\*\*Ejercicio A: Subconsultas\*\*

1. Obtener los nombres de los clientes que han realizado al menos tres extracciones en el último mes:

```sql

SELECT DISTINCT c.nombre

FROM clientes c

WHERE (

    SELECT COUNT(\*)

    FROM transacciones t

    WHERE t.cliente\_id = [c.id](http://c.id)

    AND t.tipo = 'extraccion'

    AND t.fecha >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH)

) >= 3;

```

2. Obtener el cliente con el mayor número de cuentas:

```sql

SELECT c.nombre, COUNT([cu.id](http://cu.id" \t "_blank)) as numero\_cuentas

FROM clientes c

INNER JOIN cuentas cu ON [c.id](http://c.id" \t "_blank) = cu.cliente\_id

GROUP BY [c.id](http://c.id), c.nombre

HAVING COUNT([cu.id](http://cu.id" \t "_blank)) = (

    SELECT COUNT(cuenta\_id)

    FROM cuentas

    GROUP BY cliente\_id

    ORDER BY COUNT(id) DESC

    LIMIT 1

);

```

3. Obtener las ciudades donde todos los clientes tienen más de 50 años:

```sql

SELECT DISTINCT ciudad

FROM clientes c1

WHERE NOT EXISTS (

    SELECT 1

    FROM clientes c2

    WHERE c2.ciudad = c1.ciudad

    AND TIMESTAMPDIFF(YEAR, c2.fecha\_nacimiento, CURDATE()) <= 50

);

```

\*\*Ejercicio B: Procedimientos\*\*

4. Crear un procedimiento para insertar un nuevo cliente y una cuenta asociada:

```sql

DELIMITER //

CREATE PROCEDURE insertar\_cliente\_cuenta(

    IN p\_nombre VARCHAR(100),

    IN p\_apellido VARCHAR(100),

    IN p\_fecha\_nacimiento DATE,

    IN p\_ciudad VARCHAR(100),

    IN p\_tipo\_cuenta VARCHAR(50),

    IN p\_saldo\_inicial DECIMAL(10,2)

)

BEGIN

    DECLARE v\_cliente\_id INT;

    -- Insertar el nuevo cliente

    INSERT INTO clientes (nombre, apellido, fecha\_nacimiento, ciudad)

    VALUES (p\_nombre, p\_apellido, p\_fecha\_nacimiento, p\_ciudad);

    -- Obtener el ID del cliente recién insertado

    SET v\_cliente\_id = LAST\_INSERT\_ID();

    -- Insertar la nueva cuenta asociada

    INSERT INTO cuentas (cliente\_id, tipo\_cuenta, saldo)

    VALUES (v\_cliente\_id, p\_tipo\_cuenta, p\_saldo\_inicial);

    SELECT 'Cliente y cuenta creados exitosamente' AS mensaje;

END //

DELIMITER ;

```

5. Procedimiento para generar un reporte de las 10 transacciones más grandes de un cliente:

```sql

DELIMITER //

CREATE PROCEDURE top\_10\_transacciones\_cliente(

    IN p\_cliente\_id INT

)

BEGIN

    SELECT

        t.fecha,

        t.tipo,

        t.monto,

        c.numero\_cuenta

    FROM transacciones t

    INNER JOIN cuentas c ON t.cuenta\_id = [c.id](http://c.id)

    WHERE c.cliente\_id = p\_cliente\_id

    ORDER BY t.monto DESC

    LIMIT 10;

END //

DELIMITER ;

```

\*\*Ejercicio C: Teoría\*\*

6. La respuesta correcta es la opción a):

"El procedimiento solo actualiza el stock si hay suficiente cantidad disponible"

Esto es correcto porque el procedimiento verifica que v\_stock\_actual + p\_cantidad no sea menor que 0 antes de realizar la actualización.

7. Las respuestas verdaderas sobre Stored Procedures son:

- b) Son procedimientos almacenados en la base de datos y se ejecutan con CALL

- c) Se puede invocar a un Stored Procedure dentro de otro Stored Procedure

- d) Los parámetros pueden ser de ENTRADA, SALIDA y ENTRADA/SALIDA