

DB2 for i - Enhancements in 7.2

Power Systems 2014

Tech Talk

Scott Forstie
forstie@us.ibm.com
DB2 for i Business Architect



DB2 for i

- Standard compliant
- Secure
- Scalable
- Functionally Advanced
- Excellent Performance
- Easier to use
- Easier to maintain

Value Proposition

Continual Investment and Innovation

V5R1

SQL triggers
Java Functions
DRDA DUW TCP/IP
2 GB LOBs
1 Terabyte Table
Journal Minimal Data
Two-phase over TCP/IP
DDL Journaling
Database Navigator
Generate SQL

V5R2

SQE Stage 1
IASPs
Identity columns
Savepoints
UNION in views
Scalar subselect
UDTFs
DECLARE GLOBAL TEMPORARY TABLE
Catalog views
JDBC V3.0
DRDA Kerberos
Journal Standby

V5R3

Partitioned tables
UFT-8 and UTF-16
ICU sort sequence
MQTs
Sequences
Implicit char/numeric
BINARY/VARBINARY
GET DIAGNOSTICS
DRDA Alias
DECIMAL(63)
SQE Stage 3
Ragged SWA
QDBRPLAY
Online Reorganize

V5R4

WebQuery
SSD Memory Preference
On Demand Performance Center
Health Center
Completion of SQL Core
Scalar fullselect
Recursive CTE
INSTEAD OF triggers
Descriptor area
XA over DRDA
DDM 2-phase
Scrollable cursor
2M SQL statement
1000 tables in a query
SQE Stage 5
Implicit journaling enhancements

6.1

Omnifind
MySQL storage engine
DECFLOAT
Grouping sets /super groups
INSERT in FROM
VALUES in FROM
Extended Indicator Variables
Expression in Indexes
ROW CHANGE
TIMESTAMP
Statistics catalog views
CLIENT special registers
SQE Stage 6
DDM and DRDA IPv6
Deferred Restore of MQT and Logicals
Environmental limits

7.1

XML Support
Encryption enhancements (FIELDPROCS)
Result set support in embedded SQL
CURRENTLY COMMITTED
MERGE
MQ Functions
Global variables
Array support in procedures
Partition table enhancements
Three-part names and aliases
SQE Logical file support
SQE Adaptive Query Processing
EVI enhancements
Inline functions
CREATE OR REPLACE

7.2

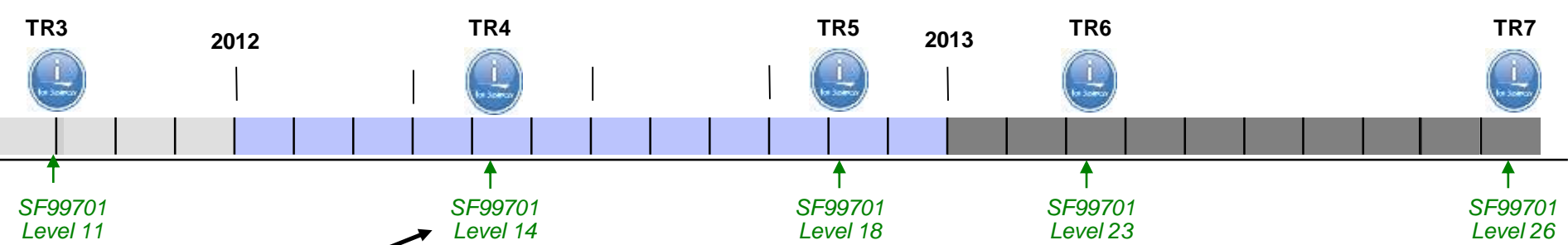
Row and Column Access Control
XMLTABLE
CONNECT BY
TRANSFER OWNERSHIP
Named arguments and defaults for parameters
Obfuscation of SQL routines & triggers
Array support in UDFs
Timestamp precision
Multiple-action Triggers
Built-in Global Variables
Record movement between partitions on UPDATE
1.7 Terabyte Indexes
Health Center – Non-database limits
Navigator Graphing and Charting
SQE I/O Costing model improvement
TRUNCATE

Agenda

- **DB2 for i – Software delivery strategy**
- **DB2 for i & 7.2 - Business Value**
- **Database Enhancements in 7.2:**
 - RCAC (safe, secure, usable, accountable, ...)
 - Performance (faster, faster, faster, ...)
 - Application development (new capabilities)
 - Performance Analyst (new insights)
 - Navigator (your favorite tool has been improved)

DB2 for i – Enhancements delivered via DB2 PTF Groups

IBM i 7.1



TR4 timed Enhancements:

- Performance enhancements for large numbers of row locks
- Automatic management of SQL Plan Cache size
- Field Procedure Masking
- CPYTOIMPF order by
- Many Others...

TR5 timed Enhancements:

- InfoSphere Guardium V9.0 – DB2 for i
- SQE enhancement for Encoded Vector Indexes defined with INCLUDE
- SQL metrics in Collection Services
- Many Others...

TR6 timed Enhancements:

- Database Reorganization (User specified starting point)
- Tracking System Limits (Phase 1)
- Index Advice for ORs
- Many Others...

TR7 timed Enhancements:

- 1.7 TB Indexes
- Tracking System Limits (Phase 2)
- RDB Alias support for 3 part names
- Deferred restore for journals
- Plan Cache properties and controls
- Many Others...

Enhancements delivered by PTF are documented here:

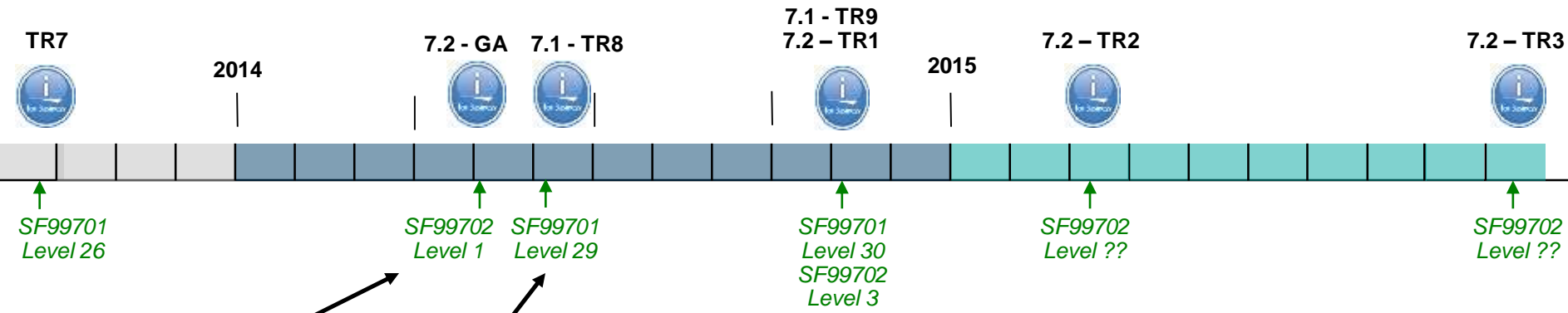
www.ibm.com/developerworks/ibmi/techupdates/db2

& in this article “A Hit Parade of DB2 for i Enhancements”

<http://iprodeveloper.com/database/hit-parade-db2-i-enhancements>

DB2 for i – Enhancements delivered via DB2 PTF Groups

IBM i 7.1 & 7.2



7.2 - GA

- First DB2 PTF Group for 7.2
- All IBM i 7.1 Enhancements thru TR8

7.1 - TR8 timed Enhancements:

- High priority enhancements based upon customer requests
- Enhanced ability to programmatically analyze performance
- New DB2 for i services
- And more...

Redpaper - IBM i 7.2 & RCAC

IBM i 7.2 – Landing page for all 7.2 enhancements:
www.ibm.com/developerworks/ibmi/techupdates/i72

Enhancements delivered by PTF are documented here:

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

DB2 for i enhancements in IBM i 7.2

Security

- Protect business critical data using data-centric design with RCAC
- Secure remote journaling with SSL

Application development

- Improved ability to build, maintain and extend complex database application architectures

Database Engineering (DBE)

- Timestamp precision control (0 thru 12)
- Partitioned tables automatically reposition a row when updated

Performance

- SQL Query Engine (SQE) supports OPNQRYF & Query/400
- Other SQE enhancements
- Use SQL to see temporary storage consumption

Scalable SQE

Data Centric

RCAC

Easy to use

Bet your business on us

Encoded Vector Indexes

Open for Business

Easy to maintain

Intelligent SSD

Secure Proven

DB2 for i Reliable

DB2 for i - 7.2 Enhancements by role - Security

Security & DB2 for i

- Column Masks
→ Deploy “need to know” logic
- Row Permissions
→ Simpler, faster security
- Violation clause for CHECK constraints
→ Protect data integrity
- Secure remote journal using SSL
→ Achieve HA & DR objectives without exposure
- SQL alternative to CHGOBJOWN
→ Embrace separation of duty, using SQL
- Use adopted authority identity within business logic
→ “Who am I?” conditional code using SQL

Data-Centric Security

- Comprehensive
- Auditable
- Sustainable
- Scalable
- Manageable



Data is an asset... what's your data worth?

DB2 for i - 7.2 Enhancements by role - Performance

Database Performance

- SQE enhancements:
 - Support for Native Queries
 - Improved I/O Costing Model
 - Enhanced implementation for IN list processing
- Navigator enhancements:
 - New PDI perspectives
 - Native Queries in the SQL Plan Cache
 - Enhanced SQL Plan Cache detail and tuning
- New system level resource...
Temporary Storage consumption
- And other enhancements that were brought back to IBM i 7.1 alongside Technology Refreshes

Out of the Box... what's **faster?

- OPNQRYF & Query/400 complex queries
- I/O intensive queries
- Queries with long IN lists

** Disclaimer: Realized performance gains depends upon many factors



New Services

- QSYS2/SYSTMPSTG catalog
- QSYS2/DUMP_SNAP_SHOT_PROPERTIES() procedure

DB2 for i - 7.2 Enhancements by role – App Dev

Database Application Development

- Increased timestamp precision
- Named and Default parameter support on UDF/UDTFs
- Use of ARRAYs within UDF/UDTFs
- Obfuscation of SQL triggers
- Built-in Global Variables
- Expressions on PREPARE & EXECUTE IMMEDIATE
- Autonomous procedures
- CURRENT USER special register
- Constants in LANGUAGE SQL routines
- Unified debugger support for SQL functions
- Datetime scalar function improvements
- And other enhancements that were brought back to IBM i 7.1 alongside Technology Refreshes

New SQL Statement

- TRUNCATE

New Built-in Functions

- LPAD()
- RPAD()

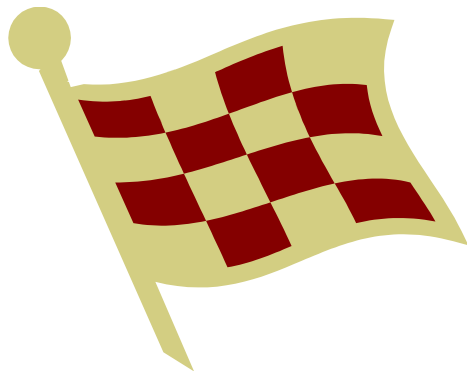
New capabilities for solving business problems with **SQL** and **DB2 for i**



SQL Query Engine (SQE) – Progression

SQE Characteristics

- Object Oriented Design
- Enhanced Performance for complex queries
- Enhanced Optimization Engine
- Separate Statistics Management
- Maintained Temporary Indexes
- Encoded Vector Indexes Enhancements
- Single, System-wide Plan Cache
- and much more...



6.1 → SQE

- Translation support
- Lateral Correlation
- UDTF support
- Optimization time improvements
- Other miscellaneous performance
- Simple Logical File support

7.1 → SQE

- Logical File support
- Adaptive Query Processing (AQP)
- EVI Aggregate capability
- Global Statistics Cache
- Other miscellaneous performance

7.2 → SQE

- Native Opens including Open Query File (OPNQRYP) & Query/400 commands
- Improved I/O costing
- Other miscellaneous performance

DB2 for i & IBM i 7.2 – Other enhancements

DBA/DBE

- Queued exclusive locks control
- SQL Server Mode detail in collection services
- SQL Details for Jobs enhancement
- Improved VARCHAR & LOB space management
- Automatic record movement between partitions

Navigator for DBA/DBE

- **Performance Data Investigator (PDI)**
 - Investigate Data – DB2 category
 - SQL Plan Cache perspectives
 - Physical vs Logical I/O breakdowns
 - And more...
- **On Demand Performance Center**
 - Observance of Native Queries
 - Advanced Monitor Compare



Navigator for DB Application Development

- **Support of all new SQL features**
 - Permissions
 - Masks
 - Named arguments and parameter defaults
 - Obfuscation of Triggers
 - Arrays in user-defined functions
 - Create based ON

DB2 for i & IBM i 7.2 – Reasons to Upgrade



Why move to 7.2?

1. Major improvements for SQL & Native DB users

- Improved database performance, in many cases with zero changes needed by the application or user
- New capabilities to protect business critical data
- Improved insight into database workloads

2. Major improvements for SQL application development

- New SQL statements, special registers, built-in global variables and catalogs
- Enhanced SQL behavior
- Extended capabilities in Navigator & PDI

3. Position your company to receive future DB2 for i enhancements

- Additional DB2 for i enhancements are coming to 7.2 via DB2 PTF Groups on the Technology Refresh (TR) cadence

Note: While 7.2 will include all enhancements, some will also be PTF'd to previous releases

DB Security – 7.2 Enhancements

DB2 for i - 7.2 Enhancements by role - Security

New SQL Statements for security

- CREATE PERMISSION
- ALTER PERMISSION
- CREATE MASK
- ALTER MASK
- ALTER TRIGGER
- TRANSFER OWNERSHIP

New Boss Option

IBM Advanced Data Security for i
(**Boss option 47**)
No Charge

Some of the details...



New Catalogs

- QSYS2/SYSCONTROLS
- QSYS2/SYSCONTROLSDEP

New Built-in Function

- VERIFY_FOR_GROUP_USER()

New Journal Entry Types

For journal code D - Database file:

- M1, M2, M3 for create/drop/alter mask
- P1, P2, P3 for create/drop/alter permission

For journal code T – Audit trail:

- AX for Row and Column Access Control
- X2 for Query manager profile changes

RCAC terms

Base Table	The table (physical file) containing business critical data.
Dependent Object	Any object (file, schema, function, or other object) the permission or mask references.
Permission	<p>A row permission defines a row access control rule for rows of a table by setting an SQL search condition that describes the set of rows a user can access.</p> <p>0 to many allowed per table</p>
Mask	<p>A column mask defines a column access control rule for a specific column in a table by using SQL CASE expression that describes what column values a user is permitted to see and under what conditions.</p> <p>0 or 1 allowed per column</p>
RULETEXT	The expression to be used by the permission or mask.

Row and Column Access Control (RCAC)

```
CREATE MASK SSN MASK ON EMPLOYEE
FOR COLUMN SSN RETURN
CASE
  WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'PAYROLL') = 1)
    THEN SSN
  WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'MGR') = 1)
    THEN 'XXX-XX-' CONCAT SUBSTR (SSN, 8, 4)
  ELSE NULL
END
ENABLE;
ALTER TABLE EMPLOYEE ACTIVATE COLUMN ACCESS CONTROL;
```

```
CREATE PERMISSION PATIENT_TABLE_HMO_PERMISSION
ON HOSPITAL.PATIENT_TABLE
FOR ROWS
WHERE ( (VERIFY_GROUP_FOR_USER (SESSION_USER, 'PCP') = 1 AND
        HOSPITAL.PATIENT_TABLE.PCP_ID = SESSION_USER) OR
        VERIFY_GROUP_FOR_USER (SESSION_USER, 'ACCTGROUP') = 1 OR
        VERIFY_GROUP_FOR_USER (SESSION_USER, 'RESGROUP') = 1)
ENFORCED FOR ALL ACCESS
ENABLE;
ALTER TABLE HOSPITAL.PATIENT_TABLE ACTIVATE ROW ACCESS CONTROL;
```

IBM Advanced Data Security for i
(Boss option 47)
No Charge

Contrasting DB2 for i - Data Security

Technology	Field Procedures	Column Masks	Row Permissions	Views & Logical Files
Use case				
Supported IBM i OS releases	7.1, 7.2	7.2	7.2	6.1, 7.1, 7.2
Limit access to some/all data within a column	Yes	Yes	No	Yes
Limit access to rows	No	No	Yes	Yes
Security logic payload (customer experience)	External program (complex)	SQL rule (simple)	SQL rule (simple)	DDS or SQL (varies)
Software Vendor component	<ul style="list-style-type: none"> • Townsend Security • Linoma • Enforcive 	None at this time	None at this time	N/A
Data encrypted at rest	Yes	No	No	No
Data encrypted in journal	Yes	No	No	No
Deployed over...	SQL Table	DDS File or SQL Table	DDS File or SQL Table	DDS File or SQL Table
Masked values apply to selection criteria	Yes	No	N/A	N/A
Data-Centric Solution	Yes	Yes	Yes	No
Success factors include: Strategy, Tuning & Consulting	Yes	Yes	Yes	No

Contrasting DB2 for i - Data Compliance

Technology	Guardium Activity Monitor for DB2 for i	Audit Journal	Data Journal
Use case			
Supported IBM i OS releases	6.1, 7.1, 7.2	6.1, 7.1, 7.2	6.1, 7.1, 7.2
Analysis & Reporting	InfoSphere Guardium	Security ISVs & InfoSphere Guardium	Security ISVs
Solution infrastructure beyond IBM i	Yes	No	No
Capture SQL statements	Yes	No	No
Capture SQL host variable values and environment	Yes	No	No
Capture database specific Audit Journal details	Yes	Yes	No
Capture before and after images of data	No	No	Yes
Able to track which rows are seen by users	No	No	No
Success factors include: Strategy, Tuning & Consulting	Yes	Yes	Yes

IBM Advanced Data Security for i (Boss Option 47)

- Option 47 must be installed to:
 - CREATE PERMISSION and CREATE MASK (RCAC)
 - Open a file that has RCAC activated
- RCAC constructs exist within the table (*FILE) and get applied by SQE
- RCAC does not replace object authorization requirements
 - If you pass the object authorization check:
 - Row permissions reduce the set of rows returned
 - Column Masks limit full or partial access to sensitive column data
- RCAC is comprehensive and applies to any interface (Native DB, SQL, RPG, APIs, Commands, etc)
- Only users with **QIBM_DB_SECADM** authority can manage RCAC

```
5770SS1  47  IBM Advanced Data Security for i
```

RCAC – Performance tuning

- For performance critical tables, assess impact of RCAC
- Proper application of RCAC will also include:
 - Security strategy
 - Performance objectives
 - Index Strategy
 - Possible removal of Select/Omit logical files and other views
 - Tuning of RCAC rule text
- Review MQT design if the masks or permissions use columns that weren't originally projected from the MQT.
- Index Advisor not extended for RCAC, but RCAC tuning “opportunities” will surface through the advisor
 - Index Advice table
 - Visual Explain → Advised Indexes

Other security oriented SQL statements

TRANSFER OWNERSHIP

SQL statement that is similar to the CL command CHGOBJOWN

```
TRANSFER OWNERSHIP OF TABLE mjatst.t1 TO USER paul  
PRESERVE PRIVILEGES
```

Operation can be run under commitment control and rolled back

Grant to GROUP and USER

Compatibility with DB2 Family

```
GRANT ALL ON council TO USER frank WITH GRANT OPTION
```

```
GRANT ALL ON council TO GROUP marketing WITH GRANT OPTION
```

Identifies whether the ID is a group or a user

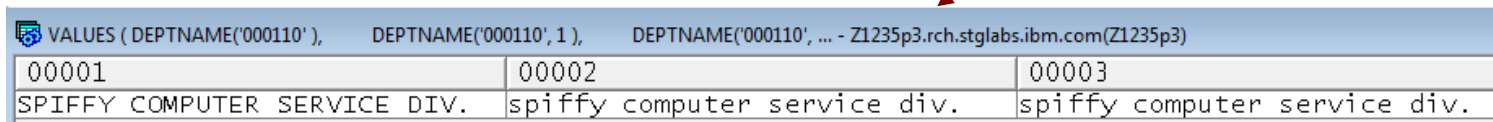
DB Application Development 7.2 Enhancements

Named Arguments & Defaults for User-Defined Function Parameters

- Similar to Named & Default parameters for procedures (IBM i 7.1 → TR5), 7.2 includes support for SQL and External User Defined Functions (UDFs)
- This enhancement brings the usability found with CL Commands to UDFs/UDTFs
- Extend existing functions without fear of breaking existing callers!

```
CREATE OR REPLACE FUNCTION DEPTNAME (
  P_EMPID VARCHAR(6) , P_REQUESTED_IN_LOWER_CASE INTEGER DEFAULT 0
)
RETURNS VARCHAR(30)
LANGUAGE SQL
D : BEGIN ATOMIC
DECLARE V_DEPARTMENT_NAME VARCHAR ( 30 ) ;
DECLARE V_ERR VARCHAR ( 70 ) ;
SET V_DEPARTMENT_NAME = (
  SELECT CASE WHEN P_REQUESTED_IN_LOWER_CASE = 0 THEN D . DEPTNAME
            ELSE LOWER(D . DEPTNAME) END CASE
        FROM DEPARTMENT D , EMPLOYEE E
        WHERE E . WORKDEPT = D . DEPTNO AND
              E . EMPNO = P_EMPID ) ;
IF V_DEPARTMENT_NAME IS NULL THEN
  SET V_ERR = 'Error: employee ' CONCAT P_EMPID CONCAT ' was not found'
;
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = V_ERR ;
END IF ;
RETURN V_DEPARTMENT_NAME;
END D ;
```

```
VALUES ( DEPTNAME('000110' ),
        DEPTNAME('000110', 1 ),
        DEPTNAME('000110',
                  P_REQUESTED_IN_LOWER_CASE=>1))
```



VALUES (DEPTNAME('000110'),	DEPTNAME('000110', 1),	DEPTNAME('000110', ... - Z1235p3.rch.stglabs.ibm.com(Z1235p3)
00001	00002	00003
SPIFFY COMPUTER SERVICE DIV.	spiffy computer service div.	spiffy computer service div.

Named Arguments and Defaults for User-Defined Function Parameters

Before 7.2 (must specify 15 parameters):

```

SELECT journal_code, journal_entry_type, object, object_type, X.*
FROM TABLE (
  QSYS2.Display_Journal(
    'PRODDATA', 'QSQRN',           -- Journal library and name
    "",                               -- Receiver library and name
    CAST(null as TIMESTAMP),        -- Starting timestamp
    CAST(null as DECIMAL(21,0)),    -- Starting sequence number
    "",                               -- Journal codes
    "",                               -- Journal entries
    "",                               -- Object library, Object name, Object type, Object member
    "","",                           -- User
    'SCOTT',                          -- User
    "",                               -- Job
    ""                                -- Program
  ) ) AS x
ORDER BY entry_timestamp DESC

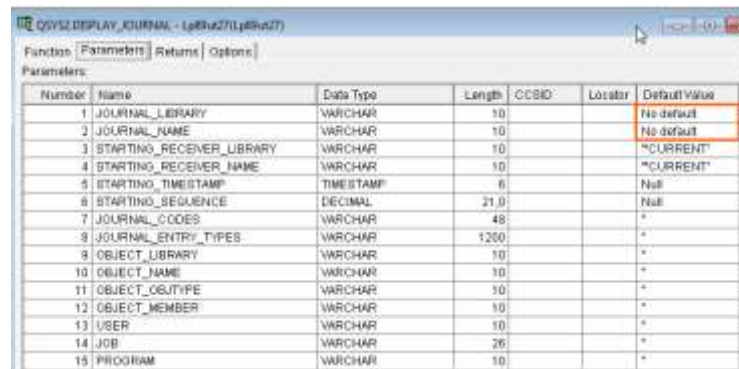
```

With 7.2 (leverage the optional parameters):

```

SELECT journal_code, journal_entry_type, object, object_type, X.*
FROM TABLE (
  QSYS2.Display_Journal(
    'PRODDATA', 'QSQRN',           -- Journal library and name
    "USER" => 'SCOTT'             -- User
  ) ) AS x
ORDER BY entry_timestamp DESC

```



Number	Name	Data Type	Length	CCSID	Locator	Default Value
1	JOURNAL_LIBRARY	VARCHAR	10			No default
2	JOURNAL_NAME	VARCHAR	10			No default
3	STARTING_RECEIVER_LIBRARY	VARCHAR	10			"CURRENT"
4	STARTING_RECEIVER_NAME	VARCHAR	10			"CURRENT"
5	STARTING_TIMESTAMP	TIMESTAMP	6			Null
6	STARTING_SEQUENCE	DECIMAL	21 0			Null
7	JOURNAL_CODES	VARCHAR	48			*
8	JOURNAL_ENTRY_TYPES	VARCHAR	1200			*
9	OBJECT_LIBRARY	VARCHAR	10			*
10	OBJECT_NAME	VARCHAR	10			*
11	OBJECT_OBJECTTYPE	VARCHAR	10			*
12	OBJECT_MEMBER	VARCHAR	10			*
13	USER	VARCHAR	10			*
14	JOB	VARCHAR	26			*
15	PROGRAM	VARCHAR	10			*

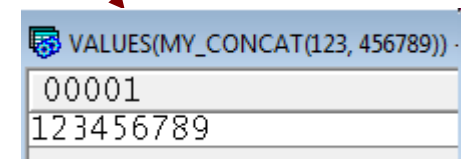
Function resolution using casting rules

- Prior to 7.2, function resolution looked an exact match
 - Match on function name
 - Match on # of parameters
 - Match on data type of parameters
- With 7.2, if DB2 for i doesn't find an exact match, it looks for the "best fit"
- Read the SQL Reference rules for details
- Basic rule, if CAST() is supported for the parameter data type mismatch, the function will be found
- Prior to this support, you would observe SQL0204 – Function not found

For example:

```
CREATE OR REPLACE FUNCTION MY_CONCAT (  
    FIRST_PART CHAR(10),  
    SECOND_PART CHAR(50))  
RETURNS VARCHAR(60)  
LANGUAGE SQL  
BEGIN  
RETURN(FIRST_PART CONCAT SECOND_PART);  
END;
```

VALUES(MY_CONCAT(123, 456789))



VALUES(MY_CONCAT(123, 456789))
00001
123456789

Function resolution using casting rules

- Character literal values are considered VARCHAR
- Passing character literal values to functions prior to 7.1 was difficult/annoying

For example:

```
CREATE FUNCTION How_Long(NAME CHAR(30))  
RETURNS INT  
RETURN LENGTH(NAME);
```

VALUES(How_Long('a b c'))

Prior to 7.2

```
> VALUES(How_Long('a b c'))
```

```
SQL State: 42704  
Vendor Code: -204
```

```
Message: [SQL0204] HOW_LONG in *LIBL type *N not found
```

With 7.2

VALUES(How_Long('a b c')) .	
00001	
30	

Array Support in User-Defined Functions

Create a type that is an array

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

Create an SQL function that uses an the array type.

```
CREATE FUNCTION myfunction1 (Input_Identifiers INTARRAY)  
  RETURNS INTARRAY  
BEGIN  
  DECLARE ids intArray;  
  DECLARE c2 CURSOR FOR SELECT * FROM UNNEST(Input_Identifiers) 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));  
  ...  
  RETURN ids;  
END
```

Timestamp Precision

Provides the ability to specify between 0 and 12 digits of precision

- Prior to IBM i 7.2, we only support 6 digits of timestamp precision
- For some applications this is no longer sufficient as systems get faster with many more processors.
- In other cases, this is more than needed
- Use ALTER TABLE to adjust existing tables
- Any precision between 0 and 12 is allowed

CREATE TABLE x

(C1 **TIMESTAMP(12)**, → Additional precision when 6 is not enough
(moving from 6→12 consumes 3 additional bytes)

C2 **TIMESTAMP(0)** → Less precision (and storage) when 6 isn't needed
(moving from 6→0 eliminates 3 bytes)

Timestamp Precision → digging deeper

Example: Compare the results of mixed precision time

```
create table corpdb.time_travel (
old_time timestamp, /* identical to timestamp(6) */
new_time timestamp(12), /* maximum precision */
no_time timestamp(0),
Last_Change TIMESTAMP NOT NULL IMPLICITLY HIDDEN FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP)
insert into corpdb.time_travel values(current timestamp, current timestamp, current timestamp)
insert into corpdb.time_travel values(current timestamp, current timestamp(12), current timestamp)

select old_time, new_time, no_time, last_change from corpdb.time_travel
```

OLD_TIME	NEW_TIME	NO_TIME	LAST_CHANGE
2014-03-14 17:59:11.591834	2014-03-14 17:59:11.591834000000	2014-03-14 17:59:11	2014-03-14 17:59:11.616310
2014-03-14 17:59:12.560759	2014-03-14 17:59:12.560759566894	2014-03-14 17:59:12	2014-03-14 17:59:12.573743

```
select new_time - last_change as new_minus_last,
new_time - old_time as new_minus_old,
new_time - no_time as new_minus_no from corpdb.time_travel
```

NEW_MINUS_LAST	NEW_MINUS_OLD	NEW_MINUS_NO
-0.024476000000	0.000000000000	0.591834000000
-0.012983433106	0.000000566894	0.560759566894

The last change time occurs after the current timestamp is captured

Both the column precision & special register precision must change to achieve greater precision

DB2 for i support includes implicit data type and precision conversion

Timestamp Precision – Application interfaces

Toolbox JDBC

- Since the Java `java.sql.Timestamp` object only supports **9 digits of precision**, a new class **AS400JDBCTimestamp** has been created to handle up to 12 digits of precision

Native JDBC

- Since the Java `java.sql.Timestamp` object only supports **9 digits of precision**, a new class, **DB2JDBCTimestamp**, has been created to handle up to 12 digits of precision

OLEDB – Full precision existed already because timestamps are represented via strings

SQL CLI

- To allow SQL CLI to fully support timestamp precision, a compatibility change has been made and is documented in the IBM i 7.2 Memorandum To Users (MTU)
- A new connection attribute (**SQL_ATTR_TIMESTAMP_PREC**) can be used to revert SQL CLI to the behavior previous to 7.2, where invalid lengths were permitted on some SQL CLI APIs.

Client ODBC

- When bound using a character type, 12 digits of precision can be used
- `SQL_TIMESTAMP_STRUCT` supports timestamp precision up to **9 digits of precision**.
- Similar to DB2 for LUW and DB2 for z/OS, looking to add a new structure and data type for precision up to 12 digits of precision

.NET – Supports timestamps with precision of 6. Looking to add support for greater precision.
[php-odbc](#) & [pdo_odbc](#) – these php extensions bind timestamps to character strings and use `SQLDescribeCol/SQLDrescribeParam` to determine the digits of precision to use, so they should work fine when the target column has 12 digits of precision.

Timestamp Precision – Programming interfaces

ILE RPG

- Timestamp precision specification: **TIMESTAMP(m)**

RPG free-form syntax matches the SQL language

<u>Column precision:</u>	<u>RPG encoding:</u>
TIMESTAMP(0)	TIMESTAMP(0)
TIMESTAMP(6)	TIMESTAMP(6)
TIMESTAMP(12)	TIMESTAMP(12)

ILE COBOL

- Timestamp precision specification: **TIMESTAMP SIZE n**

TIMESTAMP(0) → when n = 19
TIMESTAMP(n-20) → when n > 20

<u>Column precision:</u>	<u>COBOL encoding:</u>
TIMESTAMP(0)	TIMESTAMP SIZE 19
TIMESTAMP(6)	TIMESTAMP SIZE 26
TIMESTAMP(12)	TIMESTAMP SIZE 32

TRUNCATE

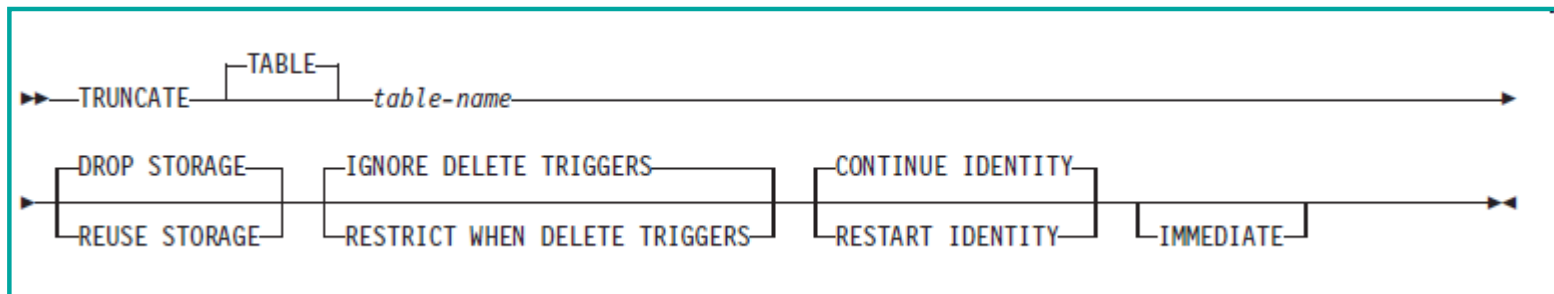
- Similar to “fast delete”
- Additional functions to
 - IGNORE or RESTRICT when delete triggers are present
 - CONTINUE or RESTART identity values
 - DROP or REUSE storage
 - IMMEDIATE performs the operation without commit even if running under commit

TRUNCATE Order_Table IGNORE DELETE TRIGGERS

TRUNCATE Order_Table RESTRICT WHEN DELETE TRIGGERS IMMEDIATE

TRUNCATE Order_Table CONTINUE IDENTITY

TRUNCATE Order_Table RESTART IDENTITY IMMEDIATE



Built-in Global Variables

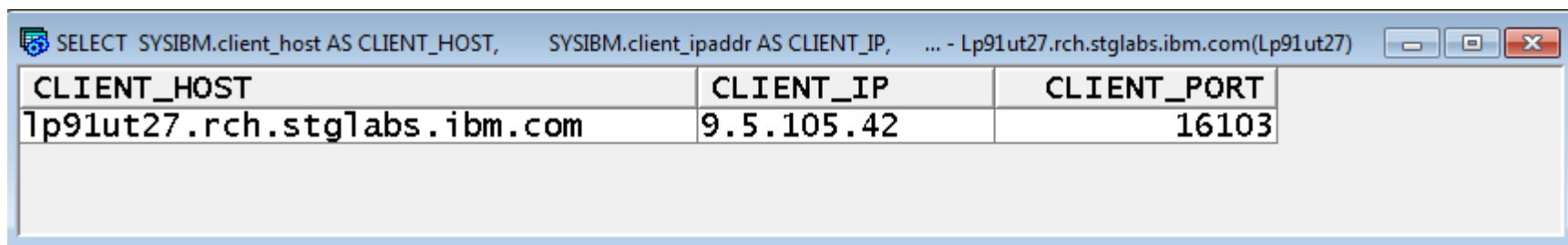
- Can be referenced anywhere a column name can be used
- DB2 for i maintains the value
- Can't be the targets of a data change operation (not settable)
- Will be set to NULL when not applicable

Variable name	Schema	Data Type	Size
CLIENT_IPADDR	SYSIBM	VARCHAR	128
CLIENT_HOST	SYSIBM	VARCHAR	255
CLIENT_PORT	SYSIBM	INTEGER	-
PACKAGE_NAME	SYSIBM	VARCHAR	128
PACKAGE_SCHEMA	SYSIBM	VARCHAR	128
PACKAGE_VERSION	SYSIBM	VARCHAR	64
ROUTINE_SCHEMA	SYSIBM	VARCHAR	128
ROUTINE_SPECIFIC_NAME	SYSIBM	VARCHAR	128
ROUTINE_TYPE	SYSIBM	CHAR	1

Built-in Global Variables – Client information

```
SELECT SYSIBM.client_host AS CLIENT_HOST,  
       SYSIBM.client_ipaddr AS CLIENT_IP,  
       SYSIBM.client_port AS CLIENT_PORT  
FROM LP92UT27.SYSIBM.SYSDUMMY1
```

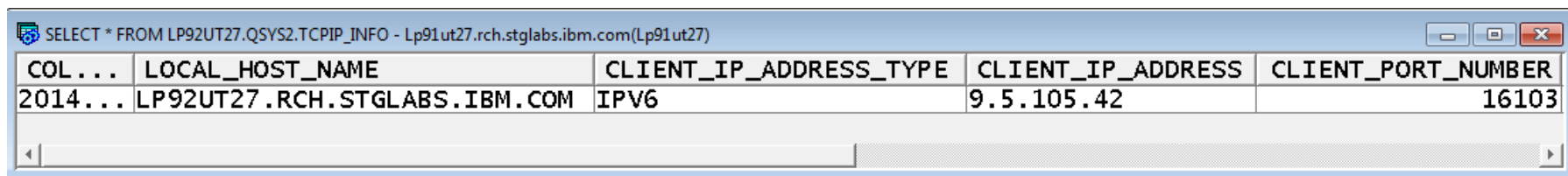
- Two ways to extract the detail
- Global variables fit nicely into View definitions & RCAC masks/permissions



SELECT SYSIBM.client_host AS CLIENT_HOST, SYSIBM.client_ipaddr AS CLIENT_IP, ... - Lp91ut27.rch.stglabs.ibm.com(Lp91ut27)

CLIENT_HOST	CLIENT_IP	CLIENT_PORT
lp91ut27.rch.stglabs.ibm.com	9.5.105.42	16103

```
SELECT * FROM LP92UT27.QSYS2.TCPIP_INFO
```



SELECT * FROM LP92UT27.QSYS2.TCPIP_INFO - Lp91ut27.rch.stglabs.ibm.com(Lp91ut27)

COL ...	LOCAL_HOST_NAME	CLIENT_IP_ADDRESS_TYPE	CLIENT_IP_ADDRESS	CLIENT_PORT_NUMBER
2014...	LP92UT27.RCH.STGLABS.IBM.COM	IPV6	9.5.105.42	16103

Built-in Global Variables – Routine information

```
create procedure scottf.show_GV_values(  
routine_schema_p out varchar(128),  
routine_specific_name_p out varchar(128),  
routine_type_p out char(1))  
language sql  
specific scottf.demonstrate_the_new_DB2_for_i_GVs  
begin  
SET (routine_schema_p,routine_specific_name_p,routine_type_p) =  
(SYSIBM.ROUTINE_SCHEMA, SYSIBM.ROUTINE_SPECIFIC_NAME,  
SYSIBM.ROUTINE_TYPE);  
end
```

- Variables reflect the currently executing routine (procedure or function)

```
CALL scottf.show_GV_values(?,?,?)
```

```
> CALL scottf.show_GV_values(?,?,?)  
Return Code = 0  
Output Parameter #1 = SCOTTf  
Output Parameter #2 = DEMONSTRATE_THE_NEW_DB2_FOR_I_GVS  
Output Parameter #3 = P
```

Built-in Global Variables – Routine information

create or replace function scottf.show_specific_name_function()

returns clob(1K)

language sql

deterministic

Begin

- Implement caller-specific logic within a trigger
- Extend your product logging detail

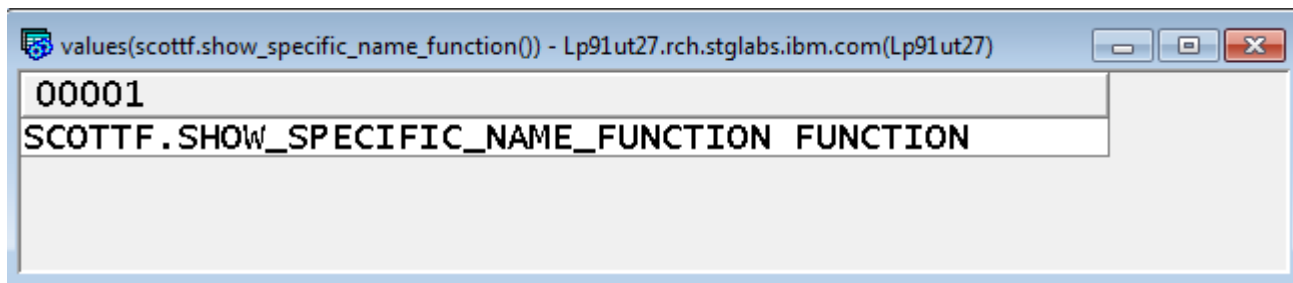
```
RETURN(SYSIBM.ROUTINE_SCHEMA CONCAT '.' CONCAT  
        SYSIBM.ROUTINE_SPECIFIC_NAME CONCAT
```

```
CASE WHEN SYSIBM.ROUTINE_TYPE = 'P' THEN ' PROCEDURE'  
      WHEN SYSIBM.ROUTINE_TYPE = 'F' THEN ' FUNCTION'
```

```
END);
```

```
end
```

```
values(scottf.show_specific_name_function())
```



CURRENT_USER special register

The CURRENT_USER special register specifies the primary authorization ID that is being used for statement authorization. In other words, in a program that adopts authority, it will return the adopted profile name. Useful in masks and permissions.

When multiple authorization IDs have been adopted within a thread, the value of the **most recently adopted authorization ID** within the thread will be returned.

```
SELECT CURRENT_USER FROM SYSIBM.SYSDUMMY1
```

```
CREATE MASK SSN_MASK ON EMPLOYEE
FOR COLUMN SSN RETURN
CASE
  WHEN (VERIFY_GROUP_FOR_USER(CURRENT_USER,'PAYROLL') = 1)
    THEN SSN
  WHEN (VERIFY_GROUP_FOR_USER(CURRENT_USER,'MGR') = 1)
    THEN 'XXX-XX-' CONCAT SUBSTR(SSN,8,4)
  ELSE NULL
END
ENABLE;
```

Special registers – similar names, different purposes

The name **CURRENT USER** could easily be misunderstood.

Special Register	Definition
USER or SESSION_USER	The <u>effective user</u> of the thread is returned.
SYSTEM_USER	The authorization ID that <u>initiated the connection</u> is returned.
CURRENT USER or CURRENT_USER	The most recently <u>adopted authorization ID</u> within the thread will be returned. When no adopted authority has occurred, the effective user of the thread is returned.

Expressions in PREPARE and EXECUTE IMMEDIATE

Before:

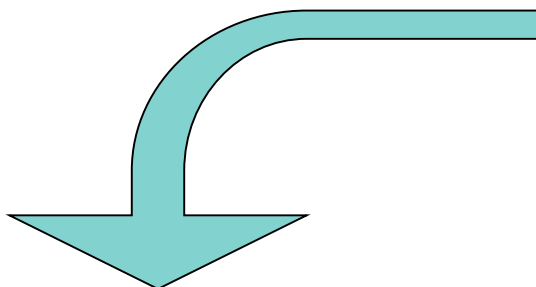
```
SET var_total_stmt = var_select_stmt CONCAT var_orderby;
PREPARE stmt1 FROM var_total_stmt;
```

After:

```
PREPARE stmt1 FROM var_select_stmt CONCAT var_orderby;
```

Simplifies use of PREPARE and EXECUTE IMMEDIATE

One statement replaces many...



```
SET INSERT_STMT = 'INSERT INTO QTEMP.TMPIDXADV SELECT * FROM QSYS2.CONDENSEDINDEXADVICE WHERE ' ;
IF ( P_LIBRARY IS NOT NULL ) THEN
    SET WHERE_CLAUSE = ' TABLE_SCHEMA = '' CONCAT
        RTRIM ( P_LIBRARY ) CONCAT '' AND ' ;
ELSE
    SET WHERE_CLAUSE = ' ' ;
END IF ;
IF ( P_FILE IS NOT NULL ) THEN
    SET WHERE_CLAUSE = WHERE_CLAUSE CONCAT ' SYSTEM_TABLE_NAME = ''
    CONCAT RTRIM ( P_FILE ) CONCAT '' AND ' ;
END IF ;
IF ( P_TIMES_ADVISED IS NOT NULL ) THEN
    SET WHERE_CLAUSE = WHERE_CLAUSE CONCAT ' TIMES_ADVISED >= '
    CONCAT P_TIMES_ADVISED CONCAT ' AND ' ;
END IF ;
IF ( P_MTI_USED IS NOT NULL ) THEN
    SET WHERE_CLAUSE = WHERE_CLAUSE CONCAT ' MTI_USED >= '
    CONCAT P_MTI_USED CONCAT ' AND ' ;
END IF ;
IF ( P_AVERAGE_QUERY_ESTIMATE IS NOT NULL ) THEN
    SET WHERE_CLAUSE = WHERE_CLAUSE CONCAT
        ' AVERAGE_QUERY_ESTIMATE >= ' CONCAT
        P_AVERAGE_QUERY_ESTIMATE CONCAT ' AND ' ;
END IF ;
SET WHERE_CLAUSE = WHERE_CLAUSE CONCAT ' NLSS_TABLE_NAME = ''*HEX'' ' ;
SET INSERT_STMT = INSERT_STMT CONCAT WHERE_CLAUSE ;
EXECUTE IMMEDIATE INSERT_STMT ;
```

```
EXECUTE IMMEDIATE 'INSERT INTO QTEMP.TMPIDXADV SELECT * FROM QSYS2.CONDENSEDINDEXADVICE WHERE ' CONCAT
CASE WHEN P_LIBRARY IS NOT NULL THEN ' TABLE_SCHEMA = '' CONCAT RTRIM ( P_LIBRARY ) CONCAT '' AND ' ELSE '' END CONCAT
CASE WHEN P_FILE IS NOT NULL THEN ' SYSTEM_TABLE_NAME = '' CONCAT RTRIM ( P_FILE ) CONCAT '' AND ' ELSE '' END CONCAT
CASE WHEN P_TIMES_ADVISED IS NOT NULL THEN ' TIMES_ADVISED >= ' CONCAT P_TIMES_ADVISED CONCAT ' AND ' ELSE '' END CONCAT
CASE WHEN P_MTI_USED IS NOT NULL THEN ' MTI_USED >= ' CONCAT P_MTI_USED CONCAT ' AND ' ELSE '' END CONCAT
CASE WHEN P_AVERAGE_QUERY_ESTIMATE IS NOT NULL THEN ' AVERAGE_QUERY_ESTIMATE >= ' CONCAT P_AVERAGE_QUERY_ESTIMATE CONCAT ' AND ' ELSE '' END CONCAT
' NLSS_TABLE_NAME = ''*HEX'' ' ;
```

Obfuscation of SQL Triggers

- Obfuscation provides the capability of optionally obfuscating proprietary SQL statements and logic within SQL procedures, functions & triggers
- **This support can be used to prevent others from seeing or changing SQL routines & triggers**

Generate SQL - Z1235p3.rch.stglabs.ibm.com(Z1235p3)

SQL will be generated for the following objects:

Name	Schema	Type
CL_TRIG1	BURNER1D	Trigger
DEFAULT_CLASS_END	BURNER1D	Trigger
GLOB_TEMP_TRIG	BURNER1D	Trigger
INSERT_EMPLOYEE	BURNER1D	Trigger
SHOW_EVPT_TRIG	BURNER1D	Trigger
SHOW_TRANS_TRIG	BURNER1D	Trigger
SHOW_TRANS_TRIG2	BURNER1D	Trigger
VALIDATE_SCHED	BURNER1D	Trigger

Output Options | Format |

Standards

ANSI/ISO

DB2 family

Extensions

Output

Statements formatted for readability

SQL privilege statements

System names for objects

Schema qualify names for objects

Column CCSID values

Drop statements

Labels and comments

Associated constraints and triggers (for table objects)

Informational messages

OR REPLACE clause

Obfuscate (for SQL function, procedure, and trigger objects)



```
-- Generate SQL
-- Version:                V7R2M0 140418
-- Generated on:           04/28/14 14:28:23
-- Relational Database:   Z1235P3
-- Standards Option:      DB2 for i
CREATE OR REPLACE TRIGGER BURNER1D.CL_TRIG1
WRAPPED QSQQO7020
aacxw8p1w8FrG8prG8VHG8Fr68VH68Vn18:d38FF3qpdw8pdw8phZ9ptj
Jx4d0LxXjXqbP5fuWetaWctaYG3Km2qs_euPemNwEOyNNQ5q41Tkent3j
83IJy2ATY8Np2mIe:HdfX7DGmeePjflIBwICJg72670tdG4BeIJ3Rwnqt
y07obdRL5TirawVVFGF05Ia5nDnETgUDDiSaIzezYrIR8QkFQGTpV:kRe
5icuopwbJFLZzrMy4bb28yam9uppSDz2CJ6dG3nf2:opWBtKuy5l_DtP
oTOUH:d:BvqXjQHQjBukEEcJ0ziZcrW:jp1P_N2q270G0i uw_Sb9Ph4hx
5EK9yxYt2kXahN:If458Q3_M7CTwRtD0cJyai0ni dr:z8_9VQyW:FnZHf
t5wy5UvcvVlKXCj1jNtIXvPa7jLh2aQTjMGC8Lkz_YUJJ2FrRVM8BXvLG
e30PIfivRFQM2vSzzQBKQrL5duthWg4A0zPIWYqcbjKmM1Uv5T1a72uP0
zjp5EkhtSJjimpYH0Bf5LlNoPvafv4c2d:5FTS0QIgvHjTaoqTSHMaDvI
refEIJDCDD0sQycVGQt3EzvLbzKkNLE:6TdHZFEhMo42mQ0AW441YCrA_
GcEYqHg3I_lBRQ2uFxpXcaEVl0jqctmQbL0PG64yavEwvKeuSX7va6Gj
gueZZbIapzQBMwhezda05Zi5dY0vUPQDDo1KGfVhoWuJxqkUsdVwc_v_c
AiqrXYJ6Ihz8D7DMEihy1DMyc4RX1zEr0EuIjo7E34X92TT6x8wLHFG0
7dDrX:PyrrPF5etEKnoPpl1TN:LKEtTqZ6SwKmV3Lhk1IdtGpzpbkVUBaU
ANpi04Dk4HWwD2c6e3m0A1oAwPippl1sF5htGk1JW_oF;

CREATE OR REPLACE TRIGGER BURNER1D.DEFAULT_CLASS_END
WRAPPED QSQQO7020
aacxw8p1w8FrG8prG8VHG8Fr68VH68Vn18:F18ph5qpdw8pdw8pdZ9:dK
u3rRixVh9ZhyLwVmkHXN_SXp10T58V10e17isGqBJTa7FF1fkvrEGZM0
T4NmyOf48YyBGRDJvPvS6U4ogATNzxvtT3RRYqqEVsGi2NcAdvOpFzw79
01p5NyZDXAdJ04sjsZBpBJdG_2Vnyk9AkmmL_9mYTERdcXM55nngAFQAEi
zR9m85_Fzf1t21djr57iUQEFwLu6HhQGVc1714o1MvbjTj0fsjSzhy85j

CREATE OR REPLACE TRIGGER BURNER1D.GLOB_TEMP_TRIG
WRAPPED QSQQO7020
aacxw8p1w8FrG8prG8VHG8Fr68VH68Vn19phW8Vd5qpdw8pdw8pd09FhI
```

**Obfuscation of Procedures & Functions
is available at IBM i 7.1**

Autonomous Procedures

An autonomous procedure is one that is executed in a unit of work that is independent from the calling application.

Similar to running **COMMIT(*NONE)** except that you can do commit or rollback inside the autonomous procedure and the commit and rollback is independent of the calling application.

```
CREATE PROCEDURE writelog (loginfo VARCHAR(1000)  
AUTONOMOUS  
BEGIN  
...  
INSERT INTO MJATST.T1 VALUES(LOGINFO);  
...  
END
```

Autonomous procedures use the named activation group 'QSQAUTOAG'.

Queued exclusive locks

- New option to direct the database to give preference to an operation that requires an exclusive lock. Use the new QAQQINI option to prevent additional conflicting locks until the operation succeeds or times out.
 - ALTER TABLE
 - CREATE TRIGGER
 - LOCK TABLE
 - RENAME TABLE

- New QAQQINI option - **PREVENT_ADDITIONAL_CONFLICTING_LOCKS**

<p>PREVENT_ADDITIONAL_CONFLICTING_LOCKS</p> <p>The following SQL DDL statements require an exclusive, no read lock on the target table.</p> <p>If the application activity cannot be quiesced, it can be hard to accomplish these operations.</p> <p>The PREVENT_ADDITIONAL_CONFLICTING_LOCKS QAQQINI option provides a control for customers to use to direct the operating system to favor a request for an exclusive, no read lock over new requests to lock the object for reading.</p>	*DEFAULT	The default value is set to *NO
	*NO	When a job requests an exclusive lock on an object, do not prevent concurrent jobs from acquiring additional locks on the object.
	*YES	When *YES is chosen, any new requests for these lower-level read locks will be kept behind the exclusive lock request and could surface to applications as the table is unavailable for use for querying. <ul style="list-style-type: none"> • ALTER TABLE (Add, Alter or Drop Column) • CREATE TRIGGER • LOCK TABLE • RENAME TABLE

DB Performance – 7.2 Enhancements

SQE - I/O Cost Model Enhancement

- **Motivation**

- Query optimizer needs to know how much time it's going to take to access an object (scan a table, probe an index)
- Original I/O Cost model assumption: 25 msec access time per I/O for all H/W
- Hardware has changed *a lot* since V5R2 (e.g. faster and smaller HDDs, external storage, SSDs)
- I/O implementation and execution algorithms diverged from model over time
- Big Data paradigm means more dependency on I/O cost model correctness

- **Costing Change @ 7.2**

- New method to **sample actual access times** rather than hard-code a fixed time; Every system will have unique I/O performance metrics tracked over time
- Sampling enables optimizer to distinguish unique performance characteristics of internal, external, and solid state storage devices

- **Result**

- **SQE has more accurate I/O detail when costing plans**



SQE - I/O Cost Model Enhancement

- Typical access times for modern disks:

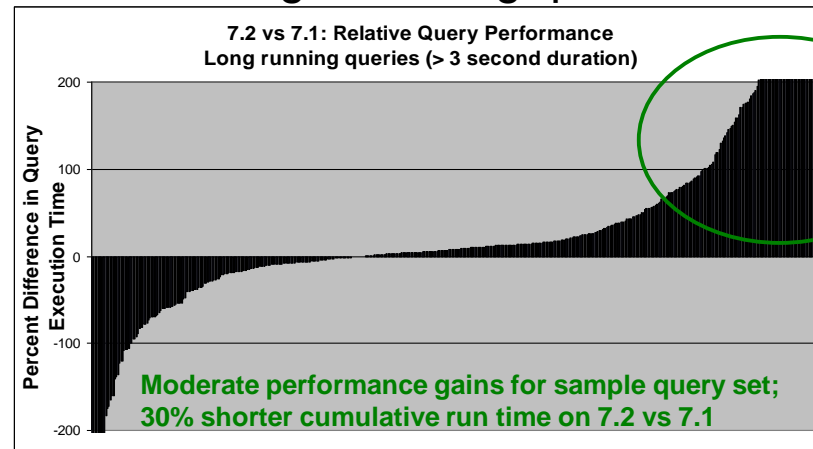
HDDs	4-10 msec
External Disks/SAN	1-6 msec
SSDs	< 1 msec
Original SQE Model	25 msec

- Relative cost of I/O is now lower; SQE processes I/O more aggressively

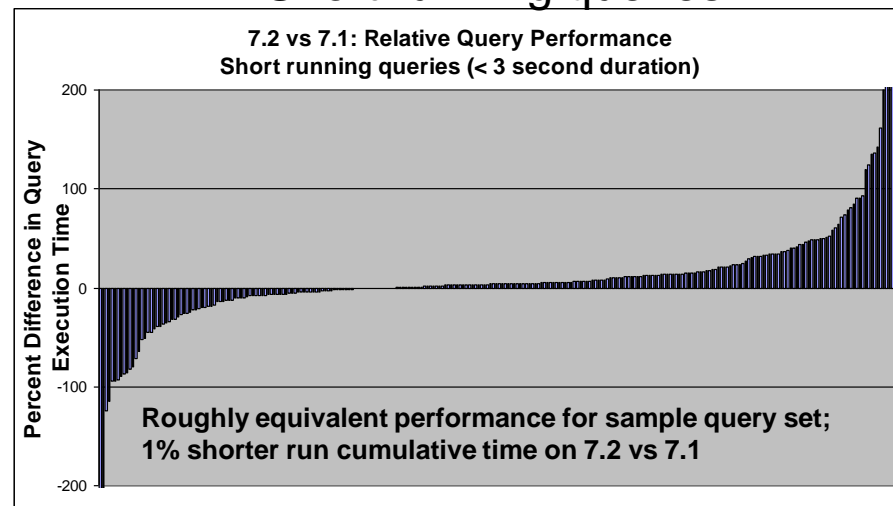
- **Potential for moderate performance gains on OLAP queries which drive significant I/O**

- Negligible benefit for short running OLTP queries which drive little I/O

Longer running queries

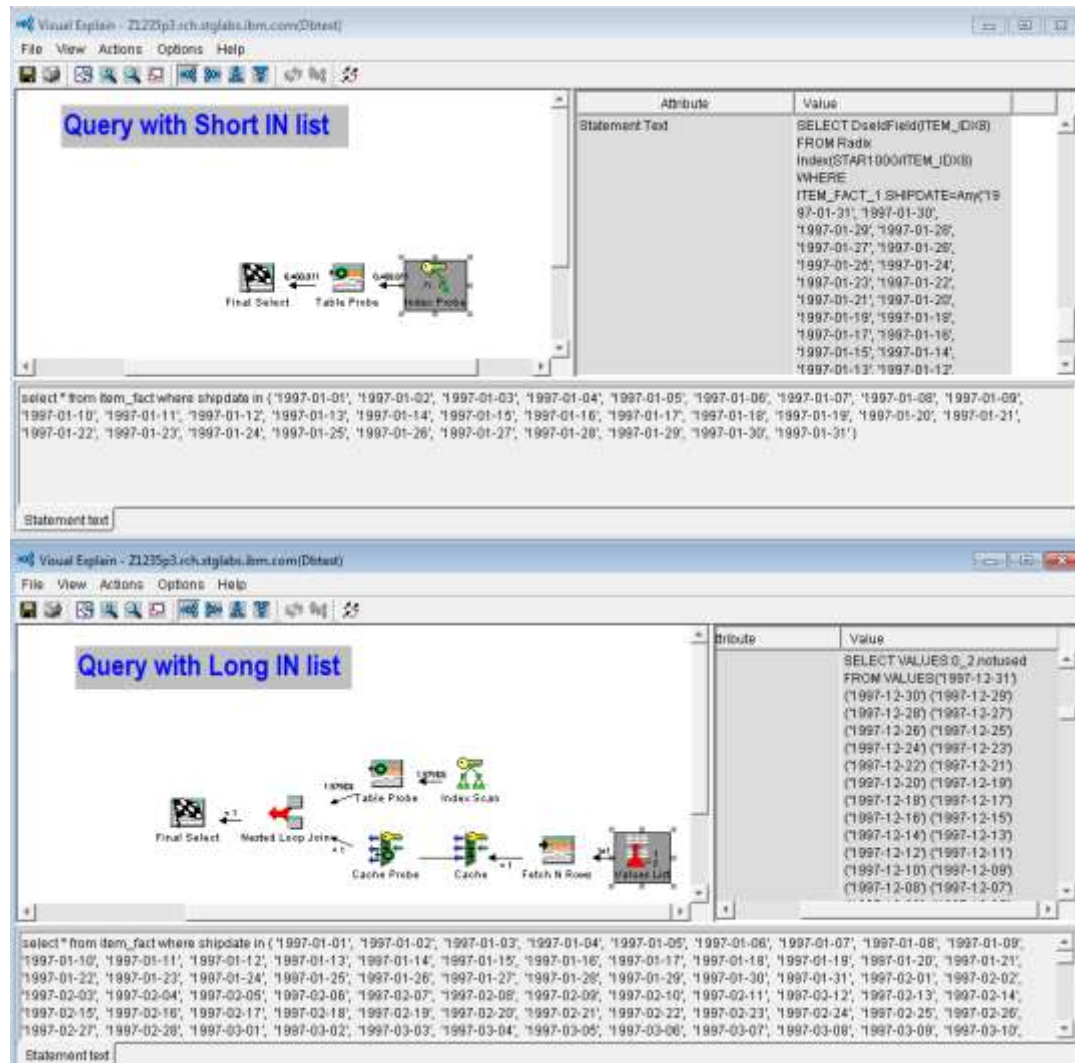


Short running queries



Long IN list queries

- SQL generators frequently use the IN predicate with a series of values based upon user selection criteria
- New with IBM i 7.2, when the number of values specified on the IN list becomes large (>80), the query is automatically converted into an INNER JOIN
- The INNER JOIN against a keyed list of values results in a more efficient implementation of the query
- **No changes or awareness are needed by the application... it just runs faster**



The top screenshot, titled "Query with Short IN list", shows a query plan with a single "Index Probe" step. The SQL statement is: `select * from Item_fact where shipdate in ('1997-01-01', '1997-01-02', '1997-01-03', '1997-01-04', '1997-01-05', '1997-01-06', '1997-01-07', '1997-01-08', '1997-01-09', '1997-01-10', '1997-01-11', '1997-01-12', '1997-01-13', '1997-01-14', '1997-01-15', '1997-01-16', '1997-01-17', '1997-01-18', '1997-01-19', '1997-01-20', '1997-01-21', '1997-01-22', '1997-01-23', '1997-01-24', '1997-01-25', '1997-01-26', '1997-01-27', '1997-01-28', '1997-01-29', '1997-01-30', '1997-01-31')`

The bottom screenshot, titled "Query with Long IN list", shows a query plan with an "Inner Join" step. The SQL statement is: `select * from item_fact where shipdate in ('1997-01-01', '1997-01-02', '1997-01-03', '1997-01-04', '1997-01-05', '1997-01-06', '1997-01-07', '1997-01-08', '1997-01-09', '1997-01-10', '1997-01-11', '1997-01-12', '1997-01-13', '1997-01-14', '1997-01-15', '1997-01-16', '1997-01-17', '1997-01-18', '1997-01-19', '1997-01-20', '1997-01-21', '1997-01-22', '1997-01-23', '1997-01-24', '1997-01-25', '1997-01-26', '1997-01-27', '1997-01-28', '1997-01-29', '1997-01-30', '1997-01-31', '1997-02-01', '1997-02-02', '1997-02-03', '1997-02-04', '1997-02-05', '1997-02-06', '1997-02-07', '1997-02-08', '1997-02-09', '1997-02-10', '1997-02-11', '1997-02-12', '1997-02-13', '1997-02-14', '1997-02-15', '1997-02-16', '1997-02-17', '1997-02-18', '1997-02-19', '1997-02-20', '1997-02-21', '1997-02-22', '1997-02-23', '1997-02-24', '1997-02-25', '1997-02-26', '1997-02-27', '1997-02-28', '1997-03-01', '1997-03-02', '1997-03-03', '1997-03-04', '1997-03-05', '1997-03-06', '1997-03-07', '1997-03-08', '1997-03-09', '1997-03-10')`

DB2 for i & In-Memory controls

➤ Set Object Access (SETOBJACC) command (tenured service)

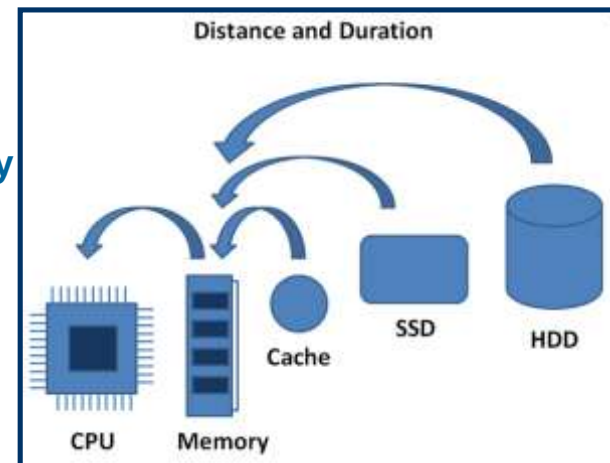
- Target physical & logical files and programs
- Object is brought into memory when the command is issued and can help with any form of access
- Separate memory pools can be used, effectively **shielding the memory** from competing applications
- The file attributes do not change
- A **single thread** brings the object into memory
- No guarantee that objects will remain in memory

➤ CL Command level memory control (added in 7.1)

- Target existing physical & logical files
CHGPF/CHGLF ... KEEPINMEM(*YES|*NO)
Database will bring the object into memory when **accessed using SQL**
- **Parallel I/O** will be considered to bring the object into memory
- Stored in the file attribute → survives IPLs, Save/Restore, etc.
- No guarantee that objects will remain in memory, but the odds are good because it happens whenever rows are fetched

➤ SQL KEEP IN MEMORY memory-attribute (new in 7.2)

- Target new or existing SQL tables & indexes
CREATE TABLE ... KEEP IN MEMORY NO or YES
Database will bring the object into memory when **accessed using SQL**
- Includes ALTER TABLE support
- Granular control for managing partition tables
- Once set, behavior matches KEEPINMEM



“In Memory...” blog image



Blog thread: **In Memory...**
<http://db2fori.blogspot.com/2013/10/in-memory.html>

Temporary storage – A mystery revealed

Improved System Management with DB2 for i

- Observe System-wide Temporary storage consumption via a new DB2 for i Service: QSYS2/SYSTMPSTG
- Read all about it in IBM Knowledge Center:

www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzajq/rzajqviewsystemstmpstg.htm

-- Which jobs are the top consumers of temporary storage?

```
SELECT bucket_current_size, bucket_peak_size,
rtrim(job_number) concat '/' concat rtrim(job_user_name) concat '/' concat
rtrim(job_name) as q_job_name
FROM QSYS2.SYSTMPSTG
WHERE job_status = '*ACTIVE'
ORDER BY
bucket_current_size desc
```



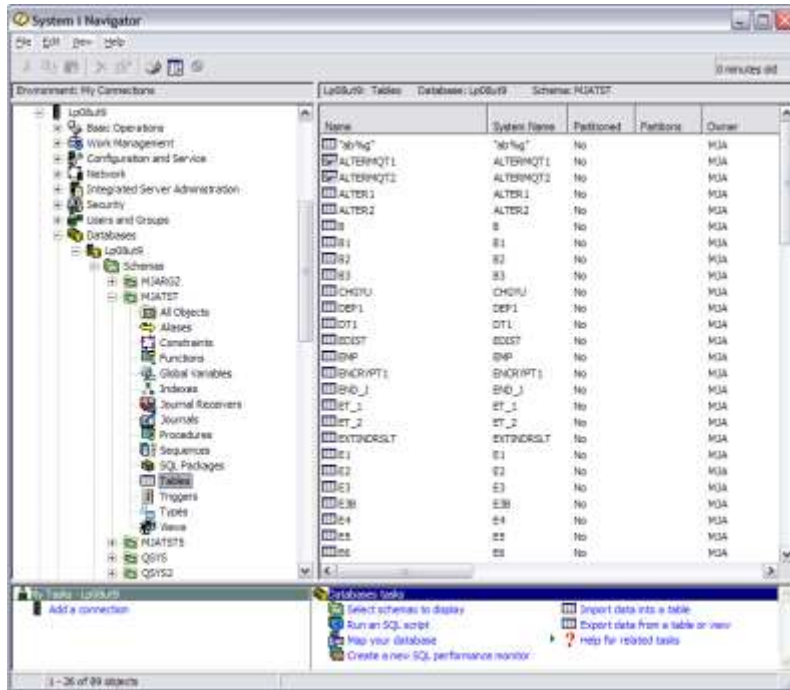
BUCKET_CURRENT_SIZE	BUCKET_PEAK_SIZE	Q_JOB_NAME
1236582400	1239851008	341402/QLWISVR/ADMIN2
706727936	789934080	342172/QDBTS/QJVAEXEC
465354752	482213888	367435/NTL/QPADEV000G
377368576	377368576	342174/QDBTS/QJVAEXEC
376946688	376946688	342175/QDBTS/QJVAEXEC
335908864	335908864	342176/QDBTS/QJVAEXEC
308379648	308379648	342177/QDBTS/QJVAEXEC
241729536	246112256	341120/QWEBADMIN/ADMIN4
226590720	2293952512	367463/QUSER/QZDASOINIT
193028096	194002944	341564/QLWISVR/SMART1113
183308288	183996416	341619/QLWISVR/SMART1114
178647040	178647040	341121/QLWISVR/ADMIN1
174456832	174456832	341266/QLWISVR/ADMIN3
167473152	167473152	341102/QYPSJSVR/QYPSJSVR
162492416	162689024	367429/QLSTAD/QDFTJOB
156450816	156450816	341651/QLWISVR/SMART1115
151515136	151560192	340925/QSYS/QTCPWRK
108703744	122335232	000000/QSYS/SCPF
98451456	114917376	368352/QUSER/QZDASOINIT

DB Navigator – 7.2 Enhancements

Navigator – what database users need to know

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, 7.1 & 7.2
Recent service level?	IBM i Access Windows Service Pack 7.1 – SI50567 → TR7	IBM HTTP SERVER FOR i PTF Group: 7.2 - SF99368 Level x 7.1 - SF99368 Level 24 → TR7 6.1 - SF99115 Level 35
Database commonality	Most features are identical, including TRx enhancements	Most features are identical, including TRx enhancements
Database differences	Run SQL Scripts Visual explain	PDI Time-based performance metrics OmniFind administration
Webpage to watch	http://www-03.ibm.com/systems/power/software/i/access/windows_sp.html	http://www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99368 http://www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99115
Next (planned) Update	May 30, 2014 → IBM i 7.1 TR8 & IBM i 7.2 GA	May 30, 2014 → IBM i 7.1 TR8 & IBM i 7.2 GA

IBM i Navigator 7.2 Enhancements



Health Center

- *System Limits*

Database Management

- *Support of all new SQL features*
 - *Permissions*
 - *Masks*
 - *Named arguments and parameter defaults*
 - *Obfuscation of Triggers*
 - *Arrays in user-defined functions*
 - *Create based ON*
- *Performance Data Investigator (PDI) Graphing and Charting*
- *Display Journal GUI (PTFed back to 7.1)*
- *and more...*

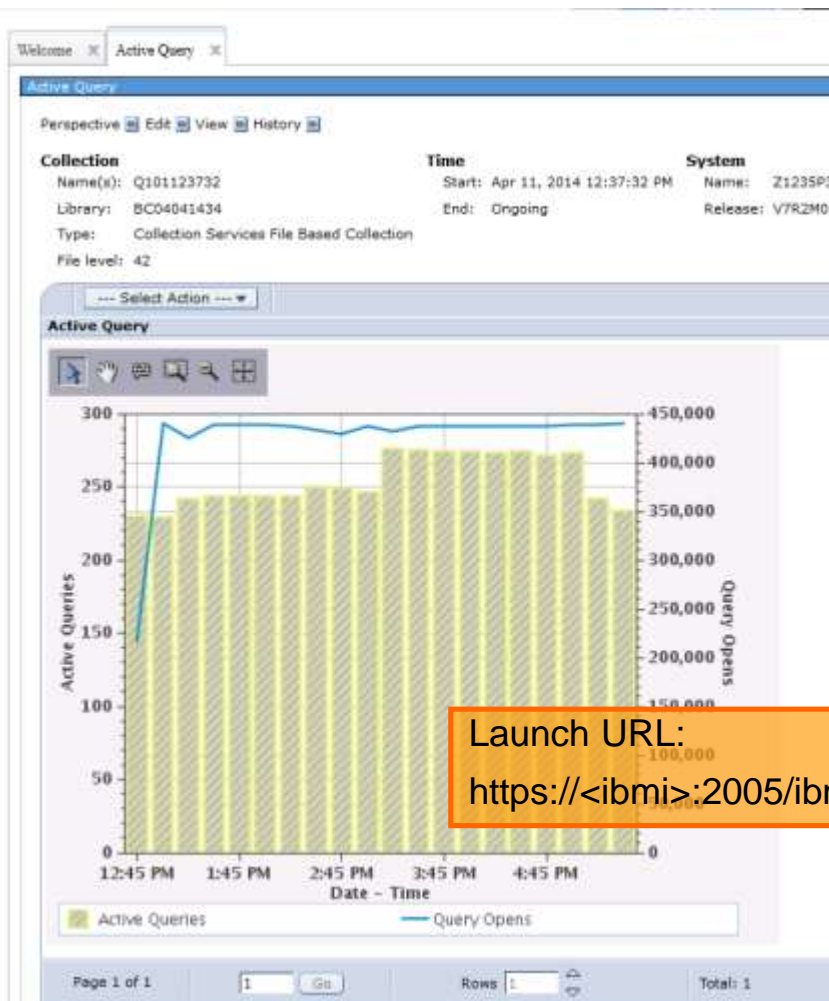
On Demand Performance Center

- *Observance of OPNQRYP & Query/400 queries*
- *Advanced Compare*
- *Enhanced Plan Cache Filtering (For Index Advice)*
- *Filter Database monitor on SQLCODE and CQE*
- *Enhanced Show Statements (PTF back to 6.1)*



Performance Data Investigator (PDI) and Database

- Abundant amount of new perspectives... helping you to quickly answer important questions

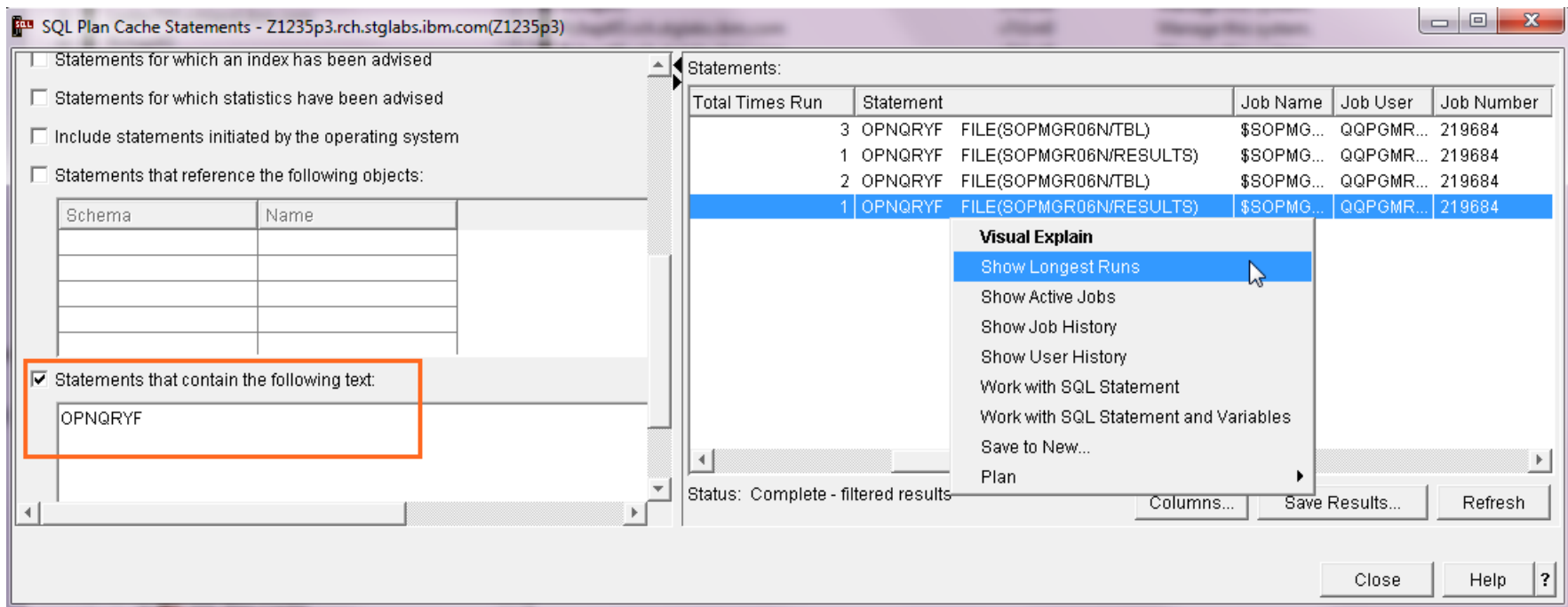



Collection	Time	System
Name(s): Q101123732	Start: Apr 11, 2014 12:37:32 PM	Name: Z123SP3
Library: BCO4041434	End: Ongoing	Release: V7R2M0
Type: Collection Services File Based Collection		
File level: 42		

Launch URL:
<https://<ibmi>:2005/ibm/console/logon.jsp>

SQL Plan Cache & 7.2

- Use 'OPNQRYF' or 'Query/400' as a Statement Text filter
- Explore Visual Explain, Show longest runs, Index Advice, and more...



SQL Plan Cache Statements - Z1235p3.rch.stglabs.ibm.com(Z1235p3)

Statements for which an index has been advised

Statements for which statistics have been advised

Include statements initiated by the operating system

Statements that reference the following objects:

Schema	Name

Statements that contain the following text:

OPNQRYF

Statements:

Total Times Run	Statement	Job Name	Job User	Job Number
3	OPNQRYF FILE(SOPMGR06N/TBL)	\$SOPMG...	QQPGMR...	219684
1	OPNQRYF FILE(SOPMGR06N/RESULTS)	\$SOPMG...	QQPGMR...	219684
2	OPNQRYF FILE(SOPMGR06N/TBL)	\$SOPMG...	QQPGMR...	219684
1	OPNQRYF FILE(SOPMGR06N/RESULTS)	\$SOPMG...	QQPGMR...	219684

Visual Explain

- Show Longest Runs
- Show Active Jobs
- Show Job History
- Show User History
- Work with SQL Statement
- Work with SQL Statement and Variables
- Save to New...
- Plan

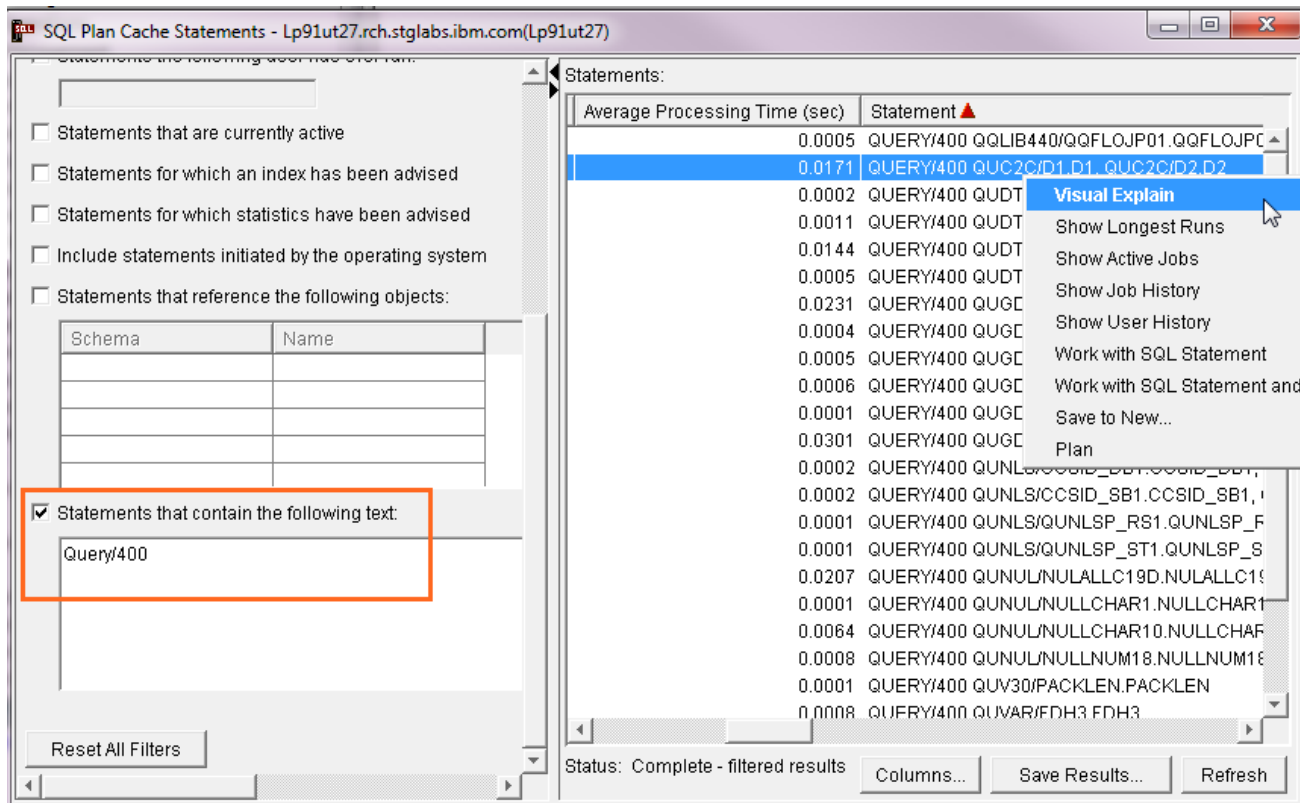
Status: Complete - filtered results

Columns... Save Results... Refresh

Close Help ?

SQL Plan Cache & 7.2

- Use 'Query/400' to find STRQRY, WRKQRY, RUNQRY activity



Statements that contain the following text:

Statements that are currently active

Statements for which an index has been advised

Statements for which statistics have been advised

Include statements initiated by the operating system

Statements that reference the following objects:

Schema	Name

Statements that contain the following text:

Query/400

Reset All Filters

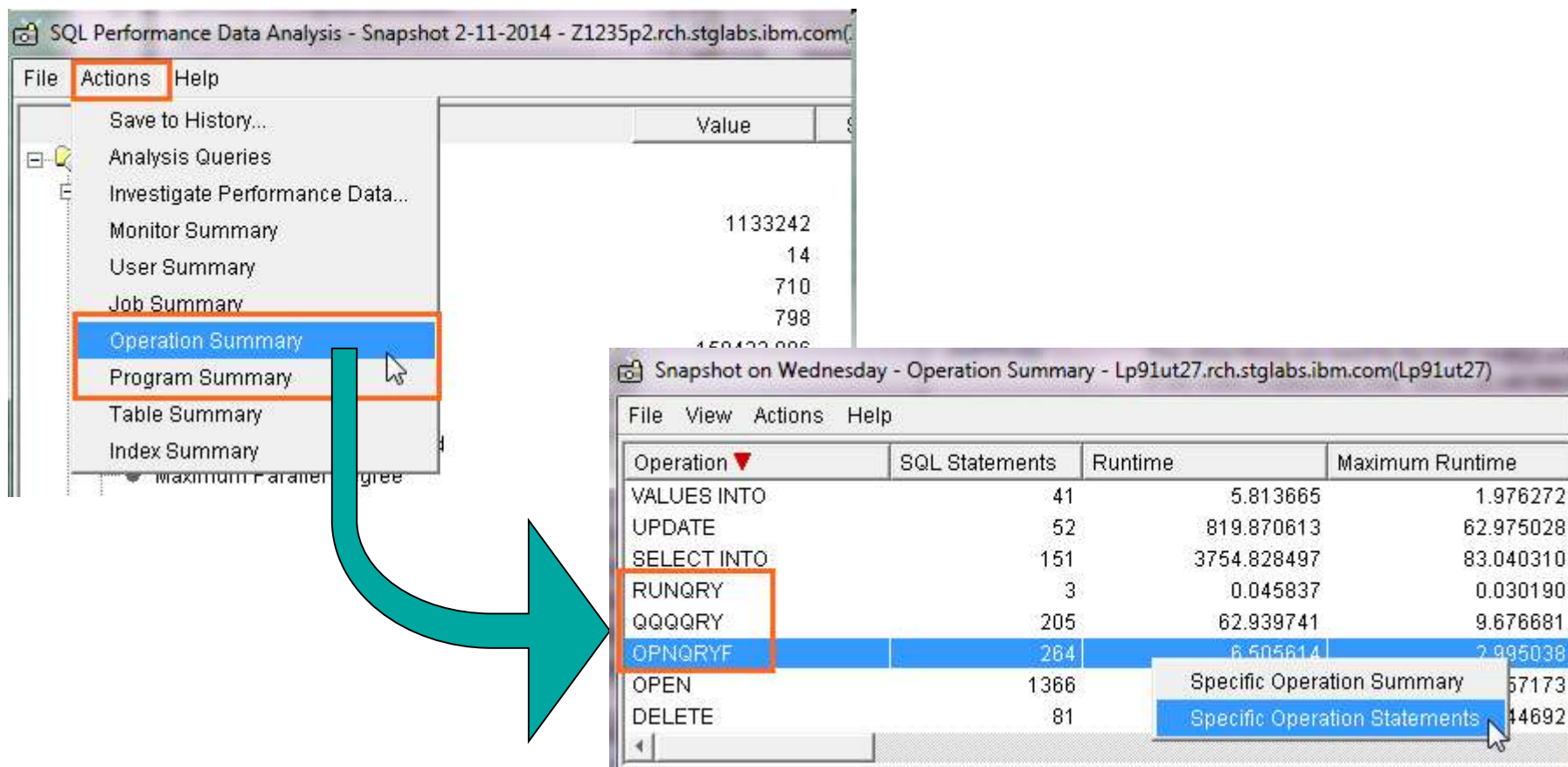
Average Processing Time (sec)	Statement
0.0005	QUERY/400 QGLIB440/QGFLOJJP01.QGFLOJPC
0.0171	QUERY/400 QUC2C/D1.D1.QUC2C/D2.D2
0.0002	QUERY/400 QUDT
0.0011	QUERY/400 QUDT
0.0144	QUERY/400 QUDT
0.0005	QUERY/400 QUDT
0.0231	QUERY/400 QUGC
0.0004	QUERY/400 QUGC
0.0005	QUERY/400 QUGC
0.0006	QUERY/400 QUGC
0.0001	QUERY/400 QUGC
0.0301	QUERY/400 QUGC
0.0002	QUERY/400 QUNLS/CCSID_SB1.CCSID_SB1
0.0002	QUERY/400 QUNLS/CCSID_SB1.CCSID_SB1
0.0001	QUERY/400 QUNLS/QUNLSP_RS1.QUNLSP_F
0.0001	QUERY/400 QUNLS/QUNLSP_ST1.QUNLSP_S
0.0207	QUERY/400 QUNUL/NULALLC19D.NULALLC19
0.0001	QUERY/400 QUNUL/NULLCHAR1.NULLCHAR1
0.0064	QUERY/400 QUNUL/NULLCHAR10.NULLCHAR
0.0008	QUERY/400 QUNUL/NULLNUM18.NULLNUM18
0.0001	QUERY/400 QUV30/PACKLEN.PACKLEN
0.0008	QUERY/400 QUVAR/FDH3.FDH3

Status: Complete - filtered results

Columns... Save Results... Refresh

SQL Plan Cache & 7.2

- Probe an SQL Plan Cache Snapshot via the 'Operation Summary'



The screenshot shows the 'SQL Performance Data Analysis' tool. The 'Actions' menu is open, and 'Operation Summary' is highlighted. A large teal arrow points from this menu item to a secondary window titled 'Snapshot on Wednesday - Operation Summary - Lp91ut27.rch.stglabs.ibm.com(Lp91ut27)'. This window displays a table with the following data:

Operation	SQL Statements	Runtime	Maximum Runtime
VALUES INTO	41	5.813665	1.976272
UPDATE	52	819.870613	62.975028
SELECT INTO	151	3754.828497	83.040310
RUNQRY	3	0.045837	0.030190
QQQQRY	205	62.939741	9.676681
OPNQRYF	284	6.505814	2.995038
OPEN	1366		67173
DELETE	81		44692

In the secondary window, the 'OPNQRYF' row is highlighted in blue, and a context menu is open over it, with 'Specific Operation Statements' selected.

SQE enhancements in IBM i 7.2

What Has Changed for SQE?

SQE can now be used to access files via native opens and native queries.

What is SQE Native Access?

- SQE Native Access refers to these flows being optimized and implemented by SQE.
- Native open refers to database file access using languages like RPG, COBOL or C, as well as CL commands such as Open Data Base File (OPNDBF).
- Native query refers to database file queries using OPNQRYF CL command or IBM Query for i (WRKQRY/RUNQRY/STRQRY).

When does SQE Native Access run?

- When any physical file or SQL table referenced by a native open or query has active Row or Column Access Control (RCAC).
- When any other functionality that requires SQE for a native open or query of an SQL view.
- When enabled via **SQE_NATIVE_ACCESS** QAQQINI option.
 - *NO – 7.1 native open or query behavior, unless one of the above conditions is true.
 - *YES – SQE will attempt to implement native open of an SQL view or native query.
 - ***DEFAULT – Same as *YES.**

Advantages of SQE support for Native Access

1. Potential **improved performance**, just by moving to 7.2
 - Native queries can experience better performance SQE by avoiding multiple cursors and getting better plans, etc.
 - Certain native open applications may actually experience performance improvements due to more aggressive SQE I/O and reduced path length.
 - Several changes made for SQE Native Access also benefit SQL (i.e. random I/O for change sensitive accesses, avoid wasted cycles duplicating cursor for opens of partitioned tables and views, etc.).
2. **Improved usability and analysis capabilities** – Greater workloads visibility in the Plan Cache
 - Advanced tooling includes Show Statements, Visual Explain, Index Advisor, and more...
3. **SQE-only capabilities** are available through more interfaces
 - RCAC requires SQE. To have a comprehensive security offering, SQE was extended to cover the majority of queries.
 - Certain features allowed by RCAC and SQL views are currently not supported by CQE such as EXCEPT, INTERSECT, sequence objects, global variables, scalar fullselects, ROLLUP, CUBE, OLAP, VERIFY_GROUP_FOR_USER, CONTAINS, and SCORE scalar functions, etc. SQE Native Access allow these files with RCAC or SQL views to be accessed via native open or native query.

Behavior Changes related to SQE Native Access

When moving from CQE to SQE, there are some behavioral differences. The two most noteworthy behavior differences are documented in the MTU. For full details on other minor behavior differences, refer to this resource and search on SQE:

<http://www.developerworks.com/ibmi/mtu>

- SQE implementation may result in a different result set ordering
 - When a query is performed without explicitly specifying an order for the results, SQE is less likely to return the results in a keyed order since it has far more advanced optimization capability than CQE.
 - Remediation:
If result ordering is important, **explicitly specify the key field(s) for ordering** (e.g. OPNQRYP KEYFLD parameter)
- Values for fields derived in an expression (i.e. SUBSTR) that are null or in error may be different when rows are read through a native interface (i.e. RPG READ)
 - When a row is read, CQE will evaluate the expression using a default value for all fields that are null or in error. SQE follows the SQL standard and does not evaluate the expression using a default value for all fields that are null or in error.
 - Either way, CQE and SQE will both return NULL indicators for fields that are null or in error.
 - Remediation:
Applications should **examine the NULL indicator** before using a field value

If a change in behavior is unacceptable, the SQE_NATIVE_ACCESS QAQQINI option can be changed to *NO so that CQE will be attempted first and SQE will only be used if CQE is unable to execute the native open or query.

Backup

SQL catalogs → Landscape view

Landscape

Catalogs

SYSCATALOGS

INFORMATION_SCHEMA_CATALOG_NAME

Schemas

SYSSCHEMAS

SQLSCHEMAS

SCHEMATA

Database Support

SQL_FEATURES
SQL_LANGUAGES
SQL_SIZING
CHARACTER_SETS

Tables Views Indexes

SYSCOLUMNS
SYSCOLUMNS2
SYSFIELDS
SYSINDEXES
SYSKEYS
SYSTABLEDEP
SYSTABLES
SYSVIEWDEP
SYSVIEWS

SQLCOLUMNS
SQLSPECIALCOLUMNS
SQLTABLES

COLUMNS
TABLES
VIEWS

Constraints

SYSCHKCST
SYSCST
SYSCSTCOL
SYSCSTDEP
SYSKEYCST
SYSREFCST

SQLFOREIGNKEYS
SQLPRIMARYKEYS

CHECK_CONSTRAINTS
REFERENTIAL_CONSTRAINTS
TABLE_CONSTRAINTS

Privileges

SYSCOLAUTH
SYSCONTROLS
SYSCONTROLSDEP
SYSPACKAGEAUTH
SYSROUTINEAUTH
SYSSCHEMAAUTH
SYSSEQUENCEAUTH
SYSTABAUTH
SYSUDTAUTH
SYSVARIABLEAUTH
SYSXSROBJECTAUTH

SQLCOLPRIVILEGES
SQLTABLEPRIVILEGES

AUTHORIZATIONS
ROUTINE_PRIVILEGES
UDT_PRIVILEGES
USAGE_PRIVILEGES
VARIABLE_PRIVILEGES

Triggers

SYSTRIGCOL
SYSTRIGDEP
SYSTRIGGERS
SYSTRIGUPD

Routines

SYSFUNCS
SYSJARCONTENTS
SYSJAROBJECTS
SYSPARMS
SYSPROCS
SYSROUTINEDEP
SYSROUTINES

SQLFUNCTIONCOLS
SQLFUNCTIONS
SQLPROCEDURECOLS
SQLPROCEDURES

PARAMETERS
ROUTINES

XML Schemas

XSRANNOTATIONINFO
XSROBJECTCOMPONENTS
XSROBJECTHIERARCHIES
XSROBJECTS

Statistics

SYSCOLUMNSTAT
SYSINDEXSTAT
SYSMQTSTAT
SYSPACKAGESTAT
SYSPACKAGESTMTSTAT
SYSPARTITIONDISK
SYSPARTITIONINDEXES
SYSPARTITIONINDEXDISK
SYSPARTITIONINDEXSTAT
SYSPARTITIONMQTS
SYSPARTITIONSTAT
SYSPROGRAMSTAT
SYSPROGRAMSTMTSTAT
SYSTABLEINDEXSTAT
SYSTABLESTAT

SQLSTATISTICS

Miscellaneous Objects

SYSPACKAGE
SYSSEQUENCES
SYSTYPES
SYSVARIABLEDEP
SYSVARIABLES

SQLTYPEINFO
SQLUDTS

USER_DEFINED_TYPES
SEQUENCES

 DB2 for i catalog views (QSYS2)
 ODBC and JDBC™ catalog views (SYSIBM)
 ANS and ISO catalog views (QSYS2)

Health Center Procedures

QSYS2.HEALTH_ACTIVITY
 QSYS2.HEALTH_DATABASE_OVERVIEW
 QSYS2.HEALTH_DESIGN_LIMITS
 QSYS2.HEALTH_ENVIRONMENTAL_LIMITS
 QSYS2.HEALTH_SIZE_LIMITS
 QSYS2.RESET_ENVIRONMENTAL_LIMITS

Utility Procedures

QSYS2.CANCEL_SQL
 QSYS2.DUMP_SQL_CURSORS
 QSYS2.EXTRACT_STATEMENTS
 QSYS2.FIND_AND_CANCEL_QSQSRVR_SQL
 QSYS2.FIND_QSQSRVR_JOBS
 QSYS2.GENERATE_SQL
 QSYS2.RESTART_IDENTITY
 SYSTOOLS.CHECK_CST
 SYSTOOLS.CHECK_SYSRoutine

Plan Cache Procedures

QSYS2.CHANGE_PLAN_CACHE_SIZE
 QSYS2.DUMP_PLAN_CACHE
 QSYS2.DUMP_PLAN_CACHE_PROPERTIES
 QSYS2.DUMP_PLAN_CACHE_topN
 QSYS2/DUMP_SNAP_SHOT_PROPERTIES
 QSYS2.END_ALL_PLAN_CACHE_EVENT_MONITORS
 QSYS2.END_PLAN_CACHE_EVENT_MONITOR
 QSYS2.START_PLAN_CACHE_EVENT_MONITOR (2)

Performance Services

SYSTOOLS.ACT_ON_INDEX_ADVICE - PROCEDURE
 SYSTOOLS.HARVEST_INDEX_ADVICE - PROCEDURE
 QSYS2.OVERRIDE_QAQQINI - PROCEDURE
 QSYS2.RESET_TABLE_INDEX_STATISTICS - PROCEDURE
 QSYS2.SYSIXADV - TABLE
 SYSTOOLS.REMOVE_INDEXES - PROCEDURE

Application Services

QSYS2.QCMDEXC - PROCEDURE
 QSYS2.OVERRIDE_TABLE - PROCEDURE
 QSYS2.DELIMIT_NAME - UDF
 SYSPROC.WLM_SET_CLIENT_INFO - PROCEDURE

Security Services

QSYS2.FUNCTION_INFO - VIEW
 QSYS2.FUNCTION_USAGE - VIEW
 QSYS2.GROUP_PROFILE_ENTRIES - VIEW
 SYSPROC.SET_COLUMN_ATTRIBUTE - PROCEDURE
 QSYS2.SQL_CHECK_AUTHORITY - UDF
 QSYS2.USER_INFO - VIEW

PTF Services

QSYS2.PTF_INFO - VIEW
 QSYS2.GROUP_PTF_INFO - VIEW

TCP/IP Services

QSYS2.TCPIP_INFO - VIEW
 SYSIBMADM.ENV_SYS_INFO - VIEW

Work Management Services

QSYS2.SYSTEM_VALUE_INFO - VIEW
 QSYS2.GET_JOB_INFO - UDTF

Object Services

QSYS2.OBJECT_STATISTICS - UDTF

Storage Services



QSYS2.SYSDISKSTAT - VIEW
 QSYS2.SYSTMPSTG - VIEW
 QSYS2.USER_STORAGE - VIEW

Journal Services

QSYS2.DISPLAY_JOURNAL - UDTF

System Health Services

QSYS2.SYSLIMTBL - TABLE
 QSYS2.SYSLIMITS - VIEW

 DB2 for i Services
 IBM i Services

<http://www.ibm.com/developerworks/ibmi/db2>

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Special notices (cont.)

IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, DB2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, AIX 5L, Chiphopper, Chipkill, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Purpose File System, , GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWER Hypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER6, POWER6+, System i, System p, System p5, System Storage, System z, Tivoli Enterprise, TME 10, Workload Partitions Manager and X-Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. UNIX is a registered trademark of The Open Group in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.

Intel, Itanium, Pentium are registered trademarks and Xeon is a trademark of Intel Corporation or its subsidiaries in the United States, other countries or both.

AMD Opteron is a trademark of Advanced Micro Devices, Inc.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).

SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECcapc, SPECchpc, SPECjvm, SPECmail, SPECimap and SPECsfs are trademarks of the Standard Performance Evaluation Corp (SPEC).

NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.

AltiVec is a trademark of Freescale Semiconductor, Inc.

Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc.

InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Other company, product and service names may be trademarks or service marks of others.