Embedded SQL

Robert C. Nickerson
ISYS 464 – Spring 2003
Topic 16

Introduction

- SQL statements can be included in programs written in other languages
- The SQL statements are called embedded SQL
- Embedded SQL statements are executed in the sequence that they appear in the program
- Language of program in which SQL statements are embedded is called host language
- Host language can be C/C++, Java, COBOL, etc.

How to Embed an SQL Statement in a Host Language Program

- Mark beginning of SQL statement with EXEC SQL
- End SQL statement with semicolon

EXEC SQL SQL statement;

- SQL statement can be SELECT, INSERT, DELETE, UPDATE, CREATE, etc.
- Effect:
  - Program executes normally until EXEC SQL is encountered
  - At that point, the SQL statement is sent to DBMS for execution
  - After SQL statement is executed, execution of program continues normally

Compilation of Programs with Embedded SQL

Software needed:

- Precompiler: Converts SQL statement following each EXEC SQL into function calls equivalent to SQL statement in host language syntax
- Compiler: Translates program in host language into machine language to create object program
- Linker: Replaces function calls in object program with functions to create executable program

Compilation of Programs with Embedded SQL

1. Precompile source program to produce precompiled program
   - Convert embedded SQL statements into host language SQL function calls using SQL-to-host-language precompiler
2. Compile precompiled program to produce object program
   - Translate precompiled program into machine language using host-language compiler
3. Link object program to produce executable program
   - Replace function calls in object program with SQL functions from SQL function library using linker
4. Execute executable program

Copyright (c) 2003 by Robert C. Nickerson.
All rights reserved. Not for general distribution.
**INCLUDE Statement**

EXEC SQL INCLUDE SQLCA;
SQLCA contains variables used to communicate between DBMS and program
Most useful variables in SQLCA:
  sqlca.sqlcode – error code returned after each SQL statement is executed
  = 0 – statement executed successfully
  < 0 – statement did not execute
  > 0 – statement executed successfully but some unusual condition occurred (e.g., no data retrieved)
sqlca.sqlerrm.sqlerrmc – error message returned after each SQL statement is executed

**WHENEVER SQLERROR Statement**

EXEC SQL WHENEVER SQLERROR statement;
statement (in host language) executed if sqlca.sqlcode < 0
statement is usually a function call to display error code and error message, then terminate program
Note: These two statements are normally placed at the beginning of the program

**Host Variables**

- Host variable:
  - Variable in syntax of host language that is used within an embedded SQL statement to identify data from database
  - Must begin with colon (:) in SQL statement but colon is not used with host variable in other parts of program
- Example: If database uses ENUM for employee number then program might use :empnum as host variable to refer to employee number.

**Single Row SELECT Statement**

Used to select data from one row of a table
Example: Find name of employee with a given number.
Assume variable searchnum is set in the program to the number of desired employee.
EXEC SQL
SELECT Employee_Number, Employee_Name
INTO :empnum, :empname
FROM Employee
WHERE Employee_Number = :searchnum;
sqlca.sqlcode = 0 - if row found matching condition
= 100 - if no row found

**DECLARE CURSOR Statement**

EXEC SQL
DECLARE cursorname CURSOR FOR selectstatement;
Cursor: Pointer to a row in a table. Must be declared before it can be used.
Example: (Usually placed at beginning of program)
EXEC SQL
DECLARE Employee_Cursor CURSOR FOR
SELECT Employee_Number, Employee_Name
FROM Employee;
Almost all features of SELECT statement can be used in DECLARE CURSOR statement
DECLARE CURSOR does not execute SELECT statement
FETCH Statement

EXEC SQL
  FETCH cursorname
  INTO hostvariablelist;

Makes the data from the row resulting from the SELECT statement in a DECLARE CURSOR statement available for processing in a program. There must be one host variable for each column retrieved by the SELECT statement.

Example:
EXEC SQL
  FETCH Employee_Cursor
  INTO :empnum, :empname;

Cursor automatically updated before each FETCH to point to next row. Program can incorporate loop to process all rows resulting from SELECT.

sqlqa.sqlcode = 0 - if FETCH retrieves row
= 100 - if cursor does not point at a row

Sample program

SCROLL Cursor

FETCH can now
  FETCH FIRST: first row
  FETCH LAST: last row
  FETCH NEXT: next row; default
  FETCH PRIOR: prior row
  FETCH ABSOLUTE n: row n
  FETCH RELATIVE +/-n: current row +/-n rows

Example: Fetch 5th row
EXEC SQL
  FETCH ABSOLUTE 5 Employee_Cursor
  INTO :empnum, :empname;

Update Statements

All update statements can be used in embedded SQL.

Examples:
EXEC SQL
  INSERT INTO Employee
    VALUES (:empnum, :empname);
EXEC SQL
  DELETE FROM Employee
    WHERE Employee_Number = :empnum;
EXEC SQL
  UPDATE Employee
    SET Employee_Name = :empname
    WHERE Employee_Number = :empnum;

Transaction Processing

Transaction: A series of database and other activities that must be completed together. If any statement in a transaction fails, the transaction should not be allowed to proceed and all previous update statements in the transaction should be rolled back.

How to know if there is a failure of a statement: If sqlqa.sqlcode < 0 after the execution of any statement. If this is the case, ROLLBACK should be executed and the program terminated.