

Instituto Tecnológico de Costa Rica

Escuela de Computación



Bases de Datos II

Grupo 20

Tarea 9

Profesor (a):

Alberto Shum Chan

Estudiante (s):

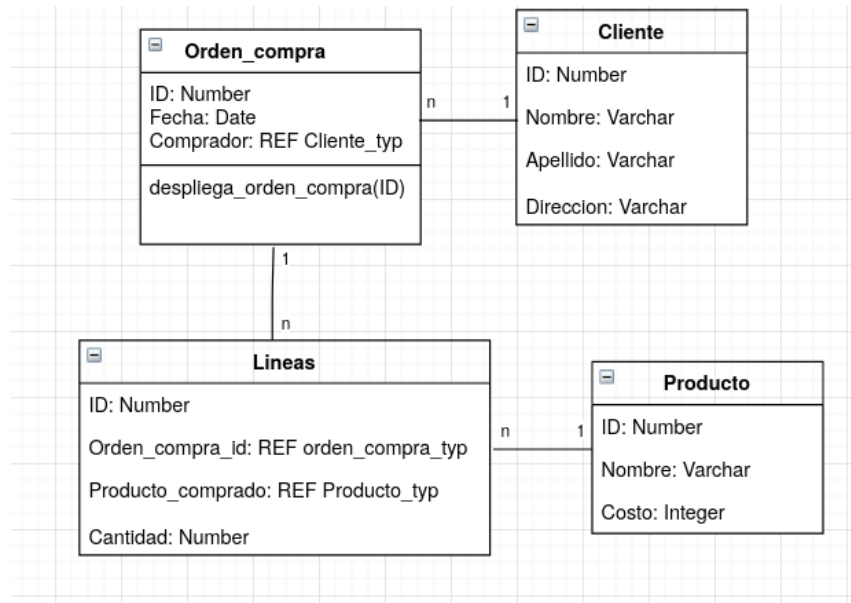
José Adrián Amador Ávila - 2016101574

Pablo Jesús Mora Barrantes - 2019205110

Jose Andrés Vargas Serrano - 2019211290

Alajuela, I Semestre 2023

A partir del siguiente diagrama se crean los tipos solicitados:



1) Tipos de datos

Tipo de dato Cliente_typ con su respectiva secuencia

```
-- Tipo de dato Cliente_typ y su secuencia
-- DROP TYPE Cliente_typ;
CREATE TYPE Cliente_typ AS OBJECT (
    id NUMBER,
    nombre VARCHAR2(20),
    apellido VARCHAR2(20),
    direccion VARCHAR2(50)
);

-- Secuencia para la tabla cliente_
-- DROP SEQUENCE s_cliente;
CREATE SEQUENCE s_cliente
    START WITH 1
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 10000000
    NOCACHE
    NOCYCLE;
```

Tipo de dato Producto_typ con su respectiva secuencia

```
-- Tipo de dato Producto_typ y su secuencia
-- DROP TYPE Producto_typ;
CREATE TYPE Producto_typ AS OBJECT (
    id NUMBER,
    nombre VARCHAR2(20),
    costo Integer
);

-- Secuencia para la tabla producto_
-- DROP SEQUENCE s_producto;
CREATE SEQUENCE s_producto
    START WITH 1
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 10000000
    NOCACHE
    NOCYCLE;
```

Tipo de dato Orden_typ con su respectiva secuencia

```
-- Tipo de dato Orden_compra_typ y su secuencia
-- DROP TYPE Orden_compra_typ;
CREATE OR REPLACE TYPE Orden_compra_typ AS OBJECT (
    id NUMBER,
    fecha DATE,
    comprador REF Cliente_typ,
    MEMBER FUNCTION despliega_orden_compra(idno NUMBER) RETURN VARCHAR2
) FINAL;
```

```
-- Secuencia para la tabla Orden_compra
-- DROP SEQUENCE s_orden_compra;
CREATE SEQUENCE s_orden_compra
    START WITH 1
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 10000000
    NOCACHE
    NOCYCLE;
```

Tipo de dato Lineas_typ con su respectiva secuencia

```
-- Tipo de dato Lineas_typ y su secuencia
-- DROP TYPE Lineas_typ;
CREATE TYPE Lineas_typ AS OBJECT (
    id NUMBER,
    orden_compra_id REF Orden_compra_typ,
    producto_comprado REF Producto_typ,
    cantidad NUMBER
);

-- Secuencia para la tabla Lineas
-- DROP SEQUENCE s_lineas;
CREATE SEQUENCE s_lineas
    START WITH 1
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 10000000
    NOCACHE
    NOCYCLE;

-- Se crea la tabla Lineas a partir del tipo Lineas_typ
```

2) Creación de las tablas a partir de los datos

Tabla cliente_ de tipo Cliente_typ

```
-- Se crea la tabla cliente_ a partir del tipo Cliente_typ
-- DROP TABLE cliente_;
CREATE TABLE cliente_ OF Cliente_typ;
```

Tabla producto_ de tipo Producto_typ

```
-- Se crea la tabla producto_ a partir del tipo Producto_typ
-- DROP TABLE producto_;
CREATE TABLE producto_ OF Producto_typ;
```

Tabla Orden_compra de tipo Orden_compra_typ

```
-- Se crea la tabla Orden_compra a partir del tipo Orden_compra_typ
-- DROP TABLE Orden_compra;
CREATE TABLE Orden_compra OF Orden_compra_typ;
-- Se agrega el scope para la referencia de la tabla Orden_compra
ALTER TABLE Orden_compra ADD SCOPE FOR (comprador) IS cliente_;
```

Tabla Lineas de tipo Lineas_typ

```
-- Se crea la table Lineas a partir del tipo Lineas_typ
-- DROP TABLE Lineas;
CREATE TABLE Lineas OF Lineas_typ;
-- Se agrega el scope para las referencias de la table Lineas
ALTER TABLE Lineas ADD SCOPE FOR (orden_compra_id) IS Orden_compra;
ALTER TABLE Lineas ADD SCOPE FOR (producto_comprado) IS producto_;
```

3) Inserción de datos en las tablas

Inserción de 3 productos y dos clientes

```
-- Agregamos datos a las tablas
-- 3 productos
INSERT INTO producto(id, nombre, costo) VALUES(s_producto.NEXTVAL, 'Jabón', 550);
INSERT INTO producto(id, nombre, costo) VALUES(s_producto.NEXTVAL, 'Arroz', 1350);
INSERT INTO producto(id, nombre, costo) VALUES(s_producto.NEXTVAL, 'Leche', 655);

-- TRUNCATE TABLE producto_;

-- 2 clientes
INSERT INTO cliente(id, nombre, apellido, direccion) VALUES(s_cliente.NEXTVAL, 'José', 'Madrigal', 'Calle Vargas, Alajuela, Costa Rica');
INSERT INTO cliente(id, nombre, apellido, direccion) VALUES(s_cliente.NEXTVAL, 'Angie', 'Ortiz', 'Barrio Amón, San José, Costa Rica');

-- TRUNCATE TABLE cliente_;
```

Inserción de 2 órdenes de compra

```
-- 2 ordenes de compra
INSERT INTO Orden_compra(id, fecha, comprador) VALUES(s_orden_compra.NEXTVAL, TO_DATE('02-MAY-2023', 'DD-MON-YYYY'), NULL);
INSERT INTO Orden_compra(id, fecha, comprador) VALUES(s_orden_compra.NEXTVAL, TO_DATE('01-MAY-2023', 'DD-MON-YYYY'), NULL);

SELECT * FROM Orden_compra;

UPDATE Orden_compra
SET comprador = (SELECT REF(a) FROM cliente_ a WHERE id = 1)
WHERE id = 1;

UPDATE Orden_compra
SET comprador = (SELECT REF(a) FROM cliente_ a WHERE id = 2)
WHERE id = 2;

-- TRUNCATE TABLE Orden_compra;
```

Inserción de 2 líneas para la primer orden de compra

```
-- 2 líneas para la primer orden de compra
INSERT INTO Lineas(id, orden_compra_id, producto_comprado, cantidad)
VALUES(s_lineas.NEXTVAL, null, null, 3);
INSERT INTO Lineas(id, orden_compra_id, producto_comprado, cantidad)
VALUES(s_lineas.NEXTVAL, null, null, 4);

UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a WHERE id = 1),
    producto_comprado = (SELECT REF(a) FROM producto_ a WHERE id = 1)
WHERE id = 1;

UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a WHERE id = 1),
    producto_comprado = (SELECT REF(a) FROM producto_ a WHERE id = 2)
WHERE id = 2;
```

Inserción de 3 líneas para la segunda orden de compra

```
-- 3 líneas para la segunda orden de compra
INSERT INTO Lineas(id, orden_compra_id, producto_comprado, cantidad)
VALUES(s_lineas.NEXTVAL, null, null, 2);
INSERT INTO Lineas(id, orden_compra_id, producto_comprado, cantidad)
VALUES(s_lineas.NEXTVAL, null, null, 2);
INSERT INTO Lineas(id, orden_compra_id, producto_comprado, cantidad)
VALUES(s_lineas.NEXTVAL, null, null, 5);

SELECT * FROM Lineas;

UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a WHERE id = 2),
    producto_comprado = (SELECT REF(a) FROM producto_ a WHERE id = 1)
WHERE id = 3;

UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a WHERE id = 2),
    producto_comprado = (SELECT REF(a) FROM producto_ a WHERE id = 2)
WHERE id = 4;

UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a WHERE id = 2),
    producto_comprado = (SELECT REF(a) FROM producto_ a WHERE id = 3)
WHERE id = 5;
```

4) Método despliega_orden_compra que recibe el ID de la orden de compra

```
-- Body de la función del tipo Orden_compra_typ
-- DROP TYPE BODY Orden_compra_typ;
CREATE OR REPLACE TYPE BODY Orden_compra_typ AS
MEMBER FUNCTION despliega_orden_compra (idno NUMBER) RETURN VARCHAR2 IS
    v_nombre_comprador VARCHAR2(20);
    v_apellido_comprador VARCHAR2(20);
    v_fecha_orden_compra DATE;
    -- declaración de variables locales
    var_productos VARCHAR2(500) := '';
    -- Variable que va a contener toda la información de la orden de compra y sus líneas
    var_detalle VARCHAR2(500);
    -- Cursor para obtener la orden de compra especificada por el id de la función
    CURSOR c_orden_compra IS
        SELECT fecha, Deref(comprador).nombre AS "nombre_comprador", Deref(comprador).apellido AS "apellido_comprador"
        FROM Orden_compra
        WHERE Orden_compra.id = idno;
    -- Cursor para obtener las líneas de la orden de compra especificada por el id de la función
    CURSOR c_lineas IS
        SELECT Deref(producto_comprado).nombre AS "nombre_producto", Deref(producto_comprado).costo AS "precio_producto",
        cantidad
        FROM Lineas
        WHERE Deref(orden_compra_id).id = idno;
BEGIN
    -- Abrimos el cursor de la orden de compra
    OPEN c_orden_compra;
    -- Extraemos el resultado en la variable r_orden_compra para poder acceder a sus datos
    FETCH c_orden_compra INTO v_fecha_orden_compra, v_nombre_comprador, v_apellido_comprador;
    -- Abrimos el cursor de las líneas asociadas a la orden de compra
    -- Extraemos todas las líneas y las guardamos como strings
    FOR r_lineas IN c_lineas
    LOOP
        var_productos := var_productos || 'Producto: ' || r_lineas.nombre_producto || ', ' ||
        || 'Precio: ' || r_lineas.precio_producto || ', ' ||
        || 'Cantidad: ' || r_lineas.cantidad || chr(13) || chr(10);
    END LOOP;
    -- Concatenamos toda la información para retornarla
    var_detalle := '-----' || chr(13) || chr(10);
    var_detalle := var_detalle || 'Fecha: ' || TO_CHAR(v_fecha_orden_compra) || chr(13) || chr(10) || 'Nombre cliente: ' || v_nombre_comprador
    || ' ' || v_apellido_comprador || chr(13) || chr(10) || var_productos;
    var_detalle := var_detalle || '-----' || chr(13) || chr(10);
    -- Cerramos el cursor de la orden de compra
    CLOSE c_orden_compra;
    RETURN var_detalle;
END;
```

Demostración de funcionamiento

The screenshot shows a SQL IDE with a query window and a modal dialog box titled "Ver Valor".

Query Window:

```
WHERE id = 4;
UPDATE Lineas
SET orden_compra_id = (SELECT REF(a) FROM Orden_compra a
WHERE id = 5;
-- TRUNCATE TABLE Lineas;
-- Pruebas de la función despliega_orden_compra()
SELECT a.despliega_orden_compra(1) FROM Orden_compra a;
-- Pruebas para desplegar los datos de las tablas
SELECT id AS LINEA, Deref(orden_compra_id).id AS "orden_compra_id",
Deref(producto_comprado).nombre, Deref(p
FROM Lineas;
```

Ver Valor Dialog Box:

Terminador de Línea: Valor por Defecto de Plataforma

Valor:

```
Fecha: 02/05/23
Nombre cliente: José Madrigal
Producto: Jabón, Precio: 550, Cantidad: 3
Producto: Arroz, Precio: 1350, Cantidad: 4
```

Buttons: Ayuda, Aceptar, Cancelar

Salida de Script x | Resultado de la Consulta 1 x | Resultado de la Consulta 2 x

Todas las Filas Recuperadas: 2 en 0.313 segundos

1

23Nombre cliente: José MadrigalProducto: Jabón, Precio: 550, Cantidad: 3Producto: Arroz, Precio: 1350, Cantidad: 4-----

