

**Tema 6. Lenguajes de Bases de Datos****EJERCICIOS. Modificación de datos en SQL****SOLUCIONES****Objetivo**

- Construir **expresiones de modificación de datos** almacenados en la base de datos, empleando las sentencias lenguaje estándar de bases de datos relacionales **SQL**.

**Esquema de base de datos 'BANCO'****SUCURSAL** (codsuc, nomsuc, activo, ciudadsuc)**CLIENTE** (codcli, nomcli, calle, ciudadcli)**BANQUERO\_PERSONAL** (cliente, nombanquero)

- Clave Ajena: cliente REFERENCIA A CLIENTE (codcli)

**CUENTA** (numcuenta, sucursal, cliente, saldo)

- Clave Ajena: sucursal REFERENCIA A SUCURSAL (codsuc)
- Clave Ajena: cliente REFERENCIA A CLIENTE (codcli)

**PRESTAMO** (numprestamo, sucursal, cliente, cantidad)

- Clave Ajena: sucursal REFERENCIA A SUCURSAL (codsuc)
- Clave Ajena: cliente REFERENCIA A CLIENTE (codcli)

**Enunciados y soluciones:**

1. **Actualizar** la calle y ciudad del cliente 'MALDONADO,D', pues se ha mudado a la calle 'LEPANTO' de 'YECLA'.

```
UPDATE CLIENTE
  SET calle = 'LEPANTO', ciudadcli = 'YECLA'
 WHERE nomcli = 'MALDONADO,D';
```

2. Modificar el activo de la sucursal SUR, para disminuirlo en 1650 euros.

```
UPDATE SUCURSAL
  SET activo = activo - 1650
 WHERE nomsuc = 'SUR';
```

3. Modificar la cantidad de cada préstamo concedido al cliente 'ESPALLARDO,E' para aumentarlo un 20%.

```
UPDATE PRESTAMO
  SET cantidad=cantidad*1.2
 WHERE cliente IN (SELECT codcli
                   FROM CLIENTE
                   WHERE nomcli = 'ESPALLARDO,E');
```

4. Añadir una columna a la tabla CLIENTE, llamada "cuantas\_cuentas" que almacenará valores numéricos de dos dígitos; su valor por defecto ha de ser 0; no admite nulos.

```
ALTER TABLE CLIENTE ADD
    cuantas_cuentas NUMBER(2) DEFAULT 0 NOT NULL;
```

5. Actualizar la columna "cuantas\_cuentas" de CLIENTE con el número de cuentas que tiene cada uno de los clientes de la base de datos.

```
UPDATE CLIENTE C
    SET cuantas_cuentas = (SELECT COUNT(*)
                           FROM CUENTA
                           WHERE cliente = C.codcli);
```

6. Añadir una columna a la tabla SUCURSAL, llamada "media\_prestamo" que almacenará valores numéricos de 9 dígitos con 2 decimales; no tiene valor por defecto; sí admite nulos.

```
ALTER TABLE SUCURSAL ADD media_prestamo NUMBER(9,2) NULL;
```

7. Actualizar la columna "media\_prestamo" para todas las sucursales, de forma que para cada sucursal que haya concedido 2 o más préstamos, almacene la cantidad media prestada por parte de dicha sucursal a los clientes.

```
UPDATE SUCURSAL S
    SET media_prestamo = (SELECT AVG(cantidad)
                           FROM PRESTAMO
                           WHERE sucursal = S.codsuc)
WHERE codsuc IN (SELECT sucursal
                 FROM PRESTAMO
                 GROUP BY sucursal
                 HAVING COUNT(*) >= 2);
```

8. **Insertar** las siguientes filas en las tablas del esquema

- El cliente llamado 'VALIENTE,T', con el código 'C10', que vive en la calle 'VENUS' de 'ALICANTE'.
- El banquero personal del cliente 'VALIENTE,T' es el banquero 'IBANEZ,J'.
- La cuenta 'Q10' del cliente 'C10', abierta en la sucursal 'S02' con un saldo de 3650 euros.
- La cuenta 'Q11' del cliente 'C10', abierta en la sucursal 'S03' con un saldo de 2000 euros.
- El préstamo 'P09' del cliente 'C10', concedido por la sucursal 'S01' por una cantidad de 600 euros.

```
INSERT INTO CLIENTE (codcli, nomcli, calle, ciudadcli)
    VALUES ('C10', 'VALIENTE,T', 'VENUS', 'ALICANTE');
INSERT INTO BANQUERO_PERSONAL
    VALUES ('C10', 'IBANEZ,J');
INSERT INTO CUENTA (numcuenta, sucursal, cliente, saldo)
    VALUES ('Q10', 'S02', 'C10', 3650);
INSERT INTO CUENTA (numcuenta, sucursal, cliente, saldo)
    VALUES ('Q11', 'S03', 'C10', 1500);
INSERT INTO PRESTAMO (numprestamo, sucursal, cliente, cantidad)
    VALUES ('P09', 'S01', 'C10', 600);
```

9. **Eliminar** los préstamos cuya cantidad sea inferior a 900 euros.

```
DELETE FROM PRESTAMO
WHERE cantidad < 900;
```

10. Eliminar las cuentas pertenecientes al cliente 'VALIENTE,T' cuyo saldo sea inferior a 2000 euros.

```
DELETE FROM CUENTA
WHERE saldo < 2000
  AND cliente IN (SELECT codcli
                  FROM CLIENTE
                  WHERE nomcli = 'VALIENTE,T');
```

11. Eliminar los clientes de Alicante que no tienen ni préstamos ni cuentas. Si surgen problemas de integridad referencial, elimine también las filas de otras tablas afectadas.

```
DELETE FROM CLIENTE
WHERE ciudadcli = 'ALICANTE'
  AND codcli NOT IN (SELECT cliente
                    FROM CUENTA)
  AND codcli NOT IN (SELECT cliente
                    FROM PRESTAMO);

-- Da error, porque ese/esos cliente/s están referenciados
-- desde BANQUERO_PERSONAL. Antes hay que ejecutar esta sentencia:
DELETE FROM BANQUERO_PERSONAL
WHERE cliente = (SELECT codcli
                 FROM CLIENTE
                 WHERE ciudadcli = 'ALICANTE'
                   AND codcli NOT IN (SELECT cliente
                                     FROM CUENTA)
                   AND codcli NOT IN (SELECT cliente
                                     FROM PRESTAMO));

-- Y ahora esta sí funciona:
DELETE FROM CLIENTE
WHERE ciudadcli = 'ALICANTE'
  AND codcli NOT IN (SELECT cliente
                    FROM CUENTA)
  AND codcli NOT IN (SELECT cliente
                    FROM PRESTAMO);
```