DB2® UDB Version 8 for z/OS SQL Enhancements

Charles Lewis
Technical Sales Specialist
DB2 UDB for z/OS
lewisc@us.ibm.com

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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.

This information contains 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.

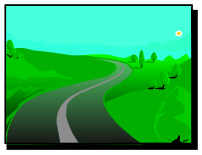
COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

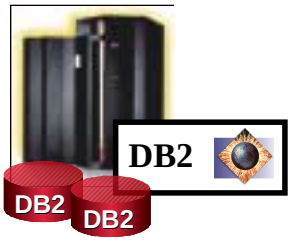
For additional information see ibm.com/legal/copytrade.phtml

Version 8 themes in DB2® for z/OS

- ✓ **Reengineering & Renaissance**
- ✓ **Breaking through Limitations**
- ✓ **SQL Function & DB2 family**
- ✓ **Performance Enhancements**
- ✓ **Continuous Availability**
- ✓ **Indexing Improvements**
- ✓ **Very Large Database**
- ✓ **WebSphere & Java**
- ✓ **SAP, PeopleSoft & Siebel**

Reengineered for e-business



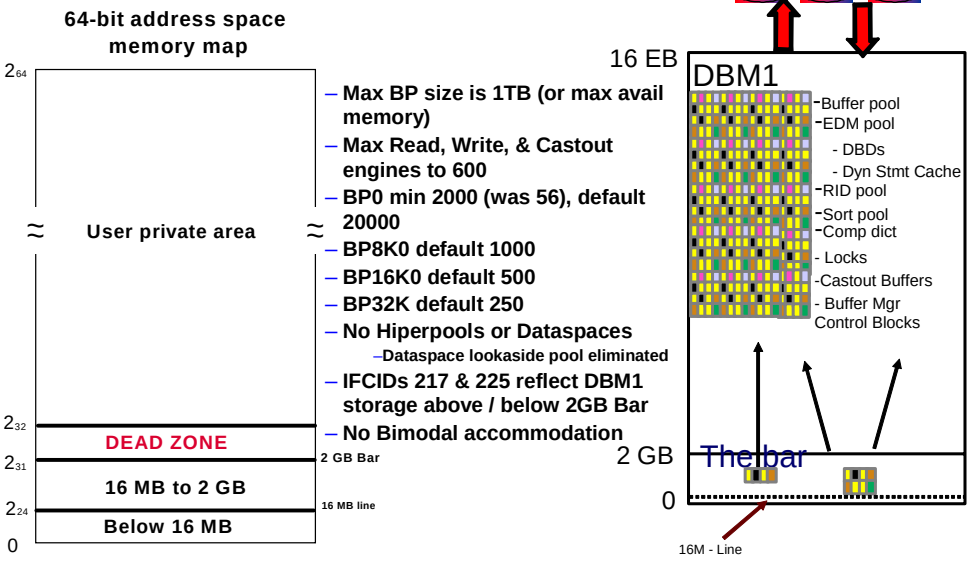


DB2 for z/OS limits: Breaking through limitations

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

– Virtual storage 2 GB	2^{31}	to	2^{64}
– Table, view, alias names	18	to	128
– Column names	18	to	30
– Partitions	254	to	4096
– SQL statement length	32 KB	to	2 MB
– Index key size	255	to	2000
– Character literals, hex	255	to	32704
– Tables in a join	15	to	225
– Active logs	31	to	93
– Archive logs	1000	to	10,000
– Special registers	Varying		
– ...			

64 bit Evolution (Virtual Storage Relief)



- Max BP size is 1TB (or max avail memory)
- Max Read, Write, & Castout engines to 600
- BP0 min 2000 (was 56), default 20000
- BP8K0 default 1000
- BP16K0 default 500
- BP32K default 250
- No Hiperpools or Dataspaces
  - Dataspace lookaside pool eliminated
- IFCIDs 217 & 225 reflect DBM1 storage above / below 2GB Bar
- No Bimodal accommodation

SQL function and DB2 Family

- Support for long names
- Multi-row INSERT and FETCH
- GET DIAGNOSTICS
- INSERT within SELECT
- IDENTITY column enhancements
- SEQUENCES
- Dynamic scrollable cursors
- CURRENT PACKAGE PATH
- Common table expressions
- SCALAR fullselect
- SQL statements can be up to 2 MB
- ... and much more

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

Multi-row INSERT - 1

- **Inserts multiple rows on one API call**
- **Can be**
  - ATOMIC or
  - NOT ATOMIC CONTINUE ON SQLEXCEPTION
- **Can be static or dynamic SQL**
- **Significant performance boost**

```
INSERT INTO T1 FOR :hv ROWS
    VALUES (:ARRAY1, :ARRAY2) ATOMIC;
```


Multi-row INSERT - 2

- **Multi-row INSERT in dynamic SQL**

```
STMT = 'INSERT INTO T1
            VALUES ( ?, ?)
            FOR MULTIPLE ROWS ATOMIC;

PREPARE S1 FROM STMT;

EXECUTE S1 FOR :HV ROWS
            USING :ARRAY1, :ARRAY2;
```

Multi-row FETCH - 1

- **Returns multiple rows on one API crossing**
- **"wide" cursor with potentially locks on multiple rows**
- **Supports scrollable and non-scrollable, static and dynamic SQL**
- **Supports positioned UPDATE/DELETE of multi-row FETCH**
- **Significant performance boost**

```
DECLARE C1 CURSOR
    WITH ROWSET POSITIONING
    FOR SELECT COL1, COL2 FROM T1;
OPEN C1;
FETCH FROM C1
    FOR :hv ROWS  INTO  :ARRAY1, :ARRAY2;
```


Multi-row FETCH - 2

- **Allows positioned UPDATE or DELETE (of a single row) on a "wide" cursor**

```
UPDATE T1 SET COL1='ABC'
    WHERE CURRENT OF  C1
    FOR ROW  :hv  OF ROWSET
```

- **Using the WHERE CURRENT OF clause on a cursor with an "active" ROWSET, updates all rows in the ROWSET**

GET DIAGNOSTICS

- **Returns SQL error information**
  - For overall statement
  - For each condition (when multiple errors occur)
- **Supports SQL error message tokens greater than 70 bytes (SQLCA limitation)**

```
INSERT INTO T1 FOR 5 ROWS VALUES(:ARRAY);
GET DIAGNOSTICS :ERR_COUNT = NUMBER;
   DO II = 1 TO ERR_COUNT;
      GET DIAGNOSTICS CONDITION :II
           :RC = RETURNED_SQLSTATE;
   END;
```

- **Supports retrieving individual errors from NOT ATOMIC multi-row INSERT statements**

SELECT from INSERT

- **Elegant technique for retrieving values created/modified by DB2 during INSERT**
  - Identity columns, sequence values
  - User-defined defaults, expressions
  - Columns modified by triggers
  - ROWIDs, CURRENT TIMESTAMP

```
SELECT C1, C2, C3, C4, C5
   FROM FINAL TABLE
      (INSERT (C1, C5) INTO T1
           VALUES('ABC', CURRENT DATE)
      );
```

Identity Column Improvements

- **ALTER support for identity columns, eg.**

```
ALTER TABLE ALTER COLUMN idencol
                         RESTART WITH 500
```

  - GENERATED ALWAYS | BY DEFAULT
  - START WITH value
  - **RESTART WITH value**
  - INCREMENT BY
  - MINVALUE | **NO MINVALUE**
  - MAXVALUE | **NO MAXVALUE**
  - CYCLE | NO CYCLE
  - CACHE | NO CACHE
  - **ORDER | NO ORDER** (**Bold** keywords are new in V8)

SEQUENCES - 1

- **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
    - ORDER/NO ORDER
  - GRANT/REVOKE privileges for the sequence object
  - NEXT VALUE FOR sequence-name
  - PREVIOUS VALUE FOR sequence-name

SEQUENCES - 2

- **An example:**

```
CREATE SEQUENCE order_seq
          START WITH 1 INCREMENT BY 1
          NOMAXVALUE
          NOCYCLE CACHE 2
```

```
INSERT INTO orders (orderno,custno)
  VALUES (NEXT VALUE FOR order_seq,123456);
```

- **or**

```
UPDATE orders
     SET orderno =NEXT VALUE FOR order_seq
     WHERE custno =123456;
```

- **or**

```
SELECT NEXT VALUE FOR order_seq
  INTO :hv_seq from sysibm.sysdummy1;
```


Dynamic Scrollable Cursors

- **Scrollable cursor that provides access to base table rather than temp tables -- allows viewing of updates (and inserts) by other users as well as your own.**
- **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;
```

CURRENT PACKAGE PATH

- **Important for SQLJ and DDF applications**
  - Less need for SET CURRENT PACKAGESET
  - Less network traffic for multiple PKLIST values
  - Easier to switch to/from JDBC and SQLJ
  - Better handling of nested stored procedure packages
- **Allows application to specify search list of package schemas (similar to PKLIST)**

```
SET CURRENT PACKAGE PATH
      =     ALPHA,
            BETA,
            PROD
```

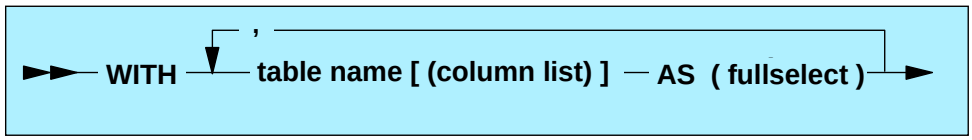

SET [CURRENT] SCHEMA

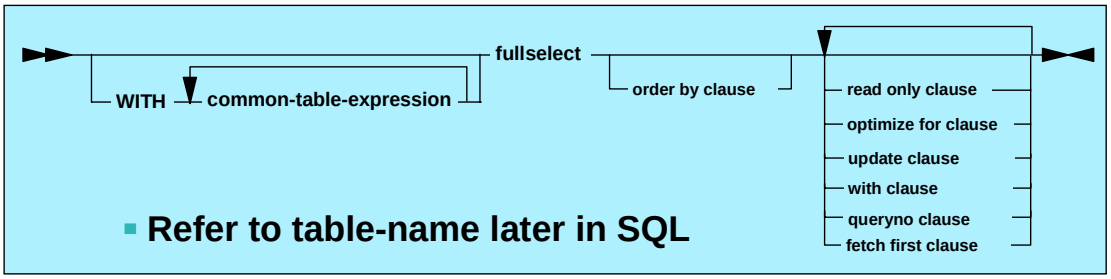
- **Set qualifier for unqualified dynamic SQL**
  - Can use SET CURRENT SQLID today
  - CURRENT SQLID has authorization implications
    - Can only be your primary or secondary auth ID
  - SET CURRENT SCHEMA is not tied to authorization
  - Improves DB2 Family compatibility

```
SET CURRENT SCHEMA  = 'MYQUAL' ;
SET SCHEMA  = 'MYQUAL' ;
```

Common Table Expressions (CTE) - 1

- **Allowed in SELECT, CREATE VIEW, INSERT**
- **Syntax:**





- **Refer to table-name later in SQL**

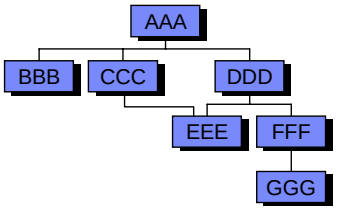
Common Table Expressions (CTE) - 2

```
WITH RPL (PART, SUBPART, QUANTITY) AS
   ( SELECT R.PART, R.SUBPART, R.QUANTITY
       FROM PARTLIST AS R
       WHERE R.PART = '01'
   )

SELECT DISTINCT PART, SUBPART, QUANTITY
  FROM RPL
  ORDER BY PART, SUBPART, QUANTITY;
```

Common Table Expressions (CTE) - 3

- **Allows for recursive SQL coding**
  - fullselect of CTE refers to itself
  - fullselect that does not refer to CTE
    UNION ALL
    fullselect that refers to CTE




```
WITH PARENT (PKEY, CKEY) AS
   ( SELECT PKEY, CKEY
        FROM HIERARCHY
        WHERE PKEY = 'AAA'
     UNION ALL
     SELECT C.PKEY, C.CKEY
        FROM HIERARCHY C ,PARENT P

        WHERE P.CKEY = C.PKEY
   )
SELECT PKEY, CKEY
  FROM PARENT;
```

PKEY	CKEY
AAA	BBB
AAA	CCC
AAA	DDD
CCC	EEE
DDD	EEE
DDD	FFF
FFF	GGG

Scalar fullselect - 1

- **Allows scalar fullselect where expressions were previously supported**

```
SELECT C1
  FROM T1
  WHERE
     T2 >  (SELECT COL1 FROM T2 ...)

SELECT C1,
            (SELECT COL1 FROM T2...),
            C3
       FROM T1
```

Scalar fullselect - 2

```
UPDATE NEW_PARTPRICE N
  SET PRICE =
    CASE
      WHEN(  (SELECT ONHAND#
                FROM INVENTORY
                WHERE PART=N.PART)  < 7  )
            THEN 1.1 * PRICE
      WHEN(  (SELECT ONHAND#
                FROM INVENTORY
                WHERE PART=N.PART) > 20  )
           THEN .8 * PRICE
      ELSE PRICE
    END;
```


SQL statements up to 2 MB

- **SQL statements can now be up to 2 MB in length**
- **Parse tree has been completely re-architected to reduce the number of cases of SQLCODE -101 (SQL too complex) due to SQL statement length**
- **Important for SQL procedure language applications**

IS NOT DISTINCT FROM

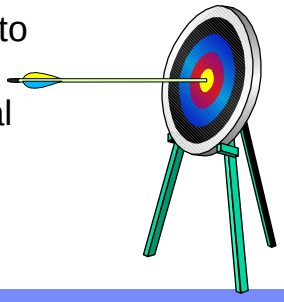
- **SQL uses three-valued logic where any given comparison can return: TRUE, FALSE, or NULL**
- **Applications can use IS NOT DISTINCT FROM to obtain a TRUE result instead of NULL when a comparing NULL values**

```
SELECT C1 FROM T1 WHERE
C1 IS NOT DISTINCT FROM :hv;
```

C1 value	:hv value	RESULT
NULL	'ABC'	FALSE
NULL	NULL	TRUE
'ABC'	'ABC'	TRUE
'ABC'	NULL	FALSE
'ABC'	'DEF'	FALSE

REOPT(ONCE)

- **Bind option that controls when the optimizer builds the access path information for dynamic SQL applications**
  - By default, access path is calculated at PREPARE
  - Using REOPT(ONCE)
    - Defers access path selection until OPEN
    - Values of host variables on OPEN are used to calculate access path
    - Resulting access path is cached in the global (prepared) statement cache

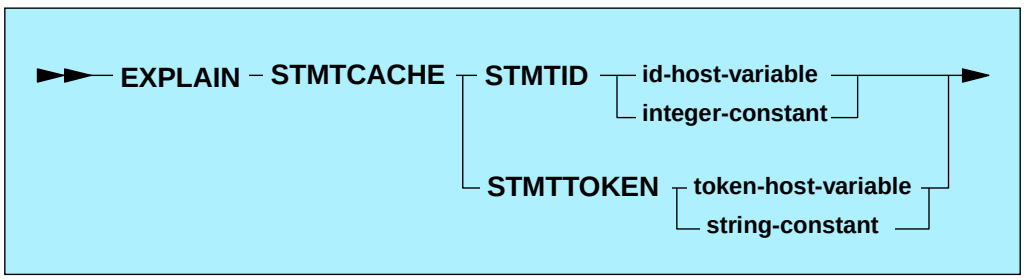


Other SQL Improvements

- **SQL procedures language extensions (see later)**
- **GROUP BY expression**
  - SELECT A+B,C
    FROM T
    ORDER BY A+B
    GROUP BY **A+B**
- **Qualified column names on SET clause**
  - UPDATE T1 SET **T1.**COL1...
- **Qualified column names on INSERT**
  - INSERT **T1.**COL1 INTO T1 VALUES...
- **Multiple DISTINCT clauses**
  - SELECT COUNT(**DISTINCT(A1)**), AVG(**DISTINCT(A2)**)
    FROM T2

Other SQL Improvements -2

- **Enhancements to the EXPLAIN statement allow you to obtain EXPLAIN information for entries in the DB2 global statement cache.**
- **Visual Explain is enhanced to exploit this new function.**



Other SQL Improvements -3

- **READ ONLY USING UPDATE LOCKS**
  - Allows WebSphere persistence layer to minimize network traffic when using searched update and pessimistic locking

```
PREPARE stmt 1 FROM
  "SELECT C1, C2, C3 FROM T1 WHERE C1=? AND C2=?
     FOR READ ONLY KEEP UPDATE LOCKS"
```

- **VOLATILE table support (see performance section for details)**

```
CREATE TABLE  TAB 1 ...  VOLATILE
```

- **Transparent ROWID column**
  - Eliminates the need to explicitly declare a ROWID column in tables that include LOBs
  - DB2 will generate a "hidden" ROWID column, which is not visible on SELECT *
  - Simplifies porting of LOB applications from other platforms

New Built-in Functions

- **ENCRYPT: encrypt a column in a table with a user-provided encryption password**
  - ENCRYPTION PASSWORD special register
- **DECRYPT_BIN**
- **DECRYPT_CHAR**
- **GET_HINT: obtain hint to help remember ENCRYPTION PASSWORD**
- **GENERATE_UNIQUE creates a CHAR(13) FOR BIT DATA value,unique across a sysplex**
- **Character-based string functions**

```
SELECT SUBSTRING(COLB,1,3,CODEUNITS32)
       AS  RESULT FROM TESTB
```

New Special Registers

- Client information for this connection
  Provided by sqleseti, Java, RRS SIGNON
  - CURRENT CLIENT_ACCTNG : accounting string
  - CURRENT CLIENT_APPLNAME : application name
  - CURRENT CLIENT_USERID : client user ID
  - CURRENT CLIENT_WRKSTNNAME : workstation name
- MQT related special registers
  - CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION
  - CURRENT REFRESH AGE
- Application enablement related special register
  - CURRENT PACKAGE PATH
  - CURRENT SCHEMA

Session variables

- **Variables set by DB2, connection or signon exit**
- **Built in function to retrieve value for a variable**
  - Use function in views, triggers, stored procedures & constraints to enforce security policy
- **Can have more general, flexible access checks**
  - Multiple columns, AND/OR logic, ...
- **Complements other security mechanism**

```
CREATE VIEW V1 AS SELECT * FROM T1
WHERE
COL5 = GETVARIABLE(SYSIBM.SECLABEL);
```

Key Java Improvements

- **DB2 Universal Driver for SQLJ and JDBC**
- **Type 2 and type 4 driver**
  - Updated to support JDBC/SQLJ 3.0 standard
    - Savepoint support
    - Connection pooling improvements
    - New metadata for PreparedStatements
    - Return autogenerated keys
    - Multiple open ResultSets for a single stored procedure
    - WITH HOLD cursors
    - Improved BLOB/CLOB support
    - More to come ...
- **SQLJ tooling support in WSAD V5.1**

UNICODE Evolution

- **UNICODE encoding scheme can represent codepoints of characters of many different geographies and languages**
  - Sometimes more than one byte to represent a single characters
  - Different languages in the same table
  - CCSID UNICODE
- **V5 ASCII data and Unicode for Java**
- **V7 UNICODE data storage and manipulation**
- **OS/390, z/OS, z/Architecture and zSeries**

WebSphere. software

Unicode Enhancements in Version 8

- **Able to join Unicode with EBCDIC or ASCII**
- **SQL in Unicode, statements, literals, object names, mixed CCSIDs / Utility statements can be in Unicode**
- **Most DB2 character catalog columns in Unicode (Unicode catalog)**
- **Collating sequence change**
- **Lengths and maximum lengths can change**
- **DB2 itself works in Unicode (eg. SQL statement parsing)**

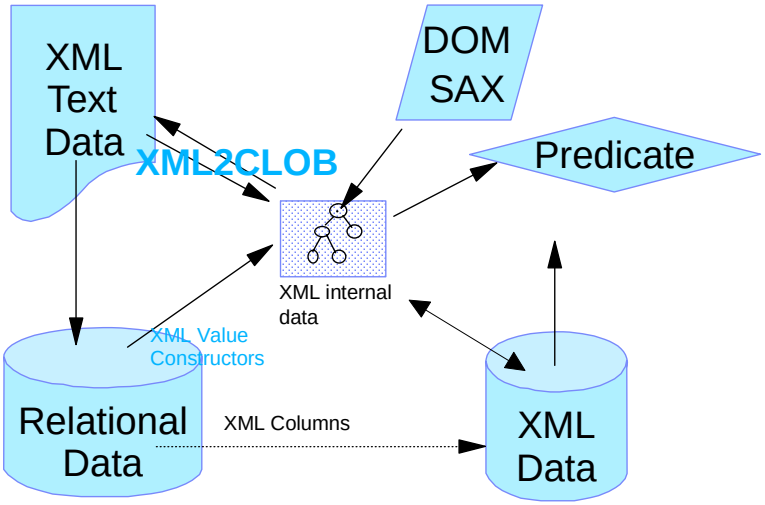
www.unicode.org

Multiple CCSIDs per SQL statement

- **A single SQL statement can now intermix EBCDIC, ASCII and Unicode tables and host variables**
- **Unlike host variables are translated to column CCSID**
- **Unlike column CCSIDs are "promoted" to Unicode**
  - May require query to be materialized/sorted

```
SELECT T1.COL_EBCDIC ,
       T2.COL_ASCII  ,
       T3.COL_UNICODE
  FROM T1,T2,T3 WHERE ...;
```

SQL/XML - XML extensions to SQL



XML value constructors:

XMLELEMENT (xmlattributes), XMLFOREST, XMLCONCAT, XMLAGG, XMLNAMESPACES

XML Publishing Functions SQL/XML

- **New built-in functions for generating XML from relational data**
  - **XMLELEMENT** generates XML element from arguments
  - **XMLATTRIBUTES** specify attributes for the XML element within XMLELEMENT
  - **XMLFOREST** creates forest of XML elements that share a specific pattern
  - **XMLCONCAT** returns a forest of XML elements that are generated from a concatenation of two or more arguments
  - **XMLAGG** returns a concatenation of XML elements from a collection of XML elements
  - **XML2CLOB** converts the new transient XML data type into UTF-8 CLOB
  - **XMLNAMESPACES** is used to declare XML Namespaces
- **Alternative to XML composition function of XML Extender**

Query Example

Query:

```
SELECT
  XML2CLOB(
    XMLELEMENT
     ( "Dept",
       XMLATTRIBUTES ( D.DEPTNO AS "deptno",
                       D.DEPTNAME AS "name" ),
      ( SELECT XMLAGG (
              XMLELEMENT (
                "Proj",
                XMLATTRIBUTES (P.PROJNO AS "projno",
                        P.PROJNAME AS "name"),
                ( SELECT XMLAGG (
                        XMLELEMENT (
                          "Emp",
                          XMLATTRIBUTES
                           (E.EMPNO as "empno"),
                        E.FIRSTNME || ' ' || E.LASTNAME
                              )
                        )
                  FROM DSN8810.EMPPROJACT EP,
                     DSN8810.EMP E
                  WHERE EP.PROJNO = P.PROJNO AND
                     EP.EMPNO = E.EMPNO
                ) )
              )
        FROM DSN8810.PROJ P
        WHERE P.DEPTNO = D.DEPTNO
      )
     ) )
FROM DSN8810.DEPT D
WHERE D.DEPTNO = 'D01';
```

Result: (formatted for easy viewing)

```
<Dept deptno="D01" name="DEVELOPMENT CENTER">
  <Proj projno="AD3100" name="ADMIN SERVICES">
    <Emp empno="000010">CHRISTINE HAAS</Emp>
  </Proj>
  <Proj projno="MA2100" name="WELD LINE AUTOMATION">
    <Emp empno="000010">CHRISTINE HAAS</Emp>
    <Emp empno="000110">VINCENZO LUCCHESI</Emp>
  </Proj>
</Dept>
```


ON COMMIT DROP enhancement

- **Declared Temporary Table improvement that causes temp table to be dropped automatically at commit:**
  - When no HELD cursors are open against the temp table
- **Significant improvement for DDF threads that use CMTSTAT=INACTIVE**
  - Stops declared temp tables from preventing thread from going INACTIVE

```
DECLARE  GLOBAL TEMPORARY TABLE
      DTT1 (COL1 CHAR (20))
      ON COMMIT DROP TABLE
```

Stored Procedures and RRS

- **Stored procedures improvements**
  - WLM management of TCBs
    - TCBs are added/removed based on WLM recommendations
  - MAX FAILURES on CREATE PROCEDURE and UDF
  - COMPJAVA (HPJ) is replaced by interpretive Java
  - No creation of, or altering to, DB2-managed SP in V8
    - Migrate to WLM-managed SP
- **RRS "implicit" connection support**
  - Similar to CAF-style implicit connections

DB2 UDB for z/OS Version 8 is

- ✓ **SQL OLTP Leadership: Multi-row Fetch & Insert, Select within Insert, Dynamic Scrollable Cursors, ...**
- ✓ **Breaking through limitations: storage, name lengths, SQL statements, partitions, logging**
- ✓ **Performance Enhancements: MQTs, index use**
- ✓ **Database changes without an outage: add partition, rotate partitions, alter cluster**
- ✓ **Integration: applications, middleware, platform**
- ✓ SAP PEOPLESOFT SIEBEL ...

Reengineered for e-business

Bibliography: DB2 for z/OS and OS/390 Redbooks

- DB2 UDB for z/OS V8 Everything you ever wanted to know, …and more, SG24-6079
- DB2 UDB for z/OS Version 8 Performance Topics, SG24-6465
- DB2 UDB for z/OS V8: Through the Looking Glass & What SAP Found There, SG24-7088
- DB2 for z/OS and OS/390: Ready for Java, SG24-6435
- Distributed Functions of DB2 for z/OS and OS/390, SG24-6952
- Moving Data Across the DB2 Family, SG24-6905
- Implementing PeopleSoft 8.4 on zSeries, SG24-6549
- Siebel 7 with DB2 for z/OS: Database Implementation & Administration, SG24-6868
- SAP on DB2 for z/OS: Multiple Components in One Database, SG24-6914
- SAP on DB2 for z/OS: High Availability Using System Automation, SG24-6836
- Squeezing the Most Out of Dynamic SQL, SG24-6418
- Large Objects with DB2 for z/OS and OS/390, SG24-6571

DB2 for z/OS on the Web

ibm.com/software/db2zos

- Full URL for DB2 for z/OS page
  http://www.ibm.com/software/data/db2/zos/index.html
- DB2 for z/OS Version 8 related material
  http://www.ibm.com/software/db2/zos/db2zosv8.html
- Direct URL for Support page
  http://www.ibm.com/software/data/db2/zos/support.html
- Presentations page:
  http://www.ibm.com/software/data/db2/zos/presentations.html
- DB2 Developer Domain articles
  http://www.ibm.com/developerworks/db2/