



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

BASE DE DATOS
ÁLGEBRA RELACIONAL

SUBCONSULTAS

INTEGRANTE
CRUZ VARGAS EMILIO

PROFESOR
ING. FERNANDO ARREOLA FRANCO

GRUPO

Índice

1. Investigar (casos de uso, restricciones, ejemplos)	2
2. Referencias	5

Lo primero que tenemos que hacer es definir qué es una subconsulta, es una consulta que se encuentra dentro de otra consulta más grande. La subconsulta se ejecuta primero y su resultado se utiliza en la consulta principal.

1. Investigar (casos de uso, restricciones, ejemplos)

1. Select Casos de uso:

- a) **Obtención de datos relacionado:** Sirve para extraer información de una tabla basada en datos de otra tabla.
- b) **Agregación de datos:** Operaciones de agregación en cuanto a conjuntos de datos que se especifiquen
- c) **Filtrado avanzado:** Filtrar resultados basados en condiciones más complejas

De las **restricciones** que tiene es que el uso excesivo de estas subconsultas puede afectar el tiempo de demora al hacer la consulta, también la **compatibilidad** ya que no todos los gestores de bases de datos aceptan las subconsultas.

Un ejemplo es que supongamos que tenemos dos tablas: `Empleados` y `Departamentos` se quiere encontrar todos los empleados que trabajan en el departamento `Ventas`:

```
SELECT Nombre, Apellido
FROM Empleados
WHERE DepartamentoID = (SELECT DepartamentoID FROM Departamentos WHERE
Nombre = 'Ventas');
```

- 2. `From` Esta subconsulta es conocida por ser una "tabla derivada." "tabla subconsulta", esta a diferencia se `SELECT` que devuelve un valor único una lista de valores, `"FROM"` devuelve un conjunto de datos que se puede utilizar como una tabla temporal.

Los casos de uso más conocidos son **combinación de datos** que sirve para combinar datos de varias tablas y crear una vista temporal de la consulta principal. También **agregación de datos** que sirve para realizar operaciones de agregación en conjuntos de datos específicos antes de unirse a la consulta principal.

Por parte de las restricciones al igual que `Select`, el abuso de esta puede ocasionar la recepción de las consultas y una muy importante, la **legibilidad** del código ya que puede afectar el entendimiento de este.

Un ejemplo de este es cuando tenemos las tablas `Empleados` y `Salarios`, de estas queremos encontrar el salario promedio del departamento por lo que se haría de la siguiente manera:

```
SELECT DepartamentoID, AVG(Salario) AS SalarioPromedio
FROM (SELECT E.DepartamentoID, S.Salario
FROM Empleados E
JOIN Salarios S ON E.EmpleadoID = S.EmpleadoID) AS
EmpleadosSalarios
GROUP BY DepartamentoID;
```

- 3. `Join` En particular esta subconsulta se puede utilizar para especificar una condición de unión más compleja. La subconsulta se ejecuta para cada fila de la tabla principal y devuelve un conjunto de

resultados que se utiliza para realizar la unión con la otra tabla.

Los casos de uso más frecuentes es **condiciones de unión complejas** las cuales implican que no pueda expresarse fácilmente utilizando solo las columnas de las tablas involucradas, también los filtros pueden ser tan específicos como se necesiten.

Por la parte de las restricciones se tienen la misma que los anteriores ya que el tiempo que podría tardar en hacer estas relaciones podría ser demasiado y no podría ser tan legible.

Un ejemplo es si tenemos las tablas `Empleados` y `Salarios` se desea encontrar empleados cuyo salario sea mayor al salario promedio de su departamento, por lo que:

```
SELECT E.Nombre, E.Apellido, S.Salario
FROM Empleados E
JOIN Salarios S ON E.EmpleadoID = S.EmpleadoID
JOIN (SELECT DepartamentoID, AVG(Salario) AS SalarioPromedio FROM
Salarios GROUP BY DepartamentoID) AS Subconsulta
ON E.DepartamentoID = Subconsulta.DepartamentoID
WHERE S.Salario > Subconsulta.SalarioPromedio;
```

4. **Where** Es una consulta que se utiliza para filtrar los resultados de una consulta principal. La subconsulta se ejecuta primero, y su resultado se utiliza como condición en la cláusula `WHERE` de la consulta principal.

Los casos de uso:

- a) **Filtrado avanzado:** Cuando necesitas aplicar condiciones más complejas que no pueden expresarse fácilmente utilizando operadores de comparación simples.
- b) **Para comparar valores de una fila con los resultados de una subconsulta.**

Las restricciones son las mismas que los casos anteriores ya que al poner tantas restricciones o tan complejas podría llegar a tomar demasiado tiempo y podría ser poco legible.

Un caso de uso podría ser si tenemos las tablas `Empleados` y `Salarios`, donde se desea encontrar empleados cuyo salario sea mayor al salario promedio de la empresa:

```
SELECT Nombre, Apellido, Salario
FROM Empleados
WHERE Salario > (SELECT AVG(Salario) FROM Salarios);
```

5. **Having** Por parte de esta subconsulta se enfoca en filtrar los resultados después de que se hayan agrupado mediante la cláusula `GROUP BY`, por lo que es similar a la cláusula `WHERE` pero esta se aplica a grupos de filas individuales.

Casos de uso:

- a) **Selección de grupos:** Para poder encontrar grupos específicos que cumplan las condiciones que se especifican
- b) **Filtrado de resultados agregados:** Sirve para aplicar condiciones a los resultados agregados después de agruparlos

Por la parte de las restricciones es que como se especificó en la definición, esta cláusula se ejecuta después de la cláusula `GROUP BY` para poder referirse a columnas agregadas o agrupadas; también, como las otras, puede llegar a afectar al rendimiento de la consulta general.

Un ejemplo es que si nosotros suponemos que tenemos la tabla `Ventas` se quiere encontrar los departamentos donde el total de ventas es mayor a \$10,000. Se podría hacer:

```
SELECT DepartamentoID, SUM(MontoVenta) AS TotalVentas
FROM Ventas
GROUP BY DepartamentoID
HAVING SUM(MontoVenta) > 10000;
```

6. Correlacionadas Hablando de la subconsulta correlacionadas.^{es}tán referenciadas por una columna de la consulta principal, estas a fiderencia de las que no son correlacionadas se ejecutan cada vez por fila procesada por la consulta principal.

casos de uso:

- a) **Filtrado basado en valores de la fila actual:** En casos de que se necesite filtrar los resultados de la subconsulta basándose en valores específicos de la fila actual de la consulta principal.
- b) **Comparaciones por fila:** Como ya se mencionó, para hcer comparaciones entre las filas de la subconsulta y las filas de la consulta principal.

Por la aprte de las restricciones es claro que la **eficeincia** es una de estas ya que si se tiene una tabla ddemasiado grande, las comparaciones que está va a hacer con las subconsultas serán demasiadas.

Un ejemplo claro es cuando tenemos la tabla .^{Em}pleadosz "Salarios" donde se quiere encontrar todos los empleados cuyo salario sea mayor al promedio de su departamento, entonces:

```
SELECT Nombre, Apellido, DepartamentoID, Salario
FROM Empleados E
WHERE Salario > (SELECT AVG(Salario) FROM Empleados WHERE DepartamentoID
= E.DepartamentoID);
```

2. Referencias

1. IBM documentation. (s. f.-b). <https://www.ibm.com/docs/es/qmf/12.1.0?topic=ddfmtuss-creating-subquery-retrieve-data-from-more-than-one-table>
2. Tema 5 Las subconsultas. (s. f.). https://www.aulalic.es/sql/t_5_1.htm
3. Subconsultas en SQL. (s. f.). DesarrolloWeb.com. <https://desarrolloweb.com/articulos/2337.php>