SQL programming

The DB2® for IBM® i database provides a wide range of support for Structured Query Language (SQL).

The examples of SQL statements shown in this topic collection are based on the sample tables and assume that the following statements are true:

- They are shown in the interactive SQL environment or they are written in ILE C or in COBOL. EXEC SQL and END-EXEC are used to delimit an SQL statement in a COBOL program.
- Each SQL example is shown on several lines, with each clause of the statement on a separate line.
- SQL keywords are highlighted.
- Table names provided in the sample tables use the schema CORPDATA. Table names that are not found in the Sample Tables should use schemas you create.
- Calculated columns are enclosed in parentheses, (), and brackets, [].
- The SQL naming convention is used.
- The APOST and APOSTSQL precompiler options are assumed although they are not the default options in COBOL. Character string literals within SQL and host language statements are delimited by single-quotation marks (').
- A sort sequence of *HEX is used, unless otherwise noted.

Whenever the examples vary from these assumptions, it is stated.

Because this topic collection is for the application programmer, most of the examples are shown as if they were written in an application program. However, many examples can be slightly changed and run interactively by using interactive SQL. The syntax of an SQL statement, when using interactive SQL, differs slightly from the format of the same statement when it is embedded in a program.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 491.

Related concepts:

Embedded SQL programming

Related reference:

"DB2 for i sample tables" on page 463
These sample tables are referred to and used in the SQL programming and the SQL reference topic collections.

DB2 for i5/OS SQL reference

What's new for IBM i 7.1

Read about new or significantly changed information for the SQL programming topic collection.

Result sets returned to programs

You can return a result set from a stored procedure to a program and have the program consume the result set. For more information, see "Writing a program or SQL procedure to receive the result sets from a stored procedure" on page 174.

XML

The XML data type has been added to SQL. This includes internal handling of XML values and XML schema decomposition. For more information, see “SQL statements and SQL/XML functions” on page 257.

Field procedures

A field procedure is a user-written exit routine to transform values in a single column. When values in the column are changed, or new values inserted, the field procedure is invoked for each value, and can transform that value (encode it) in any way. The encoded value is then stored. When values are retrieved from the column, the field procedure is invoked for each value, which is encoded, and must decode it back to the original value. For more information, see “Defining field procedures” on page 28.

MERGE statement

The MERGE statement can be used to either insert new rows or update existing rows in a table or view, depending on whether they already exists in the table or view. For more information, see “Merging data” on page 119.

Global variables

Global variables can be created and used in SQL. For more information, see “Creating and using global variables” on page 53.

Arrays in SQL procedures

An array data type has been added for use in SQL procedures. For more information, see “Array support in SQL procedures” on page 226 and “Debugging an SQL routine” on page 227.

WebSphere® MQ with DB2

DB2 provides an application programming interface to the WebSphere MQ message handling system through a set of external user-defined functions, which are called DB2 MQ functions. You can use these functions in SQL statements to combine DB2 database access with WebSphere MQ message handling. For more information, see “WebSphere MQ with DB2” on page 448.

Other functional changes to the SQL programming information

- An option to replace an existing object has been added to many DDL CREATE statements. See “Replacing existing objects” on page 54.

What's new as of April 2015

The OR REPLACE option has been added to the CREATE TABLE statement. For more information, see “Using CREATE OR REPLACE TABLE” on page 47.

What's new as of October 2014

Pipelined table functions

A pipelined SQL table function is a more flexible version of a table function. For more information, see “Example: SQL table UDFs” on page 193.

SQL variable debug for SQL routines

Debugging of SQL routines allows you to display values of SQL variables. For more information, see "Debugging an SQL routine" on page 227.

What's new as of February 2013

Multiple event triggers

A trigger can be defined for more than one event. For more information, see "Multiple event SQL triggers" on page 221.

System name for tables, views, and indexes

When creating a table, view, or index, the system name for the object can be specified on the create statement. For more information, see "Creating a table" on page 17 and "Creating and using views" on page 48.

What's new as of October 2012

Defaults for procedure parameters and using parameter names in CALL

You can define parameters for SQL and external procedures to have default values. Parameters with default values can be omitted when calling the procedure. The CALL statement can specify parameter names for any arguments. For more information, see "Defining a procedure with default parameters" on page 156.

CREATE TABLE referencing a remote table

You can create a local table with the definition and data pulled from a non-local table. For more information, see "Creating a table with remote server data" on page 23.

SQL and external routine management

Procedures and functions are tied to system objects that can be administered with CL commands. For more information, see "Managing SQL and external routine objects" on page 230.

RUNSQLSTM OPTION parameter

The RUNSQLSTM command does not need to generate a listing. For more information, see "Using the SQL statement processor" on page 421.

What's new as of April 2012

Obfuscation

The content of an SQL procedure or SQL function can be obfuscated. For more information, see "Obfuscating an SQL routine" on page 228.

XMLTABLE

The XMLTABLE table function provides a way to work with XML content as a relational table. For more information, see "Using XMLTABLE to reference XML content as a relational table" on page 284. In addition, SQL XML Programming contains all the new and existing XML information that is related to SQL.

Insert from a remote table

You can insert into a local table with data pulled from a non-local table. For more information, see "Inserting data from a remote database" on page 110.

RUNSQL

This new CL command runs a single SQL statement. For more information, see "Using the RUNSQL CL command" on page 425.

What's new as of April 2011

Connect by

Hierarchical queries can be defined using the CONNECT BY syntax. For more information, see "Using recursive queries" on page 83.

How to see what's new or changed

To help you see where technical changes have been made, the information center uses:

- The  image to mark where new or changed information begins.
- The image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

To find other information about what's new or changed this release, see the Memo to users.

PDF file for SQL programming

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select SQL programming (about 1,714 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (http://get.adobe.com/reader/)  .

Introduction to DB2 for i Structured Query Language

Structured Query Language (SQL) is a standardized language for defining and manipulating data in a relational database. These topics describe the System i® implementation of the SQL using the DB2 for i database and the IBM DB2 Query Manager and SQL Development Kit for i licensed program.

SQL manages information based on the relational model of data. SQL statements can be embedded in high-level languages, dynamically prepared and run, or run interactively. For information about embedded SQL, see Embedded SQL programming.

SQL consists of statements and clauses that describe what you want to do with the data in a database and under what conditions you want to do it.

SQL can access data in a remote relational database, using the IBM Distributed Relational Database Architecture™ (DRDA).

Related concepts:

Distributed database programming

Related reference:

"Distributed relational database function and SQL" on page 427
A *distributed relational database* consists of a set of SQL objects that are spread across interconnected computer systems.

SQL concepts

DB2 for i SQL consists of several main parts, such as SQL runtime support, precompilers, and interactive SQL.

- SQL runtime support

  SQL run time parses SQL statements and runs any SQL statements. This support is part of the IBM i licensed program, which allows applications that contain SQL statements to be run on systems where the IBM DB2 Query Manager and SQL Development Kit for i licensed program is not installed.
- SQL precompilers

  SQL precompilers support precompiling embedded SQL statements in host languages. The following languages are supported:
  - ILE C
  - ILE C++
  - ILE COBOL
  - COBOL
  - PL/I
  - RPG III (part of RPG)
  - ILE RPG

  The SQL host language precompilers prepare an application program that contains SQL statements. The host language compilers then compile the precompiled host source programs. For more information about precompiling, see Preparing and running a program with SQL statements in the Embedded SQL programming information. The precompiler support is part of the IBM DB2 Query Manager and SQL Development Kit for i licensed program.
- SQL interactive interface

  The SQL interactive interface allows you to create and run SQL statements. For more information about interactive SQL, see "Using interactive SQL" on page 410. Interactive SQL is part of the IBM DB2 Query Manager and SQL Development Kit for i licensed program.
- Run SQL Scripts

  The Run SQL Scripts window in System i Navigator allows you to create, edit, run, and troubleshoot scripts of SQL statements.
- Run SQL Statements (RUNSQLSTM) CL command

  The RUNSQLSTM command can be used to run a series of SQL statements that are stored in a source file or a source stream file. For more information about the RUNSQLSTM command, see "Using the SQL statement processor" on page 421.
- DB2 Query Manager