

# SQL: SUBCONSULTAS

Bases de Datos II - 3<sup>o</sup>2<sup>a</sup> DS

Prof. Ing. Juan Marquez



## SUBCONSULTAS

- Son consultas SELECT que están incluidas en otras consultas.
- Muchas veces pueden ser reemplazadas por JOINS. Se debe considerar qué sintaxis facilita la lectura.
- En ciertas ocasiones son imprescindibles.
- No hay límites en los niveles de anidación, pero dificulta la lectura

Se pueden utilizar en distintas partes de la query:

- En la lista de campos que va después de la palabra SELECT.
- En la parte de FROM (o JOIN)
- En las condiciones del WHERE o HAVING.

Podemos distinguir tres tipos de subconsultas:

- Las que devuelven **un solo valor**. Se incluyen en cualquier parte de la query en donde pondríamos un valor.
- Las que devuelven **una lista de valores** (es decir, muchos registros pero una sola columna). Suelen incluirse en el WHERE o HAVING, utilizando IN.
- Las que devuelven **una matriz** (muchos registros con muchas columnas). Normalmente se usan en la parte del FROM (o JOIN)

## EJEMPLO

Tomando la BD de la inmobiliaria:  
Mostrar id y precio de venta de las operaciones  
cuyo precio de venta sea superior al promedio.

```
SELECT id, precio_venta  
FROM operaciones  
WHERE precio_venta >  
      (SELECT AVG(precio_venta) FROM operaciones);
```

## OTRO EJEMPLO

Mostrar id, fecha de venta y precio de venta de las operaciones cuya fecha de venta haya sido posterior a la última venta en Entre Ríos.

**¿Cuándo fue la última venta en Entre Ríos?**

```
-- Última venta en Entre Ríos:  
SELECT MAX(o.fecha_venta)  
FROM operaciones o JOIN provincias p USING(id_provincia)  
WHERE p.descripcion LIKE "Entre Ríos";
```

```
+-----+  
| MAX(o.fecha_venta) |  
+-----+  
| 2007-04-14        |  
+-----+  
1 row in set (0.01 sec)
```

## Copiamos la query de la diapositiva anterior "dentro" de la query:

```
SELECT id, fecha_venta, precio_venta
FROM operaciones
WHERE fecha_venta > (
  -- Última venta en Entre Ríos
  SELECT MAX(o.fecha_venta)
  FROM operaciones o JOIN provincias p USING(id_provincia)
  WHERE p.descripcion LIKE "Entre Ríos"
);
```

**Otro ejemplo:** Mostrar el id y la fecha de venta de las ventas que **no** correspondan a los vendedores que han vendido más de 450 millones.



## ¿Cuáles son los id de los vendedores que han vendido mas de 450 millones?

```
SELECT v.id_vendedor, v.nombre, v.apellido, SUM(o.precio_venta)
FROM vendedores v JOIN operaciones o USING(id_vendedor)
GROUP BY v.id_vendedor
HAVING SUM(o.precio_venta) > 450000000;
```

id_vendedor	nombre	apellido	SUM(o.precio_venta)
15	Carmen	López	472591537
17	Joaquín	Iriarte	454648735

2 rows in set (0.02 sec)

## Reescribimos la consulta para que muestre solamente los id.

```
SELECT id_vendedor
FROM operaciones
GROUP BY id_vendedor
HAVING SUM(precio_venta) > 450000000;
```

```
+-----+
| id_vendedor |
+-----+
|          15 |
|          17 |
+-----+
```

Repetimos la consigna inicial: Mostrar el id y la fecha de venta de las ventas que **no** correspondan a los vendedores que han vendido más de 450 millones.

```
SELECT id, fecha_venta, id_vendedor
FROM operaciones
WHERE id_vendedor NOT IN (
    --Esta es la query de la diapositiva anterior:
    SELECT id_vendedor
    FROM operaciones
    GROUP BY id_vendedor
    HAVING SUM(precio_venta) > 450000000
);
```

## SUBCONSULTAS **NO** CORRELACIONADAS

Hasta aquí, hemos trabajado con **subconsultas completamente independientes** de la consulta principal:

- Primero se ejecuta la subconsulta.
- Luego se "ubica" el resultado de la misma en la consulta principal
- Finalmente se ejecuta la consulta principal.

## SUBCONSULTAS **NO** CORRELACIONADAS

- SELECT \*
- FROM operaciones
- WHERE precio\_venta >
- (SELECT AVG(precio\_venta) FROM operaciones);
- 1203087.5254

## SUBCONSULTAS CORRELACIONADAS

Estas subconsultas **no** son independientes de la consulta principal, sino que se ejecutan una vez por cada registro devuelto por la consulta principal.

## SUBCONSULTAS CORRELACIONADAS

Ejemplo: Mostrar nombre y apellido de los vendedores, con la cantidad de ventas de cada uno **y la fecha de su primer venta de más de un millón de pesos.**

```
SELECT id_vendedor, v.nombre, v.apellido, COUNT(*), (  
    SELECT MIN(fecha_venta)  
    FROM operaciones  
    WHERE id_vendedor = o.id_vendedor  
    AND precio_venta > 1000000  
    GROUP BY id_vendedor  
) as primer_millonaria  
FROM vendedores v JOIN operaciones o USING(id_vendedor)  
GROUP BY id_vendedor;
```

## SUBCONSULTAS CORRELACIONADAS

Ejemplo: Mostrar nombre y apellido de los vendedores, con la cantidad de ventas de cada uno **y la fecha de su primer venta de más de un millón de pesos.**

```
SELECT id_vendedor, v.nombre, v.apellido, COUNT(*), (  
    SELECT MIN(fecha_venta)  
    FROM operaciones  
    WHERE id_vendedor = o.id_vendedor  
    AND precio_venta > 1000000  
    GROUP BY id_vendedor  
) as primer_millonaria  
FROM vendedores v JOIN operaciones o USING(id_vendedor)  
GROUP BY id_vendedor;
```



RESULTADO:

id_vendedor	nombre	apellido	COUNT(*)	primer_millonaria
15	Carmen	López	387	2004-06-16
16	Pedro	Páramo	348	2004-06-02
17	Joaquín	Iriarte	388	2004-06-21
18	Jesús	Méndez	319	2004-06-19
19	María	Liberotti	336	2004-05-23
20	Luisa	Huilque	341	2004-04-21
22	Diego	Zapata	346	2004-04-19

## METODOLOGÍA PARA CONSULTAS COMPLEJAS

1. Expresar el problema **en castellano**.
2. Usar pseudocódigo para resolver parcialmente el problema.
3. Codificar las subconsultas y **probarlas** para asegurarse de que retornan el resultado esperado.
4. Codificar y probar la consulta completa.

**Fuente:** Murach's MySQL, 2ª ed.

1. Expresar el problema **en castellano**.

Después de que se realizó la última venta en Entre Ríos, ¿qué ventas se han realizado? Mostrar id, fecha, y precio de venta.

## 2. Usar pseudocódigo para resolver parcialmente el problema.

```
SELECT id, fecha_venta, precio_venta
FROM operaciones
WHERE fecha_venta > (
  -- Fecha de la última venta en Entre Ríos
);
```

### 3. Codificar las subconsultas y **probarlas** para asegurarse de que retornan el resultado esperado.

```
-- Última venta en Entre Ríos:  
SELECT MAX(o.fecha_venta)  
FROM operaciones o JOIN provincias p USING(id_provincia)  
WHERE p.descripcion = "Entre Ríos";
```

## 4. Codificar y probar la consulta completa.

```
SELECT id, fecha_venta, precio_venta
FROM operaciones
WHERE fecha_venta > (
  -- Última venta en Entre Ríos
  SELECT MAX(o.fecha_venta)
  FROM operaciones o JOIN provincias p USING(id_provincia)
  WHERE p.descripcion = "Entre Ríos"
);
```