

ACCESO A DATOS – EJERCICIO 11

Realizado por:

Juan Carlos Tauroni Roldán,

Ignacio Pereiro Poyato,

Jorge María Huguet

Contenido

1.	Modelo lógico para una base de datos relacional	3
2.	Diseña el modelo objeto-relacional	3
3.	Crea el cuerpo para la función total_venta	4
4.	Persiste los objetos	4
5.	Inserta los datos.....	4
6.	Consultas	5
6.1.1	Visualizar todas las líneas de venta para la venta id 2.....	5
6.1.2	Haz la consulta de otra forma sin sacar el nombre del objeto, usando TABLE.....	5
6.2	Visualizar todas las líneas de venta para la venta id 2, obteniendo los productos en vez de su oid.	5
6.3	Visualizar todas las líneas de venta de todas las ventas.....	6
6.4	Consulta el nombre del cliente idcliente 2.	6
6.5	Modifica el nombre del cliente 2 por Rosa Serrano.....	6
6.6	Consulta la dirección del cliente 2 y modifica la calle por calle Estopa,34.....	6
6.7	Consulta todos los datos del cliente 1 y añade un nuevo teléfono a su lista de teléfonos. Haz la consulta de otra forma usando value.	7
6.8.1	Visualiza el nombre del cliente que ha realizado la venta.	7
6.8.2	Haz la misma consulta usando Deref	7
6.9	Visualiza todos los datos del cliente anterior, que ha realizado la venta 2.	8
6.10	Visualizar el número de venta y el total de ventas hechas por el cliente 1.....	8
6.11	Visualiza las ventas de todos los clientes.	8
6.12	Crea un procedimiento que reciba como parámetro un id de venta y visualice los datos de la venta cuyo identificador recibe.....	9

1. Modelo lógico para una base de datos relacional

Se crean las tablas.

```
create table cliente(
    idcliente number primary key,
    nombre varchar(50),
    direccion varchar(50),
    poblacion varchar(50),
    cdpostal varchar(20),
    provincia varchar(40),
    nif varchar(9) NOT NULL
);

create table telefonos(
    idcliente number,
    telefono number,
    constraint pk_idcliente_tel primary key (idcliente,telefono),
    constraint fk_idclientes foreign key (idcliente) references cliente(idcliente)
);

create table productos(
    idproducto number primary key,
    descripcion varchar(80),
    pvp number,
    stockactual number default 0
);

create table ventas(
    idventas number primary key,
    idcliente number,
    fechaventa date default sysdate,
    constraint fk_idcliente foreign key (idcliente) references cliente(idcliente)
);

create table lineasventas(
    idventas number,
    numerolinea number,
    idproducto number,
    cantidad number,
    constraint pk_idventas_linea primary key (idventas,numerolinea),
    constraint fk_idventas foreign key (idventas) references ventas(idventas),
    constraint fk_idproducto foreign key (idproducto) references productos(idproducto)
);
```

Se introducen los datos.

2. Diseña el modelo objeto-relacional

```
create or replace type tip_telefonos as table of varchar2(15);
/

create or replace type tip_direccion as object(
    calle varchar2(50),
    poblacion varchar2(50),
    codpos varchar2(20),
    provincia varchar2(40)
);
/

create or replace type tip_cliente as object(
    idcliente number,
    nombre varchar2(50),
    direc tip_direccion,
    nif varchar2(9),
    telef tip_telefonos
);
/

create or replace type tip_producto as object(
    idproducto number,
    descripcion varchar2(80),
    pvp number,
    stockactual number
);
/

create or replace type tip_linea_venta as object(
    numerolinea number,
    idproducto ref tip_producto,
    cantidad number
);
/

create or replace type tip_lineas_venta as table of tip_linea_venta;
/

create or replace type tip_venta as object(
    idventa number,
    idcliente ref tip_cliente,
    fechaventa date,
    lineas tip_lineas_venta,
    member function total_venta return number
);
/
```

3. Crea el cuerpo para la función total_venta

```
create or replace type body tip_venta as
member function total_venta return number is
    total number:=0;
    linea tip_linea_venta;
    product tip_producto;
begin
    for i in 1..lineas.count loop
        linea:=lineas(i);
        select deref(linea.idproducto) into product from dual;
        total:= total+ linea.cantidad * product.pvp;
    end loop;
    return total;
end;
end;
```

4. Persiste los objetos

```
-- Crea la tabla tabla_clientes y modifica idcliente a primary key
drop table tabla_clientes;
create table tabla_clientes of tip_cliente nested table telef store as t_telefono;
/
alter table tabla_clientes add primary key (idcliente);

-- Crea la tabla tabla_productos y modifica idproducto a primary key
create table tabla_productos of tip_producto;
/
alter table tabla_productos add primary key (idproducto);

-- Crea la tabla tabla_ventas y modifica idventa a primary key
create table tabla_ventas of tip_venta nested table lineas store as tlineas;
/
alter table tabla_ventas add primary key (idventa);
```

5. Inserta los datos

```
insert into tabla_clientes values(1,'Luis Garcia',
                                tip_direccion('calle Las Flores,23','Guadalajara','19003','Guadalajara'),
                                '34343434L',
                                tip_telefonos('949876655','949876655')
                                );

insert into tabla_clientes values(2,'ana Serrano',
                                tip_direccion('calle Galiana,6','Guadalajara','19004','Guadalajara'),
                                '76767667F',
                                tip_telefonos('94980009')
                                );

insert into tabla_productos values(1, 'caja de cristal de murano',100,5);
insert into tabla_productos values(2, 'bicicleta city',120,15);
insert into tabla_productos values(3, '100 lapices de colores',20,5);
insert into tabla_productos values(4, 'ipad',600,5);
insert into tabla_productos values(5, 'ordenador portatil',400,10);
```

6. Consultas

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

```
select * from the(select tv.lineas from tabla_ventas tv where tv.idventa=2);
```

NUMEROLINEA

IDPRODUCTO

CANTIDAD

1
0000220208ED2140F1136A948EE050520D34281BD3ED2140F11361948EE050520D34281BD3
1
2
0000220208ED2140F1136B948EE050520D34281BD3ED2140F11361948EE050520D34281BD3
2

6.1.2 Haz la consulta de otra forma sin sacar el nombre del objeto, usando TABLE.

```
select lin.* from tabla_ventas tv, table(tv.lineas) lin where tv.idventa=2;
```

NUMEROLINEA

IDPRODUCTO

CANTIDAD

1
0000220208ED2140F1136A948EE050520D34281BD3ED2140F11361948EE050520D34281BD3
1
2
0000220208ED2140F1136B948EE050520D34281BD3ED2140F11361948EE050520D34281BD3
2

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

```
select deref(lin.idproducto) from tabla_ventas tv, table(tv.lineas) lin where tv.idventa=2;
```

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

TIP_PRODUCTO(3, '100 lapices de colores', 20, 5)
TIP_PRODUCTO(4, 'ipad', 600, 5)

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

```
select deref(lin.idproducto) from tabla_ventas tv, table(tv.lineas) lin;
```

```
DEREF(LIN.IDPRODUCTO)(IDPRODUCTO, DESCRIPCION, PVP, STOCKACTUAL)
-----
TIP_PRODUCTO(1, 'caja de cristal de murano', 100, 5)
TIP_PRODUCTO(2, 'bicicleta city', 120, 15)
TIP_PRODUCTO(3, '100 lapices de colores', 20, 5)
TIP_PRODUCTO(4, 'ipad', 600, 5)
```

6.4 Consulta el nombre del cliente idcliente 2.

```
select nombre from tabla_clientes where idcliente=2;
```

```
NOMBRE
-----
ana Serrano
```

6.5 Modifica el nombre del cliente 2 por Rosa Serrano.

```
update tabla_clientes set nombre='Rosa Serrano' where idcliente=2;
```

```
SQL> update tabla_clientes set nombre='Rosa Serrano' where idcliente=2;
1 row updated.

SQL> select nombre from tabla_clientes where idcliente=2;

NOMBRE
-----
Rosa Serrano
```

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

```
select tc.direc.calle from tabla_clientes tc where idcliente=2;
update tabla_clientes tc set tc.direc.calle='calle Estopa,34' where idcliente =2;
```

```
SQL> select tc.direc.calle from tabla_clientes tc where idcliente=2;

DIREC.CALLE
-----
calle Galiana,6

SQL> update tabla_clientes tc set tc.direc.calle='calle Estopa,34' where idcliente =2;
1 row updated.

SQL> select tc.direc.calle from tabla_clientes tc where idcliente=2;

DIREC.CALLE
-----
calle Estopa,34
```

6.7 Consulta todos los datos del cliente 1 y añade un nuevo teléfono a su lista de teléfonos. Haz la consulta de otra forma usando value.

```
select * from tabla_clientes cli where idcliente=1;

select value(cli) from tabla_clientes cli where idcliente=1;

update tabla_clientes set telef=tip_telefonos('949876655','949876655','666888555')where idcliente=1;
```

```
IDCLIENTE NOMBRE
-----
DIREC(CALLE, POBLACION, CODPOS, PROVINCIA)
-----
NIF
-----
TELEF
-----
1 Luis Garcia
TIP_DIRECCION('calle Las Flores,23', 'Guadalajara', '19003', 'Guadalajara')
34343434L
TIP_TELEFONOS('949876655', '949876655')
```

SQL>

1 row updated.

SQL> select value(cli) from tabla_clientes cli where idcliente=1;

```
VALUE(CLI)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOS, PROVINCIA), NI
-----
TIP_CLIENTE(1, 'Luis Garcia', TIP_DIRECCION('calle Las Flores,23', 'Guadalaj
ara', '19003', 'Guadalajara'), '34343434L', TIP_TELEFONOS('949876655', '9498
76655', '666888555'))
```

6.8.1 Visualiza el nombre del cliente que ha realizado la venta.

```
select tv.idcliente.nombre from tabla_ventas tv;
```

```
IDCLIENTE.NOMBRE
-----
Luis Garcia
Rosa Serrano
```

6.8.2 Haz la misma consulta usando Deref

```
select deref(idcliente).nombre from tabla_ventas;
```

```
DEREF(IDCLIENTE).NOMBRE
-----
Luis Garcia
Rosa Serrano
```

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

```
select deref(idcliente) from tabla_ventas where idventa=2;
```

```
DEREF(IDCLIENTE)(IDCLIENTE, NOMBRE, DIREC(CALLE, POBLACION, CODPOS, PROVINCI
-----
TIP_CLIENTE(2, 'Rosa Serrano', TIP_DIRECCION('calle Estopa,34', 'Guadalajara
', '19004', 'Guadalajara'), '76767667F', TIP_TELEFONOS('94980009'))
```

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

```
select idventa, tv.total_venta() from tabla_ventas tv where tv.idcliente.idcliente =1;
```

```
select idventa, tv.total_venta() from tabla_ventas tv where deref(idcliente).idcliente =1;
```

```

IDVENTA TV.TOTAL_VENTA()
-----
1          340
```

6.11 Visualiza las ventas de todos los clientes.

```
select idventa,deref(idcliente).idcliente cliente, lin.idproducto.descripcion producto, lin.cantidad
from tabla_ventas tv, table(tv.lineas) lin;
```

IDVENTA	CLIENTE	PRODUCTO	CANTIDAD
1		1 caja de cristal de murano	1
1		1 bicicleta city	2
2		2 100 lapices de colores	1
2		2 ipad	2

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

```
select lin.numerolinea, lin.cantidad, deref(lin.idproducto) from tabla_ventas tv, table(tv.lineas) lin where tv.idventa=2;

create or replace procedure mostrar(vp_idventa tabla_ventas.idventa%type) as
linesa number;
cantidad number;
importe number;
total_v number;
product tip_producto:=tip_producto(NULL,NULL,NULL,NULL);
cli tip_cliente;
linea_venta tip_linea_venta;
direccion tip_direccion;
fecha date;
cursor c1 is
select deref(lin.idproducto),lin.cantidad,lin.numerolinea from tabla_ventas tv, table(tv.lineas) lin where tv.idventa=vp_idventa;
begin
select deref(idcliente),fechaventa,tv.total_venta() into cli,fecha,total_v from tabla_ventas tv where idventa=vp_idventa;
select direc into direccion from tabla_clientes where cli.idcliente=idcliente;
DBMS_OUTPUT.PUT_LINE('Número de venta: '||vp_idventa||'Fecha de Venta: '||fecha);
DBMS_OUTPUT.PUT_LINE('Cliente: '||cli.nombre);
DBMS_OUTPUT.PUT_LINE('Dirección: '||direccion.calle);
DBMS_OUTPUT.PUT_LINE('*****');
open c1;
fetch c1 into product,cantidad,lineas;
loop
DBMS_OUTPUT.PUT_LINE(lineas||' '||product.descripcion||' '||product.pvp||' '||cantidad||' '||cantidad*product.pvp);
fetch c1 into product,cantidad,lineas;
exit when c1%NOTFOUND;
end loop;
close c1;

DBMS_OUTPUT.PUT_LINE('Total Venta: '||total_v);
end;
/
```

Activ
Ve a C

```
SQL> set serveroutput on;
SQL> execute mostrar(1);
Número de venta: 1Fecha de Venta: 10/11/22
Cliente: Luis Garcia
Dirección: calle Las Flores,23
*****
1 caja de cristal de murano 100 1 100
2 bicicleta city 120 2 240
Total Venta:340

PL/SQL procedure successfully completed.

SQL> execute mostrar(2);
Número de venta: 2Fecha de Venta: 10/11/22
Cliente: Rosa Serrano
Dirección: calle Estopa,34
*****
1 100 lapices de colores 20 1 20
2 ipad 600 2 1200
Total Venta:1220

PL/SQL procedure successfully completed.
```