This is the application programming aspects of DB2 9 for z/OS. For detailed notes, please see the redbooks.

Disclaimer/Trademarks

The information in this presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available information and documents presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available information and documentsinformation and documentspecification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available information and documentsmade available by those vendors. The document represents the author's best view at the time of writing and is subject to change. The assumptions and IBM internal assessments in this document are based on information from press reports, industry consultants, and public statements. They represent the opinion of the authors, not the opinion of the IBM corporation.

REFERENCES IN THIS PUBLICATION TO IBM PRODUCTS, PROGRAMS, OR SERVICES DO NOT IMPLY THAT IBM INTENDS TO MAKE THESE AVAILABLE IN ALL COUNTRIES IN WHICH IBM OPERATES. able and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is bBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS able and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is based on publicly available presentation concerns new products that IBM may or may not announce. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification of some of the features described in this presentation may change before the General Availability date of these products. Discussion of competitive products is b COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

DB2 SQL

z z/OS V8
common
luw Linux, Unix & Windows V8.2

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, range partitioning, data 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

luw — Updateable UNION in Views, ORDER BY/FETCH FIRST in subselects & table expressions, GROUPING SETS, ROLLUP, CUBE, INSTEAD OF TRIGGER, EXCEPT, INTERSECT, 16 Built-in Functions, MERGE, Native SQL Procedure Language, SET CURRENT ISOLATION, BIGINT data type, file reference variables, SELECT FROM UPDATE or DELETE, multi-site join, MDC

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

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, 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 and fullselect, caseless comparisons, INTERSECT, EXCEPT, not logged tables**, **range partitioning, data compression**

luw

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

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). 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 for LUW, code named Viper is already generally available. We are able to move more from the 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.

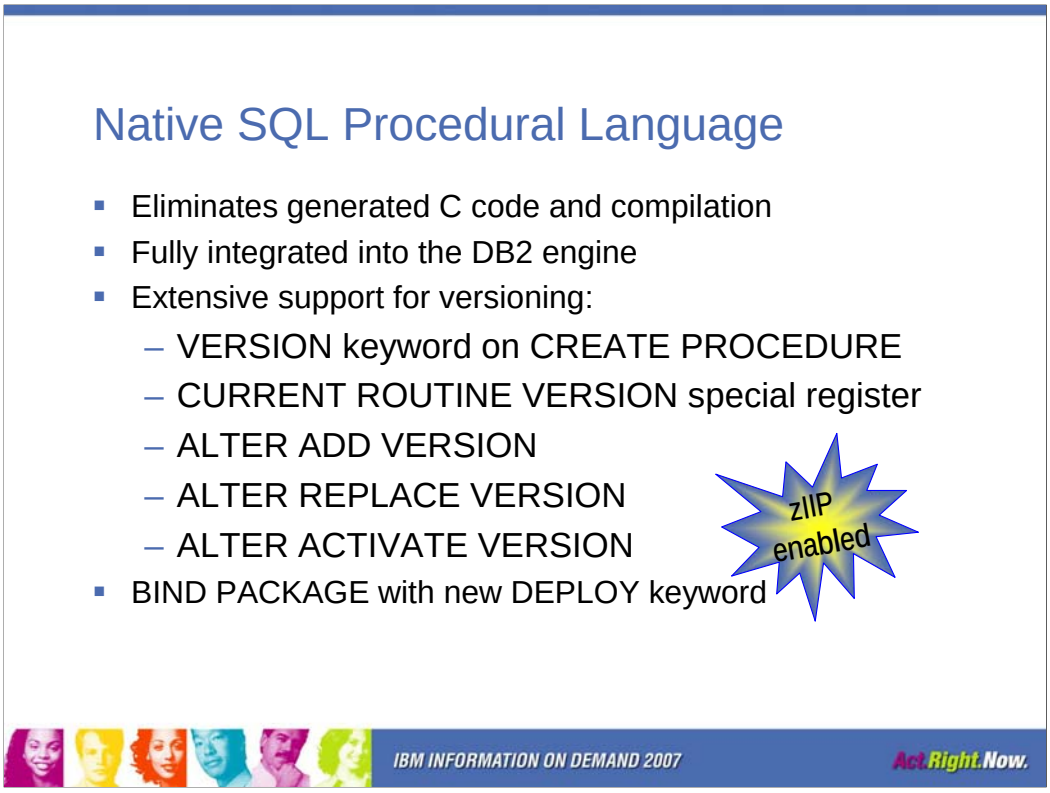
Sheryl Larsen provided the base for this information, but the mistakes are probably mine.

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). 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 for LUW, code named Viper is already generally available. We are able to move more from the 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.

Sheryl Larsen provided the base for this information, but the mistakes are probably mine.

This text just shows the direction for the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is beyond DB2 9 for z/OS and DB2 9.5 for LUW, code named Viper 2 is already generally available. We are able to move more from the z and from the luw list to the common list with changes to come.
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.
Sheryl Larsen provided the base for this information, but the mistakes are probably mine.



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. This is very similar to the changes in DB2 for luw V8.2.

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.

Native SQL stored procedures are eligible to have a portion run on zIIP processors when they are invoked from a remote client.

Optimistic Locking Support

- Built-in timestamp for each row or page
  - Automatically updated by DB2
  - Allows simple timestamp predicate to validate that row has not changed since last access
- Eliminates need for complex predicates on WebSphere CMP updates, improves performance

This change will help with migration of applications which use application consistency techniques. Application programmers will not need to be responsible for updating the timestamps. A simple timestamp predicate can be used to make sure that other updates are not lost.

TRUNCATE Statement

- Allows fast delete of all rows in a given table (segmented, partitioned or simple)
- Very useful for nightly refresh of summary tables, warehouses, etc.

```
TRUNCATE TABLE  TABLE-NAME

  < DROP STORAGE | REUSE STORAGE>

  < RESTRICT WHEN DELETE TRIGGERS  |
    IGNORE DELETE TRIGGERS>

  < IMMEDIATE>
```

This statement provides a fast way to delete rows with SQL, with better application portability. Truncate Table provides for the rapid removal of rows from a table. You can use this function to delete the contents of a table before applying data via LOAD or INSERT or MERGE.

TRUNCATE TABLE
customer requirements

- Delete rows from a table without firing DELETE triggers
  - triggers could be dropped and recreated . . . changes the table definition
- Have an option to LOAD REPLACE that works on a table level in a segmented table space with multiple tables

What TRUNCATE does

- Gives users an alternative way of emptying a table, with more flexibility over the current DELETE statement with no WHERE clause (i.e., a mass delete operation):
  - Delete all data rows in a designated DB2 table without activating DELETE triggers
  - DB2 catalog definition of the table (i.e., dropping and recreating of the delete triggers) is not needed for faster processing
  - Provides an option to allow the users to empty the designated DB2 table permanently without going through the current commit phase
  - Provides an option to reuse deallocated storage

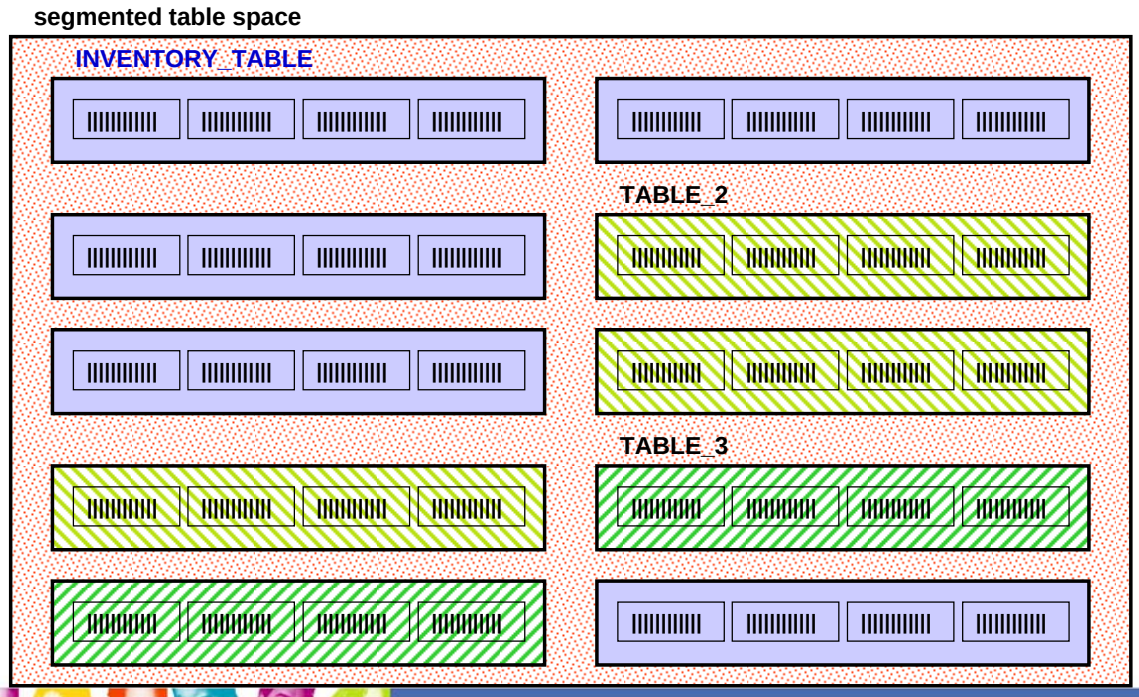
Processing modes for TRUNCATE

- *Normal way* - truncate operation must process each data page to physically delete data records from the page
  - table in a simple table space
  - table in a partitioned table space
  - any table with table attributes
    - CDC-enabled (Change Data Capture)
    - MLS-enabled (Multiple Level Security)
    - VALIDPROC-enabled

- *Fast way* truncate operation deletes data records without physically processing each data page
  - table in a segmented table space or a universal table space without the above table attributes

"Normal" way includes logging of all deleted rows

Storage --
-- 3 tables in segmented table space



INVENTORY_TABLE shares a segmented table space with two other tables, TABLE_2 and TABLE_3

TRUNCATE . . . IMMEDIATE

- **Specifies that the truncate operation is processed immediately and cannot be undone.**
- When IMMEDIATE option is specified, the table must not contain any uncommitted updates.
  - For a DGTT table object, the IMMEDIATE option does not apply to it. The truncate operation will fail since the table space contains a DGTT will be always in the update mode
  - No uncommitted DDL is allowed on the table prior to the TRUNCATE
- The truncated table is immediately available for use in the same unit of work.
- Although a ROLLBACK statement is allowed after the TRUNCATE statement, the truncate operation is not undone, and the table remains truncated. Other data changes following the TRUNCATE are rolled back.

MERGE

- Multi-row MERGE operation, using arrays
- Targets OLTP applications like SAP

```
MERGE  INTO account AS T
USING VALUES (:hv_id, :hv_amt) FOR 5 ROWS  AS S(id,amt)
ON T.id = S.id
WHEN MATCHED THEN
   UPDATE SET balance = T.balance + S.amt
WHEN NOT MATCHED THEN
   INSERT (id, balance) VALUES (S.id, S.amt)
NOT ATOMIC CONTINUE ON SQLEXCEPTION
```

Merge

Online transaction processing (OLTP) workloads that need to place data for several rows into a table will benefit from merge SQL. This capability will allow for updating of existing rows and creation of new rows through a single SQL statement.

Multi-row insert-type capability is extended, via the MERGE statement, to take an array of column values from a program and perform insert and update operations against a table. DB2 will use the matching criteria in the merge SQL to update existing rows and perform inserts for rows that do not exist, through a single SQL statement.

MERGE

- Combine UPDATE and INSERT operation to a target table or view, from a input source of host-variable-arrays modeled as a source table
  - When source rows match to target, update target rows from source
  - When source rows do not match to target, insert source rows into target

SQL Improvements – Family Compatibility

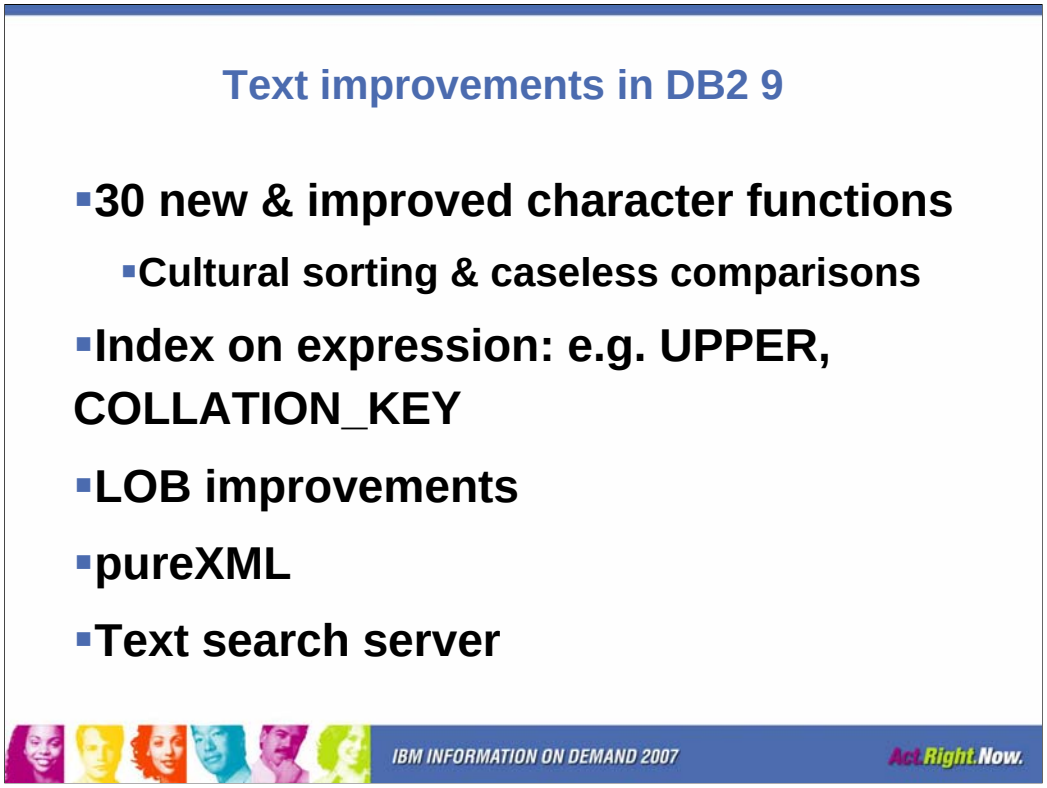
- INSTEAD OF triggers
- SELECT FROM UPDATE
- SELECT FROM DELETE
- SELECT FROM MERGE
- BIGINT, BINARY and VARBINARY data types
- ORDER BY and FETCH FIRST in subselect

Select from DELETE, UPDATE, and MERGE: The object-relational capabilities of DB2 allow for the incorporation of business logic into the database. This extends the power of SQL. Sometimes the application needs to know the results of this logic, when applied to the SQL issued. A subsequent SELECT for the data adds complexity and execution time to the application.

The *insert within select* feature of DB2 for z/OS Version 8 has been expanded to include the retrieval of columns from rows that are modified via DELETE, UPDATE, and MERGE SQL. One SQL call to DB2 modifies the table contents and returns the resultant changes to the application program.

When used with DELETE, the application now has the option to code a destructive read from a table. This is particularly useful when a table is used as a data queue, as with many vendor packages.

Subquery improvements: Correlated and non-correlated subqueries will benefit from improved optimization. They will provide added flexibility with the support of ORDER BY and FETCH FIRST clauses.



The text improvements in DB2 9 include many improvements and new functions which can provide fast search for text, including indexing, using the new index on expression with text functions like UPPER that is sensitive to locale, COLLATION_KEY, SOUNDEX and DIFFERENCE functions. Improvements for large objects and XML also help with many text applications.

Specialized text search with a separate server is noted on the next pages.

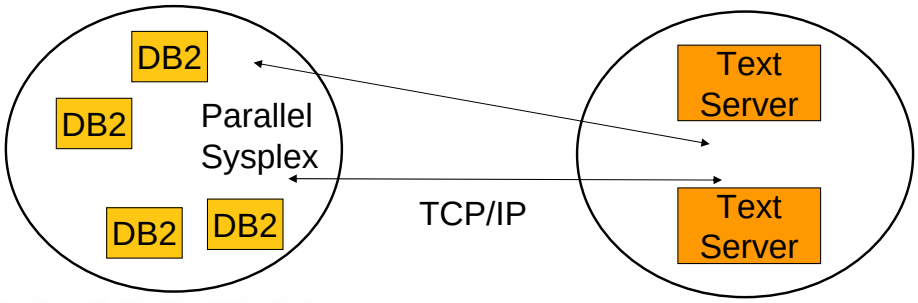
Text function improvements in DB2 9

- New built-in character functions: COLLATION_KEY,
  - ASCII_CHR, ASCII_STR, COLLATION_KEY
  - DIFFERENCE, EBCDIC_CHR, EBCDIC_STR
  - IS_IDENTICAL_TO, LEFT, LOCATE_IN_STRING
  - LPAD, NORMALIZE_STRING, OVERLAY
  - RIGHT, RPAD, SOUNDEX, UNICODE
  - UNICODE_STR, VARCHAR_FORMAT
  - XMLATTRIBUTES, XMLCOMMENT,
  - XMLDOCUMENT, XMLPARSE, XMLPI
  - XMLQUERY, XMLSERIALIZE, XMLTEXT
- Index on expression: e.g. UPPER, LOWER

The text improvements in DB2 9 include many improvements and new functions which can provide fast search for text, including indexing, using the new index on expression with text functions like UPPER that is sensitive to locale, COLLATION_KEY, SOUNDEX and DIFFERENCE functions.

Text Search Server (APAR coming)

- Text search for CHAR, VARCHAR, CLOB & XML columns
- Provide a text index server
- Efficient communication interaction with DB2 for z/OS
- Text indexes are persisted into DB2 tables for backup & recovery purposes



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.

DB2 9 Spatial Support (phase 2 APAR coming)

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
…

IBM INFORMATION ON DEMAND 2007 Act.Right.Now.

GIS = Geographic Information System. Aka Geospatial. E.g. Google Maps.
OGC = Open Geospatial Consortium

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 is anything in the real world that has an identifiable location, or anything that could be imagined as existing at an identifiable location. A 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.

DDL Porting Improvements

- Automatic selection of DATABASE and TABLESPACE when DDL omits these keywords
- Automatic CREATE of UNIQUE index for PRIMARY KEY
- Deprecated simple table space, default to segmented structure, partition by growth in NFM, segmented in CM

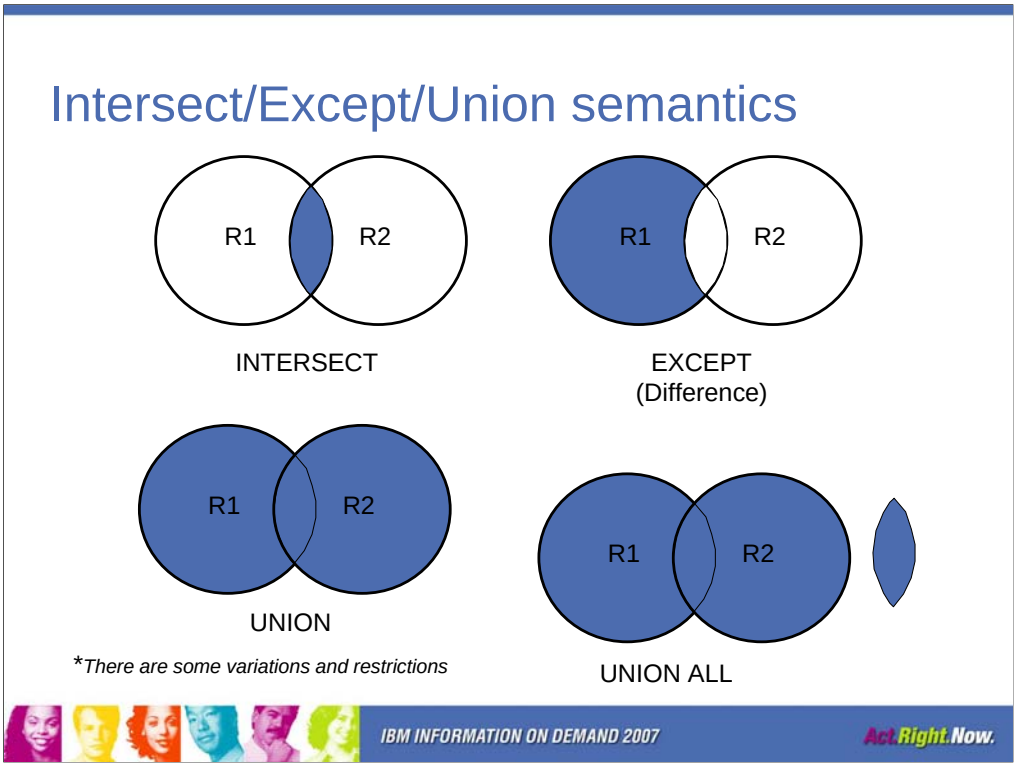
Additional family compatibility

Additional compatibility will be added for:

• Default databases and table spaces: When porting from other DBMS that do not have the same use for database and table space, providing them automatically will reduce the effort for delivering good performance.

• Automatic unique indexes are created to support defined primary keys.

•The default for a table space will change to segmented, improving performance and manageability for many customers. Existing simple table spaces are supported, but new table spaces will be segmented or partitioned.



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.

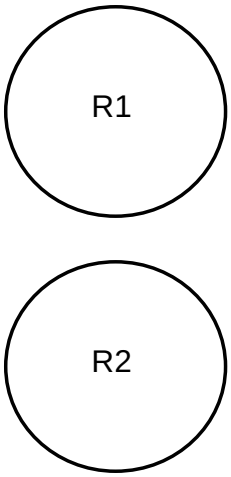
EXCEPT and INTERSECT Columns Participation

- Same as UNION today . . .
- R1 and R2 must have the same number of columns
  - data type for the n-th column of R1 must be compatible with the n-th column of R2
  - data type must not be CLOB, BLOB, DBCLOB, XML, or distinct type based on these types
- If the n-th column of R1 and the n-th column of R2 have the same name, then the n-th column of the result table has the same name; else unnamed
- Qualified column names cannot be used in the ORDER BY clause when the set operators are specified

These are the same characteristics as UNION / UNION ALL in DB2 V8

Sample query for following examples

```
SELECT LAST_NAME, FIRST_NAME, . . .
        FROM first_table
        WHERE . . .
UNION | INTERSECT | EXCEPT
SELECT LAST_NAME, FIRST_NAME, . . .
        FROM second_table
        WHERE . . .
```



Result set R1

LAST_NAME	FIRST_NAME		. . .
Castellini	Massimiliano	row 1	. . .
Castellini	Massimiliano	1	. . .
Castellini	Massimiliano	1	. . .
Cook	Ian	2	. . .
Cook	Ian	2	. . .
Cook	Ian	2	. . .
Crocker	Tom	3	. . .
Jones	Gareth	4	. . .
Jones	Gareth	4	. . .
Wilson	Mark	5	. . .

Result set R2 is similar . . . see next page

ORDER BY and FETCH FIRST in subselect

One customer has a huge table of which they want just the first 2000 rows sorted in a particular order. Unfortunately, the sort is done first, and the fetch first after. This would cause a huge sort for no reason. They had to code this using a temp table which is a lot more work than a simple select. The solution to this to allow FETCH FIRST n ROWS in subquery:

```
SELECT A, B, C FROM
   (SELECT A,B,C FROM TABLEA
    WHERE...
    FETCH FIRST 2000 ROWS ONLY ) AS TABLEB
ORDER BY C,B
```

Oracle and Microsoft allow this in some instances

DB2 for luw v7.2 has implemented these functionalities

In DB2 9

- Allow all semantically relevant clauses of the select statement to be pushed into subqueries. The original query can be taken as is and wrapped by more SQL, such as shown in the example above
- Provides more function by being able to select, e.g., the top n rows in a leg of a join, a leg of union, or a subquery.

```
(SELECT * FROM T1
  ORDER BY C1 FIRST 3  ROW ONLY)
UNION
SELECT * FROM T2
```

Restrictions

- A subselect that contains an ORDER BY or FETCH FIRST clause cannot be specified:
  - In the outermost fullselect of a view.
  - In a materialized query table
  - Unless the subselect is enclosed in parenthesis

Example:

```
CREATE VIEW V1 AS
  (SELECT * FROM T1 ORDER BY C1);
```

SQLCODE -20211

```
SELECT * FROM T1
  ORDER BY C1
UNION
SELECT * FROM T2
  ORDER BY C2
```

SQLCODE -104

Second example . . . if the two SELECT parts were in parentheses, it would be OK

New data types

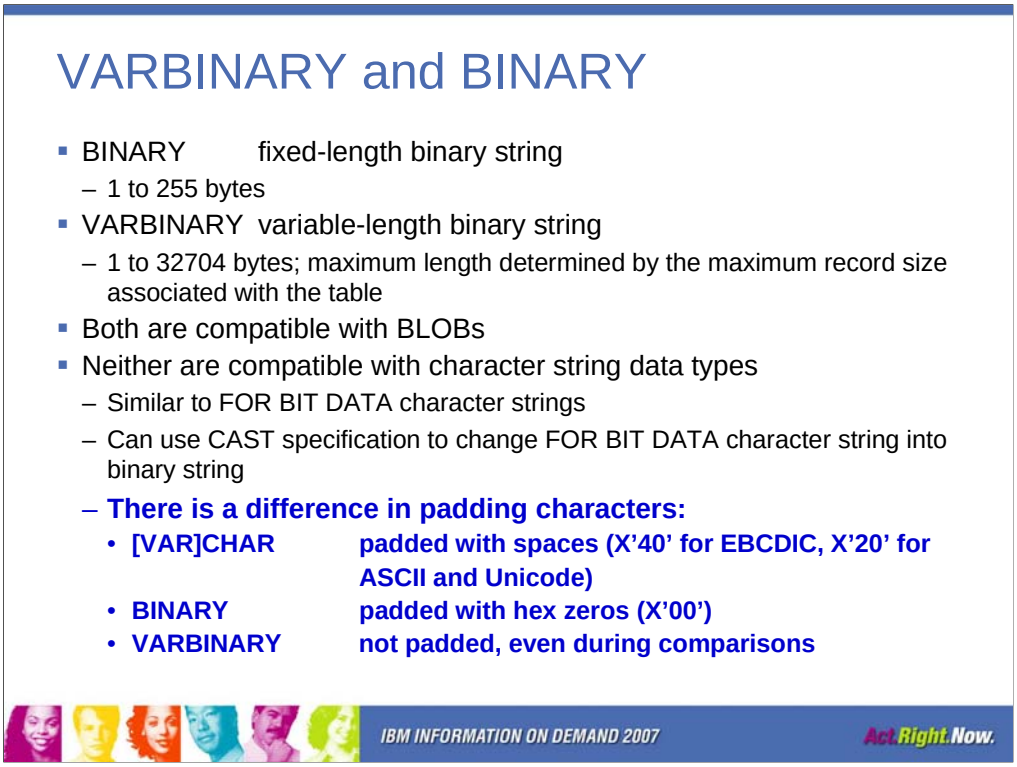
- BIGINT
- BINARY and VARBINARY
- DECFLOAT

BIGINT

- An exact numeric capable of representing 63-bit integers
  - Range:
    - -9223372036854775808 *to*
    - 9223372036854775807
- Compatible with all numeric types

BIGINT extensions to existing functions

- CHAR
- DIGITS
- LENGTH
- MOD
- MULTIPLY_ALT
- POWER
- VARCHAR



VARBINARY -- max size -- see “Maximum record size” on the description of CREATE TABLE

Line Item 625

DK015 DB2 ODBC support for BIGINT, BINARY and VARBINARY

Comparison of binary strings

- 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
  - even when the remaining bytes in the longer string are hex zeros

Hex value of operand 1	relationship	Hex value of operand 2
X'4100'	<	X'410000'
X'4100'	<	X'42'
X'4100'	=	X'4100'
X'4100'	>	X'41'
X'4100'	>	X'400000'

Functions extended for VARBINARY and BINARY

- INSERT
- LEFT
- LTRIM
- POSSTR; POSITION does not support binary types
- REPEAT
- REPLACE
- RIGHT
- RTRIM
- STRIP
- SUBSTR

Online schema and binary columns

- **A column data type could be altered only to a compatible data type.**
- However, 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 will be allowed (even though they are not considered to be compatible).
- When a CHAR FOR BIT DATA, or VARCHAR FOR BIT DATA column is altered to a BINARY or VARBINARY data type, and there is an index defined on that column, the index will be put in RBDP.
- Altering BINARY or VARBINARY data types to CHAR FOR BIT DATA or VARCHAR FOR BIT DATA will not be allowed.

Caution!!

- Caution should be taken when a CHAR FOR BIT DATA column is altered to a BINARY data type due to differences in padding.
- When a CHAR FOR BIT DATA column is altered to BINARY, the existing space characters in the table will not be changed to hexadecimal zeros (X'00). In addition, if the new length attribute is greater than current length attribute of the column, the values in the table are padded with hexadecimal zeros (X'00).

```
CREATE TABLE T1 ( C1 CHAR(5) FOR BIT DATA) CCSID EBCDIC;
INSERT INTO T1 VALUES(X'C1C2C3');
INSERT INTO T1 VALUES(X'C1C2C3C4C5');
COMMIT;

SELECT HEX( C1 ) FROM T1;
        returns:        C1C2C34040
                        C1C2C3C4C5
```

Decimal Floating Point

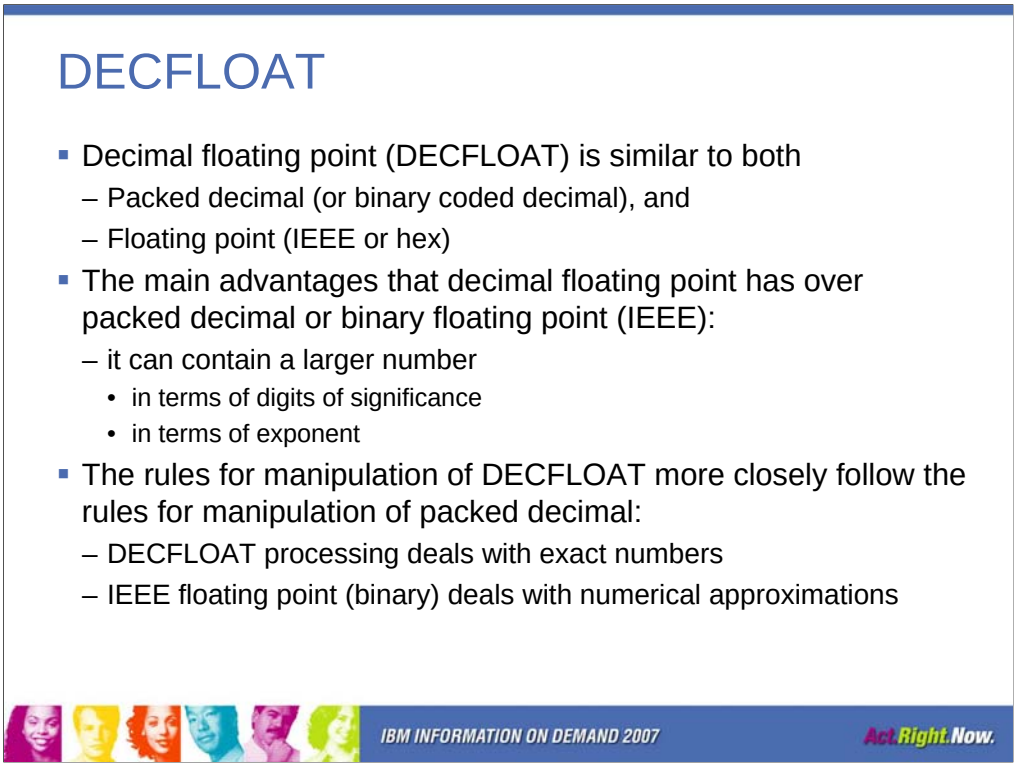
- New datatype DECFLOAT
  - –Well suited to typical customer financial calculations
  - –Similar to “calculator” mathematics
    - •Eliminates rounding errors by using base 10 math
    - •Has up to 34 digits of precision
    - •Floating point convenience with fixed point precision!!!
  - –Hardware support in System z9 (new IEEE standard)
    - •Software emulation provided for other models

New data type support is provided. Support is added for:

• Decimal floating point numbers, similar to calculator mathematics and supporting the IEEE standard. These numbers can have more precision than current floating point.

• BIGINT support of double word (8 byte) integer values

• VARBINARY, providing better comparison support for binary strings



LI 632

Requirement MR0518045946

Title Decimal Floating point data type

Description Provide users with IEEE format of floating point numbers which support larger number (in term of exponent) than the IBM 370 floating point format.

The support of larger number will be satisfied with decimal floating point data type support.

Customer Value

DECFLOAT details

- All numbers have a sign, a precision, and a scale
- The precision is the total number of decimal digits excluding the sign (either 16 or 34 for DECFLOAT)
- The scale is the total number of decimal digits to the right of the decimal point
- An exponent range of respectively 10^{-383} to 10^{+384} or 10^{-6143} to 10^{+6144}
- DECFLOAT(16):
  - Negative values: -9.999999999999999 x 10^{+384} to -1.000000000000000 x 10^{-383}
  - Positive values: 1.000000000000000 x 10^{-383} to 9.999999999999999 x 10^{+384}
- DECFLOAT(34):
  - Negative values: -9.999999999999999999999999999999999 x 10^{+6144} to -1.000000000000000000000000000000000 x 10^{-6143}
  - Positive values: 1.000000000000000000000000000000000 x 10^{-6143} to 9.999999999999999999999999999999999 x 10^{+6144}

Equality for DECFLOAT

- The DECFLOAT data type allows for multiple bit representations of the same number. Additionally, numbers with the same coefficient can have different exponents, and therefore different bit representations.
- For example 2.00 and 2.0 are two numbers with the same coefficient, but different exponent values.
- Thus, 2.00 <> 2.0 at a binary level, however the = (equal) predicate will return true for a comparison of 2.0 = 2.00. Given that 2.0 = 2.00 (the comparison is true), 2.0 < 2.00 is false. The behavior that is described here holds true whenever DB2 compares DECFLOAT data (such as for UNION, SELECT DISTINCT DECFLOAT_column, COUNT(DISTINCT DECFLOAT_column), basic predicates, IN predicates, etc)
- Example:

```
SELECT 2.0 FROM SYSIBM.SYSDUMMY1
UNION [DISTINCT]
SELECT 2.00 FROM SYSIBM.SYSDUMMY1          yields 1 row
```

Language support for DECFLOAT

- The following languages are currently supported:
  - Java
  - Assembler
  - REXX
- See the Reference Material, DB2 manuals for details

INSTEAD OF triggers: current problem and goal

- 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)

INSTEAD OF triggers

- 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

Example

```
CREATE TABLE WEATHER (CITY VARCHAR(25), TEMPF DECIMAL(5,2))
CREATE VIEW CELCIUS_WEATHER (CITY, TEMPC) AS
      SELECT CITY, (TEMPF-32)*5.00/9.00 FROM WEATHER

CREATE TRIGGER CW_INSERT INSTEAD OF INSERT ON
       CELCIUS_WEATHER
REFERENCING NEW AS NEWCW DEFAULTS NULL
FOR EACH ROW MODE DB2SQL
   INSERT INTO WEATHER VALUES (NEWCW.CITY,
                               9.00/5.00*NEWCW.TEMPC+32)

CREATE TRIGGER CW_UPDATE INSTEAD OF UPDATE ON
       CELCIUS_WEATHER
REFERENCING NEW AS NEWCW OLD AS OLDCW DEFAULTS NULL
FOR EACH ROW MODE DB2SQL
   UPDATE WEATHER AS W
     SET W.CITY = NEWCW.CITY,
         W.TEMPF = 9.00/5.00*NEWCW.TEMPC+32
     WHERE W.CITY = OLDCW.CITY
```

Restrictions

- Only 1 INSTEAD OF INSERT, UPDATE, DELETE per view
- View cannot be symmetric (i.e., no WHERE clause)
- Only has row granularity
- No WHEN clause
- Cannot specify UPDATE OF column list
- Cannot change transition variables
- Does not work with position UPDATE / DELETE
- No LOB, XML

- SELECT FROM UPDATE/DELETE/INSERT not supported
- MERGE into a view with INSTEAD OF trigger is not supported

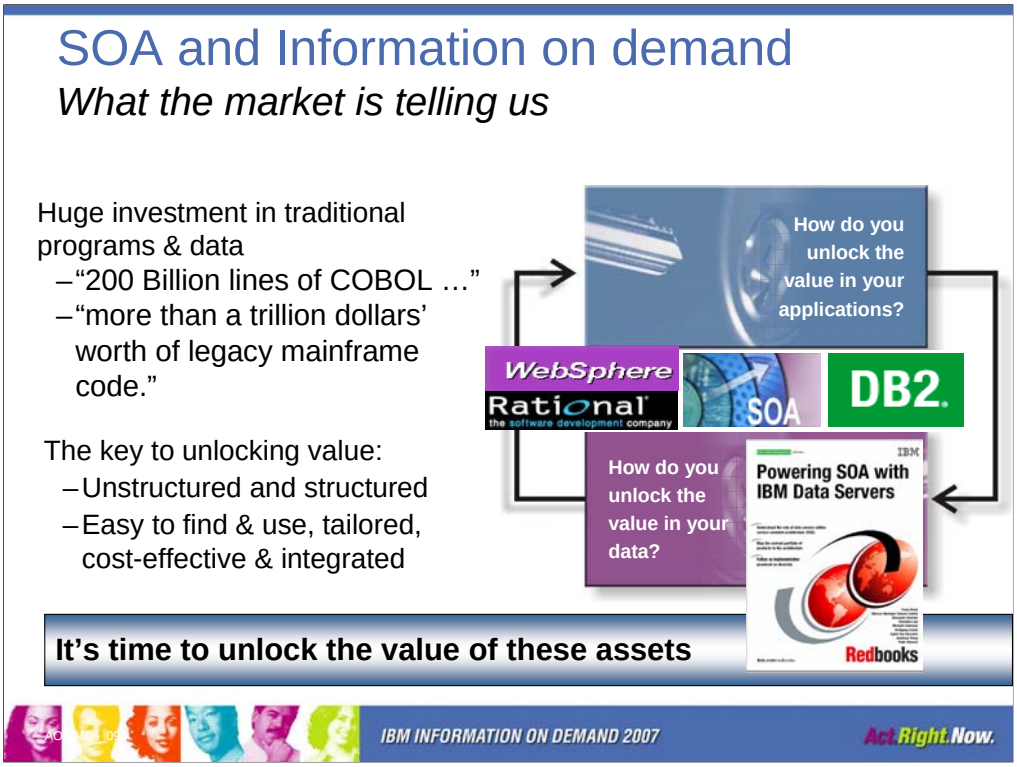
LOB Function

- SQL
  - File reference variables
  - FETCH CONTINUE
  - Automatic object creation
- Utilities
  - REORG reclaim fragmented space and improve access performance
  - REORG share level reference (read only)
  - Online CHECK LOB & CHECK DATA
  - Sample unload DSNTIAUL

Large object improvements: Large objects (LOBs) were introduced in DB2 Version 6. Usage has increased substantially in the past few years, and major enhancements have been made in DB2 Version 8.

APARs on Version 8 deliver the ability to use utilities for loading and unloading large LOB data. File reference variables are used to let the large objects be accessed from data sets instead of from storage. The abilities to reorganize and to recover space are provided.

Future changes will help with improved function and usability, DB2 family compatibility, cost of ownership, performance, and scalability.



Significant investment in traditional assets: "200 Billion lines of COBOL code in existence" eWeek May 28th 2005

"Mainframe users are sitting on more than a trillion dollars' worth of legacy mainframe code." Computerworld, 2006

"Majority of our customer data still on mainframes"

Don Greb, Mellon Financial Corp from Computerworld 2002

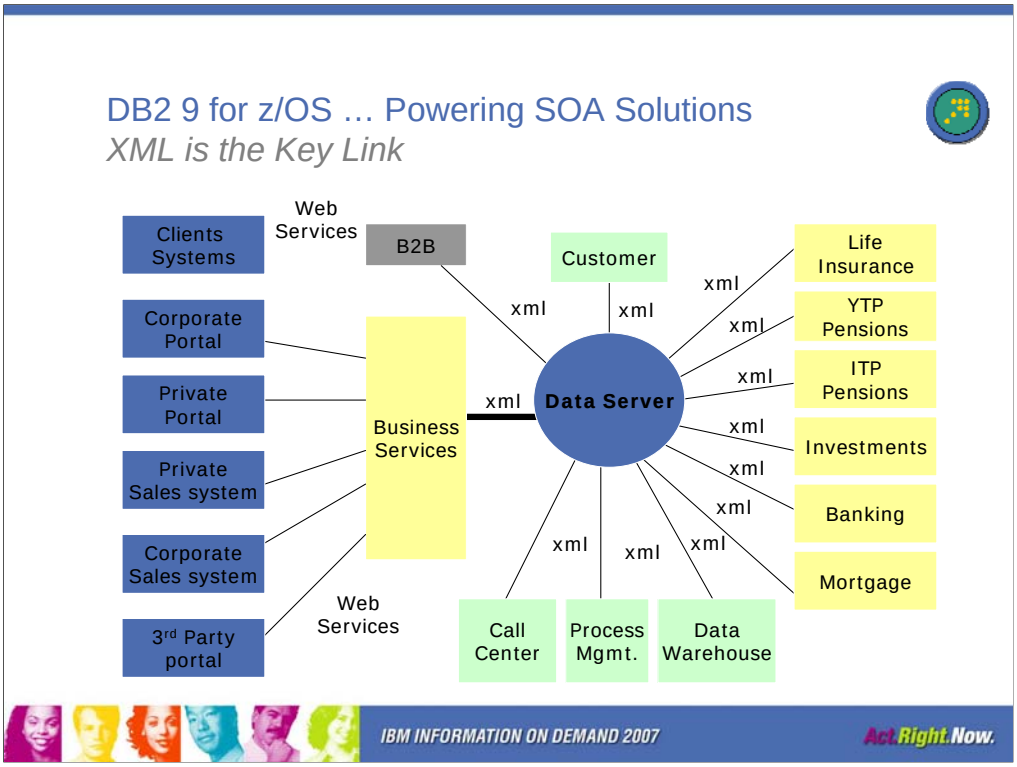
The key to unlocking the value is often in the WebSphere, Rational and DB2 portfolios:

85% of information is unstructured. 48 disparate financial systems and 2.7 ERP systems in the average $1B company. Only 1/3 of CFOs believe that the information is easy to use, tailored, cost-effective or integrated.

30% of people's time is spent searching for relevant information

Sources: IBM & Industry Studies, Customer Interviews

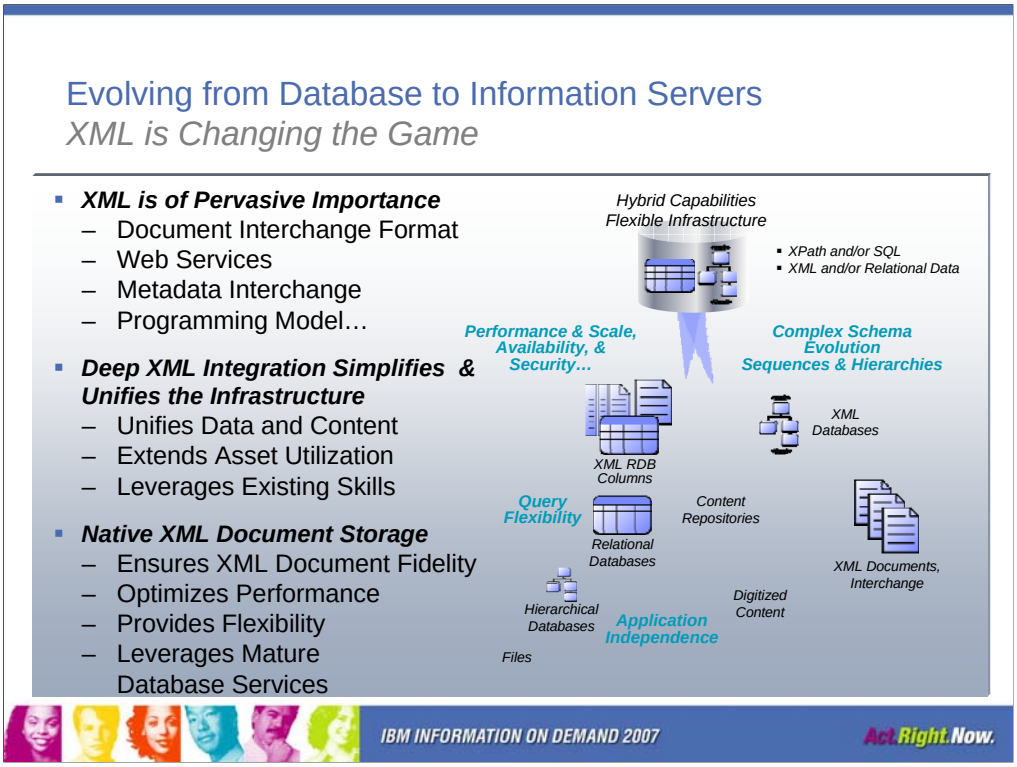
For more on SOA and how DB2 works in the services oriented architecture, see the recent redbook, Powering SOA with IBM Data Servers, SG24-7259. http://www.redbooks.ibm.com/abstracts/SG247259.html



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.

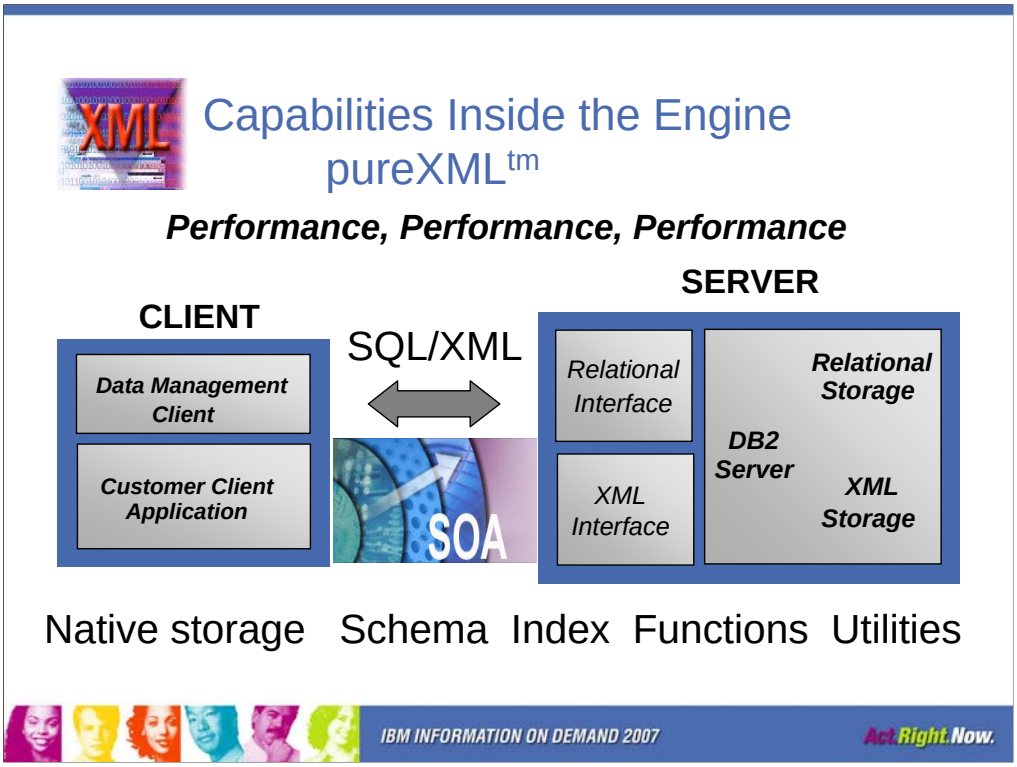
And a large Insurance client has seen the impact of pureXML to development time and cost with a 65% reduction in lines of code and more than 75% reduction in time required to develop services accessing XML data.



XML provides another model of information, and applications have embraced it. Companies, consortia and standards use XML for self-describing information, especially if there is a complex structure. Key point: 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)

IBM introduced a new generation data server with the availability of DB2 9. 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.



Powerful querying and transformation capabilities
Querying is merely finding a document that matches a certain criteria. For e.g., find all purchase orders with a certain order id. More often than not, users want to transform the data into something more relevant. For example, they may want to not return all purchase orders as-is, but instead return documents that only contain the purchase order id, and the amount of the order. DB2 already provides great support for the first two paradigms. In fact, with our XML extender, we were the first ones in the industry to support XML. We're working on completing the picture, across the DB2 family, with XPath. DB2 9 for Linux, UNIX and Windows delivers this picture, and add an XQuery interface to the data.

Interface Overview

- **Data Definition**
  create table dept(deptID int, deptdoc xml);
- **Insert**
  insert into dept(deptID, deptdoc) values (?,?)
- **Index**
  create index xmlindex1 on dept(deptdoc)
  generate key using xmlpattern '/dept/name' as sql varchar(30);
- **Retrieve**
  select deptdoc from dept where deptID = ?
- **Query**
  select deptID, xmlquery('/dept/name' passing deptdoc)
  from dept where deptID <> "PR27";

SQL/XML – work in both worlds

- Full power of SQL to address structured fields
- Join XML documents and tables
- SQL and XML Predicates
- Create XML from structured fields
- Materialize tables from XML documents


```
select d.deptID , u.headcount, xmlquery('$deptdoc/dept/name' passing d.deptdoc as
"deptdoc")
      from dept d, unit u
      where d.deptID = u.unitID and u.headcount > 200
      and xmlexists('$deptdoc/dept[@bldg = $b]' passing d.deptdoc as "deptdoc",
      u.bldg as "b")
      and xmlexists('$deptdoc/dept/employee/name' passing d.deptdoc as "deptdoc")
```

SQL/XML – XML Document Publishing

FIRSTNAME	LASTNAME	DEPARTMENT
SEAN	LEE	A00
MICHAEL	JOHNSON	B01
VINCENZO	BARELLI	A00

```
SELECT
   XMLELEMENT (NAME "Department",
   XMLATTRIBUTES (e.department AS "name" ),
   XMLAGG ( XMLELEMENT(NAME "emp", e.firstname)))  AS "department_list"
FROM employee e GROUP BY  e.department;
```

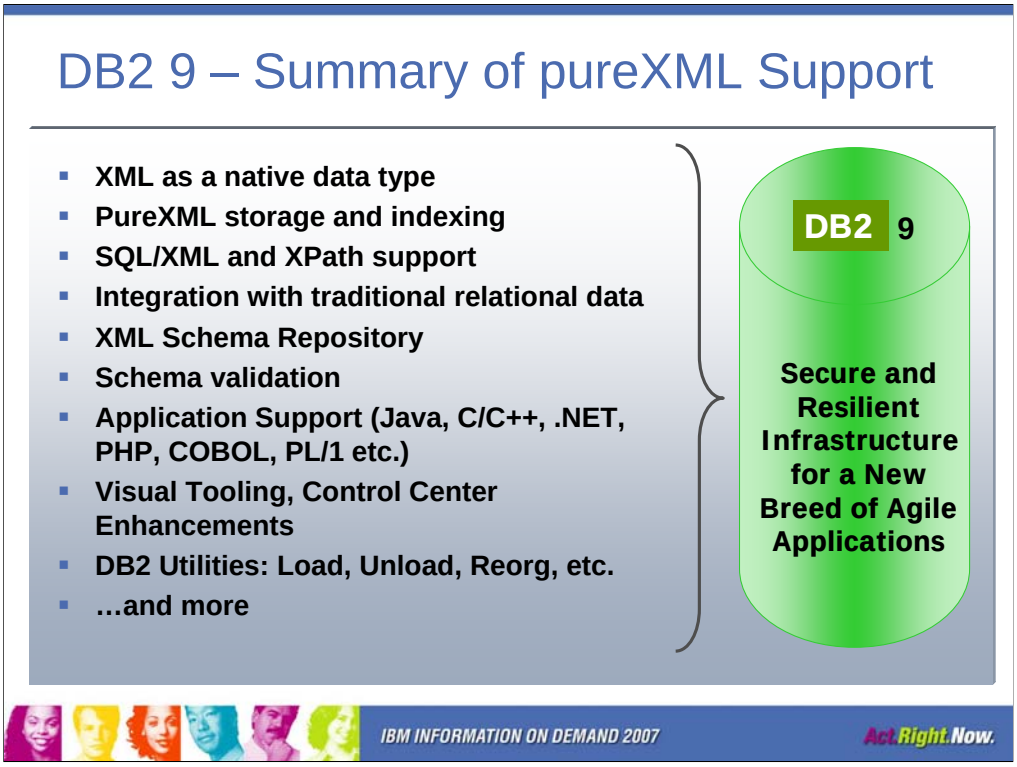
department_list

```
<Department name="A00">
    <emp>VINCENZO </emp>
    <emp>SEAN</emp>
</Department>
<Department name="B01">
    <emp>MICHAEL</emp>
</Department>
```

API Support

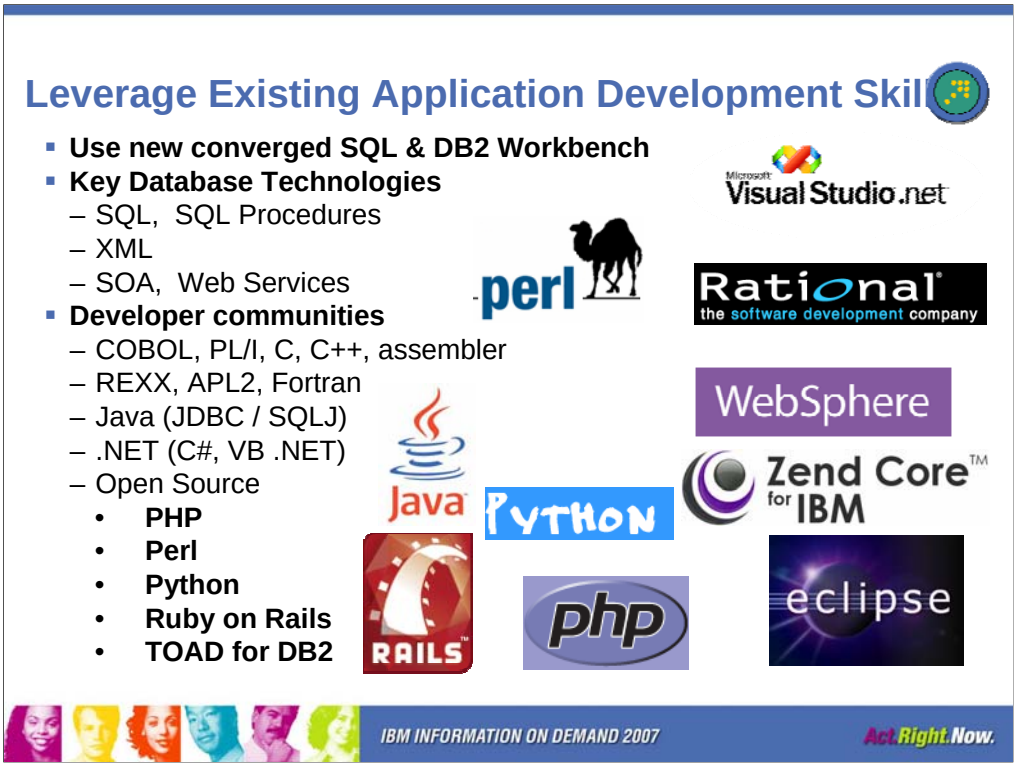
- XML type is supported in
  - Java (JDBC, SQLJ), ODBC,
  - C/C++, COBOL, PL/I, Assembly
  - .NET
- Applications use:
  - XML as CLOB(n)
  - XML as DBCLOB(n)
  - XML as BLOB(n)
  - All character or binary string types are supported
- XMLParse and XMLSerialize apply (implicitly or explicitly)

DRDA supports internally encoded or externally encoded XML type (in serialized string format).



XML is implemented as a native data type in DB2 9. XML data is stored as a parsed tree for the hierarchical structure. The trademarked term for this improvement is pureXML, with a native format for storage and the ability to have an XPath index. Both SQL access through SQL/XML with new functions and XPath query support are provided in DB2. At the same time, SQL support permits DB2 to have relational data in the same table and to integrate with other traditional relational data. The key point is that DB2 does not need to create new islands of XML data to have first class support for XML. DB2 can read the XML schema and put the result into relational tables. DB2 can validate the schema.

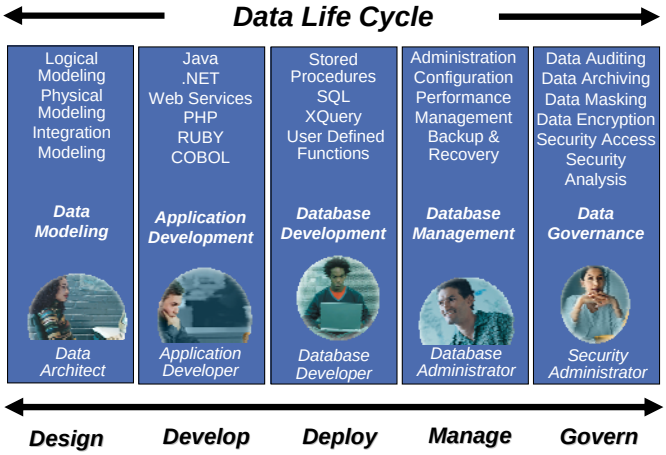
Application support is already provided in many languages, including Java, COBOL, C, C++, .NET, PHP and PL/I. Visual tooling is provided in the Developer Workbench and other products. A full set of DB2 utilities is provided to minimize support costs.

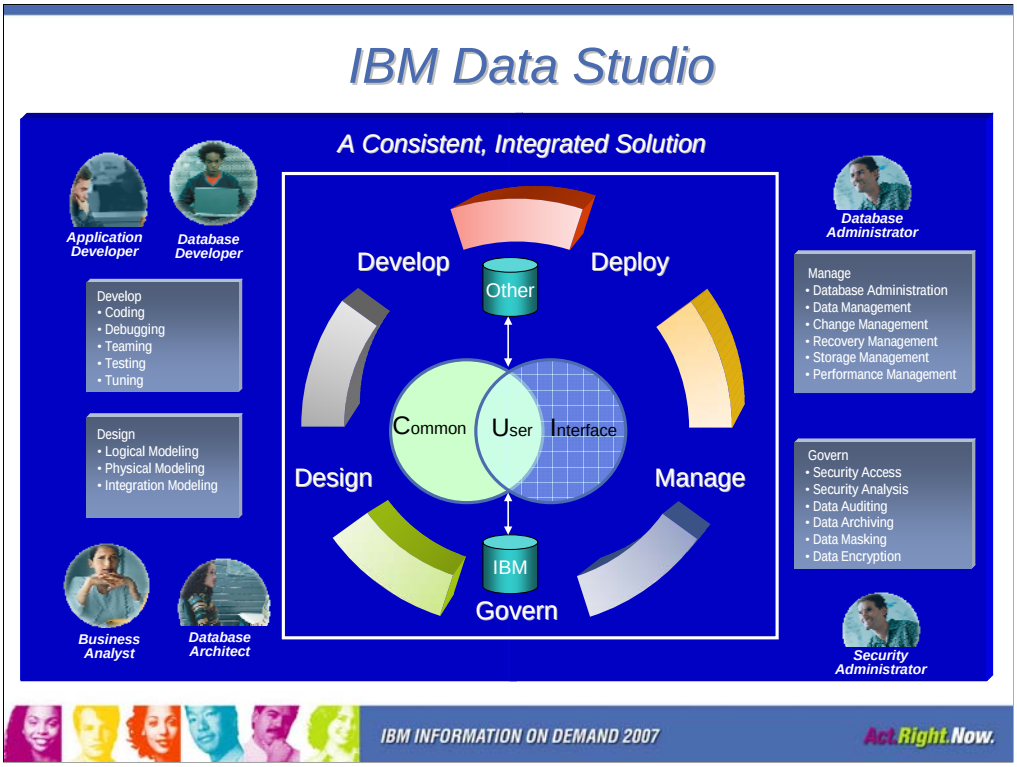


Application programming is using a wider range of tools, environments and languages. The Eclipse framework is growing strongly. We need to connect the new languages and environments to the scale and value of the existing infrastructure. The Rational and WebSphere product lines provide part of the connection, with products like Rational Data Architect and WebSphere Information Integration. DB2 clients provide more support for new environments and new languages.

IBM Data Studio

Who will use it ?





Slide objective: set up for onion peel. First we give you a comprehensive look at the total Consul portfolio

Points:

1. Consul provides the depth of security management on the mainframe, right side of the diagram and the breadth across the enterprise with comprehensive compliance management from the distributed environment on the left side of the diagram
2. Distributed portfolio is InSight
    1. Focus is on access monitoring, log management and compliance reporting
3. Mainframe portfolio is zSecure
    1. Focus is on integrated mainframe audit, monitoring, compliance and administration
    2. z/OS
    3. RACF
4. The intersection is the overlap and illustrates how Consul integrates mainframe data into InSight to provide the comprehensive compliance management view across the enterprise

IBM Data Studio v1.1

- Empowering developers and database administrators
- Complimentary and available in October of 2007
- Support for DB2 on all platforms and IDS

DB2 for LUW	DB2 for z/OS	DB2 for i5/OS	IDS
Physical Data Modeling	Physical Data Modeling	Physical Data Modeling	Physical Data Modeling
Data Distribution Viewer	Data Distribution Viewer	Data Distribution Viewer	Data Distribution Viewer
Integrated Query Editor	Integrated Query Editor	Integrated Query Editor	Integrated Query Editor
SQL Builder	SQL Builder	SQL Builder	SQL Builder
SQL Routine Debugger	SQL Routine Debugger	SQL Routine Debugger	XML Editor
Java Routine Debugger	Java Routine Debugger	Java Routine Debugger	XML Schema Editor
XML Editor	XML Editor	XML Editor	pureQuery for Java
XML Schema Editor	XML Schema Editor	XML Schema Editor	Data Web Services
pureQuery for Java	pureQuery for Java	pureQuery for Java	Object Management
Data Web Services	Data Web Services	Data Web Services	Data Management
Object Management	Object Management	Object Management	Security Access Controls
Data Management	Data Management	Data Management	Project Management
Update Statistics	Update Statistics	Security Access Controls	
Health Monitoring *	Visual Explain	Project Management	
Visual Explain	Security Access Controls		
Security Access Controls	Project Management		
Project Management			

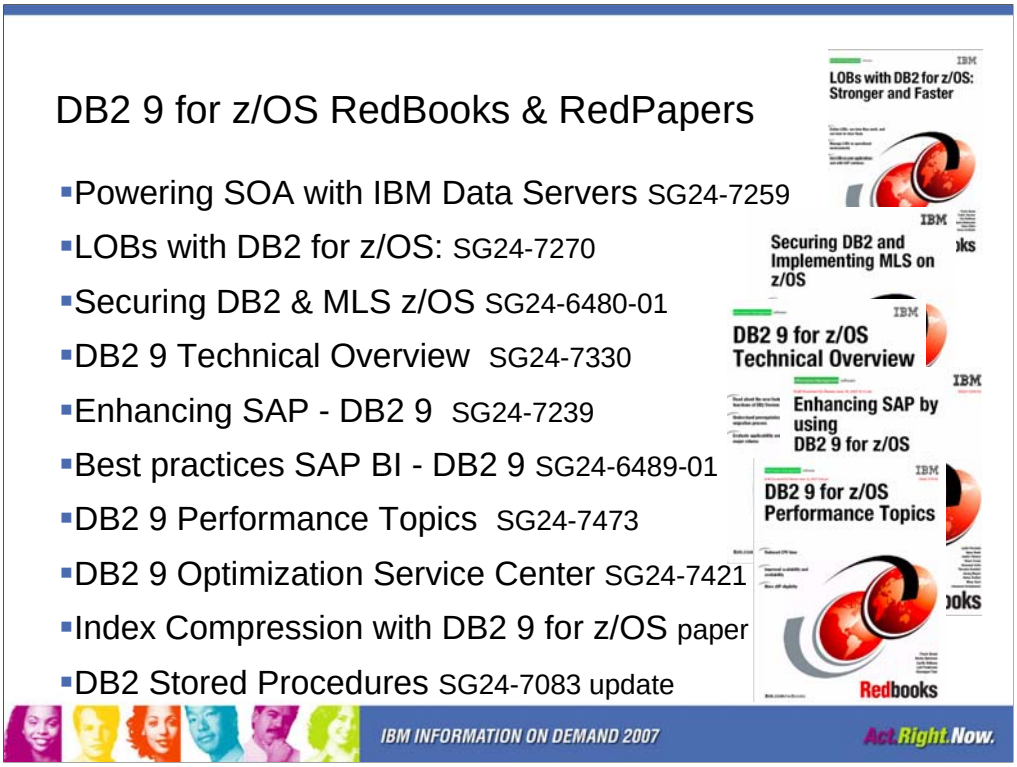
* Technical Preview

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

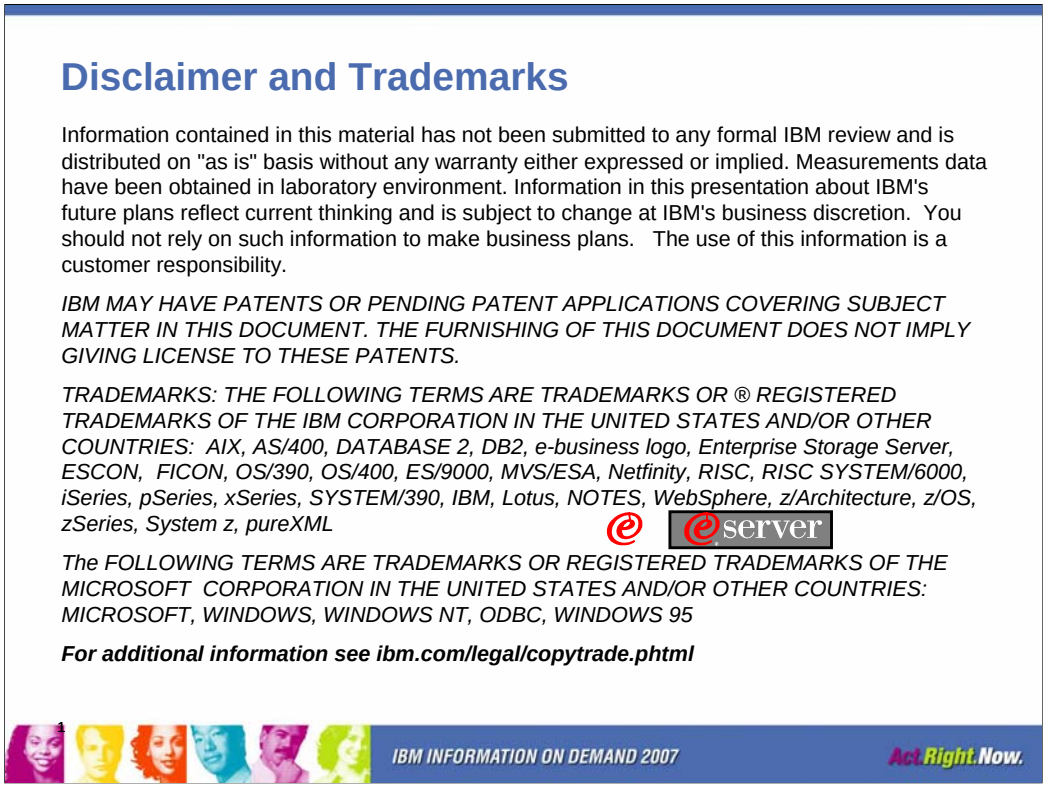
As in Version 8, there are many improvements for SQL and for XML in DB2 9. Improvements in the SQL have made migrating from other platforms, such as Unix and Windows much easier.

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.



See the DB2 library for detailed information.
http://www.ibm.com/software/data/db2/zos/library.html
Six redbooks with substantial DB2 9 content are on the web, with two drafts being reviewed, one being written and and one more in the works for later this year.

- Powering SOA with IBM Data Servers, SG24-7259
http://www.redbooks.ibm.com/abstracts/SG247259.html
- LOBs with DB2 for z/OS: Stronger & Faster SG24-7270
http://www.redbooks.ibm.com/abstracts/SG247270.html
- Securing DB2 & MLS z/OS, SG24-6480-01
http://www.redbooks.ibm.com/abstracts/sg246480.html
- DB2 9 Technical Overview, SG24-7330
http://www.redbooks.ibm.com/abstracts/SG247330.html
- Enhancing SAP - DB2 9, SG24-7239,
http://www.redbooks.ibm.com/abstracts/SG247239.html
- Best practices SAP BI - DB2 9, SG24-6489-01,
http://www.redbooks.ibm.com/redpieces/abstracts/sg246489.html
- DB2 9 Performance Topics, SG24-7473,
http://www.redbooks.ibm.com/abstracts/SG247473.html
- Index Compression with DB2 9 for z/OS, redpaper REDP4345,
http://w3.itso.ibm.com/abstracts/redp4345.html?Open
- DB2 9 Optimization Service Center, SG24-7421,
http://www.redbooks.ibm.com/abstracts/SG247421.html
- DB2 for z/OS Stored Procedures: CALL & Beyond SG24-7083-01 update later



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.

This is the Visual Explain web page. It includes some documentation and presentation material. DB2 Visual Explain can be downloaded for no additional charge and works with both V7 and V8. See the web for much more information:

http://www.ibm.com/software/data/db2/zos/downloads/ve.html

DB2 9 has a newer version of this application, with many changes, called Optimization Service Center (OSC) and a tool Optimization Expert (OE). We expect to have this function available for V8 within the next few months.

http://www.ibm.com/software/data/db2/zos/downloads/osc.html