



**Universidad Nacional Autónoma de México**



**Núñez Badillo Armando Adair**

**Bases de datos**

**Grupo 1**

**Tarea 14**

**Subconsultas**

**Semestre 2026-1**

**Profesor: Ing. Fernando Arreola**

**Fecha de entrega: 10/11/25**

# Subconsultas

## 1) Definición

Una **subconsulta** es una consulta SQL **anidada** dentro de otra (en SELECT, FROM, WHERE o HAVING) cuyo **resultado** se usa para filtrar, calcular o construir el resultado final de la consulta externa. Puede devolver **un valor escalar (1×1)**, **una columna**, **varias columnas (fila)** o **una tabla** temporal.

Una subconsulta permite que las consultas SQL sean más modulares al gestionar tareas que, de otro modo, requerirían varias consultas independientes.

Definición y finalidad

Una subconsulta SQL es una consulta anidada dentro de otra consulta SQL, que se utiliza para realizar operaciones que requieren varios pasos o una lógica compleja. La función de las subconsultas en SQL incluye lo siguiente:

- Filtrar registros basándose en datos de tablas relacionadas.
- Agregando datos y realizando cálculos dinámicamente.
- Cruzar datos entre tablas para obtener información específica.
- Seleccionar filas condicionalmente sin necesidad de uniones explícitas o lógica de código externa.

---

## 2) ¿Dónde puedo usarlas y bajo qué condiciones?

- **Tipos según cardinalidad:**
  - **Escalar:** debe devolver **una sola fila y una sola columna**; de lo contrario, error.
  - **De conjunto:** devuelve **una columna** (para IN/ANY/ALL) o varias columnas (unidas por posición con comparadores de filas).
  - **EXISTS/NOT EXISTS:** no usan el valor devuelto; solo verifican **si existen filas**.
- **Condiciones prácticas:**
  - Si la subconsulta usa columnas de la consulta externa → es **correlacionada** (se evalúa por cada fila externa).

- Si **no** referencia columnas externas → **no correlacionada** (se evalúa una vez).

## Consideraciones sobre el rendimiento y buenas prácticas

Aunque las subconsultas son potentes a la hora de escribir consultas complejas, pueden afectar al rendimiento, especialmente cuando se trabaja con grandes conjuntos de datos. Es importante tener en cuenta los escollos habituales y las mejores prácticas para mejorar el rendimiento.

Optimizar el rendimiento de las subconsultas

Optimizar el rendimiento de las subconsultas garantiza una mejora del tiempo de ejecución de la consulta y de la capacidad de respuesta de la base de datos. A continuación, se indican formas de optimizar la subconsulta.

- **Índice Columnas relevantes:** Para acelerar la recuperación de datos, asegúrate de que las columnas utilizadas en las cláusulas WHERE y JOIN y en las operaciones de comparación estén indexadas.
- **Limita el Uso de Subconsultas Correlacionadas:** Siempre que sea posible, utiliza operaciones JOIN o CTE en lugar de subconsultas correlacionadas, ya que a menudo pueden procesar los datos más rápidamente utilizando operaciones de conjunto en lugar de procesar fila por fila.
- **Limitar el número de columnas en las subconsultas:** Selecciona sólo las columnas que necesites en las subconsultas para minimizar la recuperación de datos, reducir el uso de memoria y permitir que la base de datos optimice la ejecución.
- **Utiliza EXISTE en lugar de EN:** Si una subconsulta devuelve un conjunto de datos grande, utilizar EXISTS en lugar de IN puede mejorar el rendimiento. El operador EXISTS detendrá el procesamiento de la consulta cuando encuentre una fila coincidente, mientras que el operador IN seguirá evaluando todo el resultado de la subconsulta.

---

### 3) Subconsultas correlacionadas (qué son y cuándo usarlas)

Una **subconsulta correlacionada** hace referencia a columnas de la consulta externa. Se ejecuta **por cada fila** de la externa. Útil para condiciones “por grupo/relación” difíciles de expresar con una sola agregación o cuando se busca “¿existe al menos una fila relacionada que cumpla X?”.

- Pros: expresivas, fáciles de leer para “existe/no existe”.
- Contras: pueden ser más **costosas** (se evalúan repetidamente). Procura tener **índices** en las columnas de enlace.

La siguiente consulta se ejecuta en este orden:

- Para cada fila de employees (alias e1), la subconsulta (SELECT AVG(e2.salary) FROM employees e2 WHERE e2.department\_id = e1.department\_id) calcula el salario medio de ese departamento concreto.
- A continuación, la consulta externa compara el salario de cada empleado con el salario medio del departamento e incluye sólo a los que ganan más.

```
-- Retrieves names of employees with above-average salary in their department
```

```
SELECT e1.employee_name
```

```
FROM employees e1
```

```
-- Subquery: calculates average salary for each department
```

```
WHERE e1.salary > (SELECT AVG(e2.salary) FROM employees e2 WHERE e2.department_id
```

---

## 4) Ejemplos

### *Subconsultas escalares*

Las subconsultas escalares devuelven un único valor, como una fila y una columna. Suelen utilizarse cuando se espera un único valor, como en cálculos, comparaciones o asignaciones en las cláusulas SELECT o WHERE.

En el ejemplo siguiente, la subconsulta escalar (SELECT AVG(salary) FROM employees) devuelve un único valor, el salario medio, y lo compara con el salario de cada empleado.

```
-- Example of Scalar Subquery  
-- Compares each salary to the average salary  
  
SELECT employee_name,  
       salary,  
       (SELECT AVG(salary) FROM employees) AS average_salary  
  
FROM employees  
  
WHERE salary > (SELECT AVG(salary) FROM employees);
```

### **POWERED BY**

*Subconsultas de columna*

Las subconsultas de columna devuelven una sola columna pero varias filas. Estas subconsultas se utilizan a menudo con operadores como IN o ANY, donde la consulta externa compara valores de varias filas.

Por ejemplo, la subconsulta siguiente devuelve una lista de identificadores de departamento para los departamentos situados en Nueva York, que la consulta principal utiliza después para filtrar a los empleados de esos departamentos.

```
-- Example of Column Subquery  
-- Filters based on departments in New York  
  
SELECT employee_name  
  
FROM employees  
  
WHERE department_id IN (SELECT department_id FROM departments WHERE location  
= 'New York');
```

## POWERED BY

### *Subconsultas de fila*

Las subconsultas de fila devuelven una única fila que contiene varias columnas. Estas subconsultas se suelen utilizar con operadores de comparación que pueden comparar una fila de datos, como los operadores = o IN, cuando se esperan varios valores.

La siguiente subconsulta recupera el departamento y el cargo de un directivo, y la consulta externa encuentra los empleados con valores coincidentes.

```
-- Example of Row Subquery
```

```
-- Matches department and job title with a specific manager
```

```
SELECT employee_name
```

```
FROM employees
```

```
WHERE (department_id, job_title) = (SELECT department_id, job_title FROM managers  
WHERE manager_id = 1);
```

## POWERED BY

### *Subconsultas de tablas (tablas derivadas)*

Las subconsultas de tabla, o tablas derivadas, devuelven una tabla completa de varias filas y columnas. Se suelen utilizar en la cláusula FROM como tabla temporal dentro de una consulta.

Por ejemplo, la subconsulta siguiente crea una tabla derivada de salarios medios por departamento, que luego se utiliza en la consulta externa para encontrar departamentos con un salario medio superior a un umbral especificado.

```
-- Example of Table Subquery
```

```
-- Uses derived table for average department salary comparison
```

```
SELECT dept_avg.department_id, dept_avg.avg_salary
```

```
FROM
```

```
(SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id) AS dept_avg
```

```
WHERE dept_avg.avg_salary > 50000;
```

## POWERED BY

### Comprender las subconsultas SQL

En los ejemplos anteriores, hemos visto que una subconsulta se encierra entre paréntesis. Exploraremos ahora la mecánica de una subconsulta, incluyendo su sintaxis y [orden de ejecución](#).

#### Sintaxis y estructura

La sintaxis de una subconsulta varía en función de dónde se utilice en la sentencia SQL principal, como dentro de las cláusulas SELECT, FROM o WHERE. Las subconsultas suelen ir entre paréntesis ( ), lo que indica que se trata de una consulta independiente.

El siguiente ejemplo muestra una subconsulta dentro de una cláusula WHERE, que nos permite filtrar datos en la consulta principal basándonos en los resultados de una consulta anidada.

```
-- Selects the main column to retrieve from the main table to query
```

```
SELECT column_name
```

```
FROM table_name
```

```
-- Applies a condition to filter rows based on the subquery result
```

```
WHERE column_name operator
```

```
-- Subquery retrieves data for comparison in the WHERE clause
```

```
(SELECT column_name FROM table_name WHERE condition);
```

## POWERED BY

## Referencias

- [1] DataCamp, “Subconsulta SQL: Guía completa,” *DataCamp – Tutoriales*, actualizado el 16 ene. 2025. [En línea]. Disponible en: <https://www.datacamp.com/es/tutorial/sql-subquery>. [Accedido: 11 nov. 2025].