

Bases de Datos Práctica3

BDG3304

Pedro Ballesta Garres

Antonio Rubio

Práctica 2:

Creación de tablas:

```
CREATE TABLE CONCESIONARIO (  
  cod NUMBER NOT NULL,  
  nombre VARCHAR(15) NOT NULL,  
  marca NUMBER NOT NULL,  
  direccion VARCHAR(150) NOT NULL,  
  telefono VARCHAR(15) ,  
  fecha_ina DATE NOT NULL,  
  
  CONSTRAINT PK_CONCESIONARIO PRIMARY KEY (cod)  
);
```

```
CREATE TABLE MARCA (  
  id NUMBER NOT NULL,  
  nombre VARCHAR(15) NOT NULL,  
  concesionario_principal NUMBER,  
  
  CONSTRAINT PK_MARCA PRIMARY KEY (id),  
  CONSTRAINT FK_MARCA_CONCESIONARIO FOREIGN KEY  
  (concesionario_principal) REFERENCES CONCESIONARIO(cod)  
);
```

```
ALTER TABLE CONCESIONARIO DROP CONSTRAINT  
FK_CCONCESIONARIO_MARCA;
```

```
ALTER TABLE CONCESIONARIO ADD CONSTRAINT FK_CCONCESIONARIO_MARCA  
FOREIGN KEY (marca) REFERENCES MARCA(id) INITIALLY DEFERRED;
```

```
CREATE TABLE MODELO (  
  marca NUMBER NOT NULL,  
  modelo VARCHAR(15) NOT NULL,  
  precio_recomendado NUMBER,  
  
  CONSTRAINT PK_MODELO PRIMARY KEY (marca, modelo),  
  CONSTRAINT FK_MODELO_MARCA FOREIGN KEY (marca) REFERENCES  
  MARCA(id),  
  CONSTRAINT PRECIORECOMENDADOPOSITIVO CHECK (precio_recomendado  
  >0)  
);
```

```

CREATE TABLE CLIENTE(
dni CHAR(9) NOT NULL,
nombre VARCHAR(20) NOT NULL,
direccion VARCHAR(150),
telefono VARCHAR(15),
cuenta_bancaria NUMBER(20) NOT NULL,

CONSTRAINT PK_CLIENTE PRIMARY KEY (dni)
);

```

```

CREATE TABLE CATALOGO(
concesionario NUMBER NOT NULL,
marca NUMBER NOT NULL,
modelo VARCHAR(15) NOT NULL,
precio_orientativo NUMBER,

CONSTRAINT PK_CATALOGO PRIMARY KEY (concesionario, marca,
modelo),
CONSTRAINT PRECIOPOSITIVO CHECK (precio_orientativo >0),
CONSTRAINT FK_CATALOGO_MODELO FOREIGN KEY (marca, modelo)
REFERENCES MODELO(marca, modelo) ,
CONSTRAINT FK_CATALOGO_CONCESIONARIO FOREIGN KEY (concesionario)
REFERENCES CONCESIONARIO(cod)

);

```

```

CREATE TABLE COCHE (
concesionario NUMBER,
marca NUMBER NOT NULL,
modelo VARCHAR(15) NOT NULL,
numcoche NUMBER NOT NULL,
color VARCHAR(15) NOT NULL,
puertas NUMBER(1),
cilindrada NUMBER(4),

CONSTRAINT PK_COCHE PRIMARY KEY (numcoche),
CONSTRAINT NPUERTAS CHECK (puertas IN (3,5)),
CONSTRAINT FK_COCHE_CATALOGO FOREIGN KEY (marca,
modelo,concesionario) REFERENCES CATALOGO(marca,
modelo,concesionario),
);

```

```

CREATE TABLE VENDEDOR (
dni CHAR(9) NOT NULL,
nombre VARCHAR(20) NOT NULL,
telefono VARCHAR(15) NOT NULL,
vendedor_jefe CHAR(9),
concesionario NUMBER NOT NULL,

CONSTRAINT PK_VENDEDOR PRIMARY KEY (dni),
CONSTRAINT FK_VENDEDOR_CONCESIONARIO FOREIGN KEY (concesionario)
REFERENCES CONCESIONARIO(cod),
CONSTRAINT FK_VENDEDOR_JEFE FOREIGN KEY (vendedor_jefe)
REFERENCES VENDEDOR(dni)

);

```

```

CREATE TABLE VENTA (
num NUMBER NOT NULL,
coche NUMBER NOT NULL,
cliente CHAR(9) NOT NULL,
vendedor CHAR(9),
fecha_venta DATE NOT NULL,
pvp NUMBER,
matricula VARCHAR(8),

CONSTRAINT PK_VENTA PRIMARY KEY (num),
CONSTRAINT FK_VENTA_COHE FOREIGN KEY (coche) REFERENCES
COCHE(numcoche),
CONSTRAINT FK_VENTA_CLIENTE FOREIGN KEY (cliente) REFERENCES
CLIENTE(dni),
CONSTRAINT FK_VENTA_VENDEDOR FOREIGN KEY (vendedor) REFERENCES
VENDEDOR(dni),
CONSTRAINT PVPPOSITIVO CHECK (pvp >0)
);

```

Introducción de datos:

```

--CONCESIONARIO
INSERT INTO CONCESIONARIO (cod, nombre, marca, direccion,
telefono, fecha_ina)
VALUES(1,'MERCÉDES MURCIA',1,'C/DeLaMercedes N°22', '+34 968
665522', TO_DATE('20/02/2010', 'dd/mm/yyyy') );

INSERT INTO CONCESIONARIO (cod, nombre, marca, direccion,
telefono, fecha_ina)
VALUES(2,'PEPE AUTOS',1,'C/LACASADEPEPE N°1', '+34 968 127664',
TO_DATE('11/02/2010', 'dd/mm/yyyy') );

INSERT INTO CONCESIONARIO (cod, nombre, marca, direccion,
telefono, fecha_ina)
VALUES(3,'BMW MURCIA',3,'C/PETIRROJO N°7', '+34 968 780100',
TO_DATE('05/03/2005', 'dd/mm/yyyy') );

```

```

INSERT INTO CONCESIONARIO (cod, nombre, marca, direccion,
telefono, fecha_ina)
VALUES(4,'AUDI MURCIA',2,'C/PINZON N°9', '+34 968 132546',
TO_DATE('09/02/2003', 'dd/mm/yyyy') );

INSERT INTO CONCESIONARIO (cod, nombre, marca, direccion,
telefono, fecha_ina)
VALUES(5,'TAPADERA DROGA',4,'C/VERTEDEROITALIANO N°0', '+34 968
111555', TO_DATE('14/04/2010', 'dd/mm/yyyy') );

--MARCA

INSERT INTO MARCA (id, nombre, concesionario_principal)
VALUES(1,'MERCEDES', 1);

INSERT INTO MARCA (id, nombre, concesionario_principal)
VALUES(2,'AUDI', 4);

INSERT INTO MARCA (id, nombre, concesionario_principal)
VALUES(3,'BMW', 3);

INSERT INTO MARCA (id, nombre, concesionario_principal)
VALUES(4,'FANTASICO', 5);

--MODELO

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(1,'Clase S',200000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(1,'Clase E',150000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(1,'Clase A',100000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(2,'a3',30000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(2,'a5',50000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(3,'X1',40000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(3,'X6',70000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(4,'Batmovil',2000000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(4,'Kitt',1000000);

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES(4,'Gadchetomovil',1500000);

```

```

INSERT INTO MODELO (marca, modelo, precio_recomendado)
VALUES (4, 'Barbie', 900000);

--CLIENTE

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111111L', 'Juan', 'c/ falsa1', 999333555,
21030166321234567890);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111122T', 'Pepe', 'c/ falsa2', 999365556,
21030166321234567891);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111133F', 'Pedro', 'c/ falsa3', 999333565,
21030166321234567892);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111444J', 'Sergio', 'c/ falsa5', 999333243,
21030166321234567893);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111543S', 'Fernano', 'c/ falsa6', 999333567,
21030166321234567894);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('11111563S', 'Julio', 'c/ falsa4', 999333567,
21030166321234567895);

INSERT INTO CLIENTE (dni, nombre, direccion, telefono,
cuenta_bancaria)
VALUES ('1111154Y', 'Eduardo', 'c/ falsa7', 999333521,
21030166321234567896);

--CATALOGO

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (1, 1, 'Clase S', 190000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (1, 1, 'Clase E', 150000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (1, 1, 'Clase A', 100000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)

```

```

VALUES (2,1,'Clase S',190000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (2,1,'Clase E',150000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (2,1,'Clase A',100000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (4,2,'a3',190000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (4,2,'X1',190000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (3,3,'X6',200000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (1,1,'Clase E',180000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (4,2,'a5',70000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,4,'Kitt',1900000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,4,'Barbie',1900000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,4,'Gadchetomovil',1900000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,4,'Batmovil',1900000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,1,'Clase A',120000);

INSERT INTO CATALOGO (concesionario, marca, modelo,
precio_orientativo)
VALUES (5,2,'a3',192000);

```

-- COCHE

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(1,1,'Clase S', 1, 'Blanco' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(1,1,'Clase S', 2, 'Negro' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(1,1,'Clase E', 3, 'Azul' , 5, 1500);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(2,1,'Clase S', 4, 'Rojo' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(2,1,'Clase E', 5, 'Blanco' , 5, 1800);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(3,2,'a3', 6, 'Azul' , 3, 1600);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(3,2,'a5', 7, 'Blanco' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(4,3,'X1', 8, 'Blanco' , 3, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(4,3,'X6', 9, 'Plata' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(1,1,'Clase S', 10, 'Blanco' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(5,4,'Batmovil', 12, 'Negro' , 3, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(5,4,'Gadchetomovil', 13, 'negro' , 5, 2000);
```

```
INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(5,4,'Kitt', 14, 'negro' , 5, 2000);
```



```

INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(5,4,'Barbie', 15, 'Rosa' , 3, 2000);

INSERT INTO COCHE (concesionario, marca, modelo, numcoche,
color, puertas, cilindrada)
VALUES(1,1,'Clase S', 16, 'Azul' , 5, 2000);

--Vendedores

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('88844477R','Boss Mercedes', 985664112, null , 1);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('11144477Q','Chico Mercedes', 985685712, '88844477R' ,
1);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('88844477E','chica Mercedes', 982164112, '88844477R' ,
1);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES(88889477P,'Er Pepe', 988968968, null , 2);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES(88841777R,'Boss Audi', 985664562, null , 3);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('88844477Y','chico Audi', 980364112, 88841777R , 1);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('98785423A','Al Capone', 985946112, null , 5);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('88846537R','Boss BMW', 985524112, null , 4);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('8851477R','chico BMW', 985129872, '88846537R' , 4);

INSERT INTO VENDEDOR (dni, nombre, telefono, vendedor_jefe,
concesionario)
VALUES('88835564R','Vito Corleone', 985668322, null , 5);

--VENTA

```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(1,2,'11111111L','11144477Q',TO_DATE('11/06/2013',  
'dd/mm/yyyy'),150000,'3495BFC');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(2,1,'11111133F','11144477Q',TO_DATE('21/09/2011',  
'dd/mm/yyyy'),200000,'3489BTR');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(3,3,'11111122T','11144477Q',TO_DATE('12/01/2012',  
'dd/mm/yyyy'),100000,'9659BIY');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(4,13,'11111154Y','98785423A',TO_DATE('24/06/2013',  
'dd/mm/yyyy'),2000000,'4706GTR');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(5,14,'11111444J','98785423A',TO_DATE('14/07/2013',  
'dd/mm/yyyy'),1800000,'4357DGR');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(6,15,'11111563S','98785423A',TO_DATE('23/02/2011',  
'dd/mm/yyyy'),2300000,'2576MQW');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(7,8,'11111133F','88531477R',TO_DATE('21/03/2011',  
'dd/mm/yyyy'),30000,'4526DOT');
```

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(8,9,'11111444J','88531477R',TO_DATE('25/06/2011',  
'dd/mm/yyyy'),70000,'7135HAP');
```

1. Listar por pantalla, en dos columnas, los nombres de todos los vendedores y de todos los clientes. Ambas columnas han de estar ordenadas alfabéticamente y mostrarse de la forma que se indica a continuación. La última línea ha de mostrar el número total de vendedores y de clientes. Tenga en cuenta que, lógicamente, puede haber distinto número de vendedores que de clientes. No hay que escribir procedimientos ni funciones.

VENEDORES

CLIENTES

<nombre vendedor_1> <nombre cliente_1>

... ..

<nombre vendedor_n> ...

<nombre cliente_m>

TOTAL: <n> VENEDORES Y <m> CLIENTES

Código:

```
SET SERVEROUTPUT ON;
DECLARE
CURSOR vendedores IS
SELECT nombre
FROM VENDEDOR
ORDER BY nombre;
CURSOR clientes IS
SELECT nombre
FROM CLIENTE
ORDER BY nombre;
vender VENDEDOR.nombre%TYPE;
client CLIENTE.nombre%TYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE('Vendedores          Clientes');
DBMS_OUTPUT.PUT_LINE('-----          -----');
OPEN vendedores;
OPEN clientes;

LOOP
FETCH vendedores INTO vender;
FETCH clientes INTO client;

EXIT WHEN (vendedores%NOTFOUND OR clientes%NOTFOUND);
DBMS_OUTPUT.PUT_LINE(RPAD(vender, 35) || client);
END LOOP;
IF (vendedores%NOTFOUND) THEN

IF (clientes%FOUND) THEN
```

```

LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(' ', 35) || client);
FETCH clientes INTO client;
EXIT WHEN (clientes%NOTFOUND);
END LOOP;
END IF;

ELSE
LOOP
DBMS_OUTPUT.PUT_LINE(vender);
FETCH vendedores INTO vender;
EXIT WHEN (vendedores%NOTFOUND);
END LOOP;
END IF;
DBMS_OUTPUT.PUT_LINE('-----
-----');
DBMS_OUTPUT.PUT_LINE('TOTAL: ' ||
vendedores%ROWCOUNT || ' Vendedores Y ' ||
clientes%ROWCOUNT || ' Clientes');
CLOSE vendedores;
CLOSE clientes;
END;

```

Resustado:

bloque anónimo terminado

Vendedores	Clientes
-----	-----
Al Capone	Eduardo
Boss Mercedes	Fernano
chica Mercedes	Juan
Chico Mercedes	Julio
Vito Corleone	Pedro
	Pepe
	Sergio

TOTAL: 5 Vendedores Y 7 Clientes	

2. Hacer un listado que para cada (nombre de) marca muestre sus concesionarios, indicando cuál de ellos es el concesionario principal de la marca, y para cada concesionario muestre los modelos de coche que tiene en su catálogo, indicando el recuento de coches de ese modelo que tiene el concesionario (vendidos o no, da igual). Siga el formato que se indica a continuación. Note que el listado debe estar ordenado alfabéticamente por nombre de marca y modelo, y que también debe incluir todos los cálculos que se indican (número de concesionarios de cada marca, cuántos modelos de cada marca en cada concesionario, recuento de coches de cada marca/modelo...). En este ejercicio obligatoriamente hay que usar cursores parametrizados. No hay que escribir procedimientos ni funciones.

```
MARCA <nombre marca_1>
--- <número de> CONCESIONARIOS:
----- <nombre concesionario_1>
----- <número de> MODELOS:
----- <modelo_1>. COCHES: <recuento>
...
----- <modelo_n>. COCHES: <recuento>
...
----- <nombre concesionario_i> ES EL PRINCIPAL
----- <número de> MODELOS:
----- <modelo_1>. COCHES: <recuento>
...
----- <modelo_p>. COCHES: <recuento>
...
----- <nombre concesionario_c>
...
...
MARCA <nombre marca_m>
...
```

Código:

```
SET SERVEROUTPUT ON;
DECLARE

CURSOR concesionarios (marc MARCA.id%type) IS
SELECT nombre, cod
FROM CONCESIONARIO
WHERE marca=marc;

CURSOR modelos (conc CONCESIONARIO.cod%type) IS
SELECT modelo
FROM CATALOGO
WHERE concesionario=conc;

CURSOR marcas IS
SELECT id, nombre, concesionario_principal
FROM MARCA;
numModelos NUMBER:=0;
numConcesionarios NUMBER:=0;
```

```

BEGIN

FOR m IN marcas LOOP

SELECT COUNT(*) INTO numConcesionarios
FROM CONCESIONARIO
WHERE marca=m.id;

DBMS_OUTPUT.PUT_LINE('MARCA ' || m.nombre);
DBMS_OUTPUT.PUT_LINE('-----' || numConcesionarios || '
CONCESIONARIOS:');
FOR co IN CONCESIONARIOS (m.id) LOOP

SELECT COUNT(*) INTO numModelos
FROM CATALOGO
WHERE co.cod=concesionario;

DBMS_OUTPUT.PUT('-----' || co.nombre);
IF (co.cod=m.concesionario_principal) THEN
DBMS_OUTPUT.PUT_LINE(' ES EL PRINCIPAL');

ELSE DBMS_OUTPUT.NEW_LINE;
END IF;
DBMS_OUTPUT.PUT_LINE('-----' || numModelos || '
MODELOS:');
FOR mo IN modelos(co.cod) LOOP
DBMS_OUTPUT.PUT_LINE('-----' || RPAD(mo.modelo, 15));
END LOOP;
END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;
END;

```

Resultado:

bloque anónimo terminado

MARCA MERCEDES

-----2 CONCESIONARIOS:

-----MERCEDES MURCIA ES EL PRINCIPAL

-----3 MODELOS:

-----Clase A

-----Clase E

-----Clase S

-----PEPE AUTOS

-----3 MODELOS:

-----Clase A

-----Clase E

-----Clase S

MARCA BMW

-----1 CONCESIONARIOS:

-----BMW MURCIA ES EL PRINCIPAL

-----1 MODELOS:

-----X6

MARCA FANTASICO

-----1 CONCESIONARIOS:

-----TAPADERA DROGA ES EL PRINCIPAL

-----4 MODELOS:

-----Barbie

-----Batmovil

-----Gadchetomovil

-----Kitt

3. Escribir un **package** llamado **P_CON** con el siguiente contenido:

- Una **función** llamada **compras_cliente**, que dado un dni de un cliente devuelva 1) '**CLIENTE SIN COCHES**' si ese dni corresponde a un cliente que no ha adquirido ningún coche, 2) la matrícula del coche comprado, si el cliente ha adquirido sólo un coche, 3) '**CLIENTE CON <número de> COCHES**' si el cliente ha comprado más de un coche (sustituyendo <numero de> por el recuento de los coches que ha adquirido), y 4) '**CLIENTE INEXISTENTE**' si no existe dicho cliente. Importante: la función hay que programarla manejando excepciones. No se puede utilizar cursores explícitos ni sentencias SELECT COUNT que eviten el manejo de excepciones (sólo se puede usar para calcular el número de coches una vez se sabe que el cliente ha comprado más de uno).

- Un **procedimiento** **ventas_vendedor** tal que dado un dni de un vendedor, muestre por pantalla el nombre del vendedor, su concesionario y la marca que representa, y el listado de las ventas de coches en las que ha participado, indicando la fecha de venta, el modelo del coche y el pvp. Siga el formato indicado a continuación para la salida por pantalla.

```
VENDEDOR <nombre vendedor>
CONCESIONARIO <nombre concesionario>
MARCA <nombre marca>
<número de> VENTAS:
--FECHA:<fecha>; MODELO:<nombre modelo>. PVP: <euros> €.
...
--FECHA:<fecha>; MODELO:<nombre modelo>. PVP: <euros> €.
```

Código:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE
PACKAGE P_CON IS
    FUNCTION compras_clientes (unDNI IN venta.cliente%type)
        RETURN VARCHAR2;
    PROCEDURE ventas_vendedor (dniv IN vendedor.dni%type);
END P_CON;
```

```
create or replace
PACKAGE BODY P_CON IS
    FUNCTION compras_clientes (unDNI IN venta.cliente%type)
        RETURN VARCHAR2 IS
        dniClienteV venta.cliente%type;
        matriculaC venta.matricula%type;
        dniClienteC cliente.dni%type;
        numeroCochesVendidos NUMBER;

    BEGIN

        select matricula INTO matriculaC
        from venta
        where unDNI=cliente;
        RETURN (matriculaC);
    EXCEPTION
        WHEN TOO_MANY_ROWS THEN
```



```

BEGIN
    select count(cliente) INTO NumeroCochesVendidos
        from venta
        where unDni=cliente
        group by cliente;
    RETURN NumeroCochesVendidos;
END;
WHEN NO_DATA_FOUND THEN
    BEGIN
        select dni into dniClienteC
            from cliente
            where unDni=dni;
        RETURN ('Cliente sin coches');
    EXCEPTION WHEN NO_DATA_FOUND THEN RETURN ('Cliente
inexistente');
    END;
END;

PROCEDURE ventas_vendedor (dniv IN vendedor.dni%type)
IS
nombrevendedor vendedor.nombre%type;
concevededor concesionario.nombre%type;
marcaconce marca.nombre%type;
contador number;
dnia vendedor.dni%type;
CURSOR curs IS
    select fecha_venta, pvp, modelo
    from venta, coche
    where dniv=vendedor and numcoche=coche;
    ult curs%rowtype;

begin

    select dni into dnia
    from vendedor
    where dni=dniv;
    if (dnia<>dniv) then
        DBMS_OUTPUT.PUT_LINE('Ese vendedor no existe.');
```

```

    else
    begin
        select nombre into nombrevendedor
        from vendedor
        where dni=dniv;

        select nombre into concevededor
        from concesionario
        where cod in (select concesionario
                        from vendedor
                        where dni=dniv);
```

```

select nombre into marcaconce
  from marca
 where id in (select marca
              from concesionario
              where nombre=concevendedor);

select count (*) into contador
  from venta
 where vendedor=dniv
 group by vendedor;
  dbms_output.put_line ('VENDEDOR '|| ' ' ||
nombrevendedor);
  dbms_output.put_line ('CONCESIONARIO ' || concevendedor);
  dbms_output.put_line ('MARCA ' || marcaconce);
  dbms_output.put_line (' ' || contador || ' ' || 'VENTAS :
');

  if contador>0 then
  begin
    open curs;
    loop
      fetch curs into ult;
      exit when curs%NOTFOUND;
      dbms_output.put_line ('--- FECHA : ' || ult.fecha_venta
||'; MODELO : ' || ult.modelo ||'. PVP : ' || ult.pvp || '€ .');
    end loop;
    close curs;
  end;
  else dbms_output.put_line('ningunaventa');
  end if;
end;

end if;
end;

END P_CON;

```

Fichero de pruebas:

```

SET SERVEROUTPUT ON;
DECLARE
mensaje VARCHAR2(50);
BEGIN

mensaje:=p_con.compras_clientes('11111122T');
DBMS_OUTPUT.PUT_LINE(mensaje);
p_con.ventas_vendedor('11144477Q');

END;

```

Resultado:

bloque anónimo terminado

9659BIY

VENDEDOR Chico Mercedes

CONCESIONARIO MERCEDES MURCIA

MARCA MERCEDES

3 VENTAS :

--- FECHA : 21/09/11; MODELO : Clase S. PVP : 200000€ .

--- FECHA : 11/06/13; MODELO : Clase S. PVP : 150000€ .

--- FECHA : 12/01/12; MODELO : Clase E. PVP : 100000€ .

4. Escribir los **triggers** de base de datos necesarios para imponer que un vendedor sólo puede participar en las ventas de coches expuestos en el concesionario en el que está contratado. Tenga en cuenta que un vendedor podría cambiar de concesionario (de entre los que representan la misma marca). Considere también que podría modificarse el coche y/o el vendedor de una venta existente.

Código:

```
CREATE OR REPLACE
TRIGGER comprobacion_coordinador
BEFORE
INSERT OR UPDATE OF num, coche, vendedor
ON VENTA
FOR EACH ROW
DECLARE
concesionario_coche coche.concesionario%type;
BEGIN

SELECT concesionario INTO concesionario_coche
FROM VENDEDOR
WHERE dni=:NEW.vendedor AND concesionario =( select
concesionario
from coche
where numcoche =

:NEW.coche);
EXCEPTION
WHEN NO_DATA_FOUND THEN
```

```
RAISE_APPLICATION_ERROR(-20003,'EL vendedor no pertenece al  
mismo concesionario que el coche.');
```

END;

CREATE OR REPLACE

```
TRIGGER cambio_concesionario  
BEFORE  
UPDATE OF concesionario  
ON VENDEDOR  
FOR EACH ROW  
DECLARE  
marcas1 marca.id%type;  
marcas2 marca.id%type;  
BEGIN
```

```
SELECT marca INTO marcas1  
FROM Concesionario  
WHERE cod=:NEW.concesionario;
```

```
SELECT marca INTO marcas2  
FROM Concesionario  
WHERE cod=:OLD.concesionario;
```

```
IF (marcas1<>marcas2)  
then  
RAISE_APPLICATION_ERROR(-20004,'El Vendedor no puede cambiar de  
concesionario si no es de la misma marca.');
```

END IF;

```
EXCEPTION  
WHEN NO_DATA_FOUND THEN  
RAISE_APPLICATION_ERROR(-20003,'EL concesionari no existe.');
```

END;

Ficheros de prueba:

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta,  
pvp, matricula)  
VALUES(7,16,'11111111L','98785423A',TO_DATE('11/06/2013',  
'dd/mm/yyyy'),150000,'3765BFC');
```

```
UPDATE Vendedor set concesionario='1' WHERE dni='88835564R';
```

```
UPDATE Vendedor set concesionario='8' WHERE dni='88835564R';
```

Resultado:

TRIGGER comprobacion_coordinador compilado

Error que empieza en la línea 4 del comando:

```
INSERT INTO VENTA (num, coche, cliente, vendedor, fecha_venta, pvp, matricula)

VALUES(7,16,'11111111L','98785423A',TO_DATE('11/06/2013',
'dd/mm/yyyy'),150000,'3765BFC')
```

Informe de error:

Error SQL: ORA-20003: EL vendedor no pertenece al mismo concesionario que el coche.

ORA-06512: en "BDG3304.COMPROBACION_COORDINADOR", línea 22

ORA-04088: error durante la ejecución del disparador
'BDG3304.COMPROBACION_COORDINADOR'

TRIGGER cambio_concesionario compilado

Error que empieza en la línea 4 del comando:

```
UPDATE Vendedor set concesionario='1' WHERE dni='88835564R'
```

Informe de error:

Error SQL: ORA-20004: El Vendedor no puede cambiar de concesionario si no es de la misma marca.

ORA-06512: en "BDG3304.CAMBIO_CONCESIONARIO", línea 19

ORA-04088: error durante la ejecución del disparador
'BDG3304.CAMBIO_CONCESIONARIO'

Error que empieza en la línea 6 del comando:

```
UPDATE Vendedor set concesionario='8' WHERE dni='88835564R'
```

Informe de error:

Error SQL: ORA-20003: EL concesionari no existe.

ORA-06512: en "BDG3304.CAMBIO_CONCESIONARIO", línea 25

ORA-04088: error durante la ejecución del disparador
'BDG3304.CAMBIO_CONCESIONARIO'