Introduction

SQL return codes provided by DB2 UDB for OS/390 and z/OS can be confusing and often reference manuals are not available or close at hand when you really need them. This Instant Expert Reference Card will review SQL return code processing and common SQL error conditions you may encounter in your daily work with DB2.

Retrieving SQL Return Code Information & Messages Into Your Programs

- COBOL programs executing SQL statements communicate with DB2 via a Working Storage area called the SQL Communications Area (SQLCA).
- When DB2 executes SQL statements, it returns the results of the operation into the SQLCODE and SQLSTATE fields in the SQLCA. SQLCODE provides key information about the success or failure of SQL statement execution.
- If the SQLWARN field in the SQLCA contains 'W', DB2 has set at least one of the SQL warning flags (SQLWARN1 through SQLWARNA). These flags provide additional info about execution of specific types of SQL.
- Prior to DB2 V8, COBOL programs could call a subroutine called DSNTIAR that would convert a SQLCODE in the SQLCA into a more detailed text message with diagnostics about the return code.
- DB2 V8 New with DB2 V8, COBOL programs can now execute a GET DIAGNOSTICS statement that will return all previous SQLCA values and provide additional information about new DB2 V8 extended object names and new SQL functions.
- DB2 V8 The new GET DIAGNOSTICS function replaces existing SQLCA processing now found in most DB2 COBOL programs. GET DIAGNOSTICS also passes a text message about SQLCODE directly to programs.

SQLCODE Overview

- If SQLCODE = 0, execution was successful.
- If SQLCODE > 0, execution was successful with a warning.
- If SQLCODE < 0, execution was not successful.

More DB2 SQL Error Code Resources

- IBM DB2 Version 8 for z/OS Messages & Codes:
  http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/dsnmcj10/CCONTENTS
- **100** ROW NOT FOUND FOR FETCH, UPDATE OR DELETE, OR THE RESULT OF A QUERY IS AN EMPTY TABLE. Suggestion: If expecting data, verify WHERE clause for accuracy and completeness.

- **117** THE NUMBER OF INSERT VALUES IS NOT THE SAME AS THE NUMBER OF OBJECT COLUMNS. Suggestion: Correct SQL statement to provide only one value for each column in the table.

- **231** (DB2 V8) CURRENT POSITION OF CURSOR cursor-name IS NOT VALID FOR FETCH OF THE CURRENT ROW. Suggestion: Be certain to FETCH to position on a row after opening a cursor. If cursor is declared SENSITIVE STATIC SCROLL, the row may be a hole, from which no values can be fetched.

- **304** A VALUE WITH DATA TYPE data-type1 CANNOT BE ASSIGNED TO A HOST VARIABLE BECAUSE THE VALUE IS NOT WITHIN THE RANGE OF THE HOST VARIABLE IN POSITION position-number WITH DATA TYPE data-type2. Suggestion: Verify DCLGEN host variable definitions are current with DB2 catalog table/view attributes.

- **347** (DB2 V8) THE RECURSIVE COMMON TABLE EXPRESSION name MAY CONTAIN AN INFINITE LOOP. Suggestion: Verify predicate in the SQL WHERE clause of the form “counter_col < constant” or “counter_col < :hostvar.

- **802** EXCEPTION ERROR exception-type HAS OCCURRED DURING operation-type OPERATION ON data-type DATA, POSITION position-number. Suggestion: Check arithmetic operation for divide by zero or result to exceed size of host variable.

- **180** THE DATE, TIME, OR TIMESTAMP VALUE value IS INVALID. Suggestion: Verify the data value is in the correct range and value type.

- **181** THE STRING REPRESENTATION OF A DATETIME VALUE IS NOT A VALID DATETIME VALUE. Suggestion: Verify data format with the SQL Reference Guide.

- **204** name IS AN UNDEFINED NAME. Suggestion: Correct DB2 creator or object names located in SQL statements.

- **227** (DB2 V8) FETCH fetch-orientation IS NOT ALLOWED, BECAUSE CURSOR cursor-name HAS AN UNKNOWN POSITION (sqlcode,sqlstate). Suggestion: CLOSE and re-OPEN the cursor; For scrollable use (FIRST, LAST, BEFORE, AFTER, or ABSOLUTE) to establish valid position.

- **305** THE NULL VALUE CANNOT BE ASSIGNED TO OUTPUT HOST VARIABLE number-position-number BECAUSE NO INDICATOR VARIABLE IS SPECIFIED. Suggestion: Add null indicator variable to SELECT statement in the format of “column:hostvarind”

- **501** THE CURSOR IDENTIFIED IN A FETCH OR CLOSE STATEMENT IS NOT OPEN. Suggestion: Correct logic in application program to OPEN the cursor before the FETCH or CLOSE statement.

- **502** THE CURSOR IDENTIFIED IN AN OPEN STATEMENT IS ALREADY OPEN. Suggestion: Correct logic in application program to CLOSE the CURSOR before the OPEN statement.

- **503** A COLUMN CANNOT BE UPDATED BECAUSE IT IS NOT IDENTIFIED IN THE UPDATE CLAUSE OF THE SELECT STATEMENT OF THE CURSOR. Suggestion: Use FOR UPDATE statement in your cursor.

- **530** THE INSERT OR UPDATE VALUE OF FOREIGN KEY constraint-name IS INVALID. Suggestion: Ensure that INSERT row for DB2 PARENT table is completed before INSERT row in CHILD table.

- **532** THE RELATIONSHIP constraint-name RESTRICTS THE DELETION OF ROW WITH RID X’rid-number’. Suggestion: Change the program to DELETE CHILD table row before DELETE of row on PARENT table.

- **551** auth-id DOES NOT HAVE THE PRIVILEGE TO PERFORM operation operation ON OBJECT object-name. Suggestion: Contact the support DBA to GRANT the needed privilege.

- **803** AN INSERTED OR UPDATED VALUE IS INVALID BECAUSE THE INDEX IN INDEX SPACE indexspace-name CONSTRAINTS columns of the table so no two rows can contain duplicate values in those columns. RID of existing row is Xrid. Suggestion: Verify DB2 INDEX and, if needed, change the statement to an UPDATE.