

## Db2 SQL Debug

### Test Single SQL Statements Fast

#### Business Challenges

As companies strive to increase revenue and improve customer service it becomes increasingly important to create or modify Db2 applications as quickly as possible.

Although developers may create or modify only a small portion of an application, testing the modification still requires the time consuming process of compiling and executing the entire application.

#### Test Application SQL While Coding

During the development and coding of a Db2 application, programmers want to know if the SQL they coded is valid and whether the SQL is returning the results they expect.

SoftBase's Db2 SQL Debug component of TestBase lets a developer code and dynamically test SQL statements while editing or viewing an SQL program. The Db2 SQL Debug allows for the execution and / or Explain of the SQL without having to compile and execute the application. All Db2 developers know the importance of good access paths. Db2 SQL Debug provides faster, easier, and more reliable SQL statement testing and performance analysis.

#### Db2 SQL Debug Makes Applications More Reliable

SoftBase's Db2 SQL Debug allows individual SQL statements to be tested while coding the application program – before the program executes. Db2 SQL Debug can help developers quickly assess the quality of existing, new, or altered SQL statements during development. This helps developers catch potentially problematic SQL statements early in development when they are easier to optimize.

#### Point to the SQL while in Edit or View

Use the QQ line commands to point to the SQL you want to execute. The command can be used in block form as shown or Qn.

```
VIEW      TST.G610.TDGF.LIB.SOURCE(S3930000) - 01.11      Columns 00001 00072
Command ==>                                          Scroll ==> CSR
000524      MOVE S3-TDAT          TO TDAT-TDAT-NAME
000525      MOVE S3-ENVIRONMENT-CODE TO TDAT-ENVIRONMENT-CD
000526      MOVE S3-SYSTEM-CODE    TO TDAT-SYSTEM-CD
000527      EXEC SQL
qq0528      SELECT TIMESTAMP (TDAT_DATE , TDAT_TIME)
000529      INTO :W-CURRENT-TIMESTAMP
000530      FROM TEST DATA
000531      WHERE ENVIRONMENT_CD = :TDAT-ENVIRONMENT-CD
000532      AND SYSTEM_CD      = :TDAT-SYSTEM-CD
qq0533      AND TDAT_NAME     = :TDAT-TDAT-NAME
000534      END-EXEC
```

- Improve Developer Productivity with Faster Debugging
- See Statement Access Paths without Compiling and Executing Application
- Reduce Development and Testing Time

#### Db2 SQL Debug Program SQL Options

This panel prompts for the subsystem, how the SQL should be gathered from the program and whether the SQL should be executed or explained or both.

```
DB2 SQL Debug Program Options
Command ==>
Enter SQL Option Values:
DB2 Subsystem Name . ==> DBCG      DB2 Subsystem Name
DB2 SQLID . . . . . ==> S3DBA     Plan Table Qualifier
DB2 Schema Name . . ==> TDGF610   Qualifier for Unqualified Tables
Commit or Rollback . ==> C       C - Commit R - Rollback
Maximum Result Rows. ==> 01000    Maximum Number of Result Rows to Fetch
Explain Query Number ==> 00001    Query Number for Explain in Plan Table
Left Margin . . . . . ==> 00001   First Source Column to Process
Right Margin . . . . . ==> 00072   Last Source Column to Process

Execute . . . . . ==> Y ( Y / N ) Execute SQL
Explain . . . . . ==> Y ( Y / N ) Explain SQL
```

#### Db2 SQL Debug Host Variable Prompt

Here you can supply the host variable values before executing the SQL. Data type is shown so there is no guessing or reading of application code to determine. No quotes are needed for character data since it is a fully prepared SQL statement.

```
DB2 SQL Debug Execute Host Values
Command ==>
DB2 Subsystem Name . ==> DBCG      Host Variable 0001 of 0003
DB2 Plan Name . . . . ==> PDGF610  Scroll ==>
DB2 Schema Name . . . ==> TDGF610  DB2 SQLID . . . . . ==> S3DBA
Press END key to continue. Press CANCEL to cancel.
Enter SQL Variable Values:
Host Variable /      Data Length
Value              Type   Scale Nulls
-----
000001 TDAT-ENVIRONMENT-CD SMALLINT 0000002 Y
+00205
000002 TDAT-SYSTEM-CD     SMALLINT 0000002 Y
+00003
000003 TDAT-TDAT-NAME     CHAR     0000008 Y
CSBK
```

#### Db2 SQL Debug Results

Results of the executed SQL are shown in Testbase View

```
TestBase Table Processor
Command ==>
DB2 Location ==> DBCGLOC
DB2 Object. . ==> SEE INPUT SQL FOR TABLE(S)
Columns ==> 1 Thru 26 of 26 IX ==> Lobs ==> N
-----1-----2-----+
000001 2017-05-24-14.03.16.000000 01 Rows
Scroll ==> PAGE
DB2 Subsystem ==> DBCG
Current SQLID ==> S3DBA
```

# Db2 SQL Debug

## Db2 Access Path in Db2 SQL Debug

This shows the access path chosen for the SQL statement. It is the

```

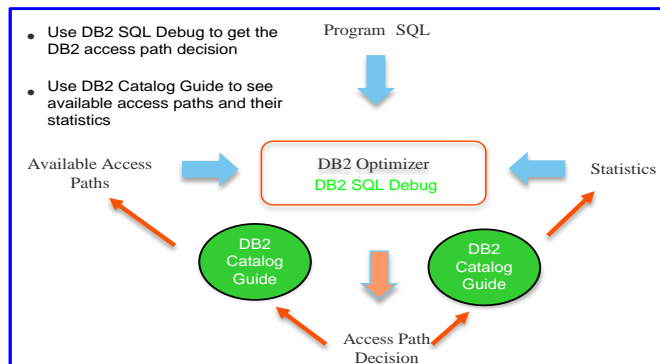
DB2 SQL Debug Explain Access Paths          Lines 1 of 3
Command ==>                               Scroll ==>
DB2 Subsystem ==> DBCG

S          T
SE        UL P      A          LM          OO SORT SORT   Line Commands:
BE L     M S B          J P F CD NEW COMP   S - Select
C A     IOE N          M F E KE UJOG UJOG   X - Expand
T N     XPQ O          / - List Commands
-----
01 01 000 01 T I N 03 0          IS NNNN NNNN
TABLE: TDAT
INDEX: TDAT01CF
    
```

data from the plan table. In this case, we have an index access path matching on 3 columns. If also licensed for TestBase Db2 Catalog Guide or TestBase View / Edit, those commands can be issued next to the table or the index. Using Db2 SQL Debug in conjunction with Db2 Catalog Guide allows for viewing all 3 determinants of access path selection:

1. SQL Statement
2. Access Paths Available
3. Statistics on those Access Paths

## Use Db2 Catalog Guide with Db2 SQL Debug



## Db2 Catalog Guide Index Information

```

TestBase DB2 Index          Lines 1 of 73
Command ==>                Scroll ==> PAGE
DB2 Location : DBCGLOC     DB2 Subsystem ==> DBCG
DB2 object  : TDGF610.TDAT01CF

Commands: Find  Export  Explode

NAME . . . . . TDAT01CF
CREATOR . . . . . TDGF610
TBNAME . . . . . TDAT
TBCREATOR . . . . . TDGF610
UNIQUERULE . . . . . P
COLCOUNT . . . . . +00003
CLUSTERING . . . . . Y
CLUSTERED . . . . . Y
DBID . . . . . +00274
OBID . . . . . +00004
ISOBID . . . . . +00005
DBNAME . . . . . TDGF6140
INDEXSPACE . . . . . TDAT01CF
FIRSTKEYCARD . . . . . -0000000001
FULLKEYCARD . . . . . -0000000001
NLEAF . . . . . -0000000001
NLEVELS . . . . . -00001
BPOOL . . . . . BPO
PGSIZE . . . . . +00004
ERASERULE . . . . . N
DSETPASS . . . . .
CLOSERULE . . . . . N
SPACE . . . . . +0000000000
IBMREQD . . . . . N
CLUSTERATIO . . . . . +00000
CREATEDBY . . . . . CSB3
IOFACTOR . . . . . +00000
PREFETCHFACTOR . . . . . -00001
STATTIME . . . . . 0001-01-01-00.00.00.000000
INDEXTYPE . . . . . 2
FIRSTKEYCARDF . . . . . -1.000000000000000E+00
FULLKEYCARDF . . . . . -1.000000000000000E+00
CREATEDTS . . . . . 2016-04-22-16.42.37.215619
    
```

This shows all the data in SYSIBM.SYSINDEXES

## Db2 Catalog Guide - Indexes on a Table

An IX (Index) command next to the table displays the following:

```

TestBase DB2 Catalog Objects          Lines 1 of 2
Command ==>                Scroll ==> PAGE
DB2 Subsystem ==> DBCG     Location: DBCGLOC

Line Commands:      Commands:
S - Select          Export - Print Results
X - Expand
/ - List Commands  Ref Object: <TB> TDGF610.TDAT

Cmd Indexes - Creator Name
-----
<IX> TDGF610.TDAT01CF
<IX> TDGF610.TDAT02NN
    
```

This shows there are 2 indexes available on this table.

## Available Commands

The available commands can be shown by using a "/". For example:

```

Command ==>
Line Comman
S - Select
X - Expans
/ - List
Cmd Indexes
<IX> TD
<IX> TD

Command Actions
Object Type: IX
Object Name: TDGF610.TDAT01CF

Command:
SG Storage Group SM Statement PL Plan
DB Database L List CL Collection
TS TableSpace X Expand EK Package
P Partition S Select EX Explain SQL
TB Table V View CK Check Const
IX Index E Edit
VW View DI Display
CO Column ST Start
AL Alias SP Stop
TG Trigger VL Volume
RL Relationship SQ View SQL

Select a Hi-lited choice and press ENTER to process command.
    
```

## Additional Db2 SQL Debug Features and Benefits

- Allows an easy prompt for host variable values without errors with respect to data type
- Executes without copying to SPUFI and making error prone changes
- Explains without binding or copying to SPUFI and substituting parameter markers for host variables
- Fully integrated with other TestBase components like Db2 Catalog Guide and View/Edit

## Accelerate Your Application

In spite of massive investments of money, time, and resources into critical Db2 for z/OS projects, most still run late and over budget. Overall application quality is poor in spite of strenuous coding and testing efforts. Db2 SQL Debug is part of SoftBase's TestBase solution that enables customers to code, test, and deploy Db2 for z/OS applications in a fraction of the time it takes now.

## About SoftBase

SoftBase is committed to creating a better Db2 development experience. By combining decades of Db2 experience with a set of proven testing and performance tuning tools and an unmatched customer support team, SoftBase delivers – helping application developers and Db2 Administrators create reliable, high quality Db2 applications faster and with ease. With SoftBase you can count on tools that work as promised and a knowledgeable support team available 24/7.

