

IBM® DB2® for i 7.1 and Mid-release Enhancements

Mark J Anderson

IBM Information Management software

Power is performance redefined

Deliver IT services faster,
with higher quality,
and with superior economics

www.ibm.com/power

Logos for Power, AIX, i, and Linux.

© 2011 IBM Corporation

DB2 for i 7.1 Enhancements

SQL Enhancements

- XML support
- MERGE
- Global variables
- Array support in routines
- Result set support in embedded SQL
- Encryption enhancements (FIELDPROCS)
- Partition table enhancements
- MQ Series functions
- Parameter marker enhancements
- Expressions in CALL
- Three-part Aliases
- Currently Committed
- CREATE OR REPLACE
- BIT scalar functions
- SSD support (6.1 PTF)
- SYSTOOLS procedures (6.1 PTF)
- QSYS2.SQL_CANCEL procedure (V5R4 PTF)

XML Support

XML data type (table columns, routine parameters, variables, etc.)
 XML document decomposition and annotation
 Document search without decomposition, supported via Omnifind
 Client and language API support for XML (CLI, ODBC, JDBC, etc.)
 XML Publishing functions

Create a table with an XML type

```
CREATE TABLE newtable (c1 INTEGER, c2 XML )
```

Alter a table and add a column with an XML type

```
ALTER TABLE newtable ADD COLUMN c3 XML
```

Define an XML variable in an SQL procedure

```
CREATE PROCEDURE xyz (IN p1 XML)  
BEGIN  
    DECLARE xmlv1 XML;  
    ...  
END
```

*Allows you to define
XML columns in a table
and XML variables in an
application*



XML XSR Objects

Procedure	Description
XSR_REGISTER	Begins registration of an XML schema. You call this stored procedure when you add the first (primary) XML schema document to an XML schema
XSR_ADDSCHEMADOC	Adds additional XML schema documents (other than the primary) to an XML schema that you are in the process of registering. You can call SYSPROC.XSR_ADDSCHEMADOC only after SYSPROC.XSR_REGISTER and before SYSPROC.XSR_COMPLETE
XSR_COMPLETE	The XSR_COMPLETE procedure is the final stored procedure to be called as part of the XML schema registration process. It completes the registration of an XML schema. You must call SYSPROC.XSR_COMPLETE each time you call SYSPROC.XSR_REGISTER. During XML schema completion, DB2 resolves references inside XML schema documents to other XML schema documents
XSR_REMOVE	The XSR_REMOVE procedure is used to remove all components of an XML schema
XDBDECOMPXML	The XDBDECOMPXML procedure is used to extract values from serialized XML data and populates relational tables with the values

Grant to and Revoke from an XSR Object

```
GRANT USAGE ON XSROBJECT xsr1 TO mja
REVOKE USAGE ON XSROBJECT xsr1 FROM mja
```

Add a description of the XSR Object

```
COMMENT ON XSROBJECT xsr1 IS 'A long description ...'
LABEL ON XSROBJECT xsr1 IS 'A short description ...'
```



XML Functions

Scalar function	Description
XMLATTRIBUTES	Returns an XML sequence that contains an attribute node for each non-null argument
XMLCOMMENT	Returns an XML value with a single comment node from a string expression
XMLCONCAT	Returns an XML value that represents a forest of XML elements generated by concatenating a variable number of arguments
XMLDOCUMENT	Returns an XML value with a single document node and zero or more nodes as its children
XMLELEMENT	Returns an XML value that represents an XML element
XMLFOREST	Returns an XML value that represents a forest of XML elements that all share a specific pattern
XMLPI	Returns an XML value with a single processing instruction node
XMLNAMESPACES	Returns the declaration of one or more XML namespaces
XMLROW	Returns an XML value with a single document node containing one top-level element node
XMLTEXT	Returns an XML value with a single text node that contains the value of the argument

Scalar function	Description
XMLPARSE	Returns an XML value from parsing the argument as an XML document
XMLSERIALIZE	Returns an SQL character string or a BLOB value from an XML value
XMLTRANSFORM	Converts XML data into other formats, including the conversion of XML documents that conform to one XML schema into documents that conform to another schema
XMLVALIDATE	Returns a copy of the input XML value augmented with information obtained from XML schema validation, including default values and type annotations

Aggregate function	Description
XMLAGG	Returns an XML type that represents a concatenation of XML elements from a collection of XML elements
XMLGROUP	Returns an XML value with a single document node containing one top-level element node from a group of rows

MERGE

If a description has been changed for an activity in the ACTIVITIES table, update the description or the same activity in the ARCHIVE-AR table. Otherwise, insert the new activity in the ARCHIVE_AR table.

```
MERGE INTO archive ar
USING (SELECT activity, description FROM activities) ac
ON (ar.activity = ac.activity)
WHEN MATCHED THEN
    UPDATE SET description = ac.description
WHEN NOT MATCHED THEN
    INSERT (activity, description) VALUES (ac.activity, ac.description)
```

Allows you to update, delete, or insert into a table based on values from a source table.

Global variables

Create a global variable to indicate the department where an employee works.

```
CREATE VARIABLE MJATST.DEPT CHAR(3)
DEFAULT ((SELECT WORKDEPT FROM EMPLOYEE
WHERE LASTNAME = SESSION_USER))

LABEL ON VARIABLE MJATST.DEPT IS 'Department variable'
COMMENT ON VARIABLE MJATST.DEPT IS 'A long description...'
```

Grant and revoke privileges from a global variable.

```
GRANT READ ON VARIABLE MJATST.DEPT TO SCOTTF
REVOKE READ ON VARIABLE MJATST.DEPT FROM SCOTTF
```

Reference a global variable. The first reference instantiates the value.

```
INSERT INTO MJATST.TEST VALUES( MJATST.DEPT, ...)
SET MJATST.TEST = 'NA '
```

Allows you to define variables that are scoped to the job.

Array Support in Routines

Create a type that is an array

```
CREATE TYPE INTARRAY AS INTEGER ARRAY[20]
```

Create an SQL procedure that uses an the array type.

```
CREATE PROCEDURE MYPROC (OUT P1 INTARRAY)
BEGIN
  DECLARE ids intArray;
  DECLARE c2 CURSOR FOR SELECT * FROM UNNEST(ids) AS x;

  SELECT ARRAY_AGG (name ORDER BY id) INTO ids FROM persons;
  SET ids = ARRAY[5,6,7] ;
  SET ids[4] = 8;
  SET (maxcardo, cardo) = (MAX_CARDINALITY(ids), CARDINALITY(ids));
END
```

Allows you to use arrays in SQL routines both as SQL variables and SQL parameters to routines.

Result set support in embedded SQL

Access a result set in ILE RPG after calling procedure P1.

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

Allows you to access result sets from procedure calls in embedded programs.

Encryption enhancements (FIELDPROCS)

Create a CUSTOMER table where the CCNBR column has a FIELDPROC.

```
CREATE TABLE CUSTOMER (
  NAME VARCHAR(50),
  ADDRESS VARCHAR(100),
  CCNBR CHAR(16) FIELDPROC MJATST.ENCRYPTPGM1 )
```

Allows for transparent encryption or encoding of data accessed through SQL or native.

Alter an existing CUSTOMER table to add a FIELDPROC to the CCNBR column.

```
ALTER TABLE CUSTOMER
  ALTER COLUMN CCNBR SET FIELDPROC MJATST.ENCRYPTPGM1
```

Restrictions

Users must have EXECUTE privilege on the field procedure program
 The field procedure must be an ILE program
 ROWID and DATALINK FILE LINK CONTROL columns are not allowed
 Identity columns and row change timestamp columns are not allowed
 Columns with a FIELDPROC cannot be referenced in a CHECK constraint
 Columns with a FIELDPROC cannot be referenced in a partitioning key

Additional masking capability in TR4

Partitioned Table Enhancements

- Allow Identity columns in a partitioned table
- Allow a referential constraint on a partitioned table

Allows you to partition tables that use referential integrity or identity columns.

Restrictions

- The identity column cannot be a partitioned key
- The parent table must either be:
 - Non-partitioned
 - Partitioned where the unique index used for the unique constraint is non-partitioned
- Partitioned tables with RI or an identity cannot be saved to a previous release



MQ Series functions

Scalar function	Description
MQREAD	Returns a message from a specified MQSeries location (return value of VARCHAR) without removing the message from the queue
MQREADCLOB	Returns a message from a specified MQSeries location (return value of CLOB) without removing the message from the queue
MQRECEIVE	Returns a message from a specified MQSeries location (return value of VARCHAR) with removal of the message from the queue
MQRECEIVECLOB	Returns a message from a specified MQSeries location (return value of CLOB) with removal of the message from the queue
MQSEND	Sends a message from a specified MQSeries location

Table function	Description
MQREADALL	Returns a table containing the message and message metadata from a specified MQSeries location (return value of VARCHAR) without removing the messages from the queue
MQREADALLCLOB	Returns a table containing the message and message metadata from a specified MQSeries location (return value of CLOB) without removing the messages from the queue
MQRECEIVEALL	Returns a table containing the message and message metadata from a specified MQSeries location (return value of VARCHAR) with removal of the messages from the queue
MQRECEIVEALLCLOB	Returns a table containing the message and message metadata from a specified MQSeries location (return value of CLOB) with removal of the messages from the queue

Allows an easy way to send, read, and receive MQ messages in an SQL statement.

Send a This example sends the string 'Testing 678' to the service MYSERVICE, using policy MYPOLICY, with correlation ID TEST3.
VALUES MQSEND (MYSERVICE, MYPOLICY, 'Testing 678', TEST3)



Three-part Aliases

Create an alias that points to a table on another logical partition (relational database).

CREATE ALIAS MJATST.t1 FOR RDB2.MJATST.T1

SELECT * FROM MJATST.T1

SELECT * FROM RDB2.MJATST.T1

Eliminates the need to explicitly use CONNECT and SET CONNECTION.

Restrictions

- All object references in a single SQL statement must reside in a single relational database
- The alias name must be the same as the remote name but can point to another alias on the remote



Currently Committed

Currently committed data can now be selected instead of waiting for locks on rows currently being updated. Currently committed can be specified on several interfaces:

- Statement level (select-statement, SELECT INTO, PREPARE ATTRIBUTES, UPDATE, and DELETE.
- Precompilers (CONACC(*CURCMT))
- JDBC connection property concurrentAccessResolution
- CLI connection attribute SQL_ATTR_CONCURRENT_ACCESS_RESOLUTION
- QAAQINI SQL_CONCURRENT_ACCESS_RESOLUTION

Reduces contention when dirty data is not allowed, but data in progress of an update is not important.

WARNING:

- If an option is not specified, the INI option will control concurrent access

Select only committed data, but updates in progress can be ignored and the currently committed data returned.

SELECT * FROM Inventory WHERE ItemNumber = 123456 USE CURRENTLY COMMITTED

Select only committed data, but wait for row locks (updates in progress).

SELECT * FROM Inventory WHERE ItemNumber = 123456 WAIT FOR OUTCOME



SQL Query Engine (SQE)

SQE Characteristics

- Parallel to Normal Release Enhancements
- Object Oriented Design
- Enhanced Performance primarily for complex queries
- Enhanced Optimization Engine
- Enhanced Statistics
- Encoded Vector Indexes Enhancements

SQE Delivery

First Wave	V5R2 GA
Second Wave	mid-V5R2 (Check APAR II13486)
Third Wave	V5R3 GA
Fourth Wave	mid-V5R3 4/29/2005
Fifth Wave	V5R4 GA
Sixth Wave	6.1 GA
Seventh Wave	7.1 GA

SQE (Stage 7)

- Logical File support
- Adaptive Query Processing (AQP)
- Other miscellaneous performance enhancements

Performance

SQL Runtime Performance Improvements

- SQL function enhancements (inline functions)
- OVRDBF and REUSEDLT(NO)
- EVI enhancements

```
CREATE ENCODED VECTOR INDEX idx2 ON sales (region)  
INCLUDE ( SUM(sales) )
```

```
SELECT region, SUM(sales) FROM sales GROUP BY region
```

Database Maintenance Performance Improvements

- ALTER TABLE performance improvements for partitioned tables

Limits/Availability/Recovery

SQL Application and Runtime Limits

- 128-byte schema names

Database

- Allow transactions to span *SYSBAS and IASPs
- CHGPF CST CHECK(*YES *NO)
- CHGSR CF TEXT(*FROM MBR) SRCTYPE(*FROM MBR)
- Additional Unicode catalog support
- Preserve statistics on ALTER TABLE
- Deflated table support
- CICS support in IASPs

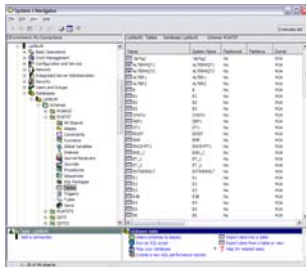
Journaling

- Additional filtering based on generic names on STRJRNLIB and CHGJRNOBJ to control what journaling implicitly gets started
- Additional control on whether journal entries for an object(s) should be filtered on remote journaling

IBM i Navigator

On Demand Performance Center

- Database monitor
 - Client register filters
 - Initial allocation
- Authority changes
- Show Statements
 - Save statements
 - Work Statement With Variables
- Run SQL Scripts – Visual Explain Options
- SQL Details for a Job
 - Result set information
 - Start database monitor
- Index Advisor enhancements
 - MTI information
 - First Advised



Health Center

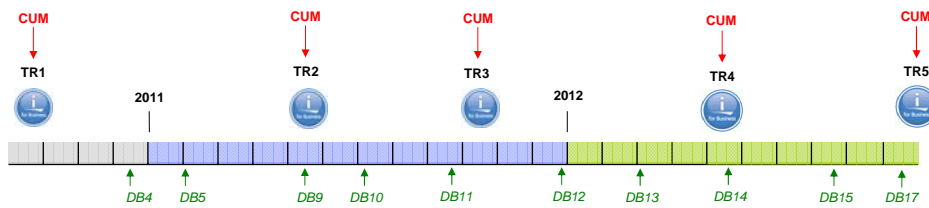
- SQL0901 Logger
- Random and Sequential I/O Counts

Database Management

- Support for new Database Features
 - XML
 - XML Schema Repository (XSR)
 - Global Variables
 - Array Types
 - Field Procedures
 - 3-part alias
 - Enable constraints without checking
- Support for Omnifind
- Generate SQL - include GRANTS
- Long Running Operation Status
 - Reorganize status enhancements
 - Index build status
 - Text Search Index build status
 - ALTER TABLE status
- Object list enhancements
 - Save folder contents
 - Object list filtering
 - List Large list and table list performance
 - Long schema name
- Last build info in SYSPARTITIONINDEXSTAT
- New *SSD and In Memory info

DB2 for i Mid-Release Enhancements – TR5

DB2 for i Technology Updates – 7.1



- Database groups come out most frequently
- CUMs include the last database group
- CUMs may come out more frequently than TRs
- New CUMs are built at the same time as T3s

www.ibm.com/developerworks/ibmi/techupdates/db2

developerWorks Technology Updates wiki

The screenshot shows the developerWorks Technology Updates wiki page. The page title is "DB2 for i - Technology Updates". The content includes a list of updates and a section titled "DB2 for i updates by PTF Group and year". Annotations highlight specific features:

- Easy navigation:** Points to the left sidebar navigation menu.
- Enhancements delivered in DB2 PTF Groups:** Points to a bullet point stating "Not all DB2 for i improvements are listed here. Only the more significant enhancements are listed. Does not include many performance enhancements since they are typically automatic. Performance enhancements are only listed if they are not automatic or are too significant not to mention."
- Organized by type of enhancement:** Points to the "DB2 for i updates by category" section, which lists categories like Functional Enhancements, Security Enhancements, Performance Enhancements, Database Management Enhancements, Availability/Recovery Enhancements, and OmniFind for IBM i.
- Subscription options:** Points to the "Subscribe to this page" link at the bottom of the page.

DB2 for i – Enhancements for TR5

Functional enhancements:

Named Arguments and Defaults for Parameters – Procedures
 CREATE TABLE with remote SUBSELECT
 Qualified name option added to Generate SQL
 OmniFind for IBM i – searching Multiple Member source physical files
 System naming convention expanded to permit (slash) and (dot) qualifiers
 Modification of Global Variables within triggers and functions
 New QAQQINI option – SQL_GVAR_BUILD_RULE
 CPYTOIMPF and CPYFRMIMPF commands – Include column headings
 RUNSQLSTM command - OPTION parameter added to control listing
 JTOpen Lite and JTLite – enabling mobile devices which use java
 Java stored procedures and functions - System naming option

Performance enhancements:

SQE enhancement for Encoded Vector Indexes defined with INCLUDE
 Index Advisor - Show Statements - improved query identification
 Add Physical IO and other SQL metrics to Collection Services
 Performance Data Investigator – New Database Perspective
 Performance improvements for temporary tables
 Performance improvement for partitioned tables
 Improved performance of privileges catalogs, ODBC & JDBC MetaData APIs
 Show Statements – Index Creates and Statistics Advised

Database Availability and Recovery enhancements:

Improved catalog management for procedures and functions
 Navigator – System Name column added to Show Related and All Objects
 Improved NULLID package management

Security enhancements:

Infosphere Guardium V9 – target DB2 for i as a data source

Primary Beneficiaries:

Application teams

DBA & Performance Analyst

High Availability, DBA & users in general

Auditors and more

Infosphere Guardium

Real-time Database Protection and Compliance

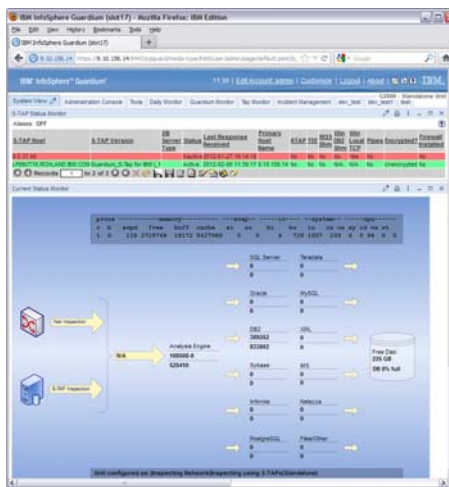
- Standardized auditing policies across three DBMS platforms (Oracle, Microsoft SQL Server, IBM DB2)
- Proactive security such as real-time alerts when critical tables are changed
- Labor and capital cost savings compared to using native logging utilities with in-house scripts
- Simplified compliance processes that rapidly meet auditors' requirements with preconfigured reports and real-time controls
 - HIPAA
 - PCI
 - SOX
 - Etc.

Current support

- SPAN port session to capture "network" activity
- Monitor Local Traffic through journal
- Upload QAUDJRN journal into Guardium Appliance

Enhancements in progress

- Two sources of data that will be sent Guardium
 - Database monitor data (using monitor over a view to send only the pertinent data to the appliance)
 - Audit entries from QSYS/QAUDJRN





Navigator – what database users need to know

- Database users will find reasons to use both Navigator options

What are the choices?	IBM i Navigator (aka System i Navigator)	IBM Navigator for i
Where does it run?	Windows PC Install	Browser Served from IBM i 6.1 or 7.1
Minimum service level?	IBM i Access Windows Service Pack 7.1 - SI47412	IBM HTTP SERVER FOR i PTF Group: 7.1 - SF99368 Level 16 6.1 - SF99115 Level 28
Database commonality	Most features are identical, including TR5 enhancements	Most features are identical, including TR5 enhancements
Database differences	Run SQL Scripts Visual explain	SQL Performance Metrics
Target Availability	December, 2012	December, 2012



JTOpen Lite / JTLite – enabling mobile devices which use java

- An alternative to the JT400 and JTOpen java toolkits, providing better performance and a small footprint (~420K) to application development targeting mobile devices.

JTOpen and JT400	JTOpen Lite / JTLite
Database -- JDBC (SQL) and record-level access (DDM)	Database -- JDBC (SQL) and record-level access (DDM)
Integrated File System	Integrated File System – open, read, write, delete
Program calls (RPG, COBOL, service programs, etc)	Program calls (RPG, COBOL, service programs, etc)
Commands	Commands
Data queues	
Data areas	
Print/spool resources	
Product and PTF information	
Jobs and job logs	Job Information
Messages, message queues, message files	
Data queues	
Users and groups	User and group Information
User spaces	
System values	
System status	

Named Arguments and Defaults for Parameters - Procedures

- Named and default parameters supported for SQL and external procedures – making it possible to run more SQL applications on IBM i.
- Extend procedures without fear of breaking existing callers and simplify the calling requirements by adding default values.
- Procedure call invocation has the same type of support as CL commands.
- With this enhancement:
 - 1) Parameters may be **omitted** if the routine was defined with a default value
 - 2) Parameters may be specified in **any order** by specifying the name in the call
 - 3) Works with LANGUAGE SQL and EXTERNAL procedures
- CREATE PROCEDURE p1 (i1 INT, i2 INT **DEFAULT** 0, i3 INT **DEFAULT** -1)...
- CALL p1(55)
- CALL p1(55, i3=>33)

default-clause:



Automatic management of catalogs when librarian commands target SQL routines

- When a procedure or function is created, the routine information is stored within the *PGM or *SRVPGM. Previously, when Librarian commands were used to copy/move/rename the object, the QSYS2/SYSROUTINE, SYSPARMS and SYSRTNDEP catalogs weren't updated to stay in sync with the executable.
- The following commands (and their API counterparts) are changed to keep the catalogs in sync with the executable object for **LANGUAGE SQL** procedures and functions:
 1. Create Duplicate Object (CRTDUPOBJ)
 2. Copy Library (CPYLIB)
 3. Rename Object (RNMOBJ)
 4. Move Object (MOV OBJ)
- Includes coverage for Librarian APIs or other operations built upon these commands.

CREATE TABLE with remote SUBSELECT

- CREATE TABLE AS and DECLARE GLOBAL TEMPORARY TABLE are enhanced to allow the select to reference a single remote database which is different than the current server connection
- An implicit remote connection is established and used by DB2 for i
- The remote query can reference a single remote homogeneous or heterogeneous table

Example:

```
CREATE TABLE DATALIB.MY_TEMP_TABLE AS (SELECT CURRENT_SERVER
CONCAT ' is the Server Name', IBMREQD
FROM X1423P2.SYSIBM.SYSDUMMY1) WITH DATA
```

```
SELECT * FROM DATALIB.MY_TEMP_TABLE
```

- ✓ 7.1 Base – 3-part name support
- ✓ TR4 – INSERT WITH remote SUBSELECT
- ✓ TR5 – CREATE TABLE WITH remote SUBSELECT

SERVER_NAME	DATA_VALUE
X1423P2 is the Server Name	Y

7.1 Backup

XML Support

Specify how whitespace should be handled when implicitly parsing XML data

```
SET CURRENT IMPLICIT XMLPARSE OPTION = 'PRESERVE WHITESPACE'
```

Convert an XML value into another data type.

```
XMLCAST(NULL AS XML )
```

Define an XML variable in an embedded language

```
SQL TYPE IS XML AS CLOB (10K) xmlhv  
...  
INSERT INTO newtable VALUES(1, :xmlhv)
```

Parameter marker enhancements

Prepare a statement that uses a parameter marker in the select list.

```
SET STMT1 = 'SELECT * FROM T1 WHERE C1 > ? + ?';  
PREPARE PREPSTMT1 FROM STMT1;
```

Previously, this would have to be something like:

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

Eliminates the need to specify a CAST of a parameter maker to a specific type in an SQL statement in most cases.

Expressions in CALL

Call a procedure and pass as arguments PARAMETER1 folded to uppercase and PARAMETER2 divided by 100.

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

Eliminates the need to run a query or SET statement to evaluate the expressions before the CALL.

BITAND, BITANDNOT, BITOR, BITXOR, and BITNOT

New bit functions based on numeric data types.

```
SELECT ITEMID FROM ITEM
  WHERE BITAND(PROPERTIES, 4) = 4
SELECT ITEMID FROM ITEM
  WHERE BITAND(PROPERTIES, 40) <> 0
UPDATE ITEM
  SET PROPERTIES = BITANDNOT(PROPERTIES, 2048)
  WHERE ITEM = 3412
UPDATE ITEM
  SET PROPERTIES = BITOR(PROPERTIES, 16)
  WHERE ITEM = 3412
UPDATE ITEM
  SET PROPERTIES = BITXOR(PROPERTIES, 1024)
  WHERE ITEM = 3412
VALUES BITNOT(CAST(2) AS SMALLINT)
```

New bit manipulation functions.

QSYS2.SQL_CANCEL Procedure

CANCEL_SQL

1) JOB_NAME - The name of the job

Restrictions:

- Will only cancel SQL database operations
- Will not cancel a commit or rollback
- Usually will return SQL0952, but may also return SQL0901
- *JOBCTL or QIBM_DB_SQLADM authority required

Ability to cancel a specific long running database request in another job.

```
CALL QSYS2.CANCEL_SQL ('197968/QUSER/QZDASOINIT')
```

CREATE OR REPLACE

Create an object and if it exists replace it.

To replace an object, the user must have:

- *OBJEXIST to the object
- *EXECUTE for the library
- Privileges to create the object

Privileges on the existing object are preserved.

CREATE OR REPLACE ALIAS Alias1 ...

CREATE OR REPLACE FUNCTION Function1 ...

CREATE OR REPLACE PROCEDURE Procedure1 ...

CREATE OR REPLACE SEQUENCE Sequence1 ...

CREATE OR REPLACE TRIGGER Trigger1 ...

CREATE OR REPLACE VARIABLE Variable1 ...

CREATE OR REPLACE VIEW View1 ...

Makes it easier to either create or replace an object.

7.1 DB2 Enhancements Summary

IBM **Information Management** software

7.1 DB2 Enhancements

SQL

- Queries**
 - Result set support in embedded SQL ★
 - CURRENTLY COMMITTED
- Data Change**
 - MERGE
- Scalar Functions**
 - MQ Series functions
 - BIT scalar functions
 - TIMESTAMP_FORMAT and VARCHAR_FORMAT enhancements
- Schema statements (DDL)** ★
 - XML support ★
 - Global variables
 - Array support in procedures
 - Encryption enhancements (FIELDPROCS) ★
 - Partition table enhancements
 - REPLACE option on CREATES
- Miscellaneous**
 - Three-part names and aliases
 - Parameter marker enhancements
 - SYSTOOLS procedures
 - QSYS2.SQL_CANCEL procedure

Performance

- SQE (Stage 7)**
 - Logical File support ★
 - Adaptive Query Processing (AQP) ★
 - Other miscellaneous performance
- SQL Runtime Performance Improvements**
 - SQL function enhancements (inline functions)
 - OVRDBF and REUSEDLT(NO) ★
 - EVI enhancements
- CREATE ENCODED VECTOR INDEX idx2 ON sales (region)**
INCLUDE (SUM(sales))
- SELECT region, SUM(sales) FROM sales GROUP BY region**
- Database Maintenance Performance Improvements**
 - ALTER TABLE performance improvements for partitioned tables
 - ALTER TABLE concurrent performance improvements



7.1 DB2 Enhancements

Limits/Availability/Recovery

SQL Application and Runtime Limits

- 128-byte schema names

Database

- Allow transactions to span *SYSBAS and IASPs
- CHGPF CST CHECK(*YES *NO)
- CHGSR CF TEXT(*FROM MBR) SRCTYPE(*FROM MBR)
- Additional Unicode catalog support
- Preserve statistics on ALTER TABLE
- Deflated table support
- CICS support in IASPs

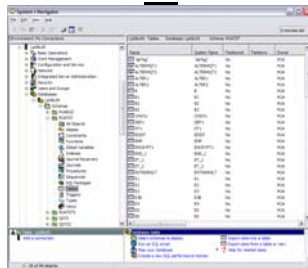
Journaling

- Additional control on whether journal entries for an object should be filtered on remote journaling ★
- Additional filtering based on generic names on STRJRNLIB and CHGJRNBJ to control what journaling implicitly gets started

IBM i Navigator 7.1

On Demand Performance Center

- Database monitor
 - Client register filters
 - Initial allocation
- Authority changes (PTFed to 6.1) ★
- Show Statements
 - Save statements
 - Work Statement With Variables
- Run SQL Scripts – Visual Explain Options
- SQL Details for a Job
 - Result set information
 - Start database monitor
- Index Advisor enhancements
 - MTI information
 - First Advised



Health Center

- SQL0901 Logger ★
- Random and Sequential I/O Counts

Database Management

- Support for new Database Features
 - XML
 - XML Schema Repository (XSR)
 - Global Variables
 - Array Types
 - Field Procedures
 - 3-part alias
 - Enable constraints without checking
- Support for Omnifind
- Generate SQL - include GRANTS
- Long Running Operation Status ★
 - Reorganize status enhancements
 - Index build status
 - Text Search Index build status
 - ALTER TABLE status
- New *SSD and In Memory info
- Object list enhancements
 - Save folder contents
 - Object list filtering
 - List Large list and table list performance
 - Long schema name
- Last build info in SYSPARTITIONINDEXSTAT

