

## Subconsulta SQL

Dannae Sanchez Duran

09 de Mayo del 2024

**SELECT:** Una subconsulta con SELECT en SQL es una consulta dentro de otra consulta. Se utiliza para realizar operaciones más complejas y obtener resultados específicos al combinar datos de múltiples tablas o realizar cálculos condicionales

Por ejemplo, supongamos que tienes dos tablas: "empleados" y "departamentos". Quieres encontrar a todos los empleados que trabajan en el departamento de "Ventas". Puedes hacerlo mediante una subconsulta con SELECT de la siguiente manera:

```
SELECT nombre
FROM empleados
WHERE departamento_id = (SELECT id FROM departamentos WHERE nombre =
'Ventas');
```

En este ejemplo, la subconsulta (SELECT id FROM departamentos WHERE nombre = 'Ventas') se ejecuta primero para encontrar el ID del departamento de Ventas. Luego, este resultado se utiliza en la consulta principal para filtrar los empleados que trabajan en ese departamento.

Un caso de uso utilizando subconsulta de SELECT es el siguiente:

Supongamos que tienes una base de datos de una tienda en línea que contiene tablas para productos y pedidos. Quieres encontrar todos los productos que se han vendido al menos una vez en la última semana.

```
SELECT nombre, precio
FROM productos
WHERE id IN (
    SELECT producto_id
```

```
FROM pedidos
WHERE fecha >= DATE_SUB(CURDATE(), INTERVAL 7 DAY)
);
```

En este ejemplo:

1. La subconsulta (SELECT producto\_id FROM pedidos WHERE fecha >= DATE\_SUB(CURDATE(), INTERVAL 7 DAY)) selecciona los IDs de los productos que se han vendido en los últimos 7 días.
2. La consulta principal selecciona el nombre y el precio de los productos que tienen un ID que coincide con los IDs obtenidos en la subconsulta.

Así, obtienes una lista de productos que se han vendido al menos una vez en la última semana, lo que puede ser útil para realizar análisis de ventas o tomar decisiones comerciales basadas en la demanda reciente.

Las subconsultas con SELECT en SQL tienen algunas restricciones que debes tener en cuenta al utilizarlas:

1. Ubicación: Las subconsultas pueden aparecer en diferentes partes de una consulta SQL, como en la cláusula SELECT, FROM, WHERE, HAVING o incluso en la cláusula SET para actualizaciones. Sin embargo, cada tipo de subconsulta tiene sus propias restricciones y reglas.
2. Tipo de resultado: Una subconsulta SELECT debe devolver un solo valor, una lista de valores o una tabla de registros. La cantidad de columnas y el tipo de datos devueltos deben ser coherentes con el contexto en el que se utiliza la subconsulta.
3. Operadores de comparación: Las subconsultas SELECT pueden utilizar operadores de comparación como "=", ">", "<", etc., pero se debe tener en cuenta que si la subconsulta devuelve más de un valor, el operador de comparación debe ser adecuado para manejar conjuntos de datos.
4. Correlación: Las subconsultas correlacionadas son aquellas en las que la subconsulta hace referencia a una tabla externa a ella misma. En este caso, la subconsulta se ejecuta una vez por cada fila de la tabla externa. Sin embargo, no todas las bases de

datos admiten subconsultas correlacionadas o pueden tener restricciones en su uso debido a problemas de rendimiento.

5. Cláusula ORDER BY: No todas las bases de datos permiten la cláusula ORDER BY dentro de una subconsulta. Si se necesita ordenar los resultados de una subconsulta, generalmente se hace en la consulta principal que la contiene.
6. LIMIT/ROWNUM: Algunas bases de datos tienen restricciones en el uso de LIMIT (MySQL, PostgreSQL) o ROWNUM (Oracle) dentro de subconsultas. Esto puede variar según la implementación específica de la base de datos.
7. Funciones agregadas: Las subconsultas pueden contener funciones agregadas como SUM(), COUNT(), AVG(), etc. Sin embargo, estas funciones deben usarse de manera adecuada y coherente con el contexto de la consulta principal.

**FROM:** La cláusula From se puede utilizar para especificar una expresión de subconsulta en SQL. La relación producida por la subconsulta se usa luego como una nueva relación sobre la que se aplica la consulta externa.

Las variables de correlación de las relaciones en la cláusula from no se pueden utilizar en las subconsultas de la cláusula from.

Su sintaxis es de la siguiente manera:

```
SELECT column1, column2 FROM  
(SELECT column_x as C1, column_y FROM table WHERE PREDICATE_X)  
as table2, table1  
WHERE PREDICATE;
```

Un caso de uso es el siguiente: Encuentre todos los profesores cuyo salario es mayor que el presupuesto promedio de todos los departamentos.

Relación del instructor :

InstructorID	Nombre	Departamento	Salario
44547	Herrero	Ciencias de la Computación	95000
44541	Factura	Eléctrico	55000
47778	Sam	Humanidades	44000
48147	erik	Mecánico	80000
411547	Melisa	Tecnologías de la información	65000
48898	Jena	Civil	50000

Relación de departamento :

Nombre de Departamento	Presupuesto
Ciencias de la Computación	100000
Eléctrico	80000
Humanidades	50000
Mecánico	40000
Tecnologías de la información	90000
Civil	60000

La consulta que lo resuelve es la siguiente:

```

Select I.ID, I.NAME, I.DEPARTMENT, I.SALARY from
(select avg(BUDGET) as averageBudget from DEPARTMENT) as BUDGET,
Instructor as I
where I.SALARY > BUDGET.averageBudget;

```

### Producción

InstructorID	Nombre	Departamento	Salario
44547	Herrero	Ciencias de la Computación	95000
48147	erik	Mecánico	80000

Explicación : el presupuesto promedio de todos los departamentos de la relación de departamento es 70000. Erik y Smith son los únicos instructores en la relación de instructores cuyo salario es superior a 70000 y, por lo tanto, están presentes en la relación de salida.

Las subconsultas en la cláusula FROM de una consulta SQL, también conocidas como "tablas derivadas", tienen algunas restricciones específicas:

1. Sintaxis: Debes asegurarte de que la sintaxis de la subconsulta esté correctamente escrita y que produzca los resultados esperados. Esto incluye la selección de las columnas adecuadas y la aplicación de cualquier filtro necesario en la subconsulta misma.
2. Columnas de salida: La subconsulta debe devolver un conjunto de columnas con nombres y tipos de datos coherentes que se puedan usar en la consulta principal. Es importante que las columnas devueltas tengan nombres únicos si la consulta principal requiere referenciarlas explícitamente.
3. Correlación: Al igual que con otras subconsultas, las subconsultas FROM pueden ser correlacionadas, lo que significa que hacen referencia a las tablas de la consulta principal. En este caso, la subconsulta se ejecuta una vez por cada fila en la tabla externa.
4. Eficiencia: El uso de subconsultas FROM puede afectar el rendimiento de la consulta, especialmente si la subconsulta es compleja o devuelve un gran conjunto de datos. Es importante evaluar el impacto en el rendimiento y considerar alternativas si es necesario.
5. Optimización: Algunas bases de datos pueden tener limitaciones en cuanto a la optimización de consultas que involucran subconsultas FROM. Es importante entender cómo la base de datos maneja estas consultas y cómo se puede optimizar su rendimiento.

6. Operaciones permitidas: Las subconsultas FROM pueden realizarse en la mayoría de las bases de datos para realizar operaciones como JOIN, UNION o INTERSECT, pero es importante verificar la compatibilidad y las restricciones específicas de la base de datos que estés utilizando.
7. Filtrado y ordenamiento: Las subconsultas FROM pueden incluir cláusulas WHERE para filtrar filas y cláusulas ORDER BY para ordenar resultados. Sin embargo, algunas bases de datos pueden tener restricciones en el uso de estas cláusulas dentro de subconsultas FROM, así que es importante verificar la documentación de la base de datos específica.

**JOIN:** Una cláusula JOIN se utiliza cuando se necesita combinar datos de dos o más tablas en un conjunto de datos. Los registros de ambas tablas se comparan en función de una condición (también denominada predicado JOIN ) que se especifica en la cláusula JOIN. Si la condición se cumple, los registros se incluyen en la salida.

Los tipos de SQL JOIN incluyen:

1. INNER JOIN (también conocido como "simple" JOIN). Este es el tipo más común de JOIN.
2. LEFT JOIN (o LEFT OUTER JOIN)
3. RIGHT JOIN (o RIGHT OUTER JOIN)
4. FULL JOIN (o FULL OUTER JOIN)

Algunas de sus restricciones son:

1. La subconsulta devolverá una tabla con al menos una columna
2. No puede hacer referencia a columnas definidas fuera de ella.

Un ejemplo usando una subconsulta con JOIN es el siguiente:

```
SELECT empleados.nombre
FROM empleados
JOIN (
    SELECT departamento_id
    FROM departamentos
    WHERE presupuesto > 100000
) AS dept_presupuesto
```

```
ON empleados.departamento_id = dept_presupuesto.departamento_id;
```

WHERE: El uso de la cláusula WHERE en una subconsulta es común cuando queremos filtrar los resultados de la subconsulta antes de utilizarlos en la consulta principal. Esto es especialmente útil cuando queremos restringir los datos que se combinan o comparan en la consulta principal.

Se suele ocupar para los siguientes casos:

1. **Filtrado de datos:** Podemos usar una subconsulta con WHERE para seleccionar un subconjunto específico de datos de una tabla antes de unirlos o compararlos con otra tabla en la consulta principal. Esto puede ayudar a reducir el número de filas procesadas y mejorar el rendimiento de la consulta.
2. **Subconsulta correlacionada:** En una subconsulta correlacionada, la cláusula WHERE puede usarse para filtrar los resultados de la subconsulta basándose en valores de la tabla externa (de la consulta principal). Esto permite que la subconsulta esté relacionada con la consulta principal y sea más dinámica en función de los valores de las filas actuales en la consulta principal.
3. **Restricción de resultados:** Podemos usar WHERE en una subconsulta para aplicar condiciones específicas a los resultados antes de usarlos en la consulta principal. Esto puede ser útil cuando queremos limitar los datos que se combinan o comparan en la consulta principal.

Un ejemplo de aplicación es el siguiente:

```
SELECT *  
FROM tabla1 t1  
WHERE t1.columna1 > (SELECT AVG(columna2) FROM tabla2 WHERE columna3  
= t1.valor);
```

Algunas de sus restricciones son:

1. Solo debe devolver un valor o una lista de valores
2. No puede hacer referencia a columnas definidas fuera de ella.

HAVING: La cláusula HAVING se utiliza en SQL para filtrar grupos de filas devueltas por una consulta que involucra funciones de agregación, como SUM(), COUNT(), AVG(), etc. Generalmente se usa junto con la cláusula GROUP BY para especificar condiciones de filtro para los grupos resultantes. La cláusula HAVING se aplica después de que se hayan agrupado los datos según la cláusula GROUP BY.

Cuando se trata de subconsultas, la cláusula HAVING puede usarse dentro de una subconsulta para filtrar los resultados antes de que se combinen o se utilicen en la consulta principal.

Aquí hay un ejemplo de cómo se puede usar la cláusula HAVING en una subconsulta:

```
SELECT column1, COUNT(*) AS num_filas
FROM tabla1
GROUP BY column1
HAVING COUNT(*) > (SELECT COUNT(*) FROM tabla2 WHERE condicion);
```

Sus restricciones son las siguientes:

1. No todas las bases de datos admiten el uso de HAVING en subconsultas, por lo que debes consultar la documentación de tu base de datos específica para confirmar su compatibilidad.
2. Debe devolver solo un valor o una lista de valores
3. No puede hacer referencia a columnas definidas fuera de ella.

CORRELACIONADAS: las subconsultas correlacionadas son útiles cuando necesitas realizar operaciones basadas en valores de la fila actual de la consulta externa. Permiten una mayor flexibilidad y son útiles en una variedad de escenarios, desde filtraciones hasta cálculos y actualizaciones condicionales.

1. Filtrado basado en valores de la fila externa: Puedes usar una subconsulta correlacionada para filtrar los resultados de una consulta basándote en valores de la fila actual de la consulta externa.
2. Cálculos basados en valores de la fila externa: Puedes utilizar subconsultas correlacionadas para realizar cálculos basados en valores de la fila actual de la consulta externa.



```

SELECT empleado_id, SUM(ventas) AS total_ventas,
      (SELECT SUM(ventas) FROM ventas) AS total_ventas_totales,
      (SUM(ventas) / (SELECT SUM(ventas) FROM ventas)) * 100 AS
porcentaje_ventas
FROM ventas
GROUP BY empleado_id;

```

3. Actualizaciones condicionales: En actualizaciones condicionales, puedes utilizar subconsultas correlacionadas para establecer valores basados en condiciones de la fila actual.

Sus restricciones son:

1. Son menos eficientes que las no correlacionadas
2. Se ejecutan una vez por cada fila de la consulta externa, por lo que pueden ser costosas en términos de rendimiento.

## Referencias

[1] “Subconsulta con select en sql”- Chat GPT. Accedido el 04 de marzo de 2024.[En línea]. Disponible en: <https://chat.openai.com/c/11e62597-ad30-44f1-9f12-42e65c985cea>

[2]”Subconsulta en la cláusula FROM”-Rudeus Greyrat. Accedido el 05 de marzo del 2024. [En línea]. Disponible en: [SQL | Subconsultas en la cláusula From – Barcelona Geeks](#)

[3]K. Koidan. “¿Cuáles son los diferentes tipos de subconsultas SQL?” Learn SQL.es. Accedido el 10 de mayo de 2024. [En línea]. Disponible: <https://learnsql.es/blog/cuales-son-los-diferentes-tipos-de-subconsultas-sql/>