Ejercicios complementarios - Serie práctica 4
Diseño de Bases de Datos Orientado a Objetos – Relacionales (BDOO-R)

Resoluciones "modelos" de los ejercicios propuestos

1.- Se crean tablas normalizadas y con claves ajenas para representar las relaciones

```
CREATE TABLE clientes (
        clinum NUMBER,
        clinomb VARCHAR2(200),
        calle VARCHAR2(200),
        ciudad VARCHAR2(200),
        prov CHAR(2),
        codpos VARCHAR2(20),
        tel1 VARCHAR2(20),
        tel2 VARCHAR2(20),
        tel3 VARCHAR2(20),
        PRIMARY KEY (clinum)
) ;

CREATE TABLE ordenes (
        ordnum NUMBER,
        clinum NUMBER REFERENCES clientes,
        fechpedido DATE,
        fechaentrega DATE,
        callent VARCHAR2(200),
        ciuent VARCHAR2(200),
        provent CHAR(2),
        codpent VARCHAR2(20),
        PRIMARY KEY (ordnum)
) ;

CREATE TABLE items (
        numitem NUMBER PRIMARY KEY,
        precio NUMBER,
        tasas NUMBER
) ;

CREATE TABLE lineas (
        linum NUMBER,
        ordnum NUMBER REFERENCES ordenes,
        numitem NUMBER REFERENCES items,
        cantidad NUMBER,
        descuento NUMBER,
        PRIMARY KEY (ordnum, linum)
) ;
```

2.- Modelo Lógico OO

```
define type Lista_Tel_T: type list(string);

define type Direccion_T: type tuple [ calle:string,
        ciudad:string,
        prov:string,
        codpos:string];

define class Clientes_T: type tuple [ clinum: integer,
        clinomb:string,
        direccion:Direccion_T,
        lista_tel: Lista_Tel_T];
```

```
define class Item_T: type tuple [ itemnum:integer,
        precio:real,
        tasas:real];

define type Linea_T: type tuple [linum:integer,
        item:Item_T,
        cantidad:integer,
        descuento:real];

define type Lineas_Pedido_T: type set(Linea_T);

define class Ordenes_T: type tuple [ ordnum:integer,
        cliente:Clientes_T,
        fechpedido:date,
        fechentrega:date,
        pedido:Lineas_Pedido_T
        direcentrega:Direccion_T];
```

3.- Definición de tipos de objetos del ítem 2 en Oracle 8

```
CREATE TYPE lista_tel_t AS VARRAY(10) OF VARCHAR2(20) ;

CREATE TYPE direccion_t AS OBJECT (
        calle VARCHAR2(200),
        ciudad VARCHAR2(200),
        prov CHAR(2),
        codpos VARCHAR2(20)
) ;

CREATE TYPE clientes_t AS OBJECT (
        clinum NUMBER,
        clinomb VARCHAR2(200),
        direccion direccion_t,
        lista_tel lista_tel_t,
) ;

CREATE TYPE item_t AS OBJECT (
        itemnum NUMBER,
        precio NUMBER,
        tasas NUMBER
) ;

CREATE TYPE linea_t AS OBJECT (
        linum NUMBER,
        item REF item_t,
        cantidad NUMBER,
        descuento NUMBER
) ;
CREATE TYPE lineas_pedido_t AS TABLE OF linea_t ;

CREATE TYPE ordenes_t AS OBJECT (
        ordnum NUMBER,
        cliente REF clientes_t,
        fechpedido DATE,
        fechentrega DATE,
        pedido lineas_pedido_t,
        direcentrega direccion_t,
) ;
```

4.- Creación de las tablas para almacenar la información

```
CREATE TABLE clientes_tab OF clientes_t
(clinum PRIMARY KEY);

CREATE TABLE items_tab OF item_t
(itemnum PRIMARY KEY) ;

CREATE TABLE ordenes_tab OF ordenes_t (
        PRIMARY KEY (ordnum),
        SCOPE FOR (cliente) IS clientes_tab
        )
        NESTED TABLE pedido STORE AS pedidos_tab ;
        ALTER TABLE pedidos_tab
        ADD (SCOPE FOR (item) IS items_tab) ;
```

5.- Inserción de datos

5.1.-

```
INSERT INTO items_tab VALUES(1004, 6750.00, 2);
INSERT INTO items_tab VALUES(1011, 4500.23, 2);
INSERT INTO items_tab VALUES(1534, 2234.00, 2);
INSERT INTO items_tab VALUES(1535, 3456.23, 2);
INSERT INTO items_tab VALUES(2004, 33750.00, 3);
INSERT INTO items_tab VALUES(3011, 43500.23, 4);
INSERT INTO items_tab VALUES(4534, 5034.00, 6);
INSERT INTO items_tab VALUES(5535, 34456.23, 5);
```

5.2.-

```
INSERT INTO clientes_tab
        VALUES (
                1, 'Lola Caro',
                direccion_t('12 Calle Lisboa', 'Nules', 'CS', '12678'),
                lista_tel_t('415-555-1212')
        ) ;

INSERT INTO clientes_tab
        VALUES (
                2, 'Jorge Luz',
                direccion_t('323 Calle Sol', 'Valencia', 'V', '08820'),
                lista_tel_t('609-555-1212','201-555-1212')
        ) ;

INSERT INTO clientes_tab
        VALUES (
                3, 'Jose Perez',
                direccion_t('12 Calle Colon', 'Castellon', 'ES', '12001'),
                lista_tel_t('964-555-1212', '609-543-1212',
                '201-775-1212','964-445-1212')
        ) ;
INSERT INTO clientes_tab
        VALUES (
                4, 'Ana Gil',
                direccion_t('5 Calle Sueca', 'Burriana', 'ES', '12345'),
                lista_tel_t()
        ) ;
```

6.- Borrado

```
DELETE FROM ordenes_tab;
DROP TABLE ordenes_tab;
```

```
DELETE FROM clientes_tab;
DROP TABLE clientes_tab;
DELETE FROM items_tab;
DROP TABLE items_tab;
DROP TYPE ordenes_t;
DROP TYPE lineas_pedido_t;
DROP TYPE linea_t;
DROP TYPE item_t;
DROP TYPE clientes_t;
DROP TYPE lista_tel_t;
DROP TYPE direccion_t;
```

7. – Creación del método para la suma

```
CREATE TYPE ordenes_t AS OBJECT (
        ordnum NUMBER,
        cliente REF clientes_t,
        fechpedido DATE,
        fechentrega DATE,
        pedido lineas_pedido_t,
        direcentrega direccion_t,

        MEMBER FUNCTION
                coste_total RETURN NUMBER,
                PRAGMA RESTRICT_REFERENCES(coste_total, WNDS, WNPS) );

CREATE TYPE BODY ordenes_t AS
        MEMBER FUNCTION coste_total RETURN NUMBER IS
        i INTEGER;
        item item_t;
        linea linea_t;
        total NUMBER:=0;
BEGIN
        FOR i IN 1..SELF.pedido.COUNT LOOP
                linea:=SELF.pedido(i);
                SELECT DEREF(linea.item) INTO item FROM DUAL;
                total:=total + linea.cantidad * item.precio;
        END LOOP;
        RETURN total;
END;
END;
```