**HOJA 2 DE EJERCICIOS TEMA 12: BLOQUES, FUNCIONES Y PROCEDIMIENTOS CON CURSORES Y EXCEPCIONES. CONTROL DE TRANSACCIONES**

**Crea el usuario t12b y utiliza los sripts del archivo TablasVentas.txt para crear las tablas**

**1. Desarrollar un procedimiento que visualice el apellido del cliente y la fecha de pedido de todos los pedidos ordenados por apellido.**

CREATE OR REPLACE PROCEDURE Apellido\_fechapedidos

is

cursor cur1 is

Select apellidos,fecha\_pedido from clientes, pedidos

where clientes.id\_cliente=pedidos.id\_cliente order by apellidos;

v\_apellido varchar2(25);

v\_fecha\_pedido date;

begin

OPEN cur1;

LOOP

FETCH cur1 INTO v\_apellido, v\_fecha\_pedido;

EXIT WHEN cur1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_apellido || ' \* ' || v\_fecha\_pedido);

END LOOP;

CLOSE cur1;

end;

begin Apellido\_fechapedidos;

end;

**2. Codificar un procedimiento que muestre la descripción de cada categoría y el número de productos que tiene.**

CREATE OR REPLACE PROCEDURE descrip\_stock

is

cursor cur1 is

Select categorias.descripcion,stock from categorias,productos

where categorias.id\_categoria=productos.id\_categoria;

v\_descripcion varchar2(25);

v\_stock number(2);

begin

OPEN cur1;

LOOP

FETCH cur1 INTO v\_descripcion, v\_stock;

EXIT WHEN cur1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_descripcion || ' \* ' || v\_stock);

END LOOP;

CLOSE cur1;

end;

begin descrip\_stock;

end;

**3. Escribir un procedimiento que reciba una cadena y visualice el apellido y el identificador de empleado de todos los empleados cuyo apellido contenga la cadena especificada. Al finalizar visualizar el número de empleados mostrados.**

CREATE OR REPLACE PROCEDURE apellido\_id(P\_Apellido varchar)

is

cursor cur1 is

Select id\_empleado from empleados

where apellidos=P\_Apellido ;

v\_id\_empleado number(3);

v\_contador number(2);

begin

OPEN cur1;

DBMS\_OUTPUT.PUT\_LINE(P\_Apellido);

LOOP

FETCH cur1 INTO v\_id\_empleado;

EXIT WHEN cur1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_id\_empleado);

v\_contador:=v\_contador+1;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(v\_contador);

CLOSE cur1;

end;

begin apellido\_id('Viescas');

end;

**4. Escribir un programa que visualice el apellido y el total gastado de los cinco clientes que más han gastado.**

CREATE OR REPLACE PROCEDURE apellido\_total

is

cursor cur1 is

select \* from(select apellidos, sum(precio\*cantidad)as total

from clientes cl natural join pedidos natural join detalles\_pedidos d

group by id\_cliente, apellidos order by total desc)where rownum<6;

v\_total number(10);

v\_apellido varchar(25);

begin

OPEN cur1;

LOOP

FETCH cur1 INTO v\_apellido,v\_total;

EXIT WHEN cur1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_apellido || '\*' || v\_total);

END LOOP;

CLOSE cur1;

end;

begin apellido\_total;

end;

**5. Codificar un programa que visualice los dos productos que menos se han vendido de cada categoría.**

create or replace

PROCEDURE productos\_menos\_vendidos

is

cursor cur1 (p\_id\_categoria number) is

select \* from(select distinct nombre, sum(cantidad) as total

from productos

natural join detalles\_pedidos

where id\_categoria=p\_id\_categoria

group by nombre

order by total )where rownum <3;

v\_id\_categoria number(3);

v\_cantidad number(10);

v\_nombre varchar(100);

begin

v\_id\_categoria:=1;

LOOP

OPEN cur1(v\_id\_categoria);

DBMS\_OUTPUT.PUT\_LINE('Categoria ' ||v\_id\_categoria);

loop

FETCH cur1 INTO v\_nombre,v\_cantidad;

EXIT WHEN cur1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_nombre || '\*' || v\_cantidad);

end loop;

v\_id\_categoria:=v\_id\_categoria+1;

CLOSE cur1;

EXIT WHEN v\_id\_categoria=7;

END LOOP;

end;

execute productos\_menos\_vendidos;

**6. Escribir un programa que muestre en formato similar a las rupturas de control los siguientes datos:**

**a. Para cada producto: nombre y precio\_venta.**

**b. Para cada categoría: número de productos y valor de los productos en stock de esa categoría.**

**c. Al final del listado: número total de productos y valor de todos los productos en stock.**

create or replace

PROCEDURE ejercicio6\_tema12

is

cursor cur1 is

select nombre, precio\_venta, stock, id\_categoria from productos

order by id\_categoria;

v\_id\_categoria number(3);

v\_id\_catcomodin number(3):=1;

v\_nombre varchar(100);

v\_precio\_venta number(10);

v\_precio\_venta\_total number(10):=0;

v\_stock number(10):=0;

v\_stock\_total number(10):=0;

v\_numeroprod number(3):=0;

v\_numeroprod\_total number(3):=0;

begin

OPEN cur1;

FETCH cur1 INTO v\_nombre,v\_precio\_venta, v\_stock,v\_id\_categoria;

DBMS\_OUTPUT.PUT\_LINE('Categoria ' || v\_id\_categoria || ':');

while(cur1%found) loop

v\_stock\_total:=v\_stock\_total+v\_stock;

v\_precio\_venta\_total:=v\_precio\_venta\_total+v\_precio\_venta\*v\_stock;

if(v\_id\_catcomodin!=v\_id\_categoria) then

v\_id\_catcomodin:=v\_id\_catcomodin+1;

DBMS\_OUTPUT.PUT\_LINE('Categoria ' || v\_id\_categoria || ':');

end if;

DBMS\_OUTPUT.PUT\_LINE(v\_nombre || '\*' || v\_precio\_venta);

v\_numeroprod:=v\_numeroprod+1;

v\_numeroprod\_total:=v\_numeroprod\_total+v\_numeroprod;

v\_numeroprod:=0;

v\_stock:=0;

v\_precio\_venta:=0;

FETCH cur1 INTO v\_nombre,v\_precio\_venta, v\_stock,v\_id\_categoria;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Stock total: '||v\_stock\_total|| ' ' ||'Precio venta total: '|| v\_precio\_venta\_total|| ' ' || 'Numero de productos total: '||v\_numeroprod\_total);

CLOSE cur1;

end;

execute ejercicio6\_tema12;

**7. Desarrollar un procedimiento que permita insertar nuevas categorías según las siguientes especificaciones:**

**- Se pasará al procedimiento la descripción de la categoría.**

**- El procedimiento insertará la fila nueva asignando como número de categoría la decena siguiente al número mayor de la tabla.**

**- Se incluirá la gestión de posibles errores.**

create or replace

PROCEDURE ejercicio7\_tema12(p\_descripcion varchar2)

is

cursor cur1 is

select distinct max(id\_categoria) from productos;

v\_id\_categoria number(3);

begin

open cur1;

fetch cur1 into v\_id\_categoria;

v\_id\_categoria:=v\_id\_categoria+10;

insert into categorias(descripcion,id\_categoria)

values(p\_descripcion,v\_id\_categoria);

close cur1;

end;

execute ejercicio7\_tema12('nuevo producto');

**8. Escribir un procedimiento que reciba todos los datos de un nuevo producto y procese la transacción de alta, gestionando posibles errores.**

**9. Codificar un procedimiento que reciba como parámetros un número de categoría, un importe y un porcentaje, y que suba el precio a todos los productos de la categoría indicada en la llamada. La subida será el porcentaje o el importe que se indica en la llamada (el que sea más beneficioso para la empresa en cada caso).**

**10. Realizar un procedimiento que suba el precio\_venta a todos los productos de la categoría indicada en la llamada (dos parámetros: id\_categoria y aumento en tanto por ciento).**

**11. Modificar el procedimiento anterior utilizando ROWID.**

**12. Escribir un procedimiento que suba el precio de todos los productos cuyo precio\_venta sea menor que el precio medio de su categoría. La subida será del 50 por 100 de la diferencia entre el precio\_venta del producto y la media de su categoría. Se deberá hacer que la transacción no se quede a medias, y se gestionarán los posibles errores.**

create or replace

PROCEDURE ejercicio12\_tema12

is

cursor cur1 is

select nombre,id\_categoria,precio\_venta from productos;

v\_nombre varchar2(100);

v\_precio\_venta number (10):=0;

v\_precio\_venta\_medio number(10):=0;

v\_id\_categoria number(3);

v\_calcular\_subida number(10):=0;

begin

open cur1;

fetch cur1 into v\_nombre,v\_id\_categoria,v\_precio\_venta;

while(cur1%found) loop

select avg(precio\_venta) into v\_precio\_venta\_medio

from productos where id\_categoria=v\_id\_categoria

group by id\_categoria;

if v\_precio\_venta<v\_precio\_venta\_medio then

v\_calcular\_subida:=v\_precio\_venta\_medio-v\_precio\_venta;

v\_calcular\_subida:=v\_calcular\_subida\*0.5+v\_precio\_venta;

update productos

set precio\_venta=v\_calcular\_subida

where nombre=v\_nombre;

DBMS\_OUTPUT.PUT\_LINE(v\_nombre||' Se ha actualizado correctamente');

end if;

fetch cur1 into v\_nombre,v\_id\_categoria,v\_precio\_venta;

end loop;

EXCEPTION

WHEN no\_data\_found THEN

raise\_application\_error(-20001,'La categoria no existe');

WHEN program\_error then

raise\_application\_error(-20000, 'Ha habido un problema');

end;

execute ejercicio12\_tema12;

**13. Diseñar una aplicación que simule un listado de los ingresos obtenidos de los pedidos de nuestros clientes según las siguientes especificaciones:**

**- El listado tendrá el siguiente formato:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Datos del pedido:……..(1) Cliente:…(2) Empleado que atendió:…….(3) Fecha: ………..(4)**

**Nombre de producto: ………..(5) unidades pedidas: ………..(6) precio por unidad: …….(7)**

**Número de líneas del pedido :…………..(8)**

**Importe total del pedido:…………..(9)**

**---------------**

**Datos del pedido:……..(1) Cliente:…(2) Empleado que atendió:…….(3) Fecha: ………..(4)**

**Nombre de producto: ………..(5) unidades pedidas: ………..(6) precio por unidad: …….(7)**

**Número de líneas del pedido :…………..(8)**

**Importe total del pedido:…………..(9)**

**---------------**

**Importe total de todos los pedidos del cliente:…………..(10)**

**- - -**

**Importe total de todos los pedidos:…………..(11)**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Donde:**

**- 1,2, 3 y 4 corresponden a número de pedido, apellidos del cliente, apellidos del empleado y fecha del pedido.**

**- 5, 6 y 7 son el nombre de producto, unidades pedidas y precio por unidad aplicado a cada línea del pedido.**

**- 8 es el número de líneas que tiene el pedido.**

**- 9 es el importe total del pedido.**

**- 10 es el importe total de los pedidos del cliente.**

**- 11 es el importe total de todos los pedidos.**

**El listado irá ordenado por apellidos del cliente.**

**14. Crear la tabla T\_INGRESOS con las columnas: *número de pedido, apellidos del cliente, apellidos del empleado que le atendió, fecha del pedido e importe total del pedido.* Modificar la aplicación anterior para que, en lugar de realizar el listado directamente en pantalla, guarde los datos en la tabla. Se controlarán todas las posibles incidencias que puedan ocurrir durante el proceso**