



# Dynamic SQL Re-Examined

William Favero  
Senior Certified IT Specialist  
DB2 for z/OS Software Sales Specialist  
IBM Sales and Distribution  
West Region, Americas

# Disclaimer

The information contained in this presentation has not been submitted to any formal IBM review and is distributed on an "As Is" basis without any warranty either expressed or implied. The use of this information is a customer responsibility.

The materials in this presentation are also subject to

- enhancements at some future date,
- a new release of DB2, or
- a Programming Temporary Fix (PTF)

*IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.*

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, iSeries, pSeries, xSeries, SYSTEM/390, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS and zSeries.

*THE FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC and WINDOWS 95.*

***For additional information visit the URL***

***<http://www.ibm.com/legal/copytrade.phtml> for “Copyright and trademark information”***

# The Basics

# Why Dynamic SQL?

- An application's menu allows multiple values for multiple screen entries and non are required.
  - What will the SQL look like?
- Is your column greater, less than, or equal to a value?
  - Non uniform distribution of data
  - Do you know the value?
  - What are your choices?
- Application needs to manipulate a number of different columns different ways with each execution
  - Do you code an SQL statement for every possible combination?

**Simplicity**

**Performance**

**Flexibility**

# Dynamic SQL Processing

Move string value into a variable DYNSQL:

```
SELECT NAME FROM TEST.CUSTOMER WHERE CUST_NBR = '12345' to DYNSQL
```

- Requires two steps: prepare and then execute
  - Prepare can be expensive and includes
    - Parsing SQL statement
    - Validating SQL syntax
    - Search catalog to
      - Validate tables
      - Validate columns
      - Insure proper set of privileges are in place for this SQL statement
    - Create optimized access path
    - Create executable statement
  - Execute prepared statement

# Dynamic SQL Flavor Examples

- **Embedded Dynamic SQL**
  - Constructed in an application program
  - COBAL, PL/I, FORTRAN, C, C++, Assembler, etc...
  - Program is compiled and bound
- **DB2 ODBC**
  - Open Database Connectivity
  - API available to C and C++
  - No precompilation or binding required
- **JDBC**
  - Java Database Connectivity
  - API to Java applications
  - Use database drivers
  - No precompilation or binding required
- **REXX**
  - REXX language interface using DSNREXX
  - No precompilation or binding required

# What About Security?

## DYNAMICRULES ( RUN -or- BIND )

- RUN (default)
  - Authid of the process running the dynamic SQL statement is checked for the proper set of privileges
  - Authid of the process is used to qualify unqualified objects
- BIND
  - Binder's authid's privilege set is used at run time
  - Binder's authid could be used to qualify unqualified object if QUALIFY is not specified
- There are other options that will not be discussed here
  - DEFINEBIND
  - DEFINERUN
  - INVOKEBIND
  - INVOKERUN

# What about caching?



# No Caching

- This is what dynamic SQL has always had
  - Prepare SQL statement ABC
  - Commit
  - SQL statement ABC must be prepared again before it can be re-executed
- Or
  - Program A prepares SQL statement ABC
  - Program A terminates
  - Program is rerun and SQL statement ABC must be prepared again

# No Caching

- Continued...
  - Program A prepares SQL statement ABC
  - Program A terminates
  - Program B starts and must prepare the exact same SQL statement ABC

# Local Cache

- Persistent PREPARE
- Bind Option - KEEP DYNAMIC(YES)
  - Prepared SQL statement persist across Commit
  - Will use dynamic statement caching if active
  - Storage is still taken from below the bar in V8
- Will require application changes to avoid unnecessary PREPAREs

# MAXKEEPD

- MAX KEPT DYN STMTS
- System wide limit
- Specify the total number of prepared, dynamic SQL statements that can be saved past a commit point by applications that run with the `KEEPDYNAMIC(YES)` bind option
- Can have a significant impact on the DBM1 address space's storage usage

# Persistent PREPARE

- Dynamic Statement Caching will affect Persistent PREPARE
- If Dynamic Caching on:
  - Prepared statement is cached
  - Cached prepared statement is used
- If Dynamic Caching off:
  - Only statement string is saved
  - DB2 prepares statement string after commit

# Prepared Statement Caching

- System wide option determined at installation time
  - DSNTIP4 - CACHE DYNAMIC SQL field
    - Acceptable values: YES, NO
    - Default: NO
    - DSNZPARM: DSN6SPRM CACHEDYN
- All DML prepared using
  - SQL PREPARE
  - SQL EXECUTE IMMEDIATE
- Prepare statement reuse if:
  - Exact match on SQL statement string
  - Authid must match (more later)
  - Some BIND options must be consistent
    - CURRENTDATA, DYNAMICRULES, ISOLATION, SQLRULES, and QUALIFIER

Dynamic Statement Cache

# Prepared Statement Caching

- Not available across data sharing members
- Be careful -
  - Caching statement may add cost to preparing statement
  - Statement skeletons take up space in the EDM pool
  - If caching is enabled (CACHEDYN YES), there is a cache search cost
- However...
  - Can have significant CPU savings for an ERP application

# CMTSTAT

- DB2 Version 7
- If DSNZPARM CMTSTAT = INACTIVE
  - DB2 writes an accounting record when a transaction commits
  - Connection qualifies to become inactive
- If using KEEP DYNAMIC(YES) bind option
  - DB2 cannot disconnect the connection from the thread (which is what happens when the connection becomes inactive), because the thread contains information about locally cached statements
  - DDF threads always (has to) remain active
- Result:
  - Accounting records are not cut at transaction boundaries. Likewise, DDF does not re-establish WLM enclaves at transaction boundaries.



# CMTSTAT

- DB2 Version 8
- DSNZPARM CMTSTAT = INACTIVE
- KEEP DYNAMIC(YES) still prevents DDF threads from becoming inactive
- However:
  - KEEP DYNAMIC(YES) allows DB2 to cut accounting records at transaction boundaries
  - New behavior allows period based WLM goals for threads that commit frequently, but cannot become inactive only because they use KEEP DYNAMIC(YES).
  - Threads that cannot become inactive for other reasons, do not reset the WLM enclave, and period based goals are probably not right for them (as was the case in the past).
- This new behavior is supported for DRDA clients. DB2 for z/OS clients that use the DB2 private protocol are not affected by this change.
- The presence of held cursors or declared temporary tables keeps the threads active and does not allow accounting intervals or WLM enclaves to complete as it did in Version 7

# EDM Pool

- If Dynamic Statement Caching is turned on
  - statements can be cached in a data space
- DB2 determines the default size of the data space as follows:
  - If the EDM pool storage size is less than or equal to 40 MB, the EDM data space is 40 MB.
  - If the EDM pool storage size is larger than 40 MB, the EDM data space size equals the EDM pool size.
  - See next slide for overrides

# EDM Pool and Data Spaces

- EDMPOOL DATA SPACE SIZE
  - Acceptable values: 1-2097152
  - Default: based on calculations
  - DSNZPARM: DSN6SPRM EDMDSPAC
  - Starting point
  
- EDMPOOL DATA SPACE MAX
  - Acceptable values: 0-2097152
  - Default: 1048576
  - DSNZPARM: DSN6SPRM EDMDSMAX
  - Had no meaning if dynamic caching is not turned on

# EDM Pool

**In DB2 Version 8, a data space is no longer used. Storage for the statement cache is now above the 2Gb bar.**

## **EDM STATEMENT CACHE for V8**

- Acceptable values: 5000 to 1048576
- DSNZP`xxx`: DSN6SPRM EDMSTMTC (new)

Version 8 always allocates at least 5 MB of storage for the dynamic statement cache regardless of the setting of CACHEDYN. Because CACHEDYN is changeable online changeable in V8, DB2 needs an initial size to be allocated at startup time to allow the size to be changed online.

# Parameter Markers

```
SELECT *  
FROM  
WHERE col = "abc"
```

**No reuse because statements are not exactly the same**

```
SELECT *  
FROM  
WHERE col = "def"
```

```
SELECT *  
FROM  
WHERE col = ?
```

**Cache reuse**

```
SELECT *  
FROM  
WHERE col = ?
```

# Invalidating Statement Cache

## Version 8 invalidation of dynamically cached statements after CREATE INDEX

If indexes are known:

- Execute RUNSTATS utility with the options
  - UPDATE NONE REPORT NO in DB2 V8.
    - This combination only invalidate statements in the dynamic statement cache without any data access or computation cost

However, for Enterprise Application solutions, indexes are often created when maintenance is applied so their existence is not obvious to a DBA.

- If a new index is created, the affected cached statements are invalidated for both data sharing and non-data sharing, but not quiesced.
  - Not quiesced means that statements that are used in active transactions can still be employed until these transactions complete.

# Invalidating Statement Cache

- Additional Version 8 considerations
  - Cached statements are invalidated when a new partition is added to a table
  - REORG REBALANCE will invalidate statement cache

# Other Stuff



# Reoptimization For Input Variables

```
SELECT * FROM TABLE_A  
WHERE COL1 = :host_variable
```

**Dynamic  
and  
Static SQL**

**WHEN\_OPTIMIZE column  
added to PLAN\_TABLE**

**Optimizer uses default  
filter factors**

**Bind - REOPT(VARS)  
Runtime reoptimization**

# REOPT Keyword

- The BIND option REOPT(ONCE) allows re-optimization an SQL statement based on the host variable value the first time it is executed. This reduces the cost of multiple optimization when REOPT(VARS) was used in the past.
- REOPT(ONCE) reuses the statement stored in the dynamic statement cache.
- REOPT(ONCE) can be EXPLAINed because the statement is in the cache, something that could not be done with REOPT(VARS)

# IFCIDs

Tracing the full SQL statement, you have two options:

1. If the statement is in the dynamic statement cache, you can use IFCID 317 to retrieve the full SQL statement text through the READS interface of the IFI.
2. If the statement is not in the dynamic statement cache, you can use the new IFCID 350. Unlike IFCID 63, it contains the full SQL statement text.

# EXPLAIN STMTCACHE

- EXPLAIN STMTCACHE - Clause added in Version 8
- STMTID id-host-variable or integer-constant
  - Specifies that the cached statement associated with the statement ID contained in host variable id-host-variable or specified by integer-constant is to be explained.
  - The statement ID is an integer that uniquely determines a statement that has been cached in dynamic statement cache. The statement ID can be retrieved through IFI monitor facilities from IFCID 316 or 124 and is shown in some diagnostic IFCID trace records such as 172, 196, and 337.
    - For every row that the EXPLAIN statement inserts into the plan table, statement table, or function table, the QUERYNO column contains the value of the statement ID.

# EXPLAIN STMTCACHE

- EXPLAIN STMTCACHE - Clause added in Version 8
- STMTTOKEN token-host-variable or string-constant
  - The STMTTOKEN parameter takes either a host variable or a string constant. This statement token string is associated with the cached statement by the application program that originally prepares and inserts the statement into the cache. The application does this by using the RRSAF SET\_ID function or by using the sqlseti API from a remotely-connected program
  - For every row inserted into the plan table, statement table, or function table by the EXPLAIN statement, the column STMTTOKEN (which is newly added column) is given the value of the statement token, and the column QUERYNO is given the value of the statement ID for the cached statement with the statement token.

# Resource Limit Facility (RLF)

- RLF allows a system administrator to:
  - Set warning and error threshold for certain dynamic SQL statements
  - Stop a dynamic SQL statement that exceeds a predefined processor limit
  - Restrict BIND and REBIND activities
  - Restrict types of parallelism for dynamic queries
- Must create database, table space, tables and indexes
- Populate table
  - RLFFUNC – describes what row controls
  - RLFBIND – turns bind on and off
  - Many other columns
- Must be started before it can be used
  - -START RLIMIT ID=xx

# *DSNZPARM*

# IDTHTOIN

- Consider using `IDTHTOIN=0`
  - `IDTHTOIN` default is 120 (seconds) in DB2 V8. For ERP applications, consider:
    - Set to 0 (zero) to ensure that idle threads are not automatically canceled by DB2.
      - This parameter is only relevant if `CMTSTAT` is set to `INACTIVE`
    - `KEEPDYNAMIC(YES)` will cause DBATs to remain active and therefore making them susceptible to idle thread timeouts

**From the Book:** `IDTHTOIN` specifies the approximate time, in seconds, that an active server thread should be allowed to remain idle before it is canceled. The thread is canceled after the timeout value expires; its locks and cursors are released. Inactive and indoubt threads are not subject to time-out. The value that you specify for `DDF THREADS` determines whether a thread can become inactive, and thus not subject to time-out.



# Short Prepare

## DSC Short Prepare Enhancements

- **V7 Problem**

- When a matching statement is found in cache, it is copied into thread related storage before it can be used
- If no storage a GETMAIN is performed
- To keep size of local pool limited initiate storage contraction regularly

## GETMAIN/FREEMAIN make up large part of cost of short prepare

- **V8 Solution**

- Use a number of shared storage pools (assigned via hashing to minimize contention) to copy statements in
- Potentially fewer storage required (not each thread needs its own pool)
- Using "best fit" algorithm to find space in assigned pool
- Result: Fewer GETMAIN/FREEMAIN operations, faster short prepare

# Some Extra Reading

- SG24-2003-00 - High Availability Considerations DB2 for OS/390 with SAP R/3
- SG24-2078-01 - Database Administration Experiences: SAP R/3 on DB2 for OS/390
- GG24-2233-00 - DB2 for OS/390 Application Design for High Performance
- SG24-2244-00 - DB2 for OS/390 Capacity Planning
- SG24-5156-01 - Planning to Install PeopleSoft with DB2 of OS/390
- SG24-5343-00 - SAP on DB2 for OS/390: Disaster Recovery
- SG24-5486-00 - DB2 for OS/390 and Continuous Availability
- SG24-5681-00 - SAP Business Information Warehouse on OS/390
- SG24-5690-00 - SAP on DB2 for OS/390: Database Availability Considerations
- SG24-5840-00 - SAP on DB2 for OS/390: Application Servers on OS/390
- SG24-6093-00 - PeopleSoft V8 on zSeries Using Sysplex Data Sharing and Enterprise Storage Systems
- SG24-6129-00 - DB2 for z/OS and OS/390 Version 7 Selected Performance Topics
- SG24-6121-00 - DB2 UDB Server for OS/390 and z/OS Version 7 Presentation Guide
- SG24-6287-00 - SAP on DB2 for OS/390 and z/OS: System Cloning
- SG24-6300-00 - DB2 for z/OS Application Programming Topics
- SG24-6418-00 - Squeezing the Most Out of Dynamic SQL with DB2 for z/OS and OS/390
- SG24-6549-00 - Implementing PeopleSoft 8.4 on zSeries
- SG24-6836-00 - SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution using System Automation
- SG24-6847-00 - SAP on DB2 UDB for OS/390 and z/OS: Implementing Application Servers on Linux for zSeries
- SG24-6868-00 - Siebel 7 with DB2 for z/OS: Database Implementation and Administration Guide
- SG24-6871-00 - DB2 UDB for z/OS V8 - Technical Preview
- SG24-6914-00 - SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOD)
- SG24-6950-00 - SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitor with Data Sharing
- SG24-6952-00 - Distributed Functions of DB2 for z/OS and OS/390
- SG24-6079-00 - DB2 for z/OS Version 8: Everything You Ever Wanted to Know, ... and More
- SG24-7078-00 - Installing Siebel 7.5.3 for z/OS and OS/390
- SG24-7083-00 - DB2 for z/OS Stored Procedures: Through the CALL and Beyond
- SG24-7088-00 - DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There
  
- REDP0162 - DB2 for z/OS and OS/390 Version 7 Selected Performance topics
- REDP0131 - SAP R/3 on DB2 for OS/390: DB2 Features That Benefit SAP
- REDP3717 - mySAP Business Suite Managed by IBM Tivoli System Automation for Linux



# Web Resources

- DB2 for OS/390 - [www.ibm.com/software/db2zos](http://www.ibm.com/software/db2zos)
- DB2 Estimator - [www.ibm.com/software/data/db2/os390/estimate/](http://www.ibm.com/software/data/db2/os390/estimate/)
  - » Version 8 of Estimator is now available for download
- ESS - [www.ibm.com/storage/hardsoft/diskdrls/technology.htm](http://www.ibm.com/storage/hardsoft/diskdrls/technology.htm)
- DB2 support/services - [www.ibm.com/software/data/db2/zos/support.html](http://www.ibm.com/software/data/db2/zos/support.html)
- ODBC - [www.ibm.com/software/data/db2/os390/odbc/](http://www.ibm.com/software/data/db2/os390/odbc/)
- ODBC static profiling - [www.ibm.com/software/data/db2/udb/staticcli/](http://www.ibm.com/software/data/db2/udb/staticcli/)
- J2EE/JDBC specifications - [java.sun.com/products/jdbc/](http://java.sun.com/products/jdbc/)
- DB2 Java Site - [www.ibm.com/software/data/db2/java/](http://www.ibm.com/software/data/db2/java/)
- DB2 for z/OS JDBC - [www.ibm.com/software/data/db2/os390/jdbc.html](http://www.ibm.com/software/data/db2/os390/jdbc.html)
- DB2 SQLJ - [www-3.ibm.com/software/data/db2/java/sqlj/](http://www-3.ibm.com/software/data/db2/java/sqlj/)
- DB2 for OS/390 SQLJ - [www-3.ibm.com/software/data/db2/os390/sqlj.html](http://www-3.ibm.com/software/data/db2/os390/sqlj.html)

# Shameless self-promotion

- My new db2 for z/OS external blog

**[blogs.ittoolbox.com/database/db2zos](http://blogs.ittoolbox.com/database/db2zos)**

**T h a n k s !**

**William Favero**

Certified IT Software Specialist  
IBM Certified DBA - DB2 V8 for z/OS  
IBM Certified Solutions Expert: DB2 V7 DBA for OS/390  
DB2 Sales Specialist  
Western Region  
IBM Sales & Distribution