**PRÁCTICA 11.**

**Modelo relacional y objeto relacional, para la 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,cdopostal,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**

1. **Diseña el modelo objeto-relacional**

Tipo **tip\_telefonos** será un **varray** de 3 elementos varchar(15).

Tipo **tip\_direccion**

calle varchar(50)

poblacion 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

Tipo **tip\_producto**

idproducto number

descripción varchar(80)

pvp number

stockactual number

Tipo **tip\_linea\_venta**

numerolinea number

idproducto REF tip\_producto

cantidad number

Creamos un tipo tabla anidada para contener la 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 **Lineas** 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\_lineas\_venta**,

Member function total\_venta retorna number

1. **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)

1. **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\_ventascon una clave primaria idventa

1. **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’))

(2,'ana Serrano, tip\_direccion('calle Galiana,6,’Guadalajara’,’19004’,’Guadalajara’),’76767667F’,tip\_telefonos(‘94980009’))

(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)

En la tabla tabla\_ventas se define una tabla anidada para el atributo LINEAS de tipo tip\_venta, contendrá las líneas de venta.

Se puede hacer de dos formas, por pasos o de una sola vez:

En dos pasos:

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

INSERT INTO TABLA\_VENTAS SELECT 1, REF(C),SYSDATE, TIP\_LINEAS\_VENTA() FROM TABLA\_CLIENTES C WHERE C.IDCLIENTE=1;

**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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

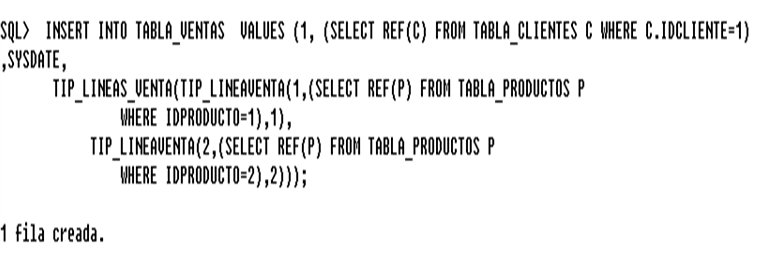
**Linea1**

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**

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);

En un solo paso:



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

En dos pasos:

INSERT INTO TABLA\_VENTAS SELECT 2, REF(C),SYSDATE, TIP\_LINEAS\_VENTA() FROM TABLA\_CLIENTES C WHERE C.IDCLIENTE=1;

**Líneas para la venta 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Linea1

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);

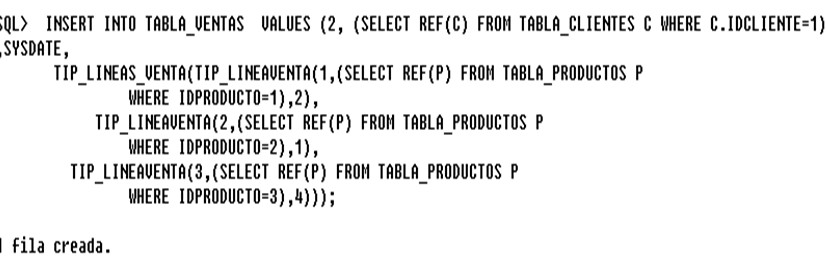
Línea 2

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);

Línea 3

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);

En un solo paso:



1. **Consultas**

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

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

* 1. **Visualizar todas las líneas de venta para la venta id 2, obteniendo los productos en vez de su oid.**
  2. **Visualizar todas las líneas de venta de todas las ventas.**
  3. **Consulta el nombre del cliente idcliente 2**
  4. **Modifica el nombre del cliente 2 por Rosa Serrano**
  5. **Consulta la dirección del cliente 2 y modifica la calle por calle Estopa,34**
  6. **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**

* 1. **Visualiza el nombre del cliente que ha realizado la venta 2.**

**Haz la misma consulta usando DEREF**

* 1. **Visualiza todos los datos del cliente anterior, que ha realizado la venta 2**
  2. **Visualizar el numero de venta y el total de ventas hechas por el cliente 1**
  3. **Visualiza las ventas de todos los clientes**
  4. **Crea un procedimiento que reciba como parámetro un id de venta y visualice los datos de la venta cuyo identificador recibe.**

**La salida será como la siguiente:**

