



CONCESIONARIO COCHES PL-SQL

MARIO JIMÉNEZ MARSET

ÍNDICE

1. ENUNCIADO – OBJETIVOS	3
2. DESARROLLO – PROCEDIMIENTOS.....	3

1. ENUNCIADO – OBJETIVOS

En esta práctica se pedía realizar, a partir del código PL-SQL facilitado, añadir a las tablas más coches y vendedores, añadir un trigger que haga aparecer el inventario por pantalla, además de los costes de los coches que aún no se han vendido.

Además, se pedía llamar a los procedimientos ya existentes en el código para realizar ventas por los distintos vendedores hasta finalizar el stock de coches.

Por último se muestran las ventas realizadas por cada vendedor.

2. DESARROLLO – PROCEDIMIENTOS

Se muestra el código completo con comentarios explicativos:

```
--en los tres bloques anónimos, se muestra cómo, si existe la tabla se borra
--para posteriormente ser creada
BEGIN
EXECUTE IMMEDIATE 'DROP TABLE Vendedor';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
/
BEGIN
EXECUTE IMMEDIATE 'DROP TABLE Venta';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
/
BEGIN
EXECUTE IMMEDIATE 'DROP TABLE Coche';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
/
--creación de la tabla coche
CREATE TABLE Coche
(
id_coche NUMBER(10),
matricula VARCHAR2(20),
color VARCHAR2(10),
marca VARCHAR2(10),
modelo VARCHAR2(10),
valor NUMBER(10),
vendido CHAR(1)
);
--creación de la tabla vendedor
CREATE TABLE Vendedor
(
id_vendedor NUMBER(10),
nombre VARCHAR2(10),
comision NUMBER(10)
);
```

--creación de la tabla venta

```
CREATE TABLE Venta
(
  id_venta NUMBER(10),
  valor_venta NUMBER(10),
  id_coche NUMBER(10),
  id_vendedor NUMBER(6)
);
```

--inserción de los datos en las tablas

--se introducen diferentes valores para visualizar posteriormente la diferencia del valor - de venta

```
INSERT INTO Coche VALUES (1,'matricula1', 'rojo','seat','model1',10,'F');
INSERT INTO Coche VALUES (2,'matricula2', 'negro','dacia','model2',30,'F');
INSERT INTO Coche VALUES (3,'matricula3', 'azul','volkswagen','model3',80,'T');
INSERT INTO Coche VALUES (4,'matricula4', 'rojo','seat','model4',20,'T');
INSERT INTO Coche VALUES (5,'matricula5', 'verde','dacia','model5',120,'F');
INSERT INTO Coche VALUES (6,'matricula6', 'azul','volkswagen','model6',310,'T');
INSERT INTO Coche VALUES (7,'matricula7', 'morado','opel','model7',42,'F');
INSERT INTO Coche VALUES (8,'matricula8', 'rosa','pontiac','model8',16,'T');
```

--se insertan cuatro vendedores con comisiones distintas

```
INSERT INTO Vendedor VALUES (1,'Juan',20);
INSERT INTO Vendedor VALUES (2,'Pedro',20);
INSERT INTO Vendedor VALUES (3,'Robustiano',30);
INSERT INTO Vendedor VALUES (4,'Crispulo',40);
```

/

--se crea un procedimiento donde se insertan las ventas

```
CREATE OR REPLACE PROCEDURE InsertaVta(idventa
Venta.id_venta%TYPE,idvendedor Vendedor.id_vendedor%TYPE, idcoche
Coche.id_coche%TYPE)
IS
```

--variable para la comisión que tiene cada vendedor

```
comision NUMBER(10);
```

--variable para guardar el precio del coche con el que vamos a trabajar

```
coche_precio NUMBER(10);
```

--variable que guarda el valor total de la venta

```
valor_venta NUMBER(10);
```

--cursores que extraen los datos que necesito de las tablas Vendedor y Coche

```
CURSOR c_leer_comision IS SELECT comision FROM Vendedor WHERE
id_vendedor = idvendedor;
```

```
CURSOR c_leer_coche_valor IS SELECT valor FROM Coche where id_coche =
idcoche;
```

```
BEGIN
```

--Abrimos los cursores

```
OPEN c_leer_comision;
```

```
OPEN c_leer_coche_valor;
```

--Recorremos los cursores almacenando el resultado en las variables

--previamente creadas

```
FETCH c_leer_comision INTO comision;
```

```
FETCH c_leer_coche_valor INTO coche_precio;
```

--Cálculo de la venta más la comisión

```
valor_venta:=(coche_precio/100*comision)+coche_precio;
```

--Insertamos en la tabla Ventas el resultado de la venta final

```
INSERT INTO Venta VALUES (idventa,valor_venta,idcoche,idvendedor);
```

--Actualizamos y marcamos que es coche se encuentra vendido

```
UPDATE Coche SET vendido='T' WHERE Coche.id_coche = idcoche;
```

```

CLOSE c_leer_coche_valor;
CLOSE c_leer_comision;
--Cerramos los cursores y se acaba el procedimiento
END InsertaVta;
/

--se crea el trigger que se dispara para hacer que aparezca el inventario y costes de los
--coches que siguen sin venderse
create or replace trigger cochesSinVender
--for each row, por cada fila
after insert on Venta for each row
--se declara el cursor y la variable en la que hacer el rowtype
declare
    cursor c_leerInventario is select *from Coche where vendido='F';
    inventario c_leerInventario%rowtype;
begin
--se abre el cursor, se recorre con el fetch (dentro un while y un loop) y se imprime por
--pantalla la información
    dbms_output.put_line('El inventario de coches sin vender es ');
    open c_leerInventario;
    fetch c_leerInventario into inventario;
    WHILE c_leerInventario%FOUND LOOP
        dbms_output.put_line(inventario.id_coche || ', ' || inventario.matricula || ', ' ||
inventario.color || ', ' || inventario.marca || ', ' || inventario.modelo || ', ' || inventario.valor ||
', ' || inventario.vendido);
        dbms_output.put_line('-----');
        fetch c_leerInventario into inventario;
    end loop;

    dbms_output.put_line('+++++++');
    ++');
    end c_leerInventario;
    -- end;
--end;
--se insertan ocho ventas diferentes
/
BEGIN
    InsertaVta(1,1,2);
    InsertaVta(2,2,4);
    InsertaVta(3,3,7);
    InsertaVta(4,4,8);
    InsertaVta(5,1,3);
    InsertaVta(6,2,1);
    InsertaVta(7,4,5);
    InsertaVta(8,3,6);
END;
/

--Creamos el proceso que recibe por parámetro el id de vendedor para mostrar sus
--ventas
CREATE OR REPLACE PROCEDURE mostrar_ventas_vendedor(idvend
Vendedor.id_vendedor%TYPE)
IS
--Creamos un cursor que guarda las ventas con el id que hemos pasado por
--parámetros
CURSOR c_mostrar_venta IS SELECT * FROM venta WHERE id_vendedor=idvend;

--Creamos un cursor que guarda el valor total de la suma de un campo de la tabla

```

```
--Venta donde se cumple una condición
CURSOR c_suma_total IS SELECT sum(valor_venta) FROM Venta WHERE
id_vendedor=idvend;

v_venta c_mostrar_venta%ROWTYPE;
suma_t NUMBER(20);

BEGIN
--Mostramos por pantalla una lista de las ventas realizadas como el cabecero de la
tabla creada
dbms_output.put_line('VENTAS REALIZADAS POR EL VENDEDOR ' || idvend);
dbms_output.put_line('+++++');
--Abrimos el cursor
OPEN c_mostrar_venta;
OPEN c_suma_total;

--Recorremos el cursor almacenando dentro de venta el resultado
FETCH c_mostrar_venta INTO v_venta;

--Creamos un bucle para que recorra todas las filas de las tablas, las almacene y las
vaya mostrando una por una y así finalmente tener el resultado de todas
WHILE c_mostrar_venta%FOUND LOOP

dbms_output.put_line(v_venta.id_venta || '|' || v_venta.valor_venta || '|' ||
v_venta.id_coche);
dbms_output.put_line('_____');
FETCH c_mostrar_venta into v_venta;

--Cerramos el bucle
END LOOP;
FETCH c_suma_total INTO suma_t;
dbms_output.put_line(' Suma total del valor de las ventas: '||suma_t);

--Cerramos el cursor
CLOSE c_suma_total;
CLOSE c_mostrar_venta;
END mostrar_ventas_vendedor;
/

--Muestra las ventas del vendedor 1
BEGIN
mostrar_ventas_vendedor(1);
END;
/

--Muestras las ventas del vendedor 2
BEGIN
mostrar_ventas_vendedor(2);
END;
/

--Muestras las ventas del vendedor 3
BEGIN
mostrar_ventas_vendedor(3);
END;
/
```

```
--Muestras las ventas del vendedor 4  
BEGIN  
mostrar_ventas_vendedor(4);  
END;  
/
```

3. RESULTADOS

En este apartado se muestran los resultados demandados en la práctica. En primer lugar, se muestra el inventario de coches con los coches sin vender hasta que no queda ninguno:

```

Statement processed.
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
2, matricula2, negro, dacia, model2, 30, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
7, matricula7, morado, opel, model7, 42, F
-----
+++++++
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
7, matricula7, morado, opel, model7, 42, F
-----
+++++++
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
7, matricula7, morado, opel, model7, 42, F
-----
+++++++
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
+++++++
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
-----
+++++++
El inventario de coches sin vender es
1, matricula1, rojo, seat, model1, 10, F
-----
5, matricula5, verde, dacia, model5, 120, F
-----
+++++++
El inventario de coches sin vender es
5, matricula5, verde, dacia, model5, 120, F
-----
+++++++
El inventario de coches sin vender es
+++++++

```

Después, se muestran las ventas realizadas por cada uno de los cuatro vendedores:


```
Statement processed.
VENTAS REALIZADAS POR EL VENDEDOR 1
+++++
1|36|2
-----
5|96|3
-----
Suma total del valor de las ventas: 132

Statement processed.
VENTAS REALIZADAS POR EL VENDEDOR 2
+++++
2|24|4
-----
6|12|1
-----
Suma total del valor de las ventas: 36

Statement processed.
VENTAS REALIZADAS POR EL VENDEDOR 3
+++++
3|55|7
-----
8|403|6
-----
Suma total del valor de las ventas: 458

Statement processed.
VENTAS REALIZADAS POR EL VENDEDOR 4
+++++
4|22|8
-----
7|168|5
-----
Suma total del valor de las ventas: 190
```