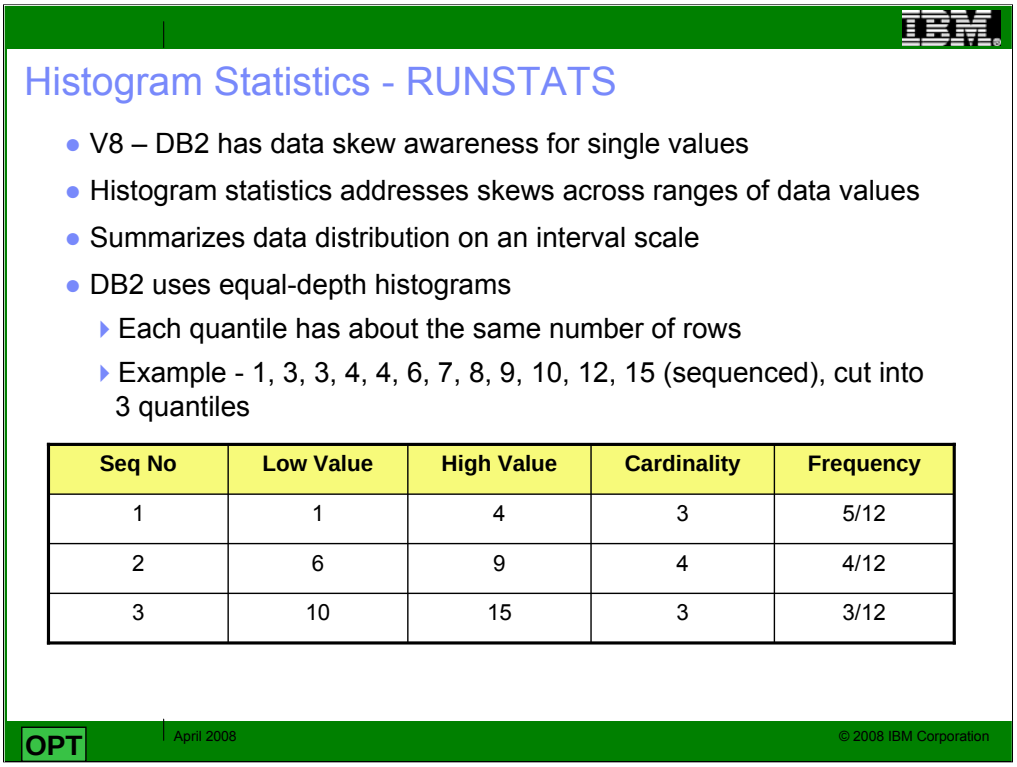

DB2 9 provides many improvements for Data Warehouse. Today's complex applications include both transactions and reporting, so performing both well is imperative. The key improvements for reporting are optimization enhancements to improve query and reporting performance and ease of use. More queries can be expressed in SQL with new SQL enhancements. Accompanying DB2 9 is a new product, IBM DataQuant, which has the critical mass of features that allow it to be positioned as a viable BI and data analytics offering with support for both graphical reports and interactive visual dashboards. It provides a sophisticated graphics engine, supporting dozens of charts and layouts and over 100 built in functions. DataQuant provides very granular security- limit information on a per user/group, also tailor look and feel for users. It also provides a rich client or web-based development/runtime environment. IBM DataQuant is deally suited to the rapid fulfillment of 'everyday' dashboard and reporting requirements. It is simple to develop and deploy – quick turnaround at low development cost. It also allows IT groups / analysts to quickly respond to custom requirements.
Where IBM DataQuant fits: Where there's a need to distribute data using straightforward graphical reports and information dashboards Where quick prototyping and rapid development is more important than complex analytical features In QMF and/or z-based environments where tracking, governing and z-based deployment are valued factors For customers that find competitor solutions too complex and costly
Index compression or also deep compression provide a significant reduction in storage. The index compression relies upon page level compression instead of row-level compression (technique used for tables). Indexes with 32K page sizes can save up to 8x on disk space with the compression feature turned on. Improved index compression with minimal overhead resulting in Beta customers reporting 50% or more savings in disk space
The key performance improvements in DB2 9 are reduced cpu time in the utilities, improved LOB performance and scalability, improved optimization for SQL, the zIIP processing for remote native SQL procedures, reduced cpu time for data with varying length and better sequential access.
Significant CPU time reduction in most utilities:
5% - 30% in Load, Reorg, Rebuild
35% in Load Partition
10% - 20% in Copy, Recover Table Space
20% - 60% in Check Index
40% - 50% in Reorg Index
Additional 10% to 15% improvement in virtual storage. Remote Native SQL stored procedures can take advantage of zIIP. Improved data is provided for the optimizer, with improved algorithms. New optimizer techniques and more SQL functions enhance DB2 for z/OS as a DWH platform. New OmniFind text search functions provide efficient communication interactions with DB2 for z/OS. OmniFind text indexes are persisted into DB2 tables for backup/recovery purposes.
1 - Winter Corporation's "2005 Top Ten" awards - http://www.wintercorp.com/index.html




Better performance for star join queries.

Easier to design indexes – multi-col indices no longer needed.

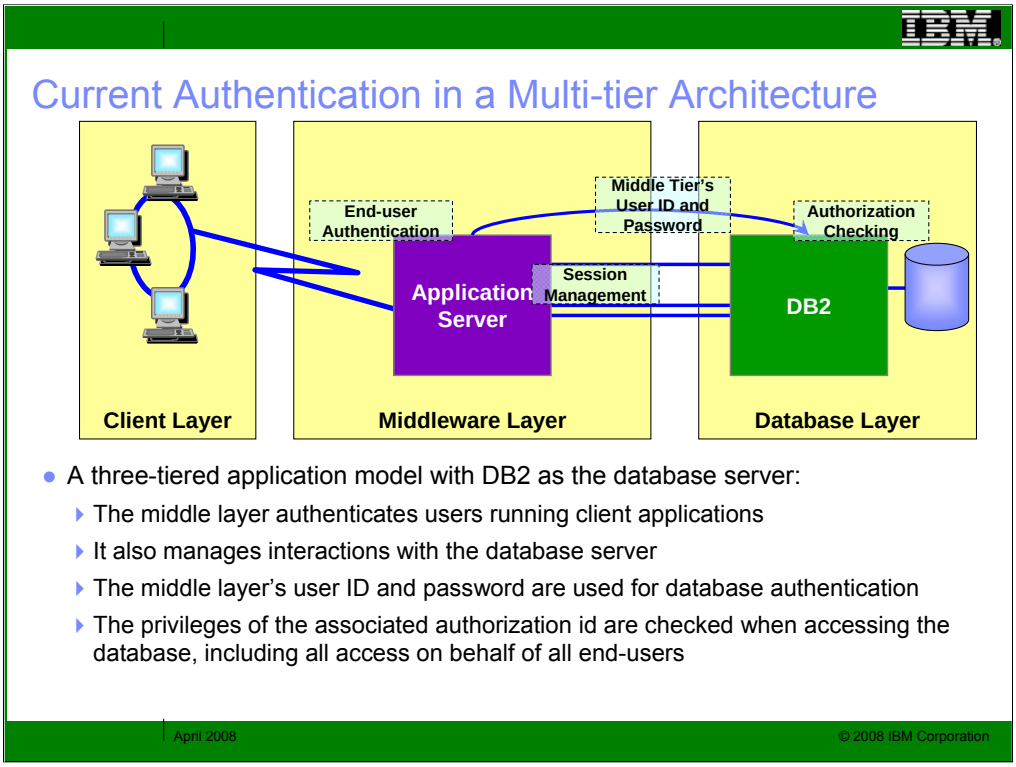
More aggressive use of parallelism, more zIIP eligible.




RUNSTATS

- Maximum 100 quantiles for a column
- Same value columns WILL be in the same quantile
- Quantiles will be similar size but:
  - Will try and avoid big gaps between quantiles
  - A column high value and low value may have separate quantiles
  - Null WILL have a separate quantile
- If less than 100 column values, reverts to Distribution Stats
- Not supported with LOAD and REORG

Supports column groups as well as single columns

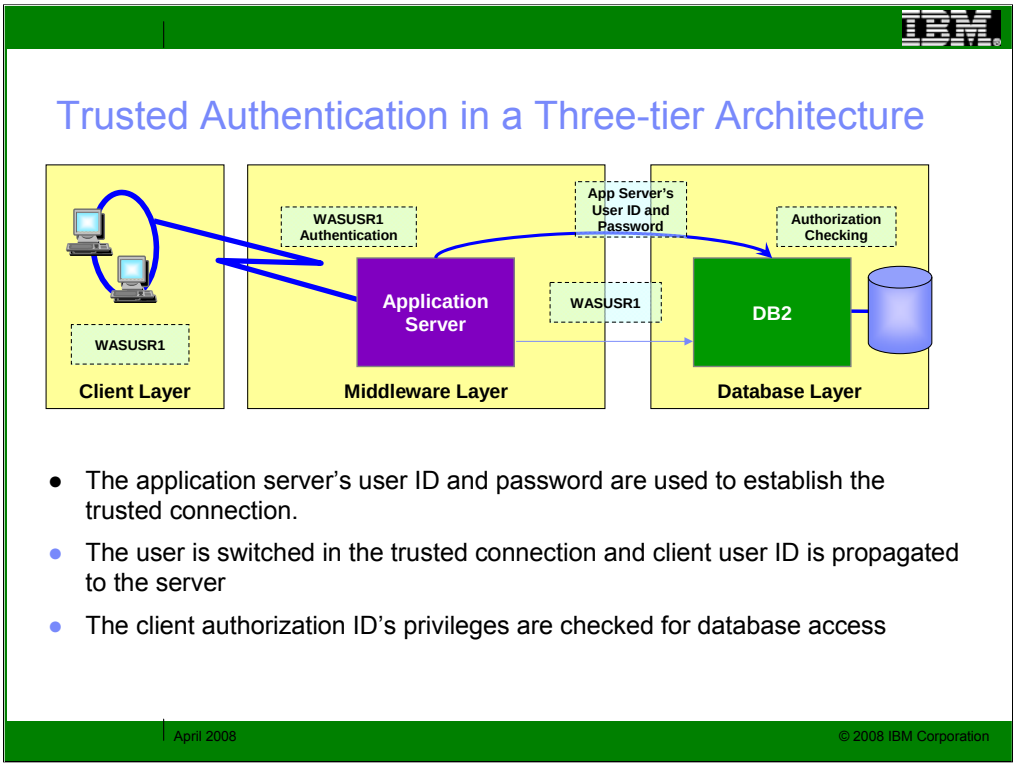



In a three-tier architecture model all interactions with the database server occur under middle-tier authorization ID. This results in loss of end-user identity, over granting of privileges to the middle-tier authorization ID, and weakened security. Hence, need better access control from application servers with end to end auditing of users.

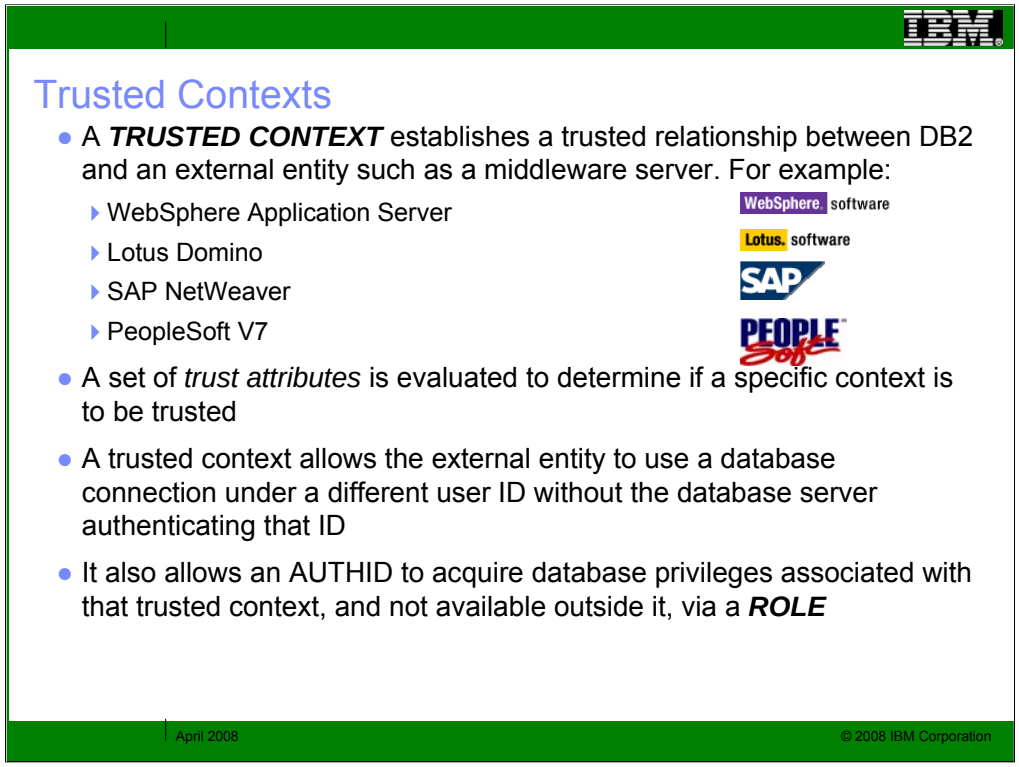
When a user creates an object, the user becomes the owner. So, when the user leaves the company, in order to remove privileges from the user the object has to be dropped and recreated.

In some customer shops, a generic TSO id is created and granted SYSADM authority. This is done in order to avoid the cascading effect of revoking SYSADM when individual team members leave the organization. Unfortunately the use of the generic TSO user id does not provide the individual accountability that is now required for Sarbanes-Oxley (SOX) compliance. A function similar to shared DBADM/SYSADM is needed with auditing capability.

Also, DBADM can create view/alias for another ID, but has no DROP/ALTER privilege




Trusted context addresses the problem of establishing a trusted relationship between DB2 and an external entity, such as a middleware server.
A series of trust attributes are evaluated at connect time to determine if a specific context is to be trusted.
The relationship between a connection and a trusted context is established when a connection to the server is first created.
Once established, a trusted connection provides the ability to:
-Use the trusted connection for a different user without authentication.
-Acquire special set of privileges by an authorization ID, that are not available to it outside the trusted context. This is accomplished by associating a role with the trusted context.
-Allow a role to own objects, if objects are created in a trusted context with role defined as the owner.
-Acquire security label (RACF SECLABEL) to be used for multi-level security verification. Multi-level security restricts access to an object or a row based on the security label of the object or row and the security label of the user.




A ***TRUSTED CONTEXT*** establishes a trusted relationship between DB2 and an external entity such as a middleware server. For example:

WebSphere Application Server

Lotus Domino

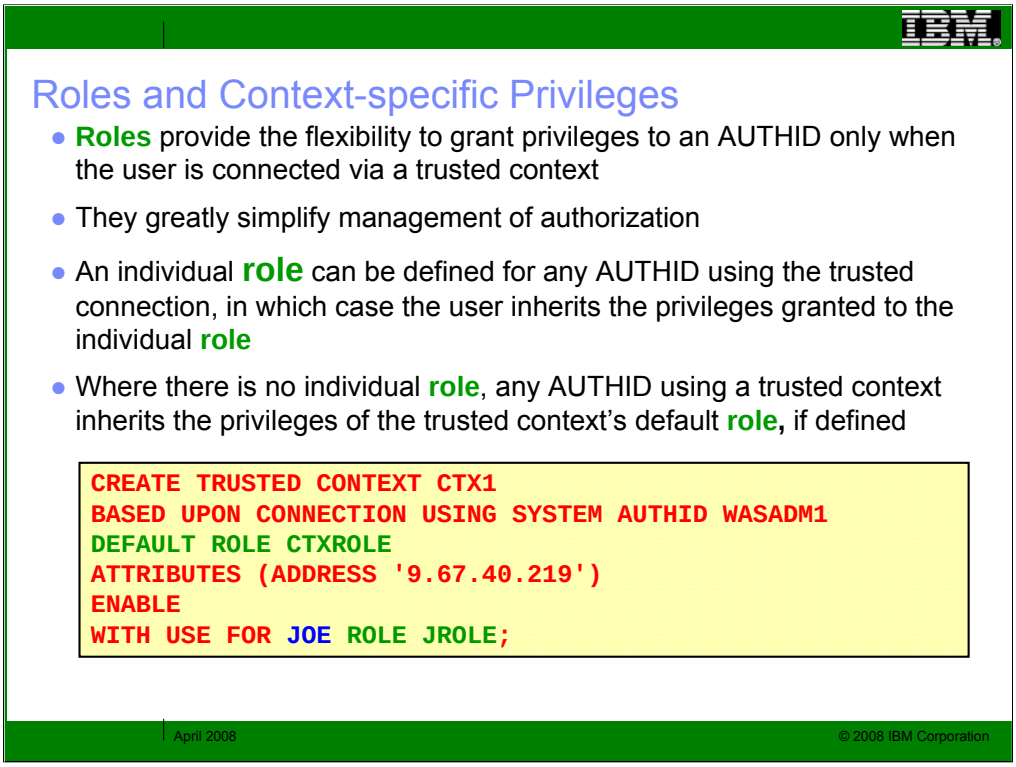
SAP NetWeaver

PeopleSoft V7

A set of *trust attributes* is evaluated to determine if a specific context is to be trusted

A trusted context allows the external entity to use a database connection under a different user ID without the database server authenticating that ID

It also allows an AUTHID to acquire database privileges associated with that trusted context, and not available outside it, via a ***ROLE.***



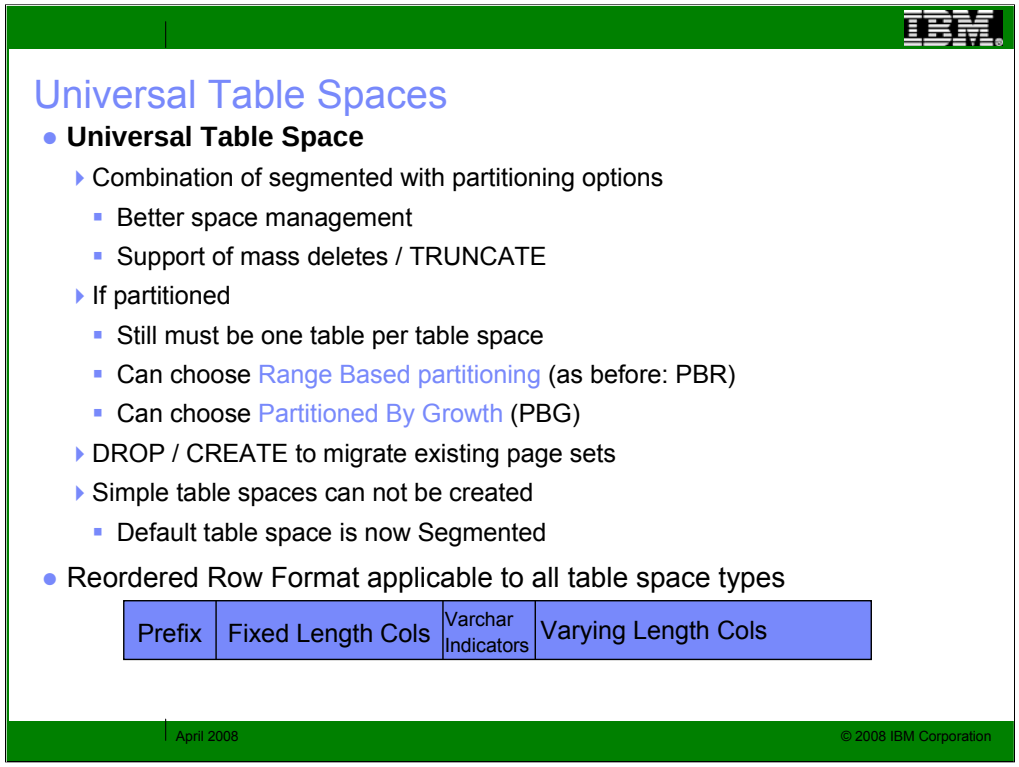

Trusted context allows for the assignment of a default role to a trusted context and assignment of a role to the user of the context. Same role can be assigned in different trusted contexts.

In current DB2 security model, privileges assigned to an id is universally available. By assigning privileges to a role (for ex: SELECT) in a trusted context and allowing access to the user only when connected from certain location provides better control and flexibility on where and when and how DB2 privileges can be exercised

Roles can be assigned and removed from individuals via trusted context. This allows DBA to perform object maintenance during a change control window and then lose the role privilege when the window is shut. This is similar to shared SYSADM or DBADM user IDs, but avoids the audit compliance problems associated with shared user IDs.

Auditing trails of the work completed during the maintenance window

Drop role: Role should not own objects or part of trusted context definition or associated with the current thread.

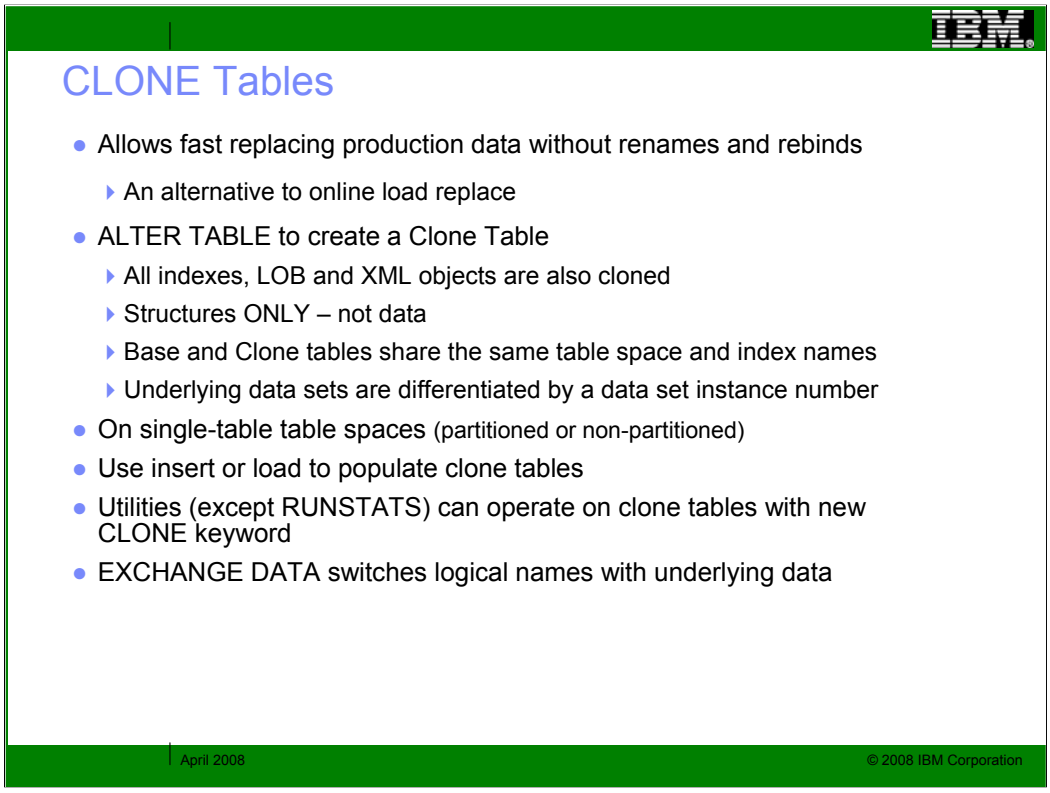



We need a partitioning table space with some of the segmented table space features.
The advantages of segmented space maps for partitioned table spaces ?
Universal Table space (Partition By Range) A partitioned segmented table space.
Partitioning column required. One table per table space.
Universal Table space (Partition By Growth) A partitioned segmented table space.
No partitioning column required. One table per table space.

**Partition By Growth (PBG)**

Single-table table space, where each partition contains a segmented page set (allows segmented to increase from 64GB to 16TB or 128 TB with 32K pages)
Eliminates need to define partitioning key and assign key ranges
Partitions are added on demand. A new partition is created when a given partition reaches DSSIZE. DSSIZE defaults to 64G. Up to MAXPARTITIONS
Retains benefits of Utilities and SQL parallelism optimizations for partitioned tables
SEGSIZE defaults to 4 & LOCKSIZE defaults to ROW
Considerations: Single-table table space Always defines as LARGE
Need PBR for query partition elimination
No LOAD PART, ALTER ADD PART, or ROTATE PART All indexes are NPSIs
Automatic repositioning of Variable columns to end of row
Length attributes replaced with Indicators positioned after fixed length columns

Any table space created in DB2 9 NFM
To Convert: REORG or LOAD REPLACE a table space or partition ADD PARTITION
No EDITPROCs or VALIDPROCs
PIT RECOVER will set the table space to the row format of the PIT
Catalog / Directory remains in Basic Row Format (BRF)




## Fast replacement of a table with another

This function will allow you to generate a copy of a current table with the same attributes and same data, in the same table space. It appears to an application or user as a very fast replacement of data within a table. Web-based applications, striving for maximum availability will benefit from the option of implementing a pair of tables that are clones of each others' structure. Copies for application testing and auditing can be easily created. These *clone* tables will have the unique ability to change names quickly. Applications can quickly and almost transparently switch between dual mirror tables.

On single-table table spaces (partitioned or non-partitioned)

Use insert or load to populate clone tables

Utilities (except RUNSTATS) can operate on clone tables with a new CLONE keyword

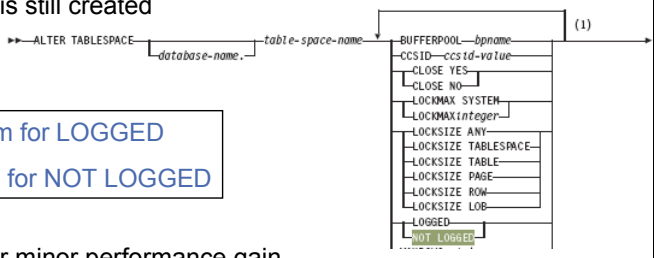
Use EXCHANGE to switch logical names with underlying data

FLIGHTS now references the data that was in FLIGHTS_CLONE

Indicated in SYSTABLESPACE .CLONE

## NOT LOGGED Tables

- Is actually NOT LOGGED tables, indexes, LOB, XML
- ALTER / CREATE a TABLESPACE as NOT LOGGED
  - ALTER not allowed if in same UOW with an update to the table space
- Indexes, LOB, and XML inherent the logging attribute of the base
  - This are considered "*Linked*" objects
- Effects the UNDO / REDO records
  - Control information is still logged
- LOB continue to log system pages & auxiliary indexes
- A Unit of Recovery (UR) is still created




LOG YES is a synonym for LOGGED

LOG NO is a synonym for NOT LOGGED

Recommendation:
Don't sacrifice recovery for minor performance gain.




Cannot explicitly specify for XML & Index objects
LOBs can be set independent of the base table. However, if a LOB is LOGGED, the base must also be logged. This "*dissolves the link*" with the base. Not compatible with CHANGE DATA CAPTURE attribute Applies to any tables in the table space Not allowed for DSNDB06 or Work File database
SYSCOPY activity

ALTER LOGGED to NOT LOGGED creates a recoverable point
ALTER NOT LOGGED TO LOGGED marks the object COPYP for the base table space
Frequent ALTERing may require more SYSCOPY space

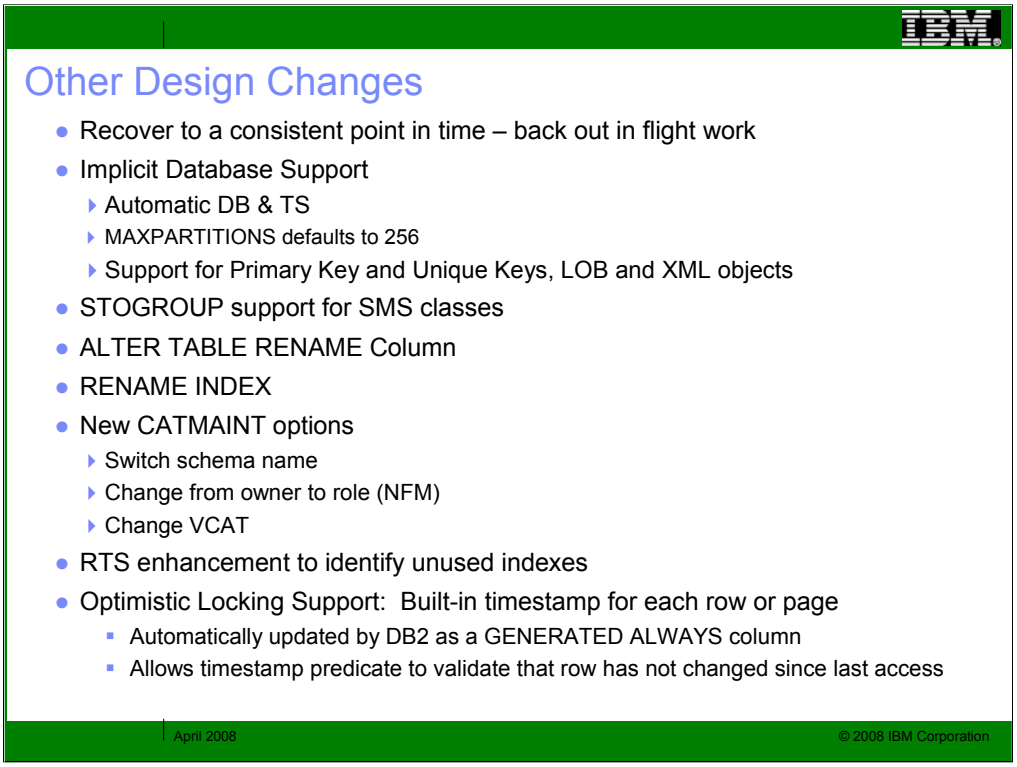
A FULL COPY should be taken

Just before ALTERing to NOT LOGGED
Just after ALTERing to LOGGED

If changes are made while NOT LOGGED

The space is marked ICOPY
An ALTER to LOGGED will set COPYP

Image copies can be SHRLEVEL NONE or REFERENCE Full or incremental
Be careful with any ROLLBACK or CANCEL command that impacts a thread acting on NOT LOGGED objects:
CANCEL, ROLLBACK, LOAD RESUME failures, and Restart Can cause the object (and XML space) to end up in a RECP state and in the LPL. Indexes often end up in RBDP & in the LPL

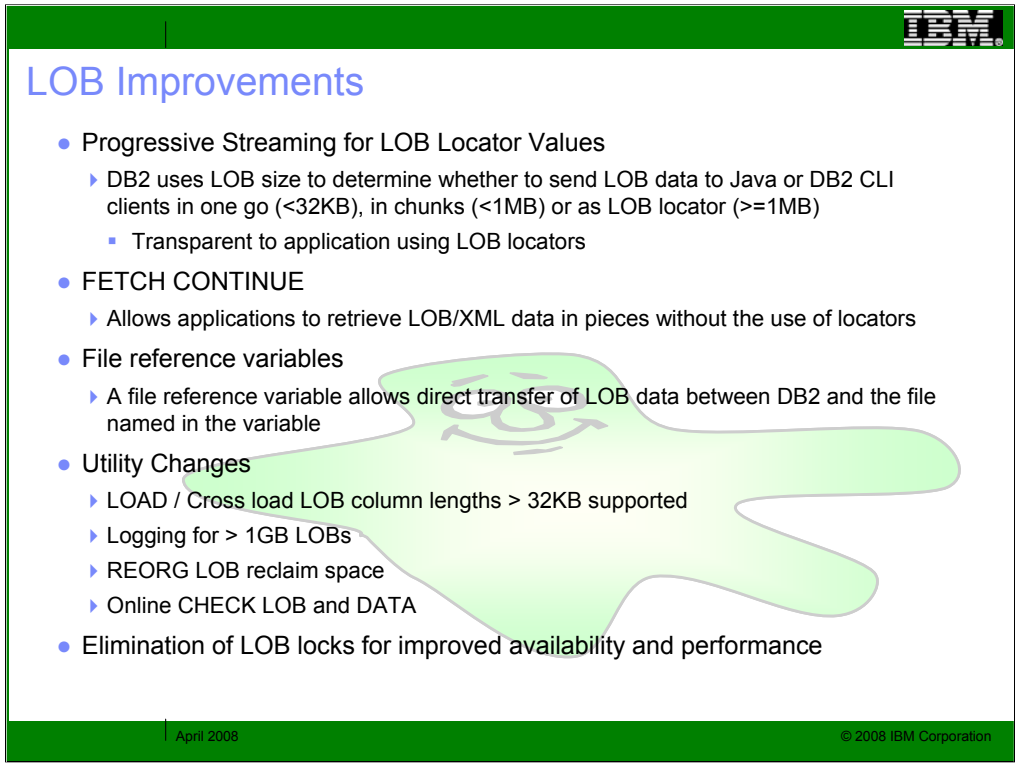



Implicit Database Support
- Automatic DB & TS
  - IMPDSDEF defaults to YES to allow CREATE IMPLICIT DATABASE data sets
  - IMPTSCMP default to NO for Compression of implicitly created table spaces in implicitly create databases
  - DSNxxxxx, where xxxxx is a 5 digit number incremented with each implicit create
- MAXPARTITIONS defaults to 256
- Support for Primary Key and Unique Keys
- LOB table space, auxiliary table, and auxiliary index support

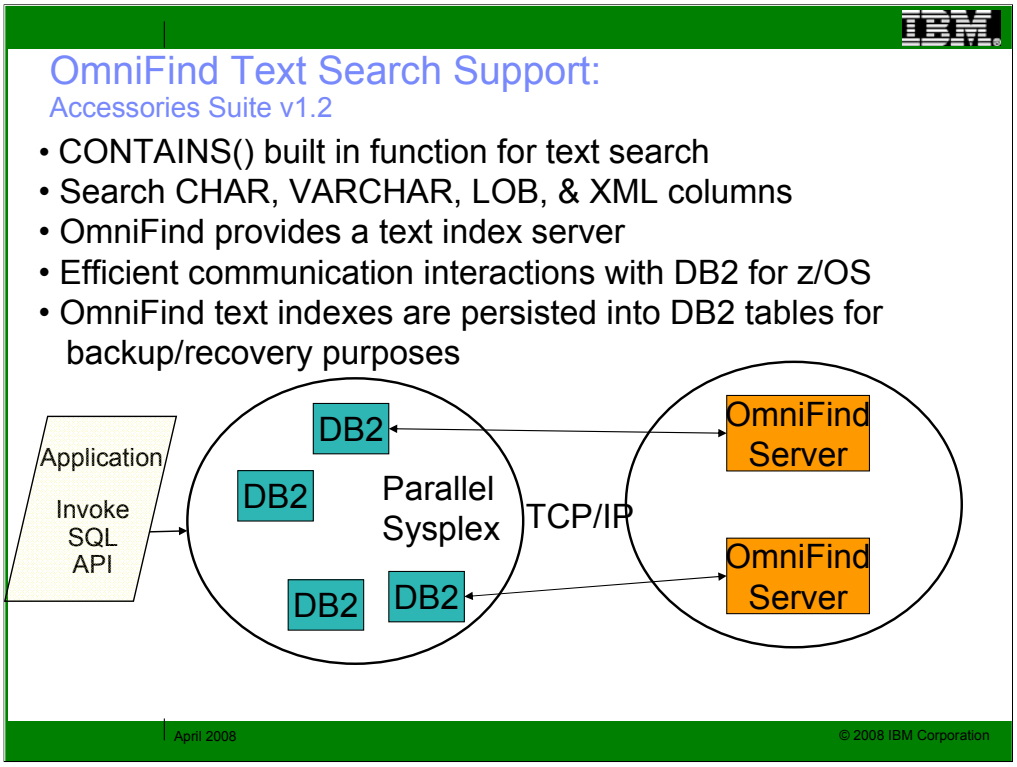
ALTER TABLE … RENAME COLUMN
- Restricted if the Column
  - Is referenced in a View
  - Has an Index on Expression defined
  - Has a Check Constraint or Field Procedure defined
- Restricted if the Table
  - Has a trigger
  - Is an MQT or reference by an MQT
  - Has a Valid Procedure or Edit Procedure
  - Is a Catalog Table

RENAME INDEX
- Existing PLAN_TABLE contents remain unchanged

New CATMAINT options
- Switch schema name
- Change from owner to role (NFM)
- Change VCAT




For Java and DB2 CLI programs that use locators with LOBs. Improves performance and less network traffic for LOBs that are less than 1MB Default behavior if using DB2 9 for z/OS Requires DB2 Connect 9.1 FP 1
No changes required to programs using locator values
DB2 Client and Type-4 driver manage progressive streaming of data to program
DB2 for z/OS determines whether to flow LOB values or Locators to client based on size thresholds for JDBC, SQLJ, and CLI
    For small LOBs, (Default <= 32KB) the performance should approximate that of retrieving a VARCHAR column of comparable size
    Medium size LOBs (Defaults > 32KB and <= 1MB)
    For large LOBs (Default over 1MB) locators are still used
Specific FETCH that contains LOB or XML columns
    Used with programs that materialize LOBs
    Application uses a buffer that might not be large enough to hold the entire LOB or XML value.
    If any of the fetched LOB or XML columns do not fit, DB2 returns information about truncated columns and the actual length.
Retrieve LOB or XML data in multiple pieces without use of locators
    Must specify WITH CONTINUE on initial FETCH
    Subsequent fetches use FETCH CURRENT CONTINUE
    Application must manage buffers & reassemble data
    Not required to fetch entire object before moving to next
    SQLCA indicates whether data is truncated
LOAD / Cross load LOB column lengths > 32KB supported
Logging for > 1GB LOBs
REORG LOB reclaim space
    SHRLEVEL(REFERENCE)
    Allows LOG NO
    SHRLVEL(NONE) is still an option.
Online CHECK LOB and DATA
Elimination of LOB locks
    Now using LRSN & page latching for consistency checks
    Prior to DB2 9, LOB locks were held until commit
        Even for UR
Space search for LOB allocation
    No LOB locks acquired for space search
    Read LSN granularity improved to page level in LOB table space
Improved availability & performance
    Particularly for UR readers
Requirements: NFM “Locking protocol 3” GBP changes Automatic in non-data sharing
    Clean group-wide shutdown in data sharing once NFM enabled until PK62027




We added a new built-in function, CONTAINS() to allow better text searching. This is the basic architecture of our solution. A specialized text search engine on a separate server is expected to come in DB2 9, but will deliver after general availability. While the server is separate, the text indexes are saved into DB2 tables to provide improved backup and recovery and consistency with DB2 data.

## DB2 9 Spatial Support

**Enabling Open Geospatial Consortium (OGC) compliant geospatial applications**

- **Spatial data types**
- **Spatial functions and predicates**
- **Spatial indexes**
- **Spatial search**
- **OGC-compliant spatial catalog**

Map
GPS
Address
Street
State
…

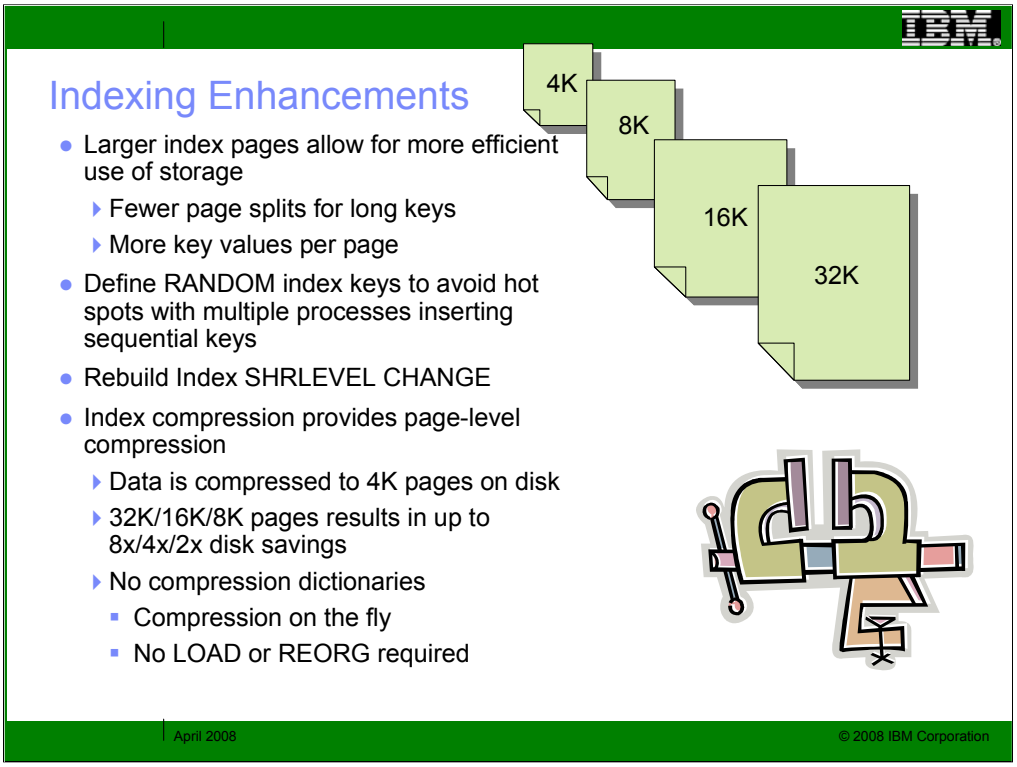
Phase II delivered in APARs PK51020, PK54451, …

Get updated IBM Spatial Support for DB2 for z/OS User's Guide and Reference

April 2008



You can use IBM Spatial Support for DB2 for z/OS to generate and analyze spatial information about geographic features, and to store and manage the data on which this information is based. A geographic feature can be:
An object (that is, a concrete entity of any sort); for example, a river, forest, or range of mountains.
A space; for example, a safety zone around a hazardous site, or the marketing area serviced by a particular business.
An event that occurs at a definable location; for example, an auto accident that occurred at a particular intersection, or a sales transaction at a specific store.
IBM Spatial Support for DB2 for z/OS makes available facts and figures about the locations of geographic features. You can use functions to provide spatial data. IBM Spatial Support for DB2 for z/OS provides services to import spatial data in Shapefile and SDE Export formats.
Basically it allows you to add a spatial column (e.g. LOCATION) to a DB2 table (that has customer info including an address). You can then use a GEOCODER to translate the address into coordinates. The cordinates are stored in the spatial column LOCATION.  You can then use ESRI's Arc Explorer to combine a map with your DB2 table to visualize where your customers are for example. There are many other uses and possibilities.  ESRI announced on 3/21/07 that they will be providing support for DB2 for z/OS spatial functions.  Phase II support is now available. GIS = Geographic Information System.  Aka Geospatial.  E.g. Google Maps.  OGC = Open Geospatial Consortium




Larger index pages allow for more efficient use of storage

- Fewer page splits for long keys
- More key values per page

Multiple processes inserting sequential keys can create hot spots on indexes

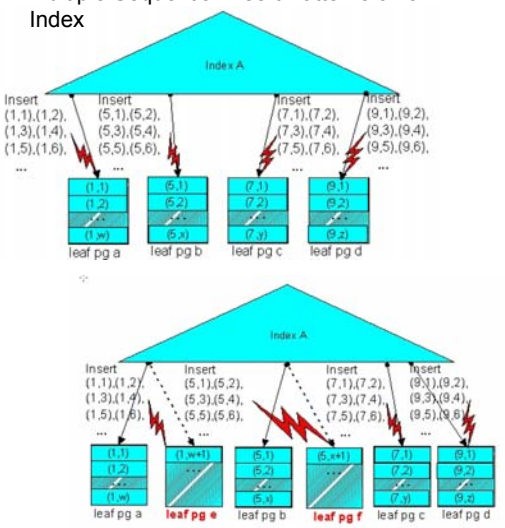
- Randomized index keys avoid hot spots
- Application throughput improved via avoidance of locking conflicts

Rebuild Index

- SHRLEVEL CHANGE

## Asymmetric Index Page Splits

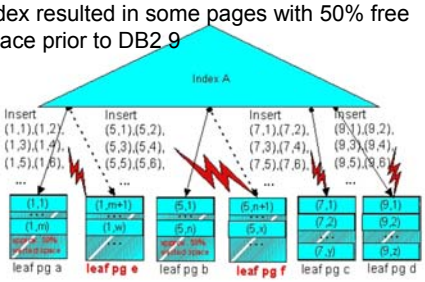
Multiple Sequential Insert Patterns on an Index







Sequential inserts into the middle of an index resulted in some pages with 50% free space prior to DB2 9

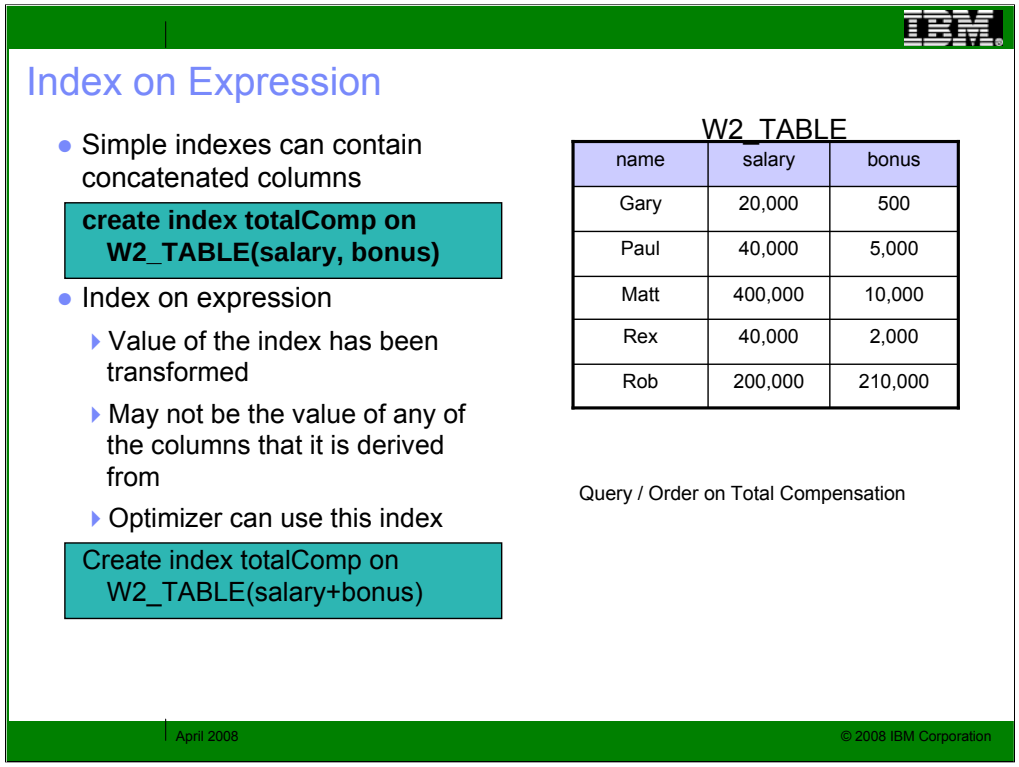



New algorithm dynamically accommodates a varying pattern of inserts




Performance for inserting is expected to increase substantially, through a wide range of improvements. Logging performance is improved substantially, with data sharing improvements and archiving. The newer disk and channel changes (DS8000, 4 Gb per second channels, MIDAW), improve the data rates substantially. Indexes are improved, with larger page sizes to reduce the number of page splits and also a better page split. Where performance should be optimized for inserts, rather than for later retrieval, the append option can be used. If the data need to be randomized to avoid insert hot spots, the new randomized index key is useful.

The segmented space structure is more efficient, so adding universal table space structure for the large partitioned table spaces helps DB2 scale.



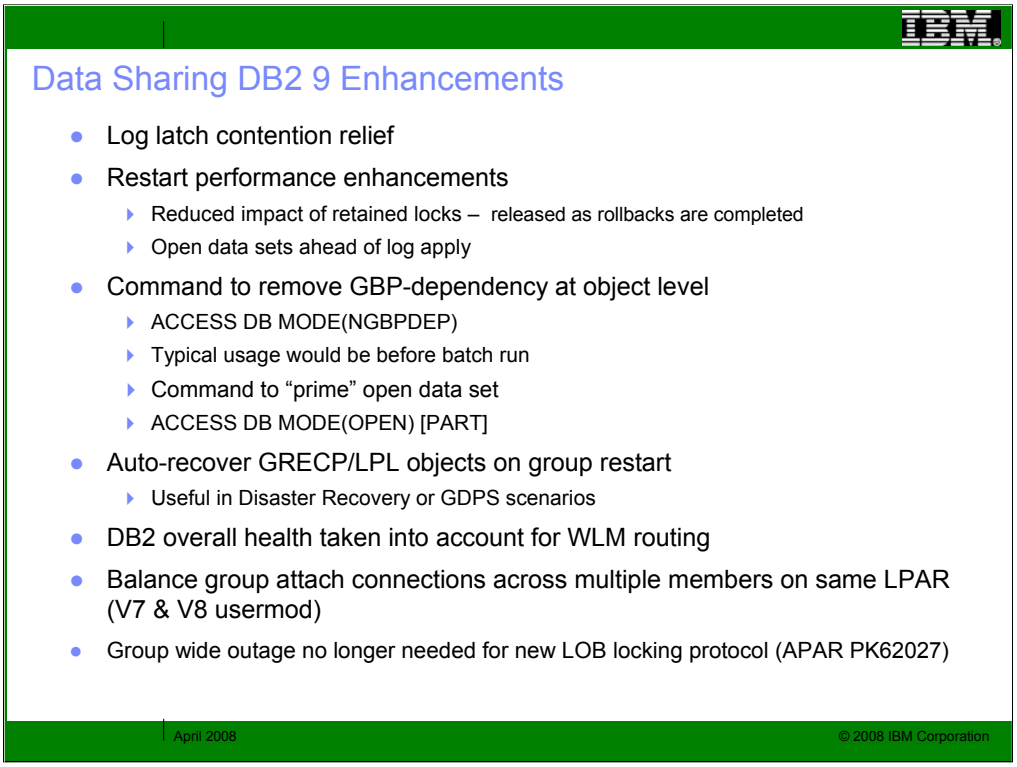

Test query WITH PAYTOT(FIRSTNME,LASTNAME, TOTALPAY) AS (SELECT FIRSTNME,LASTNAME,SUM(SALAR

Y) + SUM(BONUS) FROM DBA032.EMP GROUP BY LASTNAME, FIRSTNME) SELECT FIRSTNME FR

OM PAYTOT WHERE TOTALPAY=(SELECT MAX(TOTALPAY) FROM PAYTOT) applied to this example is WITH PAYTOT(name, totalpay)

AS (select name,SUM(salary) + SUM(bonus) from W2_TABLE GROUP BY name)

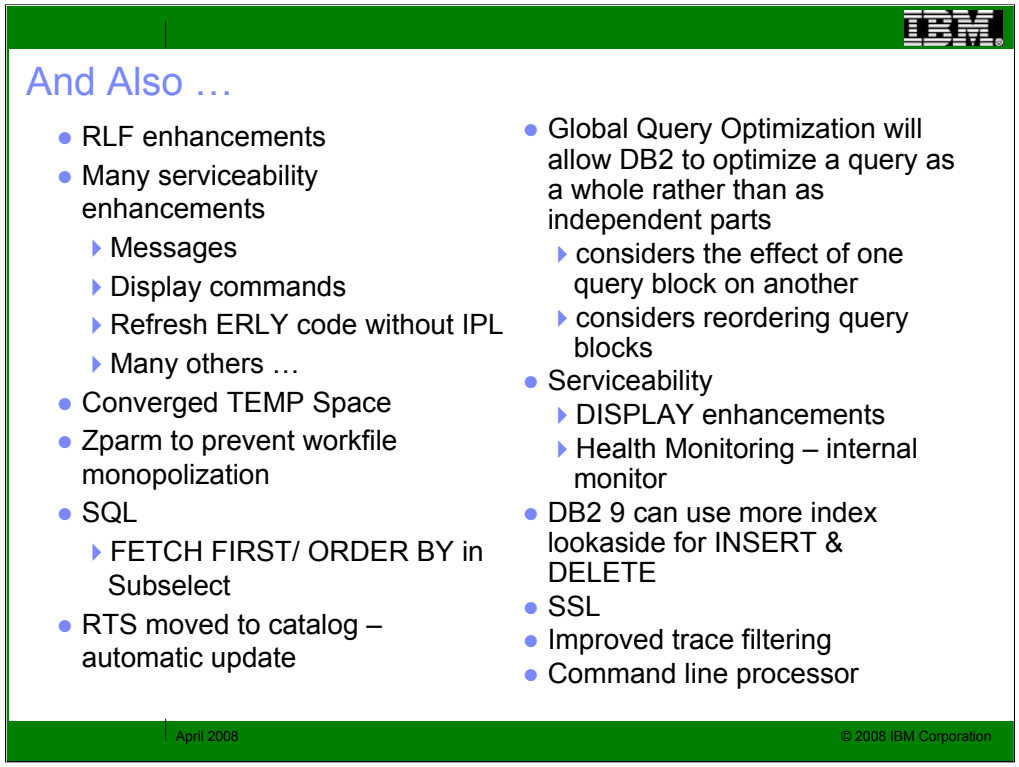
SELECT name from PAYTOT WHERE totalpay=(SELECT MAX(totalpay) from PAYTOT)



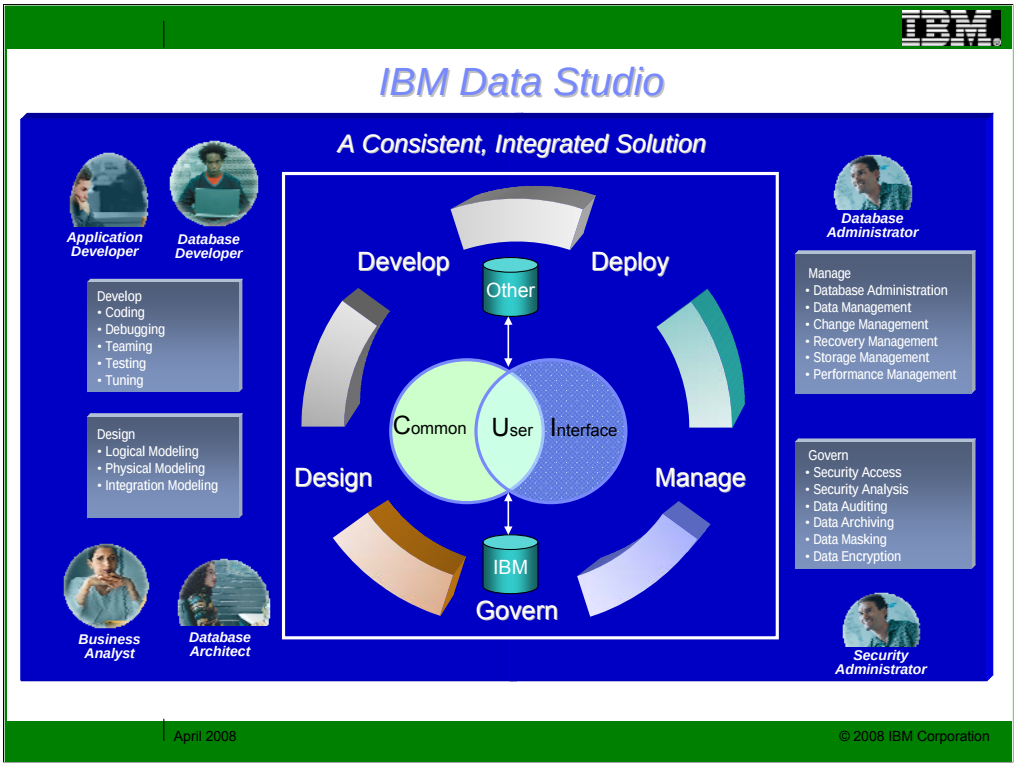

-ACCESS DB….,

**Mode (OPEN)** Forces the physical opening of the page set or partition on just the local member. This moves the overhead of the physical open from an SQL thread to the command thread. This improves the transaction rate for the first SQL thread to reference a given page set or partition.

**Mode (NGBPDEP)** Converts the specified page set or partition, non-disruptively to a non-group buffer pool dependent. You should use this before running large batch processes against a particular page set or partition to improve performance in a data sharing environment. Only issue this command to the member on which you plan to run the batch programs.



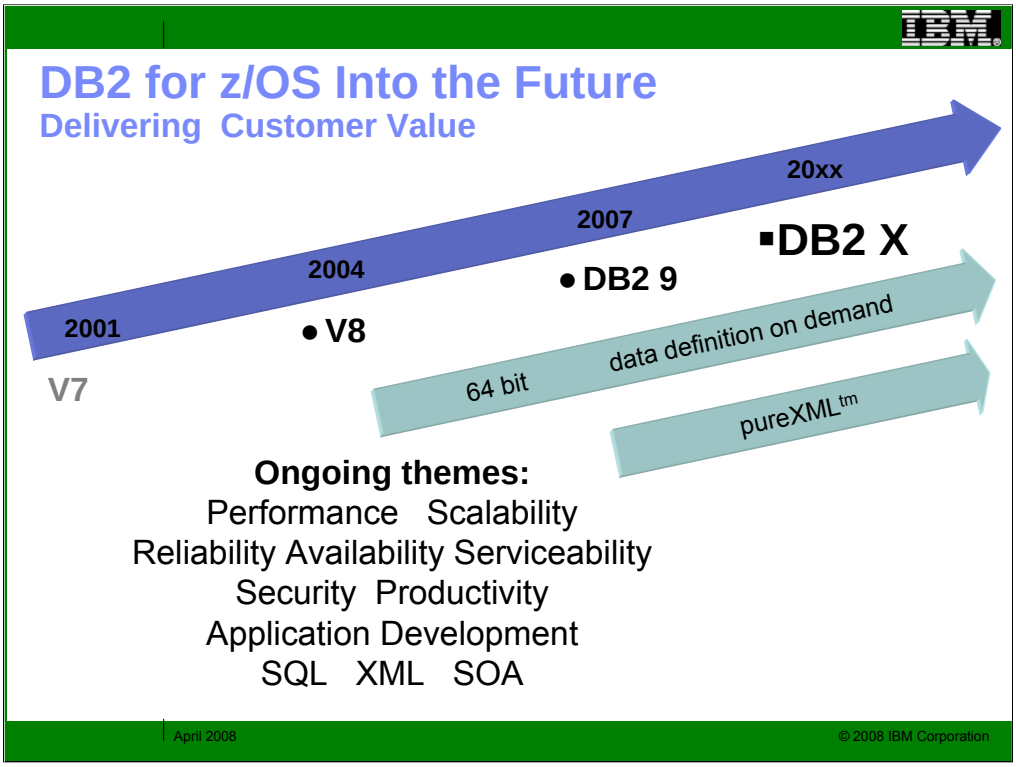

There is much more to talk about, but not much time to do it. Many customers have a favorite item that's in this list. Improvements for security, performance, operations and programmers are included.




IBM Data Studio is for designers, developers, and administrators who work across the entire data life cycle and across the full set of IBM relational DBMS: DB2 for LUW, DB2 for z/OS, DB2 for i5/OS and IDS. Almost every customer will need the tools included with DB2, and many will need the additional ones. See the IBM Data Studio web page for all the changes in IBM Data Studio. Watch closely, as this area is changing fast.

http://www.ibm.com/software/data/studio/

ftp://ftp.software.ibm.com/software/data/db2zos/IOD1298_ADadminTrendsDirections_CotnerOct2007.pdf

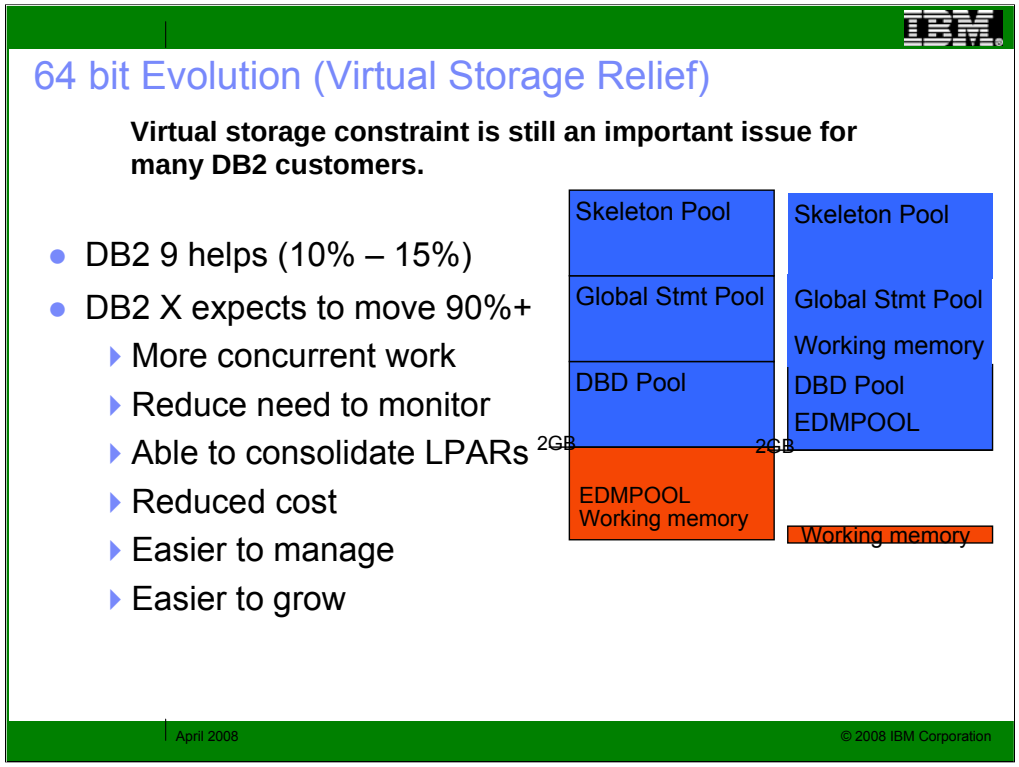



DB2 for z/OS V7 became generally available (GA) March 2001, and V8 delivered three years later. DB2 9 became generally available in March 2007, three more years. We expect the next version will be 2.5 to 3 years from DB2 9 GA to DB2 10 or DB2 X or whatever the name becomes.

The themes for future versions will continue to focus on core platform strengths of performance, scalability, reliability, stability, availability, resilience, and security. PureXML and Schema evolution or data definition on demand will be ongoing for a long time. In contrast, most of the 64 bit evolution should be completed in DB2 X.

The key interfaces for customers and vendors expand for both XML and for SQL. Information is a key leg of the SOA platform, and DB2 for z/OS provides many advantages for data management in SOA.

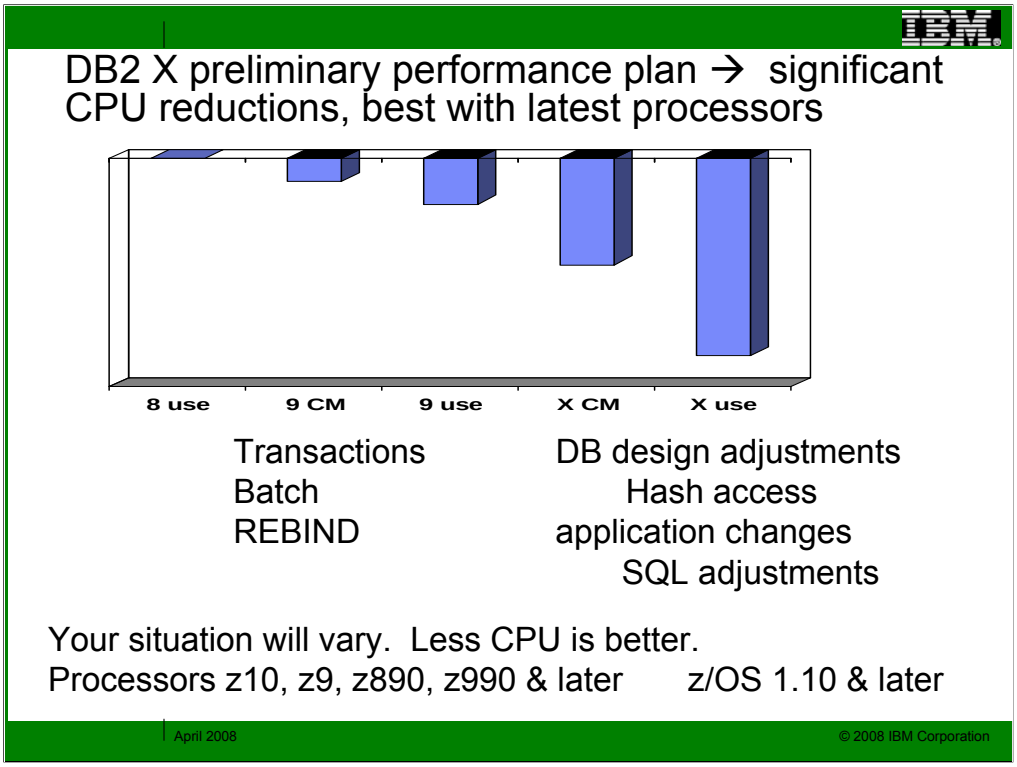
Standards, interoperability, portability and security along with secure access using the latest technologies are key touch points. Productivity improvements for application developers and for database administrators are very important as data grows in scale and complexity.




The DB2 9 virtual storage objective was 10-15% relief. The DB2 X target is more than 90% of the DBM1 address space and also in the DDF address space. We expect the result to be the ability to run much more concurrent work, with an early guess of 3 to 5 times more threads.

Storage monitoring should be drastically reduced. Customers are consolidating LPARs. Sometimes they need to have more than one DB2 subsystem on an LPAR, costing real storage and CPU. With these changes, work can run in one DB2 subsystem, rather than needing more members.

The net for this change is expected to be reduced cost, improved productivity, easier management, and the ability to grow DB2 use much more easily.




Reducing CPU from DB2 9 to DB2 X without significant administration or application changes is the primary thrust of the performance work.

This work is very preliminary, but the performance plan for DB2 X is much more aggressive than in any recent version. The last version which contained significant improvements for reducing CPU time in transactions and batch was Version 2 in 1988. Versions 3 to 9 made improvements in queries and in utility CPU time and provided many scalability improvements, but little reduction in transaction CPU time.

We expect DB2 X to run only on z10, z9, z890, z990, and later processors, and to provide CPU reductions from the beginning, with improvements in CM, but more dramatic reductions for applications that can take advantage of the improvements in application design.

## Availability: extend DB2 industry leadership

- **Increasing need to make database changes online**
- **DB2 X:**
  - More online schema change: ALTER & online REORG
    - Table space type to universal
    - Page size & Member cluster
    - Drop pending changes
  - REORG SHRLEVEL(CHANGE) for LOBs
  - Consistent image copy without quiesce
  - Inline copies to allow for dataset-level FlashCopy
  - Online REORG usability and performance enhancements

April 2008



Continuous availability requirements continue to escalate. Nice big batch windows are a thing of the past. DBAs increasingly need the ability to make all changes and to do all maintenance activities online or around the clock.

**DB2 X** allows more online schema changes with an ALTER for a PENDING change, then an online REORG to take effect. X also allows table space type to be altered to universal for a table space containing a single table (simple, segmented or partitioned). Page size and member clustering can be altered. Index changes become less disruptive. Pending changes which have not been completed with a REORG can be dropped.

REORG SHRLEVEL(CHANGE) for LOBs

Consistent image copy without quiesce

Inline copies to allow for dataset-level FlashCopy

Online REORG usability and performance enhancements

## Business Security & Compliance Needs

- **Data retention**
- **Protect sensitive data from privileged users**
- **Separate authority to perform security related tasks**
- **Allow EXPLAIN without execute privilege or ability to access data**
- **Audit privileged users use**


DB2 X

- Temporal or versioned data, 'as of' query
- Row level and column level access
  - Allow masking of value
- Finer granularity administrator privileges

April 2008



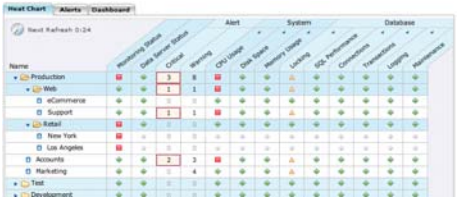
Customers are being pressed for a wide range of improved security and compliance. Data retention is a growing need. Protecting sensitive data from the privileged users and administrators is required. Separation of authority for security, access, and some common tasks, like EXPLAIN will help. Auditing for privileged users can also make compliance simpler.

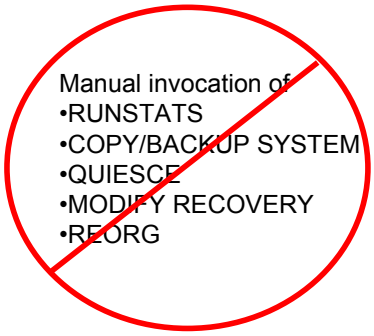
In DB2 X, we expect to have a form or temporal data or the ability for a table to contain both current and historical data, and to query the information as of a specific point in time.

Access control is refined in several ways with better granularity for the administrative privileges and with finer grained access control at the row and column level, including the ability to mask access to fields. A separate privilege is provided for security administration and for explain. Auditing is also enhanced.

## DB2 X Utilities and Tools: Productivity – Doing More with Less!

- Web-based Data Studio administrative console
- Reduce or eliminate tasks
  - Auto-stats
  - Auto-storage management
  - Auto-compression
  - Recover to a different table space in a different DB2 with consistency
  - Reduce RUNSTATS resource consumption
  - Dataset-level FlashCopy
  - UTSERIAL elimination
  - REORG Enhancements (LOBs, etc.









Some of the improvements come with Data Studio administration – stronger cross-platform graphical interfaces for improved productivity.

Some of the improvements come within DB2 for z/OS. DB2 utilities have a strong focus on making DB2 easier to use by automating tasks and eliminating tasks where possible. Avoiding the manual invocations can also help avoid problems for running the function too often or not often enough.


This text just shows the projected relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. The key at the top has the shorthand, the platform and the level.

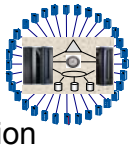
This step in the process is DB2 X for z/OS and DB2 for luw Y. The projected DB2 X for z/OS and DB2 for luw code named Y are being developed, not announced or generally available for some time. We expect to be able to move more from the z and the luw list to the common list with these changes to DB2.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. The changes in a specific version are not consistent. As we introduce new function, sometimes it will be on one platform first, but movement from unique lists into the common list continues to be the strongest trend.

## Beyond DB2 9 Improvement Focus Areas

- **Performance: OLTP, Query, Index & LOBs**
- **Scalability & 64 Bit Memory**
- **SQL & SQL Procedures**
- **XML**
- **SOA**
- **Security**
- **Availability**
- **Autonomics**
  - Administration
  - Security



April 2008



Beyond DB2 9, many of the focus areas are familiar. We need to have substantial improvements in performance, so that transactions and queries can improve, as well as improvements in LOBs and XML.

More work is needed in scalability, so that more threads can be run, with less work to manage and tune virtual storage.

SQL and SQL procedure language continue to need enhancements to improve programmer productivity and make porting from other DBMS much faster and easier.

XML made a huge stride in DB2 9, and customer usage will show many improvements needed.

Service Oriented Architecture requires many other changes to complement the SQL and XML changes.

Availability improvements continue, reducing planned outages with more changes that do not need an outage.

Autonomics help with productivity, even as they improve performance and availability by reducing or eliminating tasks in administration and security.

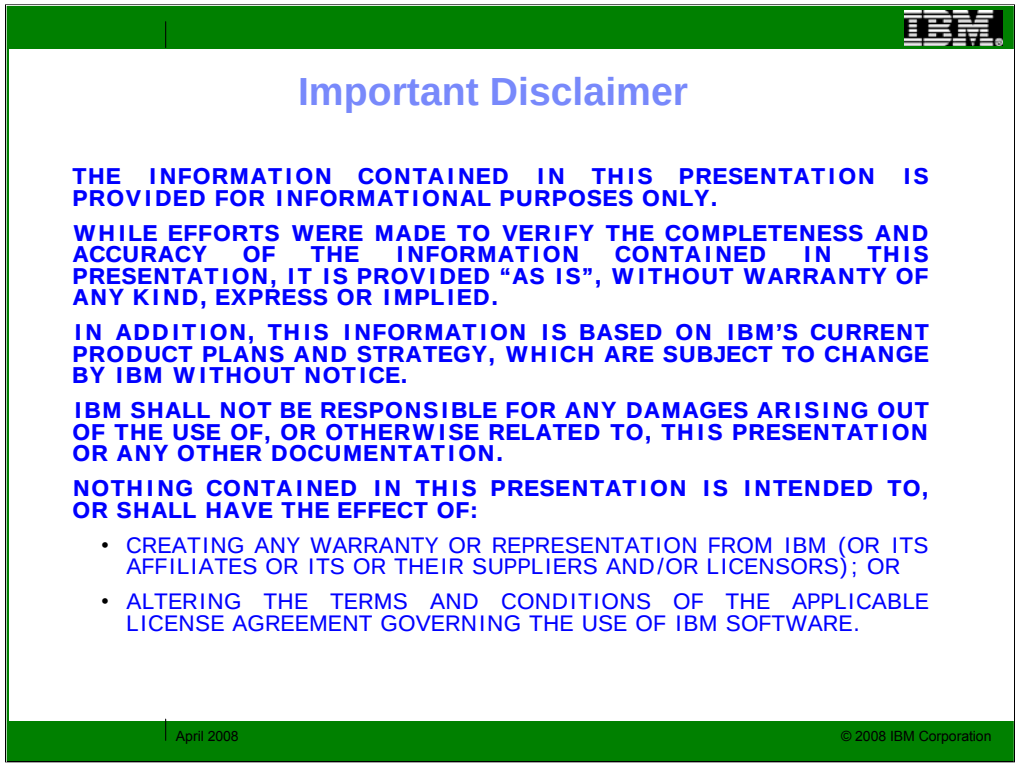
DB2 for z/OS Directions

# Questions?


References and memory in migration section
System z, zIIP in TCO section
Utility information section




Thanks for coming. We appreciate your questions and are looking forward to your feedback.




This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in the operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

*IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.*

*TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/400, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, System z,*

*The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, ODBC*