PRÁCTICAS DE SISTEMAS DE BASES DE DATOS

GRADO EN INGENIERÍA INFORMÁTICA
2º CURSO

SESIÓN 8: SQL inmerso en C. Introducción

Ana Belén Gil González
Ana de Luis Reboredo

Departamento de Informática y Automática
Universidad de Salamanca

Location: Salamanca

8. SQL INMERSO EN C. INTRODUCCIÓN

8.1. Introducción

El SQL inmerso, permite utilizar el lenguaje SQL para realizar accesos a una base de datos desde un programa escrito en un lenguaje de alto nivel (C/C++, Pascal, COBOL, ...). Concretamente en estas sesiones nosotros vamos a utilizar SQL inmerso en C dentro del entorno Oracle. Escribiremos un programa "casi C " con extensión .pc, que incluirá tanto las sentencias habituales de C, como sentencias SQL inmersas. Este programa deberá ser precompilado por Pro*C/C++ (Precompilador de SQL de Oracle).

Sintaxis básica: `EXEC SQL <sentencia SQL>;`

Para diseñar un programa con SQL inmerso, se tendrá en cuenta que:

- Dichos programas (con extensión `.pc`) incluyen una o varias sentencias de SQL inmerso
- Las sentencias de SQL inmerso pueden realizar accesos a una Database, o pueden ser simplemente declarativas
- Todas las sentencias que impliquen acceso a una base de datos deben ejecutarse dentro de una conexión

Sintaxis para Conexión y Desconexión:

```
EXEC SQL CONNECT :username IDENTIFIED BY :password ;

...

EXEC SQL COMMIT WORK RELEASE; /* ó EXEC SQL ROLLBACK
RELEASE */
```

Para las prácticas utilizaremos una sintaxis abreviada de conexión:

```
char oracleid = '/';

EXEC SQL CONNECT :oracleid;
```

8.2. Variables del Lenguaje

Son la clave para la comunicación entre el programa y la base de datos. Se declaran conforme a la sintaxis del lenguaje C al estilo de la declaración regular de una variable. Los **tipos de datos que pueden utilizarse con el Oracle** son: char , char[*n*], int, short, long, float, double , VARCHAR[*n*].
Algunas de las sentencias SQL hacen referencia a variables del lenguaje (variables declaradas como variables C). Para lo cual se deben cumplir las siguientes condiciones:

1. Las variables deben estar definidas en una sección de declaración.

2. En las sentencias SQL , dichas variables aparecerán **precedidas de dos puntos ":"**

Una **sección de declaración** es un conjunto de sentencias de definición de variables y tipos comprendidas entre las sentencias EXEC SQL BEGIN DECLARE SECTION; y EXEC SQL END DECLARE SECTION;

Sintaxis:

```
EXEC SQL BEGIN DECLARE SECTION;
/* declaración de las variables */
...
EXEC SQL END DECLARE SECTION;
```

8.3. Control de Errores

Para controlar los errores después de la ejecución de una sentencia SQL, el programa puede verificar el estado de ejecución del programa simplemente comprobando el **área de comunicaciones del SQL** (SQLCA)

La SQLCA (SQL *Communications Area*) es utilizada para detectar errores y cambios de estado a lo largo de la ejecución de un programa. Esta estructura contiene campos que son actualizados por Oracle tras la ejecución de cualquier sentencia SQL en tiempo de ejecución de dicho programa.

Para utilizar la SQLCA es necesario incluir el fichero de cabecera **sqlca.h** con la directiva #include:

```
#include <sqlca.h>
```

ó bien:

```
EXEC SQL INCLUDE sqlca;
```

8.4. Acceso a la Base de Datos sin Cursores

Sentencia SELECT

Sólo permite recuperar una fila de la base de datos

Sintaxis:

```
EXEC SQL
SELECT expresión {, expresión }
INTO :variable {, :variable }
FROM tabla {, tabla }
[WHERE cláusula ]
```

```
[GROUP BY cláusula ]
[HAVING cláusula ]
[ORDER BY cláusula ]
```

Sentencias UPDATE / INSERT / DELETE

Actualizan, insertan y borran filas en una tabla de la base de datos.

Sintaxis:

```
EXEC SQL
  UPDATE table
  SET columna = expression {, columna = expression }
  [ WHERE condición ] ;

EXEC SQL
  INSERT INTO table [( lista_columnas )]
  VALUES ( lista_valores ) ;

EXEC SQL
  INSERT INTO table [( lista_columnas )] subselect;

EXEC SQL
  DELETE FROM table
  [ WHERE condición ] ;
```

EJERCICIOS PROPUESTOS

Introducción al SQL inmerso en C. Acceso a la Base de Datos sin Cursores

1. Obtener el nombre y apellidos de un lector a partir de su código, que es introducido por el usuario.
2. Obtener el número de autores pertenecientes a una nacionalidad a partir del código de nacionalidad introducido por el usuario.
3. Obtener el número de autores pertenecientes a una nacionalidad a partir de la nacionalidad introducida por el usuario.
4. Obtener la localidad en la que está ubicada una sucursal a partir del código que es introducido por el usuario.
5. Obtener toda la información de un autor a partir del código de éste, introducido por el usuario.
6. Obtener el número de préstamos en un año concreto contabilizados para una sucursal a partir del código de la sucursal y el año elegido.
7. Obtener el número de préstamos no finalizados contabilizados para una sucursal a partir del código de la sucursal.

8. Codificar un programa que utilizando SQL inmerso en C permita modificar la dirección, población y provincia de un lector. El programa, inicialmente, debe pedir al usuario el código del lector. Si dicho lector no existe mostrará un mensaje de error indicando que no existe y por tanto no se puede modificar; en caso contrario pedirá al usuario los nuevos valores para los atributos dirección, población y provincia y actualizará dichos datos para el lector especificado. Si el usuario no proporciona valor para alguno de los atributos, estos deben actualizarse con valor nulo.

9. Codificar una función utilizando SQL inmerso en C que permita añadir una nueva sucursal en la base de datos conociendo su código, dirección, población y provincia.
   El prototipo de la función será el siguiente:
   int **insSucursal**(char *codigo, char *dirección, char *problacion, char *provincia)
   Donde:
   a. El primer parámetro identifica el código, el segundo la dirección de la nueva sucursal y así con los demás parámetros
   b. Si la función acaba correctamente deben comprometerse todas sus modificaciones e indicarse que el alumno fue matriculado devolviendo un cero.
   c. Si se viola alguna restricción de integridad o se realiza alguna operación contra la base de datos sin éxito, se deben anular todos los efectos realizados por la función y devolver el código de error apropiado.

   La función debe insertar valores nulos en aquellos atributos de la tabla para los que no se proporciona valor.

NOTA: Realizar la comprobación de errores en todos y cada uno de los ejercicios enunciados