

📌 Subqueries en SQL (Consultas Anidadas)

Una **subquery** es una consulta dentro de otra consulta. Se usa para recuperar datos que luego se utilizan en la consulta principal.

Se pueden usar en:

- ✓ SELECT → Para filtrar o calcular valores.
- ✓ FROM → Como una tabla temporal.
- ✓ WHERE → Para filtrar resultados.

◆ Tipos de Subqueries

1. Subquery en WHERE (para filtrar datos)

Ejemplo: Encontrar clientes que han realizado un pedido.

```
SELECT Nombre
FROM Clientes
WHERE ClienteID IN (SELECT ClienteID FROM Pedidos);
```

📌 Explicación:

- La subquery obtiene los ClienteID que han hecho pedidos.
- La consulta principal selecciona los nombres de esos clientes.

2. Subquery en SELECT (para calcular valores)

Ejemplo: Mostrar el nombre del producto y su precio en comparación con el precio promedio.

```
SELECT Producto, Precio,
       (SELECT AVG(Precio) FROM Productos) AS Precio_Promedio
FROM Productos;
```

📌 Explicación:

- La subquery calcula el precio promedio de todos los productos.
 - Se muestra junto a cada producto para compararlo.
-

3. Subquery en FROM (como tabla temporal)

Ejemplo: Obtener el promedio de ventas por cliente y luego filtrar solo los clientes con más de 500 en promedio.

```
SELECT ClienteID, Promedio_Ventas
FROM (
    SELECT ClienteID, AVG(Precio) AS Promedio_Ventas
    FROM Pedidos
    GROUP BY ClienteID
) AS Subconsulta
WHERE Promedio_Ventas > 500;
```

📌 Explicación:

- La subquery (FROM (...) AS Subconsulta) calcula el promedio de ventas por cliente.
- La consulta principal filtra solo aquellos con promedio mayor a 500.

💡 Proyecto práctico: Encontrar empleados mejor pagados por departamento

📌 Escenario:

Una empresa quiere identificar a los empleados que tienen el **salario más alto en cada departamento**.

📌 Tabla: Empleados

EmpleadoID	Nombre	Departamento	Salario
1	Ana	Ventas	2500
2	Luis	IT	3500
3	Marta	Finanzas	2700
4	Pedro	IT	4000
5	Sofía	Ventas	2800

📌 Objetivo:

Escribir una consulta SQL que **muestre los empleados con el salario más alto en cada departamento**.

📌 **Pista:** Usa una **subquery en WHERE** para encontrar los empleados cuyo salario sea igual al salario máximo de su departamento.

🚩 **Reto:** Escribe tu propia consulta usando Subqueries

🚀 **Tu resultado debe verse así:**

Nombre	Departamento	Salario
Sofía	Ventas	2800
Pedro	IT	4000
Marta	Finanzas	2700

♦ Cuando tengas tu consulta, dime si necesitas ayuda o más desafíos 🚀 !

🚩 **Ejercicio 1: Encontrar clientes sin pedidos**

🚩 **Objetivo:** Obtener una lista de clientes que **no han realizado ningún pedido**.

Tablas disponibles:

Clientes

ClienteID	Nombre	País
1	Ana	México
2	Luis	Chile
3	Marta	Perú
4	Pedro	Argentina

Pedidos

PedidoID	ClienteID	Producto	Precio
101	1	Laptop	800
102	2	Teléfono	500
103	1	Tablet	300

♦ **Instrucciones:**

Escribe una consulta SQL para encontrar clientes que **no aparecen en la tabla de Pedidos** usando una **Subquery en WHERE**.

🚩 **Pista:** Usa NOT IN o NOT EXISTS.

🚀 **Ejercicio 2: Encontrar productos con un precio superior al promedio**

🚀 **Objetivo:** Listar los productos cuyo **precio sea mayor que el precio promedio de todos los productos**.

Tabla: Productos

ProductoID	Nombre	Precio
1	Laptop	800
2	Teléfono	500
3	Tablet	300
4	Audífonos	100

💡 **Instrucciones:**

Escribe una consulta SQL para encontrar los productos cuyo precio sea superior al **precio promedio** de todos los productos.

🚀 **Pista:** Usa una **Subquery en WHERE** para calcular el precio promedio con AVG().

🚀 **Ejercicio 3: Encontrar empleados con el segundo salario más alto**

🚀 **Objetivo:** Obtener a los empleados que tienen el **segundo salario más alto**.

Tabla: Empleados

EmpleadoID	Nombre	Salario
1	Ana	2500
2	Luis	3500
3	Marta	2700
4	Pedro	4000
5	Sofía	2800

💡 **Instrucciones:**

Escribe una consulta SQL que encuentre el **segundo salario más alto** y muestre el nombre del empleado que lo tiene.

🚀 **Pista:**

1. Usa DISTINCT para evitar duplicados.
2. Encuentra el **máximo salario que no sea el salario más alto** (MAX() dentro de una subquery).

✦ **Ejercicio 4: Mostrar clientes y el número de pedidos que han realizado**

✦ **Objetivo:** Mostrar una lista de clientes junto con la **cantidad de pedidos** que han hecho.

Tablas disponibles:

Clientes

ClienteID	Nombre
1	Ana
2	Luis
3	Marta

Pedidos

PedidoID	ClienteID
101	1
102	2
103	1

♦ **Instrucciones:**

Escribe una consulta SQL que muestre cada cliente junto con la cantidad de pedidos que ha realizado.

✦ **Pista:** Usa una **Subquery en SELECT** con COUNT().

✦ **Ejercicio 5: Encontrar los empleados mejor pagados en cada departamento**

✦ **Objetivo:** Mostrar **el empleado con el salario más alto en cada departamento**.

Tabla: Empleados

EmpleadoID	Nombre	Departamento	Salario
1	Ana	Ventas	2500
2	Luis	IT	3500
3	Marta	Finanzas	2700
4	Pedro	IT	4000
5	Sofía	Ventas	2800

♦ **Instrucciones:**

Escribe una consulta SQL que encuentre a los empleados cuyo salario sea el más alto en su departamento.

📌 **Pista:** Usa una **Subquery en WHERE** para comparar el salario con MAX() por departamento.

🔥 **Reto final: Construye tu propio análisis de datos con Subqueries**

Piensa en un problema real en el que puedas aplicar **Subqueries** y trata de escribir una consulta para resolverlo. Puede ser sobre ventas, clientes, empleados, etc.

✅ Si necesitas ayuda, dime en qué área quieres hacer el análisis y te guío 🚀