

Desarrollo de aplicaciones multiplataforma Acceso a datos

Tema 2-Práctica 12: Gestión de ventas

Veremos cómo Oracle permite proporcionar una solución relacional y otra objeto-relacional.

Modelo lógico para una base de datos relacional

CLIENTE

(idcliente, nombre, dirección, población, cdpostal, provincia, nif, tel1, tel2, tel3)

PRODUCTOS(idproducto, descripción, pvp, stockactual)

VENTAS(idventas, idcliente, fechaventa)

LINEASVENTAS(idventa, numerolinea, idproducto, cantidad)

1. Diseña el script para el modelo relacional

Modelo lógico para una base de datos objeto-relacional

2. Diseña el modelo objeto-relacional

Tipo **tip_telefonos** será un **varray** de 3 elementos varchar(15).

```
SQL> CREATE TYPE TIP_TELEFONOS AS VARRAY(3) OF VARCHAR(15);  
2 /
```

Tipo creado.

Tipo **tip_direccion**

calle varchar(50) ,
población varchar(50),
codpos varchar(20),
provincia varchar(40)

Tipo **tip_cliente**

idcliente number,
nombre varchar(50),
direc tip_dirección,
nif varchar(9),

telef tip_telefonos

```
SQL> CREATE TYPE TIP_DIRECCION AS OBJECT(  
  2  CALLE VARCHAR(50),  
  3  POBLACION VARCHAR(50),  
  4  CODPOSTAL NUMBER(5),  
  5  PROVINCIA VARCHAR(40)  
  6  );  
  7  /
```

Tipo creado.

```
SQL> CREATE or replace TYPE TIP_CLIENTE AS OBJECT(  
  2  IDCLIENTE NUMBER,  
  3  NOMBRE VARCHAR(50),  
  4  DIREC TIP_DIRECCION,  
  5  NIF VARCHAR(9),  
  6  TELEF TIP_TELEFONOS  
  7  );  
  8  /
```

Tipo creado.

Tipo **tip_producto**

idproducto number,
descripción varchar(80),
pvp number,
stockactual number

```
SQL> CREATE TYPE TIP_PRODUCTO AS OBJECT(  
  2  IDPRODUCTO NUMBER,  
  3  DESCRIPCION VARCHAR(80),  
  4  PVP NUMBER,  
  5  STOCKACTUAL NUMBER);  
  6  /
```

Tipo creado.

Tipo **tip_linea_venta**

numerolinea number,
idproducto REF tip_producto,
cantidad number

```
SQL> CREATE TYPE TIP_LINEAVENTA AS OBJECT(  
  2  NUMEROLINEA NUMBER,  
  3  IDPRODUCTO REF TIP_PRODUCTO,  
  4  CANTIDAD NUMBER);  
  5  /
```

Tipo creado.

Creamos un tipo tabla anidada para contener las líneas de una venta:
Tipo tabla anidada **tip_lineas_venta** de tipo tip_lineaventa.

Creamos un tipo venta **tip_venta** para los datos de las ventas, cada venta tendrá un atributo **Líneas** del tipo tabla anidada definida anteriormente.

En el tipo venta se definirá una función miembro **total_venta** que calcula el total de la venta de las líneas de venta que forman parte de una venta. (líneas.COUNT devuelve el número de líneas de una venta)

Tipo **tip_venta**

Idventa number

Idcliente REF tip_cliente,

Fechaventa date,

Líneas tip_líneas_venta,

Member function total_venta retorna number

```
SQL> CREATE TYPE TIP_LINEAS_VENTA AS TABLE OF TIP_LINEAVENTA;  
2 /
```

```
SQL> CREATE TYPE TIP_VENTA AS OBJECT (  
2 IDVENTA NUMBER,  
3 IDCLIENTE REF TIP_CLIENTE,  
4 FECHAVENTA DATE,  
5 LINEAS TIP_LINEAS_VENTA,  
6 MEMBER FUNCTION TOTAL_VENTA RETURN NUMBER);  
7 /
```

tipo creado.

Crea el cuerpo para la función total_venta, que calcula el total de la venta de las líneas de venta que forman parte de una venta. (líneas.COUNT devuelve el número de líneas de una venta) (Al ser una función miembro trabaja sobre objetos y no sobre las tablas, por lo que los select que se necesiten se harán del tipo from dual)

3.

```

SQL> CREATE OR REPLACE TYPE BODY TIP_VENTA AS
  2 MEMBER FUNCTION TOTAL_VENTA RETURN NUMBER IS
  3 TOTAL NUMBER:=0;
  4 LINEA TIP_LINEAVENTA;
  5 PRODUCT TIP_PRODUCTO;
  6 begin
  7   FOR I IN 1..LINEAS.COUNT LOOP
  8     LINEA:=LINEAS(I);
  9     SELECT Deref(LINEA.IDPRODUCTO) INTO PRODUCT FROM DUAL;
 10     TOTAL:=TOTAL + LINEA.CANTIDAD * PRODUCT.PVP;
 11   END LOOP;
 12   RETURN TOTAL;
 13 END;
 14 END;
 15 /

```

Cuerpo del tipo creado.

```

SQL> create or replace type body tip_ventas as
  2 MEMBER FUNCTION total_venta RETURN number IS
  3 producto tip_producto;
  4 total number;
  5 BEGIN
  6 total:=0;
  7 for i in 1..Lineas.count loop
  8   select Deref(Lineas(i).idproducto) into producto from dual;
  9   total:=total+(producto.pvp*Lineas(i).cantidad);
 10 end loop;
 11 return total;
 12 END;
 13 END;
 14 /

```

Cuerpo del tipo creado.

4. Persiste los objetos

Crear las tablas:

Tabla_clientes de tipo tip_clientes con una clave primaria idcliente

Tabla_productos de tipo tip_producto con una clave primaria idproducto

Tabla_ventas de tipo tip_ventas con una clave primaria idventa

```
SQL> CREATE TABLE TABLA_CLIENTES OF TIP_CLIENTE(IDCLIENTE PRIMARY KEY);
```

Tabla creada.

```
SQL> CREATE TABLE TABLA_PRODUCTOS OF TIP_PRODUCTO (IDPRODUCTO PRIMARY KEY);
```

Tabla creada.

```
SQL> CREATE TABLE TABLA_VENTAS OF TIP_VENTA (  
  2     IDVENTA PRIMARY KEY  
  3     )NESTED TABLE LINEAS STORE AS TABLA_LINEAS;
```

Tabla creada.

5. Inserta datos en las tablas

Inserta estos datos 2 clientes y 5 productos:

(1,'Luis Garcia', tip_direccion('calle Las Flores,23','Guadalajara','19003','Guadalajara'),'34343434L',tip_telefonos('949876655','949876655'))

```
SQL> INSERT INTO TABLA_CLIENTES VALUES(1,'Luis Garcia',
  2 tip_direccion('calle Las Flores,23','Guadalajara','19003','Guadalajara'),'34343434L',tip_telefo
nos('949876655','949876655'));
```

1 fila creada.

(2,'ana Serrano, tip_direccion('calle Galiana,6','Guadalajara','19004','Guadalajara'),'76767667F',tip_telefonos('94980009'))

```
SQL> INSERT INTO TABLA_CLIENTES VALUES (2,'Ana Serrano',
  2 tip_direccion('calle Galiana,6','Guadalajara','19004','Guadalajara'),'76767667F',tip_telefonos(
'94980009'));
```

1 fila creada.

(1, 'caja de cristal de murano',100,5)

(2, 'bicicleta city',120,15)

(3, '100 lapices de colores',20,5)

(4, 'ipad',600,5)

(5, 'ordenador portatil',400,10)

```
SQL> INSERT INTO TABLA_PRODUCTOS VALUES (1, 'caja de cristal de murano',100,5);
```

1 fila creada.

```
SQL> INSERT INTO TABLA_PRODUCTOS VALUES (2, 'bicicleta city',120,15);
```

1 fila creada.

```
SQL> INSERT INTO TABLA_PRODUCTOS VALUES (3, '100 lapices de colores',20,5);
```

1 fila creada.

```
SQL> INSERT INTO TABLA_PRODUCTOS VALUES (4, 'ipad',600,5);
```

1 fila creada.

```
SQL> INSERT INTO TABLA_PRODUCTOS VALUES (5, 'ordenador portatil',400,10);
```

1 fila creada.

En la tabla `tabla_ventas` se define una tabla anidada para el atributo `LINEAS` de tipo `tip_venta`, contendrá las líneas de venta.

Inserto la venta con `idventa 1` para el cliente `idcliente 1`:

```
SQL> INSERT INTO TABLA_VENTAS SELECT 1, REF(C),SYSDATE, TIP_LINEAS_VENTA() FROM TABLA_CLIENTES C WHERE C.IDCLIENTE=1;
```

1 fila creada.

Inserto en la tabla_ventas dos líneas de venta para el `idventa 1`, para los productos de línea 1 la cantidad es 1 y para los productos de línea 2 la cantidad es 2

Líneas para la venta 1 *****

Línea1

```
SQL> INSERT INTO TABLE (SELECT V.LINEAS FROM TABLA_VENTAS V WHERE V.IDVENTA=1)(SELECT 1,REF(P),1 FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=1);
```

Línea 2

```
SQL> INSERT INTO TABLE (SELECT V.LINEAS FROM TABLA_VENTAS V WHERE V.IDVENTA=1)(SELECT 2,REF(P),2 FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=2);
```

1 fila creada.

OTRA FORMA CON UN SOLO INSERT CON

INSERT INTO ...VALUES...

```
SQL> INSERT INTO TABLA_VENTAS VALUES (1, (SELECT REF(C) FROM TABLA_CLIENTES C WHERE C.IDCLIENTE=1),SYSDATE, TIP_LINEAS_VENTA(TIP_LINEAVENTA(1,(SELECT REF(P) FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=1),1), TIP_LINEAVENTA(2,(SELECT REF(P) FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=2),2)));
```

1 fila creada.

Inserto la venta con idventa 2 para el cliente idcliente 1:

Líneas para la venta 2 *****

```
SQL> INSERT INTO TABLA_VENTAS SELECT 2, REF(C),SYSDATE, TIP_LINEAS_VENTA() FROM TABLA_CLIENTES C WHERE C.IDCLIENTE=1;
```

1 fila creada.

Línea1

```
SQL> INSERT INTO TABLE (SELECT V.LINEAS FROM TABLA_VENTAS V WHERE V.IDVENTA=2)(SELECT 1,REF(P),2 FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=4);
```

1 fila creada.

Línea 2

```
SQL> INSERT INTO TABLE (SELECT V.LINEAS FROM TABLA_VENTAS V WHERE V.IDVENTA=2)(SELECT 2,REF(P),1 FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=1);
```

1 fila creada.

Línea 3

```
SQL> INSERT INTO TABLE (SELECT V.LINEAS FROM TABLA_VENTAS V WHERE V.IDVENTA=2)(SELECT 3,REF(P),4 FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=5);
```

1 fila creada.

OTRA FORMA CON UN SOLO INSERT into....VALUES...

```
SQL> INSERT INTO TABLA_VENTAS VALUES (2, (SELECT REF(C) FROM TABLA_CLIENTES C WHERE C.IDCLIENTE=1),SYSDATE, TIP_LINEAS_VENTA(TIP_LINEAVENTA(1,(SELECT REF(P) FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=1),2), TIP_LINEAVENTA(2,(SELECT REF(P) FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=2),1), TIP_LINEAVENTA(3,(SELECT REF(P) FROM TABLA_PRODUCTOS P WHERE IDPRODUCTO=3),4)));
```

1 fila creada.

6. Consultas

6.1 Visualizar todas las líneas de venta para la venta id 2.

```
SQL> SELECT U.LINEAS FROM TABLA_VENTAS U WHERE U.IDVENTA=2  
2 ;
```

```
LINEAS(NUMEROLINEA, IDPRODUCTO, CANTIDAD)
```

```
-----  
TIP_LINEAS_VENTA(TIP_LINEAVENTA(1, 0000220208EB6AA2DEF3704A5EA1B4340DE352B3559F2  
9B93BD4D54C6282C20BD9BFB4AEDC, 2), TIP_LINEAVENTA(2, 000022020848BBBD8CC2654809B  
109E3440739CDE39F29B93BD4D54C6282C20BD9BFB4AEDC, 1), TIP_LINEAVENTA(3, 000022020  
88633B75A473646709AC19E9CC1FF11AC9F29B93BD4D54C6282C20BD9BFB4AEDC, 4))
```

Otra forma sin sacar el nombre del objeto, usando

TABLE

```
SQL> SELECT lin.* FROM TABLA_VENTAS v, table(v.lineas) lin WHERE U.IDVENTA=2;
```

```
NUMEROLINEA
```

```
IDPRODUCTO
```

```
CANTIDAD
```

```
-----  
1  
0000220208EB6AA2DEF3704A5EA1B4340DE352B3559F29B93BD4D54C6282C20BD9BFB4AEDC  
2
```

```
2  
000022020848BBBD8CC2654809B109E3440739CDE39F29B93BD4D54C6282C20BD9BFB4AEDC  
1
```

```
NUMEROLINEA
```

```
IDPRODUCTO
```

```
CANTIDAD
```

```
-----  
3  
00002202088633B75A473646709AC19E9CC1FF11AC9F29B93BD4D54C6282C20BD9BFB4AEDC  
4
```

6.2 Visualizar todas las líneas de venta para la venta id 2, obteniendo los productos en vez de su oid.

```
SELECT lin.numerolinea, lin.cantidad, deref(lin.idproducto) FROM TABLA_VENTAS v, table(v.lineas) lin WHERE V.IDVENTA=2;
```

```
SQL> SELECT lin.numerolinea, lin.cantidad, deref(lin.idproducto) FROM TABLA_VENTAS v, table(v.lineas) lin WHERE V.IDVENTA=2;
```

```
NUMEROLINEA  CANTIDAD
```

```
-----
```

```
DEREF(LIN.IDPRODUCTO)(IDPRODUCTO, DESCRIPCION, PVP, STOCKACTUAL)
```

```
-----
```

```
          1          2  
TIP_PRODUCTO(4, 'ipad', 600, 5)
```

```
          2          1  
TIP_PRODUCTO(1, 'caja de cristal de murano', 100, 5)
```

```
          3          4  
TIP_PRODUCTO(5, 'ordenador portatil', 400, 10)
```

Otra forma de obtenerlo mismo usando la clausula THE

```
SQL> select NUMEROLINEA, CANTIDAD, Deref(IDPRODUCTO) FROM THE(SELECT T.LINEAS FROM TABLA_VENTAS T WHERE IDVENTA=2);
```

```
NUMEROLINEA  CANTIDAD
```

```
-----
```

```
DEREF(IDPRODUCTO)(IDPRODUCTO, DESCRIPCION, PVP, STOCKACTUAL)
```

```
-----
```

```
          1          2  
TIP_PRODUCTO(4, 'ipad', 600, 5)
```

```
          2          1  
TIP_PRODUCTO(1, 'caja de cristal de murano', 100, 5)
```

```
          3          4  
TIP_PRODUCTO(5, 'ordenador portatil', 400, 10)
```

6.3 Visualizar todas las líneas de venta de todas las ventas.

```
SQL> SELECT lin.numerolinea,lin.cantidad,deref(lin.idproducto) FROM TABLA_VENTAS v, table(v.lineas
) lin ;
```

```
NUMEROLINEA  CANTIDAD
-----
DEREF(LIN.IDPRODUCTO)(IDPRODUCTO, DESCRIPCION, PUP, STOCKACTUAL)
-----
          1          1
TIP_PRODUCTO(1, 'caja de cristal de murano', 100, 5)

          2          2
TIP_PRODUCTO(2, 'bicicleta city', 120, 15)

          1          2
TIP_PRODUCTO(4, 'ipad', 600, 5)
```

```
NUMEROLINEA  CANTIDAD
-----
DEREF(LIN.IDPRODUCTO)(IDPRODUCTO, DESCRIPCION, PUP, STOCKACTUAL)
-----
          2          1
TIP_PRODUCTO(1, 'caja de cristal de murano', 100, 5)

          3          4
TIP_PRODUCTO(5, 'ordenador portatil', 400, 10)
```

6.4 Consulta el nombre del cliente idcliente 2

```
SQL> SELECT NOMBRE FROM TABLA_CLIENTES WHERE IDCLIENTE=2;
```

```
NOMBRE
-----
Ana Serrano
```

6.5 Modifica el nombre del cliente 2 por Rosa Serrano

```
SQL> UPDATE TABLA_CLIENTES SET NOMBRE='Rosa Serrano' WHERE IDCLIENTE=2;
```

1 fila actualizada.

6.6 Consulta la dirección del cliente 2 y modifica la calle por calle Estopa,34

```
SQL> SELECT DIREC FROM TABLA_CLIENTES WHERE IDCLIENTE=2;
```

```
DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA)
```

```
TIP_DIRECCION('calle Galiana,6', 'Guadalajara', 19004, 'Guadalajara')
```

```
SQL> UPDATE TABLA_CLIENTES SET DIREC=TIP_DIRECCION('Calle Estopa,34','Guadalajara',19004, 'Guadalajara') WHERE IDCLIENTE=2;
```

1 fila actualizada.

```
SQL> SELECT DIREC FROM TABLA_CLIENTES WHERE IDCLIENTE=2;
```

```
DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA)
```

```
TIP_DIRECCION('Calle Estopa,34', 'Guadalajara', 19004, 'Guadalajara')
```

Otra forma accediendo a la calle

```
UPDATE TABLA_CLIENTES SET DIREC.calle= 'Calle Estopa,34' WHERE  
DIREC.calle= 'calle Galiana,6' ;
```

6.7 Consulta todos los datos del cliente 1 y añade un nuevo teléfono a su lista de teléfonos.

```
SQL> SELECT * FROM TABLA_CLIENTES WHERE IDCLIENTE=1;
```

```
IDCLIENTE NOMBRE
```

```
DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA)
```

```
NIF
```

```
TELEF
```

```
1 Luis Garcia
```

```
TIP_DIRECCION('calle Las Flores,23', 'Guadalajara', 19003, 'Guadalajara')  
34343434L
```

```
TIP_TELEFONOS('949876655', '949876655')
```

Otra forma usando value

```
SQL> SELECT value(c) FROM TABLA_CLIENTES c WHERE IDCLIENTE=1;
```

```
VALUE(C)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA), NIF,
```

```
TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalajara',  
, 19003, 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '949876655'))
```

Actualizo

```
SQL> update TABLA_CLIENTES SET TELEF=TIP_TELEFONOS ('949876655', '949876655', '900000000') WHERE ID
IENTE=1;

1 fila actualizada.

SQL> SELECT value(c) FROM TABLA_CLIENTES c WHERE IDCLIENTE=1;

VALUE(C)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA), NIF,
-----
TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalajara'
, 19003, 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '949876655', '9
00000000'))
```

6.8 Visualiza el nombre del cliente que ha realizado la venta 2

```
SQL> SELECT U.IDCLIENTE.NOMBRE FROM TABLA_VENTAS U WHERE U.IDVENTA=2;

IDCLIENTE.NOMBRE
-----
Luis Garcia
```

Haz lo mismo Usando Deref

```
SQL> SELECT Deref(IDCLIENTE).NOMBRE FROM TABLA_VENTAS U WHERE U.IDVENTA=2;

Deref(IDCLIENTE).NOMBRE
-----
Luis Garcia
```

6.9 Visualiza todos los datos del cliente anterior, que ha realizado la venta 2

```
SQL> SELECT Deref(IDCLIENTE) FROM TABLA_VENTAS U WHERE U.IDVENTA=2;

Deref(IDCLIENTE)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA
-----
TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalajara'
, 19003, 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '949876655', '9
00000000'))
```

6.10 Visualizar el numero de venta y el total de ventas hechas por el cliente 1

```
SQL> SELECT V.IDVENTA,V.TOTAL_VENTA() FROM TABLA_VENTAS V WHERE V.IDCLIENTE.IDCLIENTE=1;
```

IDVENTA	V.TOTAL_VENTA()
1	340
2	2900

Otra forma

```
SELECT V.IDVENTA,V.TOTAL_VENTA() FROM TABLA_VENTAS V  
WHERE Deref(V.IDCLIENTE).IDCLIENTE=1;
```

6.11 Visualiza las ventas de todos los clientes

```
SQL> SELECT V.IDVENTA, Deref(IDCLIENTE),V.FECHAVENTA FROM TABLA_VENTAS V ;
```

IDVENTA	Deref(IDCLIENTE)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA	FECHAVENTA
1	TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalajara', 19003, 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '949876655', '900000000'))	23/09/12

2		
---	--	--

IDVENTA	Deref(IDCLIENTE)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOSTAL, PROVINCIA	FECHAVENTA
	TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalajara', 19003, 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '949876655', '900000000'))	23/09/12

El cliente 2 no tiene ventas

6.12 Crea un bloque PL que visualice los datos de la venta cuyo identificador recibe.

```
-----PL-----
CREATE OR REPLACE PROCEDURE VER_VENTA (ID NUMBER) AS
LIN NUMBER;
CANT NUMBER;
IMPORTE NUMBER;
TOTAL_V NUMBER;
PRODUC TIP_PRODUCTO:=TIP_PRODUCTO(NULL,NULL,NULL,NULL);
CLI TIP_CLIENTE:=TIP_CLIENTE(NULL,NULL,NULL,NULL,NULL);
DIR TIP_DIRECCION:=TIP_DIRECCION(NULL,NULL,NULL,NULL);
FEC DATE;

CURSOR C1 IS SELECT lin.numerolinea,DEREF(lin.idproducto),lin.cantidad
from tabla_ventas v, TABLE(v.lineas) lin WHERE v.IDVENTA=ID;
BEGIN
    SELECT Deref(IDCLIENTE),FECHAVENTA,V.TOTAL_VENTA() INTO CLI, FEC, TOTAL_V
    FROM TABLA_VENTAS V WHERE IDVENTA=ID;
    DIR:=CLI.DIREC;
    DBMS_OUTPUT.PUT_LINE('NUMERO DE VENTA: '||ID||'*FECHA DE VENTA: '||FEC);
    DBMS_OUTPUT.PUT_LINE('CLIENTE: '||CLI.NOMBRE);
    DBMS_OUTPUT.PUT_LINE('DIRECCION: '||DIR.CALLE);
    DBMS_OUTPUT.PUT_LINE('*****');
    OPEN C1;
    FETCH C1 INTO LIN, PRODUC, CANT;
    WHILE C1%FOUND LOOP
        IMPORTE:=CANT*PRODUC.PVP;
        DBMS_OUTPUT.PUT_LINE(LIN||'*'||PRODUC.DESCRIPCION||'*'||PRODUC.PVP||'*'||CANT||'*'||IMPORTE);
        FETCH C1 INTO LIN,PRODUC,CANT;
    END LOOP;
    CLOSE C1;
    DBMS_OUTPUT.PUT_LINE('TOTAL VENTA: '||TOTAL_V);
END;
/
```

```
SQL> set serveroutput on;
SQL> begin
  2  ver_venta2(2);
  3  end;
  4  /
NUMERO DE VENTA:2*FECHA DE VENTA: 13/11/13
CLIENTE: Luis Garcia
DIRECCION: calle Las Flores,23
*****
1*ipad*600*2*1200
2*caja de cristal de murano*100*1*100
3*ordenador portatil*400*4*1600
TOTAL VENTA: 2900

Procedimiento PL/SQL terminado correctamente.
```

Otra forma:

```

SQL> CREATE OR REPLACE PROCEDURE datosVenta(pIdVenta NUMBER)
2 IS
3   v_idVenta NUMBER;
4   v_fechaVenta DATE;
5   v_nombre VARCHAR2(15);
6   v_calle VARCHAR2(30);
7   v_importe NUMBER;
8   CURSOR c_lineasVenta IS
9     SELECT lv.numlinea AS numlinea,
10            Deref(lv.idproducto).descripcion AS descripcion ,
11            Deref(lv.idproducto).ppv AS precio,
12            lv.cantidad AS cantidad
13     FROM ventas, table(lineas) lv
14     WHERE idventa= pIdVenta;
15
16   totalVenta number;
17
18 BEGIN
19   -- cabecera del cliente---
20
21   SELECT IDVENTA,FECHAVENTA,DEREF(V.IDCLIENTE).NOMBRE,
22          Deref(V.IDCLIENTE).DIREC.CALLE
23   INTO v_idVenta, v_fechaVenta, v_nombre, v_calle
24   FROM VENTAS V
25   WHERE v.idventa= pIdVenta;
26
27   DBMS_OUTPUT.PUT('NUMERO DE VENTA: ' || v_idVenta );
28   DBMS_OUTPUT.PUT_LINE('* FECHA DE VENTA: ' || v_fechaVenta);
29   DBMS_OUTPUT.PUT_LINE('CLIENTE: ' || v_nombre);
30   DBMS_OUTPUT.PUT_LINE('DIRECCION: ' || v_calle);
31   DBMS_OUTPUT.PUT_LINE('***** ');
32
33   -- lineas de venta-----
34   FOR Reg IN c_lineasVenta LOOP
35     DBMS_OUTPUT.PUT(reg.numlinea || '*');
36     DBMS_OUTPUT.PUT(reg.descripcion || '*');
37     DBMS_OUTPUT.PUT(reg.precio || '*');
38     DBMS_OUTPUT.PUT(reg.cantidad || '*');
39     v_importe := reg.cantidad * reg.precio;
40     DBMS_OUTPUT.PUT_LINE(v_importe || '*');
41   END LOOP;
42
43   -- totales de la venta
44   SELECT v.total_venta() INTO totalVenta FROM ventas v
45   WHERE v.idventa = pIdVenta;
46
47   DBMS_OUTPUT.PUT_LINE('TOTAL VENTA: ' || totalVenta);
48 END datosVenta;
49 /

```

accediendo al dato que
necesito y usando un
alias




```
SQL>
SQL>
SQL> // Bloque de prueba de procedimiento
```

Procedimiento creado.

```
SQL> SET SERVEROUTPUT ON
SQL> BEGIN
  2     datosVenta(1);
  3     DBMS_OUTPUT.PUT_LINE('-----');
  4     datosVenta(2);
  5 END;
  6 /
```

```
NUMERO DE VENTA: 1* FECHA DE VENTA: 13/11/13
CLIENTE: Luis Garcia
DIRECCION: calle Las Flores, 23
*****
1*caja de cristal de Murano*100*1*100*
2*bicicleta city*120*2*240*
TOTAL VENTA: 340
-----
```

```
NUMERO DE VENTA: 2* FECHA DE VENTA: 13/11/13
CLIENTE: Luis Garcia
DIRECCION: calle Las Flores, 23
*****
1*caja de cristal de Murano*100*2*200*
2*bicicleta city*120*1*120*
3*100 lapices de colores*20*4*80*
TOTAL VENTA: 400
```

Procedimiento PL/SQL terminado correctamente.

Otra forma con FOR:

6.12 Crea un procedimiento que reciba como parámetro un id de venta y visualice los datos de la venta

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE VISUALIZARVENTA(VENTAID NUMBER)AS
CURSOR C_DATOSVENTA IS SELECT L.*,DEREF(L.IDPRODUCTO) PROD FROM TABLA_VENTAS V, TABLE(V.LINEAS) L
WHERE V.IDVENTA = VENTAID;
V_NOMCLI VARCHAR(50);
V_CALLECLI VARCHAR(50);
V_FECHAVENTA DATE;
V_TOTALVENTA NUMBER;
BEGIN
SELECT FECHAVENTA INTO V_FECHAVENTA FROM TABLA_VENTAS V WHERE V.IDVENTA = VENTAID;
SELECT DEREF(V.IDCLIENTE).NOMBRE,DEREF(V.IDCLIENTE).DIREC.CALLE INTO V_NOMCLI,V_CALLECLI
FROM TABLA_VENTAS V WHERE V.IDVENTA = 2;
DBMS_OUTPUT.PUT_LINE('NUMERO DE VENTA: '||VENTAID||'*FECHA DE VENTA: '||V_FECHAVENTA);
DBMS_OUTPUT.PUT_LINE('CLIENTE: '||V_NOMCLI);
DBMS_OUTPUT.PUT_LINE('DIRECCION: '||V_CALLECLI);
DBMS_OUTPUT.PUT_LINE('*****');
FOR V_DATA IN C_DATOSVENTA LOOP
DBMS_OUTPUT.PUT_LINE(V_DATA.NUMEROLINEA||'*'||V_DATA.PROD.DESCRIPCION||'*'||V_DATA.PROD.PVP||
'*'||V_DATA.cantidad||'*'||V_DATA.PROD.PVP*V_DATA.cantidad);
END LOOP;
SELECT V.total_venta() INTO V_TOTALVENTA FROM TABLA_VENTAS V WHERE IDVENTA = VENTAID;
DBMS_OUTPUT.PUT_LINE('TOTAL VENTA: '||V_TOTALVENTA);
END;
/
```

```
create or replace procedure datosVenta(v_idventa number) as
v_venta tip_venta;
v_cliente tip_cliente;
begin
select value(v) into v_venta from tabla_ventas v where idventa = v_idventa;
select deref(v.idcliente) into v_cliente from tabla_ventas v where v.idventa = v_idventa;
dbms_output.put_line('NUMERO DE VENTA:' || v_venta.idventa || '*' || 'FECHA DE VENTA: ' || v_venta.fechaventa);
dbms_output.put_line('CLIENTE:' || v_cliente.idcliente);
dbms_output.put_line('DIRECCION:' || v_cliente.direc.calle);
dbms_output.put_line('*****');
for r_lineas in (select lin.numerolinea numlinea, deref(lin.idproducto) producto,
lin.cantidad cantidad from tabla_ventas v, table(v.lineas) lin where idventa = v_idventa) loop
dbms_output.put_line(r_lineas.numlinea || '*' || r_lineas.producto.descripcion
|| '*' || r_lineas.producto.pvp || '*' || r_lineas.cantidad || '*'
|| r_lineas.producto.pvp * r_lineas.cantidad);
end loop;
dbms_output.put_line('TOTAL VENTA:' || v_venta.total_venta());
end datosVenta;
/

begin
datosVenta(2);
end;
/
```

6.12 Crea un bloque PL que visualice los datos de la venta cuyo identificador recibe.

```
SQL> DECLARE
  2  codigov number:=&codigo;
  3  cursor c1 IS select v.idventa,DEREF(v.idcliente).idcliente,v.Fechaventa,v.Lineas from Tabla_ven
tas v where Deref(v.idcliente).idcliente=codigov;
  4  lin tip_lineas_venta;
  5  idv number;
  6  idcli number;
  7  Fechav date;
  8  idprod number;
  9  pdesc varchar(80);
 10  ppvp number;
 11  pstockactual number;
 12  BEGIN
 13  OPEN c1;
 14  LOOP
 15  FETCH c1 INTO idv,idcli,Fechav,lin;
 16  EXIT WHEN c1%NOTFOUND;
 17  DBMS_OUTPUT.PUT_LINE('Idventa: '||idv);
 18  DBMS_OUTPUT.PUT_LINE('Idcliente: '||idcli);
 19  DBMS_OUTPUT.PUT_LINE('Fecha Venta: '||Fechav);
 20  for j in 1..lin.count loop
 21  DBMS_OUTPUT.PUT_LINE('NumeroLinea: '||lin(j).numerolinea);
 22  select p.idproducto,p.descripcion,p.pvp,p.stockactual into idprod,pdesc,ppvp,pstockactual fr
om Tabla_productos p where p.idproducto=DEREF(lin(j).idproducto).idproducto;
 23  DBMS_OUTPUT.PUT_LINE('IdProducto: '||idprod);
 24  DBMS_OUTPUT.PUT_LINE('Descripcion Producto: '||pdesc);
 25  DBMS_OUTPUT.PUT_LINE('PVP: '||ppvp);
 26  DBMS_OUTPUT.PUT_LINE('Stock actual: '||pstockactual);
 27  DBMS_OUTPUT.PUT_LINE('Cantidad: '||lin(j).cantidad);
 28  END Loop;
 29  END LOOP;
 30  CLOSE c1;
 31  END;
 32  /
```

CON THE

```
SQL> CREATE OR REPLACE PROCEDURE VER_VENTA (ID NUMBER) AS
  2  LIN NUMBER;
  3  CANT NUMBER;
  4  IMPORTE NUMBER;
  5  TOTAL_U NUMBER;
  6  PRODUC TIP_PRODUCTO:=TIP_PRODUCTO(NULL,NULL,NULL,NULL);
  7  CLI TIP_CLIENTE:=TIP_CLIENTE(NULL,NULL,NULL,NULL,NULL);
  8  DIR TIP_DIRECCION:=TIP_DIRECCION(NULL,NULL,NULL,NULL);
  9  FEC DATE;
 10  CURSOR C1 IS SELECT NUMEROLINEA, DEREFER(IDPRODUCTO), CANTIDAD FROM THE(SELECT T.LINEAS FROM T
A_VENTAS T WHERE IDVENTA = ID);
 11  BEGIN
 12    SELECT DEREFER(IDCLIENTE),FECHAVENTA,U.TOTAL_VENTA() INTO CLI, FEC, TOTAL_U FROM TABLA_VENTA
WHERE IDVENTA=ID;
 13    DIR:=CLI.DIREC;
 14    DBMS_OUTPUT.PUT_LINE('NUMERO DE VENTA: '||ID||'*FECHA DE VENTA: '||FEC);
 15    DBMS_OUTPUT.PUT_LINE('CLIENTE: '||CLI.NOMBRE);
 16    DBMS_OUTPUT.PUT_LINE('DIRECCION: '||DIR.CALLE);
 17    DBMS_OUTPUT.PUT_LINE('*****');
 18    OPEN C1;
 19    FETCH C1 INTO LIN, PRODUC, CANT;
 20    WHILE C1%FOUND LOOP
 21      IMPORTE:=CANT*PRODUC.PUP;
 22      DBMS_OUTPUT.PUT_LINE(LIN||'*'||PRODUC.DESCRIPCION||'*'||PRODUC.PUP||'*'||CANT||'*'||IMPOR
;
 23      FETCH C1 INTO LIN,PRODUC,CANT;
 24    END LOOP;
 25    CLOSE C1;
 26    DBMS_OUTPUT.PUT_LINE('total venta: '||TOTAL_U);
 27  END;
 28  /
```

Procedimiento creado.

```
NUMERO DE VENTA:2*FECHA DE VENTA: 23/09/12
CLIENTE: Luis Garcia
DIRECCION: calle Las Flores,23
*****
1*ipad*600*2*1200
2*caja de cristal de murano*100*1*100
3*ordenador portatil*400*4*1600
total venta: 2900
```

Procedimiento PL/SQL terminado correctamente.