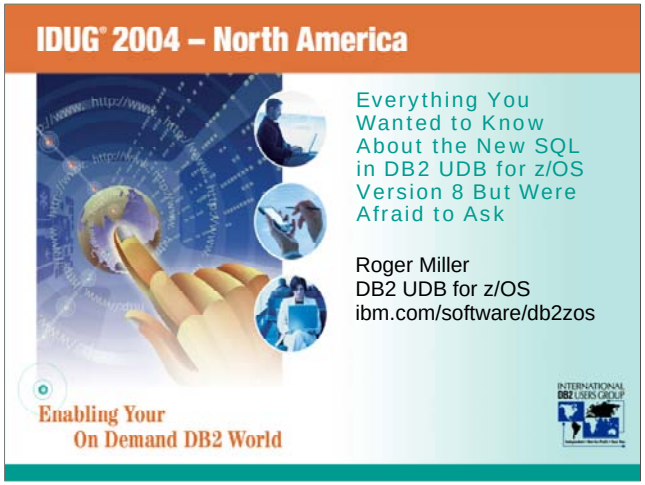


This is an overview of DB2 ® UDB for z/OS Version 8 (V8). DB2 V8 is the twelfth release and delivers more function than any release of DB2 for MVS. DB2 V8 became generally available March 26, 2004. This version brings extensive integration and synergy with zSeries hardware, with middleware and with applications. Data support, application development and query enhancements are added for e-business, building upon the traditional enterprise of choice characteristics of availability, exceptional scalability, and performance.
DB2 Version 8 has been re-engineered for e-business on demand, with many fundamental changes in architecture and structure. Key improvements enhance scalability, application porting, security, and continuous availability. Management for very large databases is made much easier, 64-bit virtual storage support makes management simpler and improves scalability and availability. This new version breaks through many old limitations in the definition of DB2 objects. These enhancements include SQL improvements, schema evolution, longer names for tables and columns, longer SQL statements, enhanced Java and Unicode support, enhanced utilities, more log data sets, and a lot more.

Version 8 is also a breakthrough in SQL, with too many new functions to list them all. We will discuss a few of them on the next pages. Add support for volatile tables, group by expression, multiple DISTINCT clauses, and qualified names for INSERT and UPDATE.

Table function improvements, trigger performance, star join sparse index improvements, cost-based parallel sort, better ability to use indexes, longer statements, longer names, longer literals and predicates, session variables, new special registers and SQL procedures improvements, provide a giant leap for SQL.

IDUG® 2004 – North America

Limits: DB2 for z/OS

Image of Earth from Moon, Source: NASA (Public Domain)

Breaking through limitations - SQL

- Table name sizes 18 to 128
- VIEW & ALIAS names 18 to 128
- Column name sizes 18 to 30
- SQL statement length 32K to 2MB
- Tables in a join 15 to 225
- Character Literals 255 to 32704
- Hex literal digits 255 to 32704
- Predicates 255 to 32704
- Index key 255 to 2000

Enabling Your On Demand DB2 World

One of the keys to reengineering is breaking through the limits of the current architecture. Increasing some limits improves scalability. Increasing other limits improves productivity, portability & family consistency. Increasing name sizes & SQL statement lengths makes porting from other DBMS much easier and improves DB2 family compatibility. Increasing the amount of virtual storage allows the longer names, larger SQL statements and increased sizes. DB2 V8 will often require a little more real memory (1% to 10% for large subsystems), but permit increased scalability and availability.

IDUG® 2004 – North America

Limits: DB2 for z/OS

Image of Earth from Moon, Source: NASA (Public Domain)

Breaking through limitations SQL ...

- Current optimization 8 to 128
- CURRENT PACKAGESET 18 to 128
- CURRENT PATH 254 to 2048
- SCHEMA 8 to 128
  - Authorization ids still 8 bytes

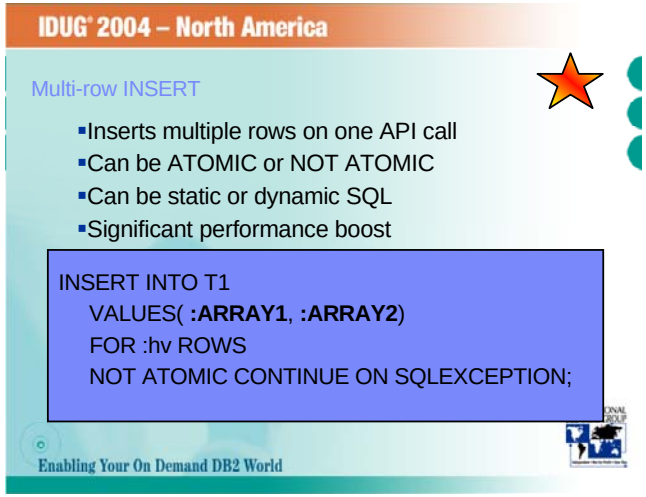
Enabling Your On Demand DB2 World

The larger number of tables in a join helps with porting applications and improves our ability to handle vendor applications.

The additional active and archive logs provides larger volumes and better flexibility for the amount of log data. Customers can keep up to 372 GB of active log data and 40 terabytes of archive log data.

Maximum table sizes increase for partitioned table spaces with 8K (to 32 TB), 16K (to 64 TB) and 32K pages (to 128 TB).

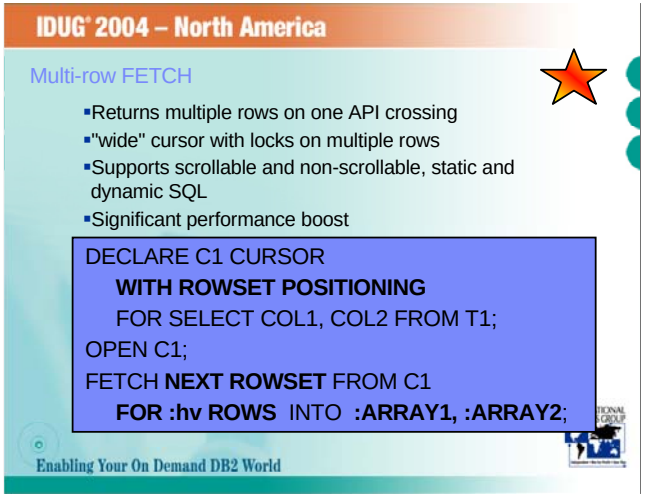
Increasing the special register lengths allows more flexibility for applications.



Being able to insert multiple rows helps in application portability and can improve performance, particularly across a network.

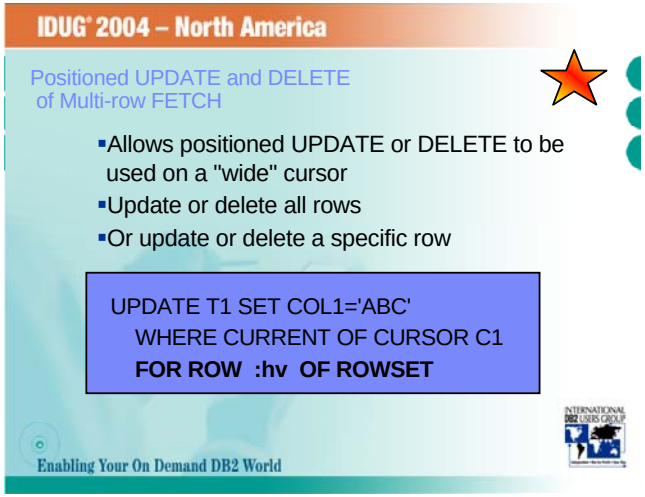
You can have atomicity (all rows must be successfully inserted or all are rolled back) or insert and then resolve any errors.

This change supports host language arrays, adding another in a range of customer enhancement requests.

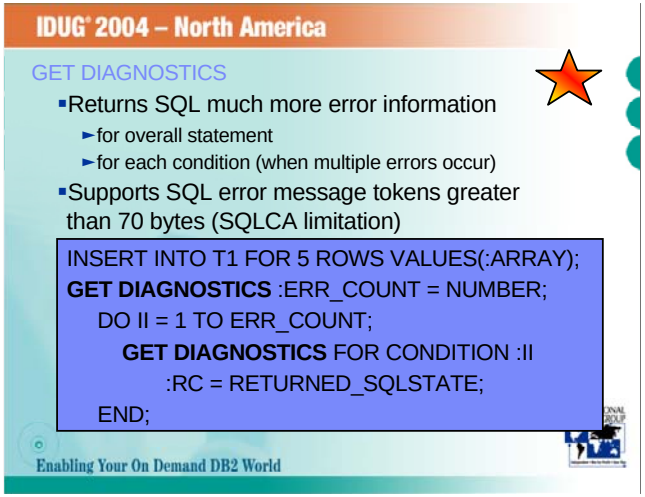


Multiple row FETCH also helps with application portability and performance. It provides a new concept called a "wide" cursor, which contains multiple rows rather than just one. A rowset is a set of rows that is retrieved through a multiple-row fetch.

Being able to FETCH multiple rows at once can make a larger percentage improvement in performance, compared to INSERT. The FETCH statement requires less processing than INSERT, generally.

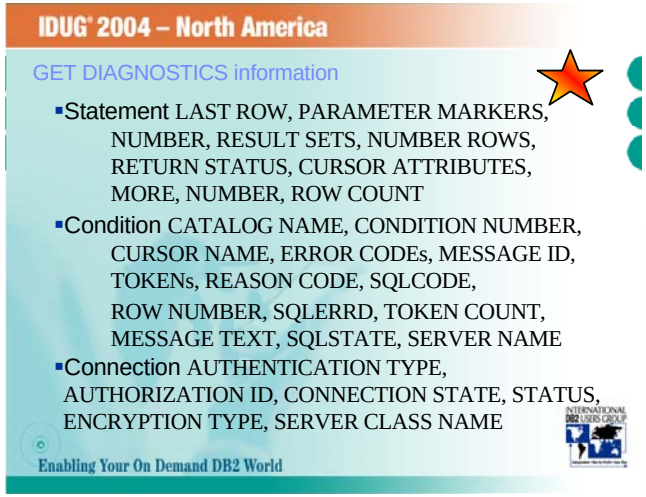


The concept of the wide cursor was extended to the UPDATE and DELETE. This shows how to update a specific row of the rowset in a wide cursor. If the cursor is positioned on a rowset, and the UPDATE is WHERE CURRENT OF cursor-name, then all rows corresponding to the rows of the current rowset are updated.

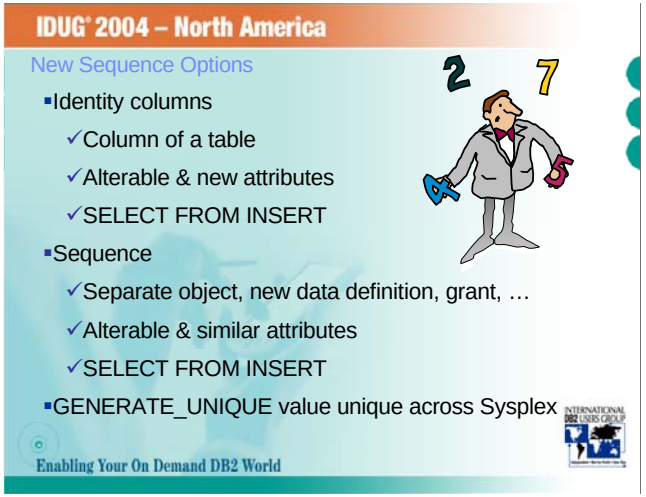


The new GET DIAGNOSTICS statement is important to provide the information from all of the extended names and new function. Most programmers will need to switch from using the less standard SQLCA and use this more standard, more capable facility for diagnostic information when their applications include long names or multi-row operations.

The GET DIAGNOSTICS statement can return a lot more information about the statement and / or about conditions and connections. It can return the longer names. It can return multiple conditions for the multi-row statements. It can return the error message associated with an error.



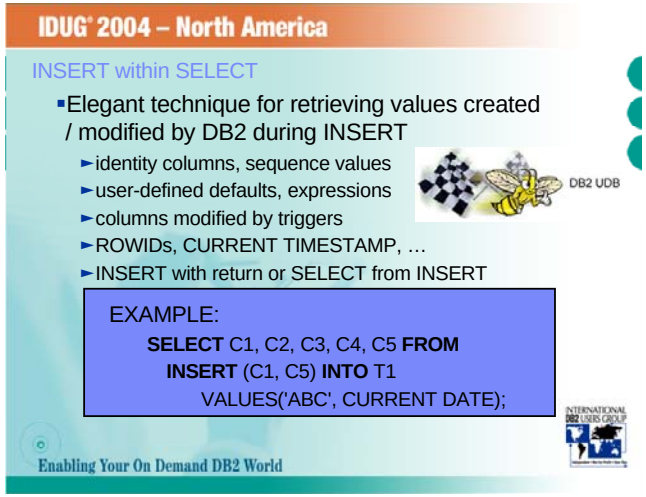
In addition to the information that can only be returned using GET DIAGNOSTICS, such as for multiple rows or for long names, there are many items that now have a standard technique for returning them to the application. There is a more standard technique for getting information about the statements, more information about any warnings or error conditions and information about the connection.



Identity columns were introduced in DB2 UDB for OS/390 Version 6, but Version 8 provides needed enhancements for usability. New attributes are added for sequences and the attributes can be altered, so that operations like LOAD are easier. A new SELECT FROM INSERT statement provides a standard way to retrieve the result of inserting an identity column.

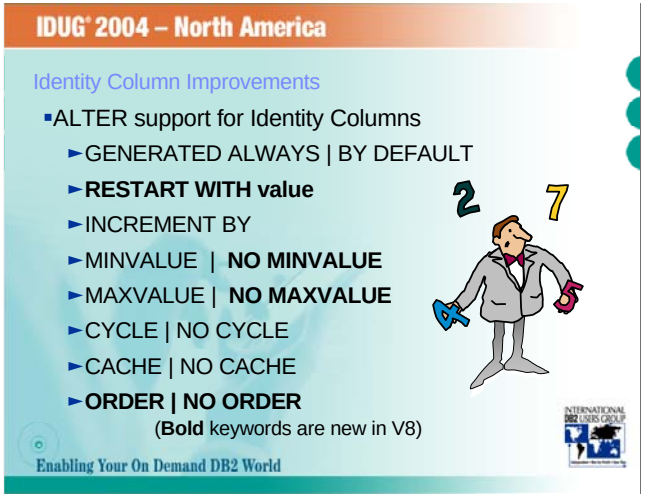
Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences. Sequences are helpful in porting Oracle applications to DB2, while identity columns are more like the SQL Server construct.

A new GENERATE_UNIQUE function is also added to generate a CHAR(13) FOR BIT DATA that is unique across a data sharing group or Parallel Sysplex.



Have you wanted to return values from the row you just inserted? Applications often need the value of the identity column, the defaults, expression results, current timestamp or the effect of a trigger. Now you can SELECT FROM the inserted row. While there have been some specialized techniques, this is more general and more elegant.

Some programmers think of this as being able to return the generated values from an insert or insert with returns.



When identity columns were provided with Version 6, customers identified some important enhancements that were needed. Many customers asked for the ability to ALTER the identity column attributes, especially the ability to change the GENERATED option. This added flexibility will allow identity columns to be used in many more situations.

New identity column attributes can be specified to aid porting from other vendor implementations: RESTART WITH, NO MINVALUE, NO MAXVALUE, NO ORDER, ORDER. The identity column is a style often used by those who use Sybase.

Sequences

- Useful for porting Oracle applications
- New SQL support:
  - CREATE SEQUENCE
  - ALTER SEQUENCE
    - RESTART WITH value
    - INCREMENT BY
    - MINVALUE, MAXVALUE
    - CYCLE/NO CYCLE
    - CACHE/NO CACHE
  - GRANT/REVOKE privileges for sequence object
  - NEXT VALUE FOR expression
  - PREVIOUS VALUE FOR expression

Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences.

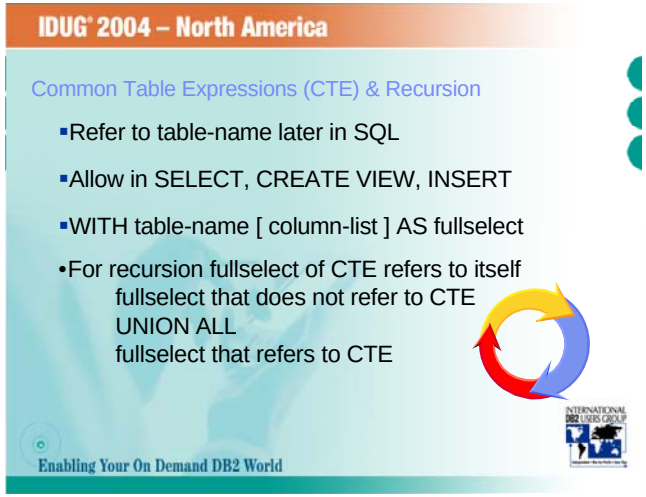
Sequences are helpful in porting Oracle applications to DB2. The language to use a sequence is to specify NEXT VALUE FOR expression or PREVIOUS VALUE FOR expression in your SQL statement.

Dynamic Scrollable Cursors

- Scrollable cursor provides access to base table rather than work file -- allows viewing of updates by others.
- Defaults to single row fetch, so DDF applications should use:
  - multi-row FETCH
  - positioned update/delete for multi-row FETCH

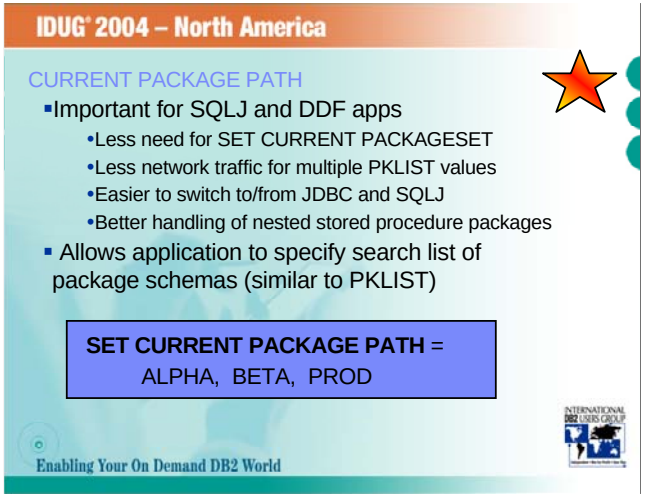
```
DECLARE C1 SENSITIVE DYNAMIC SCROLL
CURSOR FOR SELECT C1, C2 FROM T1;
```


Static scrollable cursors came in V7, with the ability to use a work file for scrolling. Dynamic cursor scrolling is performed directly on the base tables. You can use embedded SQL for scrolling, and this work complements the multi-row SQL statements.



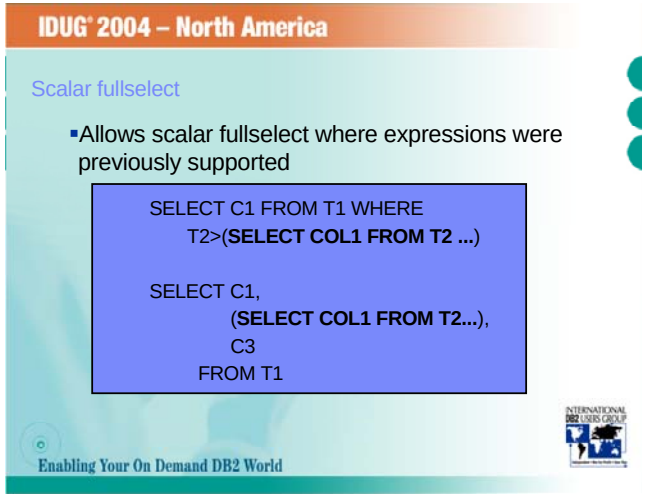
Common Table Expressions provide improved usability and DB2 family consistency. In some cases they can be used to improve performance as well, computing a value once, rather than several times.

The common table expression is defined and used within an SQL statement. Each common table expression can be referenced many times in the statement, and all references to a common table expression in an SQL statement share the same result table. This is unlike regular views or nested table expressions which are derived each time they are referenced.



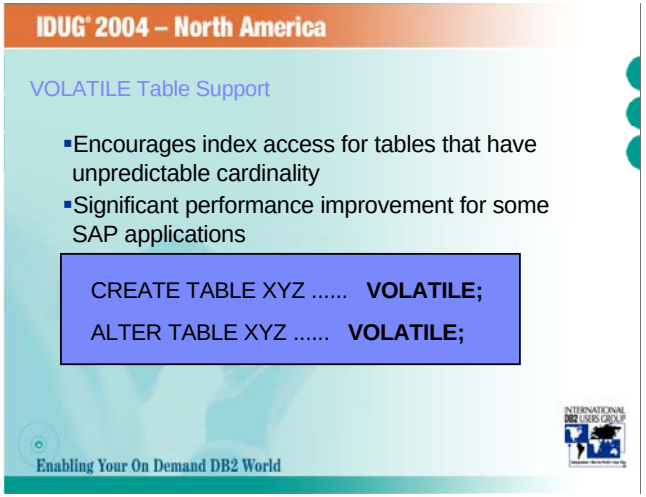
Package switching and versioning for static SQL applications is critical to DB2 for z/OS customers. SQLJ access will increase the need for these types of control. A new special register, CURRENT PACKAGE PATH, provides a means to specify a list of collections to search for the appropriate package.

The semantics are similar to the PKLIST Bind Option, except that the PACKAGE PATH list is processed at the server. This new special register will provide control for applications that do not run under a DB2 plan.



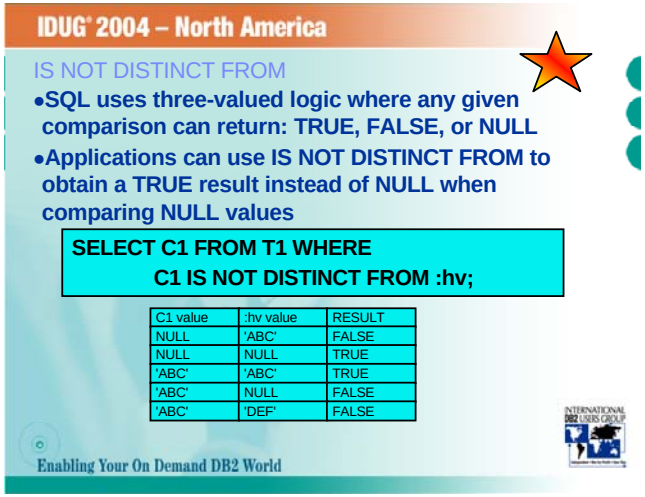
The scalar fullselect or a fullselect that results in a single scalar value to be used where an expression is allowed today. The scalar fullselect can be used in the SELECT clause, in the WHERE clause, or in a CASE expression, for instance. This improves DB2 function, the ability to pert applications and DB2 family compatibility.

A scalar fullselect, as supported in an expression, is a fullselect, enclosed in parentheses, that returns a single row consisting of a single column value. If the fullselect does not return a row, the result of the expression is the null value.

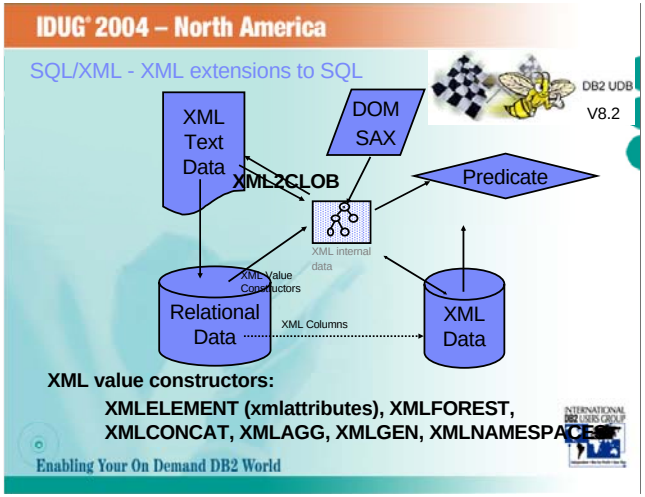


Where applications have tables that are volatile, it is difficult or perhaps impossible to find a good time to gather statistics. Volatile tables have a wide range of cardinality. For example, a table might have no rows part of the time and a million rows at other times.

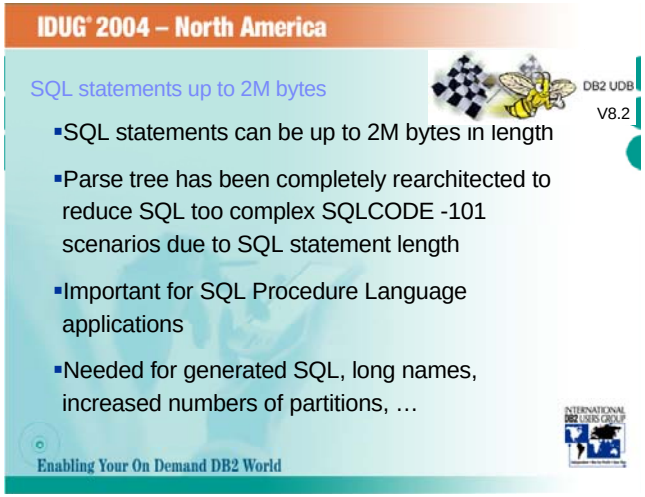
Customers or vendors can use this option on CREATE TABLE to indicate that the table is volatile, and index access should be encouraged.



This is often called the distinct predicate. By definition a null value is unknown and this makes it unequal to all other values, including other null values. The only way to test for null values is to use the IS NULL predicate, as in "WHERE col IS NULL". A predicate of the form "WHERE col = :hv :nullind" will never match a null value in "col", even if the host variable "nullind" contains a null indicator. Unfortunately, this is not intuitively obvious. The IS NOT DISTINCT FROM predicate provides an explicit technique to compare and treat two NULL values as equal.

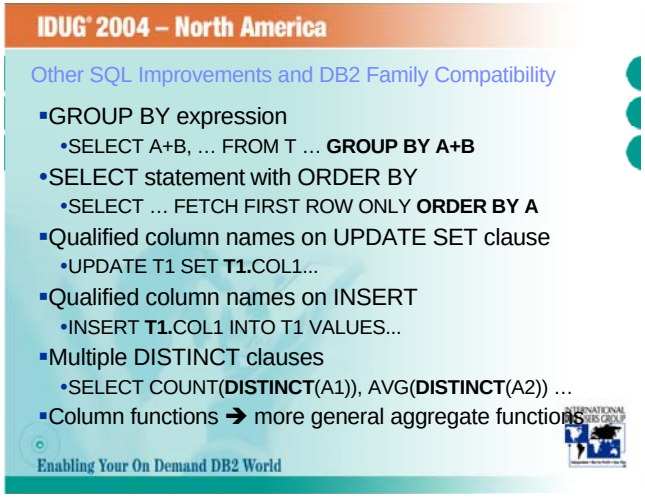


XML Publishing Functions provide a set of SQL built-in functions that allow applications to generate XML data from relational data with high performance. These functions reduce application development efforts in generating XML data for data integration, information exchange, and web services, thus enhance the leadership position of DB2 for z/OS as an enterprise database server.



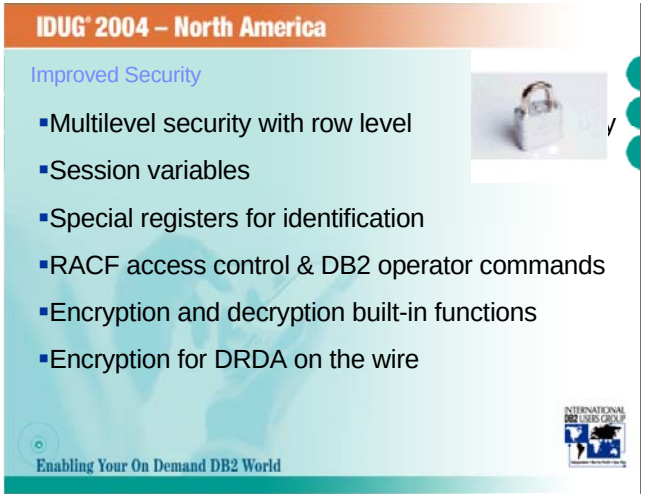
In addition to other limit-breaking support in Version 8, you can now have SQL statements that are up to 2 MB in length. A number of the Version 8 capabilities stretch the limit on the size of an SQL statement. Long names and 4096 partitions take much more space. An SQL Procedure must be completely stated in a single SQL statement, and was limited to 32K. Other changes in DB2 allow much larger structures and thus much larger statements. SQL statements that are too large or too complex should be very unusual, but still possible, with the larger statements.

Several of the changes to SQL are very important for SQL procedures. The limit of 32 kilobytes for an SQL statement was for the entire SQL procedure, and that limit is now 2 megabytes. The ability to return information to the caller of an SQL procedure is improved with the RETURN statement issued from within an SQL procedure. The GET DIAGNOSTICS statement (already valid within SQL procedures) is extended to support returning the status information from a RETURN statement of an SQL procedure (the new RETURN_STATUS keyword). Other DB2 platforms support the RETURN statement and the GET DIAGNOSTICS enhancement within SQL procedures, so these changes improved DB2 family consistency.



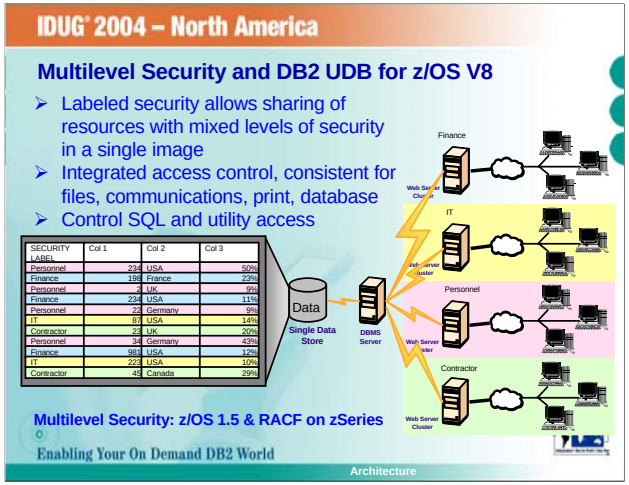
We are making big strides in SQL improvements, including: GET DIAGNOSTICS, SEQUENCE, Dynamic scrollable cursors, Scalar fullselect, Multiple DISTINCT clauses, GROUP BY expression, Qualified column names on INSERT and UPDATE SET clause, Unicode for SQL, intermixing EBCDIC, ASCII, or Unicode columns in a single SQL statement and SELECT from an INSERT statement In addition, we are making major changes in our SQL system limits by extending support for long names in SQL objects to 30 characters for column names and to 128 characters for most other SQL objects; expanding support for SQL statements up to 2 megabytes in length; increasing the length of literals and predicates to 32 kilobytes; and extending JOIN to allow up to 225 tables in a single statement.

www7b.boulder.ibm.com/dmdd/library/techarticle/0209cotner/0209cotner.html



Everyone seems to be more aware of security today. Improving integration and making security more robust and easier to manage are very important.

Customers asked for a wide range of enhancements for security. DB2 V8 provides new options for higher security, more granularity, and more information for additional flexibility in applications and SQL. Improvements include improvement for RACF access control, encryption and decryption built-in functions and encryption for data on the DRDA wire.

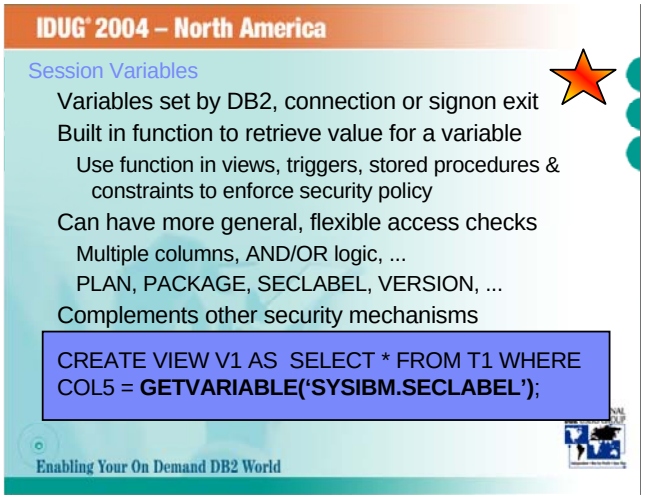


z/OS 1.5 and RACF 1.5 or Security Server add another type of security, called multilevel security, labeled security or mandatory access control (MAC) to our capabilities. The only option in the past with a high degree of separation has been physical separation. In the database world that might mean another machine or LPAR or perhaps another subsystem, another database or another table. With multilevel security, we still have a high degree of security even with data in the same table.

Access control is consistent across many types of resources using RACF, so that multilevel controls apply for data sets, for communications, for print and for database access – both objects and now with row level granularity. The DB2 controls are for both SQL access and for utility access.

For an more on multilevel security, see **Planning for Multilevel Security (GA22-7509)**

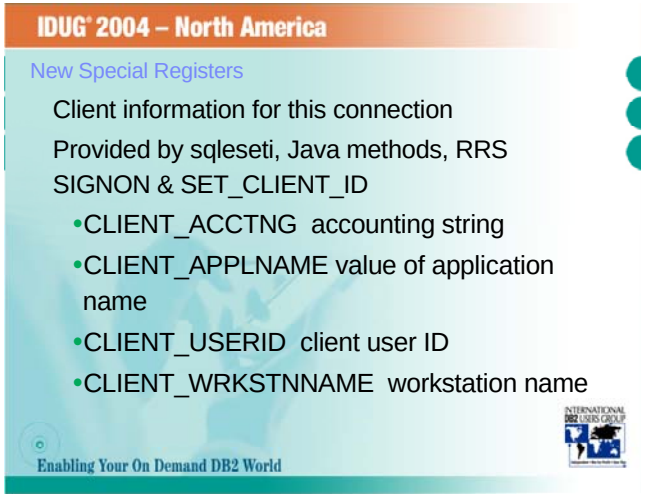
http://publibz.boulder.ibm.com/epubs/pdf/e0z2e100.pdf



Session Variables provide another way to provide information to applications. Some variables will be set by DB2. Others can be set in the connection and signon exits to set these session variables

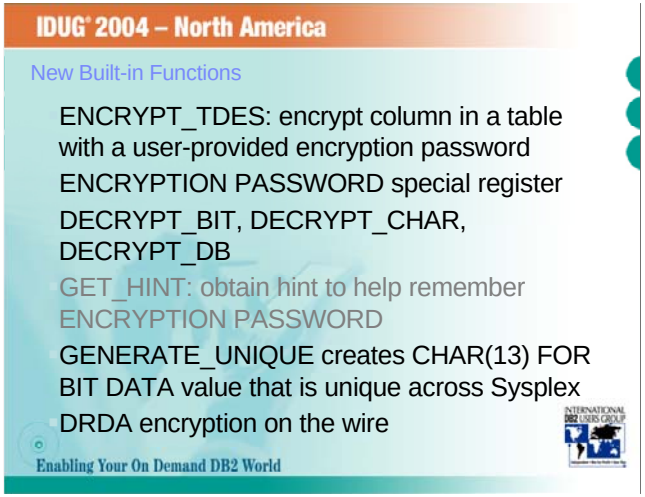
A new built-in function GETVARIABLE is added to retrieve the values of a session variable. This function can be used in views, triggers, stored procedures and constraints to help enforce a security policy. If your primary security need is more general, flexible controls, this information complements other security mechanisms.

For example, you can have a view which provides data that is at the current security label.



Four new SPECIAL REGISTERS are added to the product. These special registers are CLIENT_ACCTNG, CLIENT_APPLNAME, CLIENT_USERID, and CLIENT_WRKSTNNAME. The information is provided through a number of application programming interfaces.

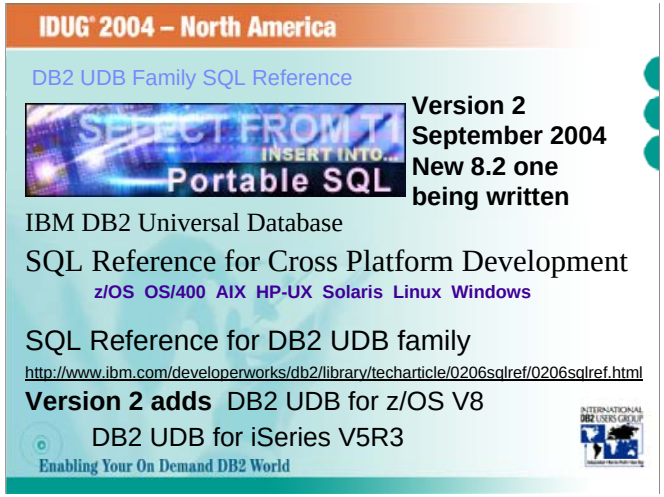
These special registers were added to DB2 for Linux, UNIX & Windows V8.



Functions ENCRYPT_TDES (triple DES), DECRYPT_BIN, DECRYPT_CHAR, and GETHINT are added. The SET ENCRYPTION PASSWORD statement allows the application to specify a password

The ability to generate a unique value is also included. These changes came in DB2 for Linux, UNIX and Windows V8, so this change improves DB2 family consistency.

DRDA is extended to allow encryption of the data being sent.



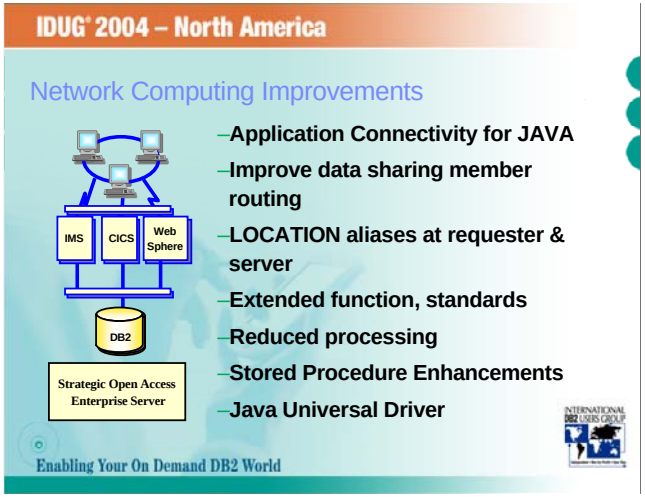
Do you want to design or write applications for the entire DB2 family, rather than for just one of the platforms? If so, you need the IBM DB2 Universal Database SQL Reference for Cross-Platform Development. Version 2 covers these product versions:

DB2 UDB for Linux, UNIX & Windows V8

DB2 UDB for z/OS V8

DB2 UDB for iSeries V5 R3

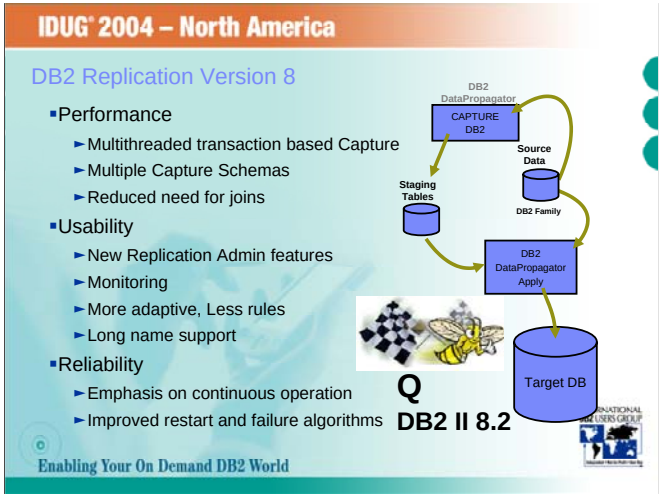
SQL consistency across the DB2 family has improved substantially in the past few versions, while significant new common function has been added. SQL that is common to the DB2 UDB relational database products and the SQL 1999 Core standard is much more comprehensive. The 852 page book describes the rules and limits for preparing portable programs for these versions. More is coming.
http://ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html



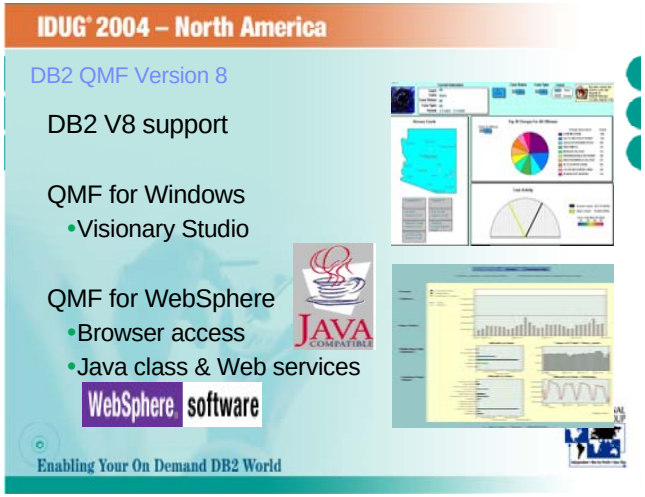
DB2 V8 also provides many improvements for network computing: security, availability, usability and performance. The improved data sharing member routine is more robust. Having the ability to use multiple names for a server adds flexibility. Increased levels of standards are implemented. Performance improvements will reduce cpu costs substantially. Improved granularity for stored procedures help with managing your work load, using the WLM.

The Java Universal Driver is used across the DB2 family.

DB2 Connect improvements are noted in the recent announcements and in the What's New in DB2 UDB Version 8.1? book. While DB2 V8 works with DB2 Connect V7.2 (fixpak 10, 11 or 12), some of the improvements only deliver with V8.1.4 or fixpak 4 or later, with Stinger. Version 7.2 end of service is September 2004, so plan to use V8.1.4 or later. Encryption support comes in Stinger.
ibm.com/software/data/db2/udb/pdfs/db2q0.pdf
Also see the web:
ibm.com/software/data/db2/db2connect/

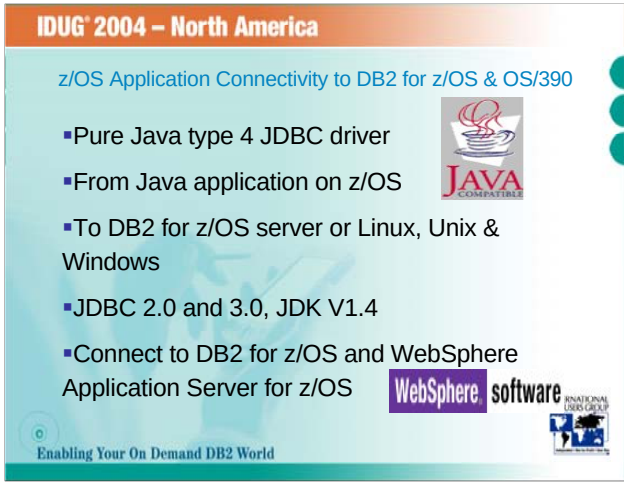


Replication is rearchitected in Version 8. The architecture for data capture has changed to improve performance and availability. See the new Redbook, The Practical Guide to DB2 UDB Data Replication V8, SG24-6828-00. V8 improves performance, throughput, has lower latency, real time alert monitor, and much improved usability and reliability. See the replication improvement and Stinger previews for the outlook on replication using queues and other improvements for more robust, higher volume, lower latency replication. With these enhancements, replication can be used for operational replication and disaster recovery situations.
publib-b.boulder.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg246828.html

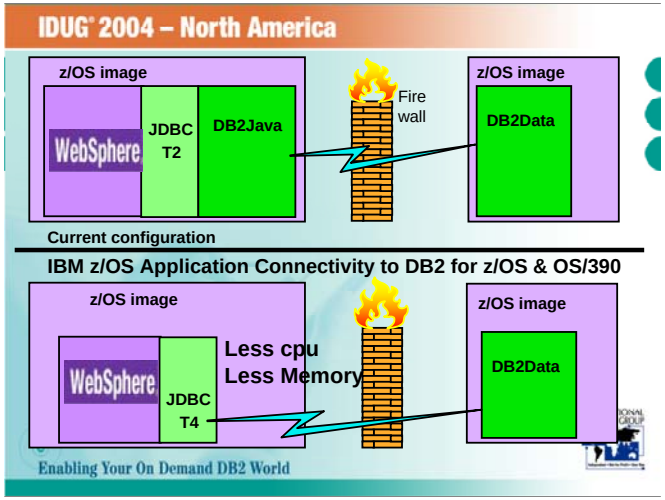


What's Coming in QMF Version 8? Easier, faster, and more global on-demand access to enterprise data and analysis through support for DB2 Version 8 plus:

QMF for Windows: - new drag-and-drop data visualization with Visionary Studio (in addition to existing summary reports, charts, and spatial data maps): across / pivot / top formatting, conditional formatting, rich HTML reports, multi-dimensional analysis (OLAP), visual query building interface, new visual database explorer, support for DB2 V8.1 features
QMF for WebSphere: new, greatly enhanced user interface for Web-based data access through an ordinary browser, visual display of customized report libraries, rapid, robust query development: Expression Builder, Java class API & Web services API for custom Web-based applications.
http://www.ibm.com/software/data/qmf/



A new IBM z/OS Application Connectivity to DB2 for z/OS and OS/390 delivers performance, scalability and flexibility to JAVA applications, as announced December 16, 2003. The announcement is for DB2 UDB for z/OS and OS/390 Version 7 and for DB2 UDB for z/OS Version 8. Please see the next foil and announcement for more information.



z/OS ® Application Connectivity to DB2 ® for z/OS and OS/390 ® is a no-charge, optional feature of DB2 Universal Database ® Server for z/OS V7 and V8. This feature consists of a component known as the DB2 Universal Database Driver for z/OS, Java ™ Edition, a pure Java, type 4 JDBC driver designed to deliver high performance and scalable remote connectivity for Java-based enterprise applications on z/OS to a remote DB2 for z/OS database server. The driver:

· Supports JDBC 2.0 and 3.0 specification and JDK V1.4 to deliver the maximum flexibility and performance required for enterprise applications

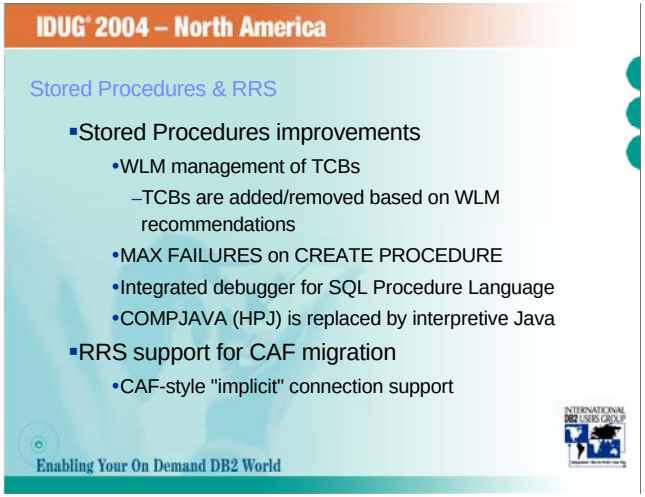
· Delivers robust connectivity to the latest DB2 for z/OS and WebSphere ® Application Server for z/OS

· Provides support for distributed transaction support

· Allows custom Java applications that don't require an application server to run in a remote partition and connect to DB2 z/OS

See the December 16, 2003 announcement or the web for more:

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/tips0356.html?Open



z/OS Workload Manager function can determine appropriate resource utilization and provide a method of changing the number of tasks within a stored procedures address space.

Customers can set a limit on how many times each stored procedure or user defined function can fail before it is stopped. We already allow the customer to specify a max abend value for all stored procedures and user defined functions on a single DB2 image.

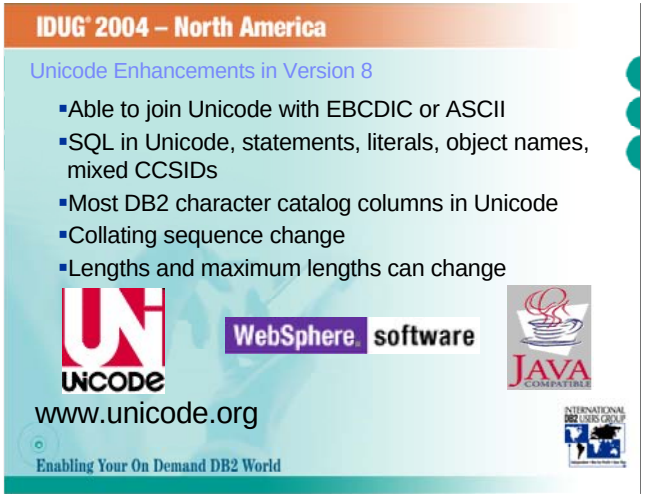
DB2 for z/OS Version 8 will no longer support LANGUAGE COMPJAVA stored procedures.

Applications that have been written to use CAF implicit connect can be easily converted to RRSAF.

Java support will be more consistent across platforms as we use a single code base across the DB2 family. The improved consistency also adds new function to DB2 and improves integration with WebSphere and Java.

The Java Universal Driver is updated to support the JDBC/SQLJ 3.0 standard, including improvements like savepoints, connection pooling improvements, the ability to reuse PreparedStatements, multiple open ResultSets for a single stored procedure, WITH HOLD cursors, and improved BLOB and CLOB support. Very substantial improvements in Unicode, allowing join of Unicode tables with EBCDIC and Ascii and converting DB2 catalog to Unicode.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/F01.pdf



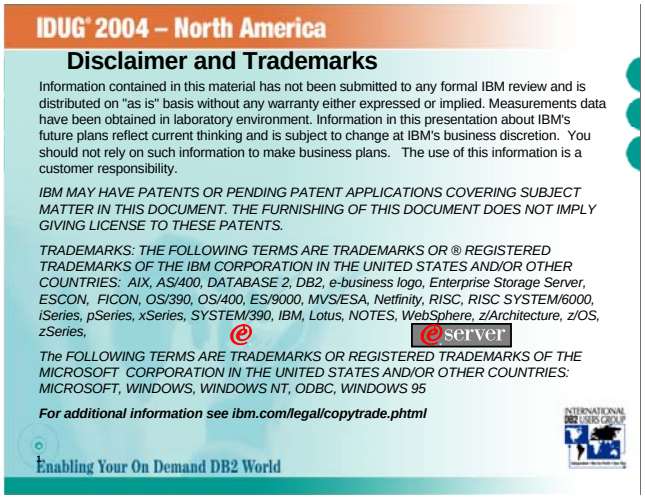
The key improvements for Unicode allow much more flexibility, with the ability to join a Unicode table to one that is ASCII or EBCDIC. SQL statements and literals can be Unicode or EBCDIC. Many of the DB2 catalog character columns will be converted to Unicode, so Unicode is for everyone. SQL is converted to Unicode before parsing to allow code-page dependent parsing.

The collating sequence for Unicode is similar to ASCII, with numbers sorting lower than letters. The lengths and maximum lengths of strings can change as they are converted to Unicode, so treat lengths as variable.

Version 8 is the twelfth and largest ever release of DB2 for z/OS. It brings new synergy with the zSeries hardware and uses the z/OS 64-bit virtual addressing capabilities. V8 improves data support, application development, and query function enhancements for e-business. It also builds on the traditional zSeries and DB2 characteristics of availability, exceptional scalability, and performance for the enterprise database management system of choice. V8 has been re-engineered for e-business on demand, with many fundamental changes in architecture and structure. Key improvements enhance scalability, application porting, security, architecture, and continuous availability. Management for very large databases is made much easier, while 64-bit virtual storage support makes management simpler and improves scalability and availability. This new version breaks through many old limitations in the definition of DB2 objects, including SQL improvements, online schema evolution, longer names for tables and columns, longer SQL statements, enhanced Java and Unicode support, enhanced utilities, more log data sets, more partitions, and many more advantages. Customers, vendors and consultants tell us that DB2 for z/OS Version 8 is exciting for them. It is a very important milestone.

Version 8 includes dozens of changes in SQL, improving family consistency in many cases, playing leapfrog in others, pushing DB2 SQL beyond current boundaries for enhanced application portability, open standards. Longer names for tables and columns mean that customers can use more meaningful names, matching standards. Longer SQL statements help with SQL that is generated or used in an SQL procedure. Here are some: multi-row INSERT, FETCH & UPDATE, GET DIAGNOSTICS, INSERT within SELECT, IDENTITY Column enhancements, SEQUENCES, CURRENT PACKAGE PATH, Dynamic Scrollable Cursors, Common Table Expressions, Scalar Fullselect, Materialized Query Tables, Unicode SQL, XML Publishing and much more. These changes improve our customer productivity, consistency across the DB2 family and ability to port applications. If you want to design or write applications for the entire DB2 family, then use the IBM DB2 Universal Database SQL Reference for Cross-Platform Development. Many barriers that limit our customers are removed: using 64 bit memory, providing consistent table and column name lengths, allowing 2 megabyte SQL statements, 4096 partitions, and three to ten times the log space. Customers encountered many limitations over the past 20 years, and lifting the limits required extensive reengineering for some. Other limits allow improved scalability and availability, such as tripling the active log size and ten times the archive logs. Allowing sixteen times the number of partitions allows a table to use one partition per day for eleven years.

Key performance enhancements deliver better family consistency and run many times faster. Being able to make database changes without an outage, such as adding a partition, is a breakthrough for availability. Alter your table and go, no need to drop and redefine. Online schema evolution is more resilient. The most important change for many customers is the ability to use ALTER in many places instead of needing to drop and redefine. We call this online schema evolution, and it can reduce outages by hours or days for a major database structure change. Database administrators can add a partition to an existing partitioned table space or rotate the partitions. Other changes in online schema evolution allow better partitioning and improved disk access, avoiding random access with more effective database designs. Many enhancements improve our integration with zSeries, z/OS and with key vendor applications like PeopleSoft, SAP and Siebel.



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.