

# EJERCICIOS DE PL/SQL

**1.** Realizar un programa que solicite un pedido de un artículo determinado, la cantidad que se pide siempre es uno, el programa debe comprobar si hay stock suficiente y si es así actualizar el stock y, en una tabla temporal insertar el código de artículo y el mensaje "Vendido". Si no hay stock se inserta en la tabla temporal el código del artículo y el mensaje "Sin existencias".

**2.** Modificar el ejercicio anterior para actualizar la tabla de pedidos y detalle. Se solicitará un código de cliente por teclado además del código de artículo que ya se pedía en el ejercicio anterior.

El primer pedido a efectuar será el 8 y el último el 15.

La cantidad será 1.

La fecha será la del sistema.

**3.** Programa que solicita por teclado un código de cliente e inserta en la tabla temporal:

numero total de pedidos de ese cliente (tot\_ped)

precio total de esos pedidos (precio\_tot)

código de cliente (cod\_cli)

nombre de cliente (nombre\_cli)

	temp		
tot_ped	precio_tot	cod_cli	Nombre

**3b.** Crearse un registro t\_reg\_emple con la siguiente estructura:

Código del empleado, nombre y el job. Insertar en una tabla temporal el nombre del empleado y job para el empleado 7782.

**4a** Insertar 50 filas en una tabla.

**4b** Recorre todas las filas de la tabla y las inserta en una tabla temporal. Crearse la tabla students e insertarle algunos datos.

```
CREATE TABLE TEMP (  
    V_INDEX NUMBER,  
    V_MAJOR VARCHAR2 (15));
```

```
V_MAJOR (-7) := 'MATEMATICAS';  
V_MAJOR (4) := 'HISTORIA';  
V_MAJOR (5) := 'ECONOMIA';
```

**4c** Llenar una tabla PL con los nombres de los clientes y cargarla en una tabla temporal de sql que contenga el índice de la tabla PL y el nombre del cliente, y visualizar ésta.

**5a.** Llenar una tabla temporal con los códigos de clientes, nombre del cliente y cod de empleado. De tres formas posibles, usando bucle LOOP, bucle WHILE y bucle FOR

**5b.** A los pedidos del cliente 2, añadirle 10 unidades más a la cantidad e insertar en una tabla temporal el código de artículo y un mensaje que diga "Diez unidades más vendidas".

**6.** Crear un procedimiento que añada nuevos pedidos a un cliente determinado.

El procedimiento recibe el idped, idart, cantidad y codcli.

**7.** Procedimiento que borra los pedidos del cliente especificado.

Recibe el codcli.

**8.** Procedimiento para cambiar el oficio de un empleado. Se pasa al codemp y el nuevo oficio. Insertará en TEMP oficio\_ant, nuevo, codemp.

**9.** Crear una función "Anual" para devolver el salario anual cuando se pasa el salario mensual y la comisión de un empleado.

Asegurarse que controla nulos. Utilizar una variable de acoplamiento para ver lo que devuelve.

**10.** Crear un paquete llamado "Actualiza" que tiene tres procedimientos y una tabla PL.

- Un procedimiento de alta de un pedido.
- Un procedimiento de baja de los pedidos y detalles de un cliente concreto.

- Una tabla PL que almacenará los códigos de los pedidos.

- Un procedimiento "Listar" que tiene como parámetro 'in' el código de cliente y como parámetro

de salida una tabla PL con los códigos de los pedidos del cliente especificado, además

se grabará en una tabla temporal el código del pedido y el código del artículo.

**11.** Crear un trigger para ver como suceden los eventos de activación.

Vamos a utilizar la tabla plantilla y vamos a cambiar a las "Enfermeras" por "Interinos".

**12.** Crear un disparador a nivel de fila tal que después de insertar, modificar o borrar un detalle de la tabla detalles introduzca en la tabla temporal el usuario, la fecha del sistema

el código de pedido que se ha modificado, insertado o borrado, así como un mensaje que diga:

"Detalle dado de alta"

"Detalle borrado"

"Detalle modificado"

Según corresponda.

**12.b** Disparador que inserta una fila en la tabla Temp con el texto 'subida de salario empleado' y el número del empleado al que se le ha subido el salario. El disparador se activará después de actualizar el salario de la tabla empleador;

**12.c** Trigger que se dispara cada vez que se borra un empleado, guardando el número empleado, apellido y departamento en la tabla TEMP

**13.** Limitar a 5 (0 al 4) el número de detalles de cada pedido.

**14.** Solicitar un código de cliente por teclado si existe se inserta su código en una tabla temporal

y el mensaje 'EXISTE', si no existe se dispara la excepción NO\_DATA\_FOUND y se inserta en la tabla temporal el mensaje 'NO EXISTE'.

**15.** Se solicita un código de artículo por teclado y se inserta en una tabla temporal el precio

y la descripción de este artículo si está pedido, es decir, si su código está en la tabla detalle.

Si el artículo no está pedido, se genera una excepción de usuario con el mensaje "El artículo (código) no lo ha pedido ningún cliente"

**16.** Modificar el ejercicio 2 añadiendo excepciones, de tal forma que en el gestor de excepciones controlemos que select ha fallado, insertando en una tabla temporal cliente no existe, o bien artículo no existe, según haya fallado uno u otro.