



Materia: Bases de datos

Profesor: Fernando Arreola Franco

Alumno: Rueda De Oliveira Chun Shik

Tarea: Tarea 14

Semestre: 2026-1

Grupo: 1

## Introducción

Las subconsultas en SQL representan una de las herramientas más poderosas y versátiles en el manejo de bases de datos relacionales. Estas consultas anidadas permiten resolver problemas complejos de extracción y manipulación de datos mediante la combinación de múltiples operaciones en una sola sentencia SQL. El dominio de las subconsultas es fundamental para cualquier profesional de bases de datos, ya que facilitan la escritura de consultas más eficientes, legibles y mantenibles, permitiendo realizar filtrados avanzados, cálculos complejos y operaciones de comparación que serían difíciles de lograr con consultas simples.

### ¿Qué es una Subconsulta?

Una subconsulta es una consulta SQL que se encuentra anidada dentro de otra sentencia SQL más grande, la cual se denomina consulta externa o consulta contenedora. También se las conoce como consultas internas o consultas anidadas.

Estas consultas siempre se encierran entre paréntesis y, por lo general, se ejecutan antes que la consulta externa. El resultado de la subconsulta (ya sea un solo valor, una lista de valores o una tabla temporal) es utilizado por la consulta externa para realizar sus operaciones.

### Dónde y Cuándo Usar Subconsultas

Puedes utilizar subconsultas en varias partes de una sentencia SQL y con diferentes tipos de comandos:

- En comandos DQL (SELECT): Es el uso más común, también se pueden emplear en las cláusulas SELECT, FROM, WHERE, JOIN y HAVING.
- En comandos DML: También pueden usarse en sentencias de modificación de datos como INSERT, UPDATE y DELETE para definir los valores o los registros a afectar.

Aquí una tabla con escenarios y ejemplos de uso

Escenario	Ejemplo de uso
Filtrar con resultados de otro SELECT	WHERE precio > (SELECT AVG(precio) FROM productos)
Comprobar existencia	WHERE EXISTS (SELECT 1 FROM pedidos WHERE cliente_id = clientes.id)
Usar resultados de un GROUP JOIN	FROM (SELECT cliente_id, SUM(total) FROM pedidos GROUP BY cliente_id) AS ventas
Crear columnas calculadas	SELECT nombre, (SELECT COUNT(*) FROM pedidos WHERE cliente_id = clientes.id) AS total_pedidos

### Subconsultas correlacionadas

Una subconsulta correlacionada es un tipo especial de subconsulta donde la consulta interna depende de la consulta externa para su ejecución. A diferencia de las subconsultas normales que se ejecutan una sola vez, las correlacionadas se ejecutan una vez por cada fila procesada por la consulta externa.

Su característica principal es que, dentro de la subconsulta, se hace referencia a una columna de la tabla que está en la consulta externa.

Ejemplo: Para encontrar a todos los clientes que no han realizado ningún pedido en el último mes, podrías usar una subconsulta correlacionada con NOT EXISTS:

```
SELECT nombre
FROM clientes c
WHERE NOT EXISTS (
    SELECT 1
    FROM pedidos p
    WHERE p.cliente_id = c.id -- ¡Refiere a la tabla 'c' de la consulta externa!
    AND p.fecha > CURRENT_DATE - INTERVAL 1 MONTH
);
```

En este ejemplo, para cada cliente en la tabla clientes, la subconsulta verifica si existe al menos un pedido reciente para ese cliente en específico.

## Ejemplos Prácticos

Ejemplo 1: Subconsulta Escalar en el WHERE

Objetivo: Listar todos los productos cuyo precio es superior al precio promedio.

```
SELECT nombre, precio
FROM productos
WHERE precio > (
    SELECT AVG(precio) -- Esta subconsulta devuelve un solo valor (el
    promedio)
    FROM productos
);
```

## Ejemplo 2: Subconsulta de Múltiples Filas con IN

Objetivo: Obtener los nombres de todos los clientes que han realizado al menos un pedido.

```
SELECT nombre, apellido
FROM clientes
WHERE id IN (
    SELECT DISTINCT cliente_id -- Esta subconsulta devuelve una lista de IDs
    FROM pedidos
);
```

## Ejemplo 3: Subconsulta en el FROM (Tabla Derivada)

Objetivo: Mostrar el nombre de cada artista junto con el total de sus ventas.

```
SELECT a.nombre, a.apellido, ventas_totales.monto
FROM artistas a
JOIN (
    SELECT artista_id, SUM(venta_precio) AS monto -- Crea una tabla
temporal agrupada
    FROM ventas
    GROUP BY artista_id
) AS ventas_totales
ON a.id = ventas_totales.artista_id;
```

## Ejemplo 4: Subconsulta Correlacionada en el SELECT

Objetivo: Listar todos los clientes y, junto a cada uno, la cantidad de pedidos que han realizado.

```
SELECT nombre,
    ( SELECT COUNT(*)
        FROM pedidos p
        WHERE p.cliente_id = c.id -- Correlacionada: cuenta los pedidos
de *este* cliente
    ) AS numero_de_pedidos
FROM clientes c;
```

## Consideraciones finales

- Legibilidad vs. Rendimiento: Las subconsultas pueden hacer que la lógica compleja sea más fácil de escribir y entender. Sin embargo, en algunos casos, especialmente con conjuntos de datos muy grandes, una sentencia con JOIN podría ser más eficiente. Es recomendable probar ambas opciones si el rendimiento es crítico.
- Regla de oro: La subconsulta debe devolver un tipo de resultado compatible con el operador de la consulta externa. Por ejemplo, si usas el operador `>`, la subconsulta debe devolver un único valor.

## Conclusión

El estudio de las subconsultas revela su importancia fundamental como herramienta indispensable en el arsenal de todo desarrollador SQL. Estas consultas anidadas no solo amplían las capacidades de consulta tradicionales, sino que permiten abordar problemas de datos complejos con soluciones elegantes y eficientes. Desde simples comparaciones con valores únicos hasta operaciones avanzadas con subconsultas correlacionadas, esta técnica demuestra su versatilidad en múltiples escenarios prácticos. Su correcta implementación, considerando tanto el rendimiento como la mantenibilidad del código, marca la diferencia entre consultas básicas y soluciones de base de datos profesionales y optimizadas que responden efectivamente a las necesidades del negocio.

## Bibliografía

Beaulieu, A. (s/f). *Aprender Snowflake SQL y Scripting*. O'Reilly Media.

Benítez, R. (s/f). *sql-cheatsheet: Resumen práctico de SQL: definición de datos, consultas, manipulación, control y transacciones*.

*¿Cómo se manejan las consultas SQL complejas con varias tablas y condiciones?* (2023, marzo 22). Linkedin.com; [www.linkedin.com](http://www.linkedin.com).  
<https://www.linkedin.com/advice/3/how-do-you-handle-complex-sql-queries-multiple?lang=es>

Correa, C. (s/f). *apuntes-sql: Apuntes y ejercicios resueltos de SQL en PostgreSQL, SQL Server y phpMyAdmin*.

Database Development. (s/f). *¿Cómo se decide cuándo usar una subconsulta o una unión en SQL?* Linkedin.com. Recuperado el 7 de noviembre de 2025, de <https://www.linkedin.com/advice/0/how-do-you-decide-when-use-subquery-join-sql?lang=es>

Database Testing. (2023, marzo 22). *¿Cómo se usan las subconsultas para filtrar datos de varias tablas?* Linkedin.com; [www.linkedin.com](http://www.linkedin.com).  
<https://www.linkedin.com/advice/0/how-do-you-use-subqueries-filter-data-from-multiple?lang=es>