

Actividad de aprendizaje:

Instrucciones:

SQL Llaves

1. Crear las tablas y llaves indicadas.

Considere el siguiente :

Crear las Tablas y Llaves Indicadas

```
-- Tabla 1: Clientes
CREATE TABLE Clientes (
    id_cliente INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    edad INT CHECK (edad BETWEEN 18 AND 85) NOT NULL
);

-- Tabla 2: Cuentas
-- La llave foránea 'id_cliente' asegura que cada cuenta
pertenezca a un cliente existente.
CREATE TABLE Cuentas (
    id_cuenta INT PRIMARY KEY,
    id_cliente INT NOT NULL,
    saldo NUMERIC(10, 2) CHECK (saldo BETWEEN -5000.00 AND
100000.00) NOT NULL,
    CONSTRAINT fk_cliente
        FOREIGN KEY (id_cliente)
        REFERENCES Clientes(id_cliente)
        ON DELETE CASCADE -- Si se borra un cliente, sus cuentas
se borran (Integridad Referencial)
        ON UPDATE CASCADE -- Si se actualiza el id_cliente, se
actualiza en Cuentas
);

-- Crear SECUENCIAS (para autogenerar IDs si la base de datos lo
requiere y no usa AUTOINCREMENT/IDENTITY)
```

```
-- Nota: La sintaxis de secuencias puede variar (PostgreSQL,  
Oracle, etc.). Aquí se usa un ejemplo genérico:  
CREATE SEQUENCE seq_cliente_id START WITH 1 INCREMENT BY 1;  
CREATE SEQUENCE seq_cuenta_id START WITH 1 INCREMENT BY 1;
```

2. Crear 5 clientes y 15 cuentas asociadas. Use saldos entre -5.000 y 100.000, ambos incluidos. Use edades entre 18 y 85 ambos incluidos -

Inserción de 5 Clientes (Edades entre 18 y 85)

```
-- Insertar Clientes (Usando sec_cliente_id.nextval si la base de  
datos lo permite, sino, se usa un número)  
INSERT INTO Clientes (id_cliente, nombre, edad) VALUES (1, 'Ana  
García', 78);  
INSERT INTO Clientes (id_cliente, nombre, edad) VALUES (2, 'Luis  
Pérez', 25);  
INSERT INTO Clientes (id_cliente, nombre, edad) VALUES (3, 'Maria  
Soto', 40);  
INSERT INTO Clientes (id_cliente, nombre, edad) VALUES (4, 'Carlos  
Ruiz', 80); -- Cliente con más edad  
INSERT INTO Clientes (id_cliente, nombre, edad) VALUES (5, 'Elena  
Torres', 32);
```

Inserción de 15 Cuentas (Saldos entre -5.000 y 100.000)

```
-- Clientes(id_cliente): 1=Ana(78), 2=Luis(25), 3=Maria(40),  
4=Carlos(80), 5=Elena(32)  
  
-- Cuentas para Cliente 1 (Ana): 3 cuentas  
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (101, 1,  
50000.00);  
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (102, 1,  
-1200.50); -- Saldo Negativo
```

```

INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (103, 1,
100.00);

-- Cuentas para Cliente 2 (Luis): 2 cuentas
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (201, 2,
850.75);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (202, 2,
-500.00); -- Saldo Negativo

-- Cuentas para Cliente 3 (Maria): 4 cuentas
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (301, 3,
15000.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (302, 3,
200.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (303, 3,
-4999.99); -- Saldo Negativo
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (304, 3,
75000.00);

-- Cuentas para Cliente 4 (Carlos - Cliente con más edad): 3
cuentas
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (401, 4,
1000.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (402, 4,
2000.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (403, 4,
3000.00);

-- Cuentas para Cliente 5 (Elena): 3 cuentas
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (501, 5,
50.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (502, 5,
120.00);
INSERT INTO Cuentas (id_cuenta, id_cliente, saldo) VALUES (503, 5,
900.00);

```

Ejemplo de Actualización (DML: UPDATE) y Borrado (DML: DELETE)

```

-- Actualizando la información (Ejemplo DML: UPDATE)
-- Aumentar el saldo de la cuenta 402 en 500.00
UPDATE Cuentas
SET saldo = saldo + 500.00
WHERE id_cuenta = 402;

```

```
-- Borrando información (Ejemplo DML: DELETE)
-- Borrar una cuenta (si es necesario)
DELETE FROM Cuentas
WHERE id_cuenta = 503; -- La cuenta de Elena con 900.00 se borra.
```

Crear las siguientes consultas:

3. Listar el saldo de cada cuenta del cliente con más años de edad.
4. Listar el promedio de edad de los clientes con saldo negativo.
5. Listar el nombre y cantidad de cuentas de quienes tienen más de una.
6. Listar el saldo combinado (suma) de cada cliente con más de una cuenta.
7. Listar todos los clientes y su saldo combinado de todos aquellos clientes que tengan al menos una cuenta con saldo negativo.