**ACTIVIDAD TEORÍA UT5**

1. ¿Qué es un **procedimiento almacenado** en una base de datos?

Es un conjunto de comandos SQL que pueden almacenarse en el servidor.

1. ¿Cuál es la diferencia entre un **procedimiento y una función**?

Un procedimiento se invoca usando un comando, y sólo puede pasar valores usando variables de salida. Una función puede llamarse desde dentro de un comando como cualquier otra función (esto es, invocando el nombre de la función), y puede retornar un valor escalar.

1. Sintaxis (**ejecuta en consola** y copia la sintaxis relacionada con las siguientes acciones relativas a procedimientos y funciones:

PROCEDIMIENTOS.

* 1. Crear un procedimiento. CREATE PROCEDURE
  2. Modificar un procedimiento. ALTER PROCEDURE
  3. Borrar un procedimiento. DROP PROCEDURE
  4. Mostrar la creación de un procedimiento. SHOW CREATE PROCEDURE
  5. Mostrar su estatus. SHOW PROCEDURE STATUS FUNCIONES.

FUNCIONES

1. Crear una función. CREATE FUNCTION
2. Modificar una función. ALTER FUNCTION
3. Borrar una función. DROP FUNCTION
4. Mostrar la creación de una función. SHOW CREATE FUNCTION
5. Mostrar su estatus. SHOW FUNCTION STATUS
6. ¿Para qué sirve la sentencia **CALL**?

Para invocar un procedimiento que ha sido invocado previamente.

1. ¿Para qué se utiliza la sentencia compuesta BEGIN…..END?

para escribir sentencias compuestas que pueden aparecer en el

interior de procedimientos almacenados y triggers.

1. ¿Para qué se utiliza la sentencia DECLARE?

Para definir varios iconos locales de una rutina.

1. Escribe la sintaxis para declarar una variable local.

DECLARE var\_name[…….] type [DEFAULT value]

1. Escribe la sintaxis cómo asignar valores a variables (SET).

SET var\_name = expr [, var\_name = expr] …

1. ¿Para qué sirve la variable SELECT ….INTO?

Para almacenar columnas seleccionadas directamente en variables.

1. ¿Qué es un cursor?

Un cursor no es más que una estructura que almacena el conjunto de filas devuelto por una consulta a la base de datos.

1. Define los tipos de cursores:
   1. Implícitos: Oracle abre implícitamente un cursor para procesar cada sentencia SQL que no esté asociada con un cursor declarado explícitamente.
   2. Explícitos: Cuando una consulta devuelve múltiples filas, podemos declarar explícitamente un cursor para procesar las filas devueltas.
2. Consulta en consola, **define** y copia la **sintaxis** de:
   1. Declarar un cursor. . Pueden definirse varios cursores en una rutina, pero cada cursor en un bloque debe tener un nombre único.

DECLARE cursor\_name CURSOR FOR select\_statement

* 1. Abrir un cursor: Este comando abre un cursor declarado préviamente.

OPEN cursor\_name

* 1. Sentencia FETCH de un cursor: Este comando trata el siguiente registro (si existe) usando el cursor abierto que se especifique, y avanza el puntero del cursor.

FETCH cursor\_name INTO var\_name [, var\_name]…

* 1. Cerrar un cursor. Este comando cierra un cursor abierto préviamente.

CLOSE cursor\_name

1. Escribe la sintaxis de las **sentencias de control de flujo**:
   1. Sentencia IF:

IF search\_condition THEN statement\_list [ELSEIF search\_condition THEN statement\_list]….

[ELSE statement\_list]

END IF

* 1. Sentencia CASE.

CASE case\_value

WHEN when\_value THEN statement\_list

[WHEN when\_value THEN statement\_list]

[ELSE statement\_list]

END CASE

* 1. Sentencia LOOP.

[begin\_label:] LOOP

Statement\_list

END LOOP [end\_label]

* 1. Sentencia LEAVE.

LEAVE label

* 1. Sentencia ITERATE.

ITERATE label

* 1. Sentencia REPEAT

[begin\_label:] REPEAT

Statement\_list

UNTIL search\_condition

END REPEAT [end\_label]

* 1. Sentencia WHILE

[begin\_label:] WHILE search\_condition DO

Statement\_list

END WHILE [end\_label]

1. ¿Qué es un **disparador o trigger**?

Es un objeto con nombre en una base de datos que se asocia con una tabla, y se activa

cuando ocurre un evento en particular para esa tabla.

1. Ejecuta en consola y escribe la sintaxis:
   1. Crear un disparador.

CREATE TRIGGER nombre\_disp momento\_disp evento\_disp ON nombre\_tabla FOR EACH ROW sentencia\_disp;

* 1. Borrar un disparador.

DRP TRIGGER [nombre\_esquema.]nombre\_disp;

1. Indicar 3 **limitaciones** en la utilización de los disparadores.

• La palabra clave BEFORE indica el momento de acción del disparador. En este caso, el disparador debería activarse antes de que cada registro se inserte en la tabla. La otra palabra clave posible aqui es AFTER.

• La palabra clave INSERT indica el evento que activará al disparador. En el ejemplo, la sentencia INSERT causará la activación. También pueden crearse disparadores para sentencias DELETE y UPDATE .

• Las sentencia siguiente,FOR EACH ROW , define lo que se ejecutará cada vez que el disparador se active, lo cual ocurre una vez por cada fila afectada por la sentencia activadora. En el ejemplo, la sentencia activada es un sencillo SET que acumula los valores insertados en la columna amount. La sentencia se refiere a la columna como NEW.amount, lo que significa “el valor de la columna amount que será insertado en el nuevo registro.”