

CONCESIONARIO

COCHES PL-SQL

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

[1. ENUNCIADO – OBJETIVOS 3](#_Toc124782069)

[2. DESARROLLO – PROCEDIMIENTOS 3](#_Toc124782070)

# 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.

# 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

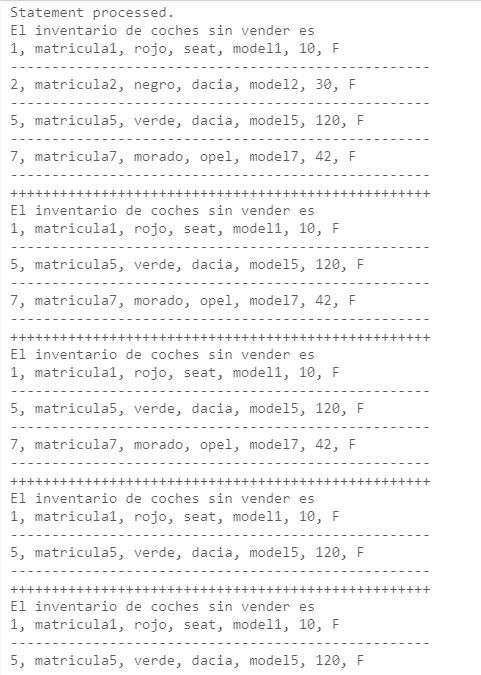
mostrar\_ventas\_vendedor(4);

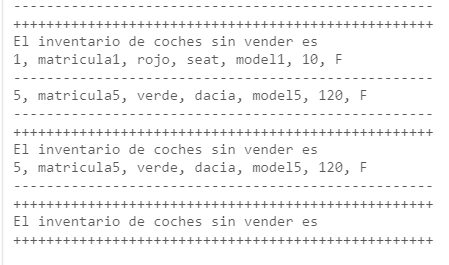
END;

/

# 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:





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

