

## Semana 3 – Consultas avanzadas: JOIN, subconsultas y vistas

### Objetivos de la semana

- Comprender la importancia de combinar información de varias tablas en una base de datos relacional.
- Aprender a usar los distintos tipos de JOIN (INNER, LEFT, RIGHT).
- Introducir el concepto y uso de las **subconsultas**.
- Conocer las **vistas** (VIEW) como herramienta para simplificar consultas.
- Aplicar lo aprendido en ejercicios integradores.

### 1. Introducción

Hasta ahora trabajamos con consultas sobre una sola tabla. Pero en una base de datos real, los datos están distribuidos en varias tablas que se relacionan entre sí. Por ejemplo: en una librería no alcanza con una tabla de *Libros*. También necesitamos tablas de *Autores*, *Clientes*, *Pedidos*, etc.

La potencia de SQL está en poder **combinar esas tablas** para responder preguntas más complejas.

### 2. JOINS

El JOIN permite **unir dos o más tablas** en una sola consulta, en base a una columna común.

#### a) INNER JOIN (o simplemente JOIN)

Trae solo los registros que tienen coincidencia en ambas tablas.

Ejemplo con tablas:

#### Autores

AutorID	Nombre
1	Isabel Allende
2	Julio Cortázar

### Libros

LibroID	Titulo	AutorID
1	La Casa de los Espíritus	1
2	Rayuela	2
3	Bestiario	2

Consulta:

```
SELECT Libros.Titulo, Autores.Nombre  
FROM Libros  
INNER JOIN Autores ON Libros.AutorID = Autores.AutorID;
```

Resultado:

Titulo	Nombre
La Casa de los Espíritus	Isabel Allende
Rayuela	Julio Cortázar
Bestiario	Julio Cortázar

### b) LEFT JOIN

Devuelve **todos los registros de la tabla izquierda**, aunque no tengan coincidencia en la tabla derecha.

Ejemplo: si tenemos un autor que todavía no publicó libros, igual aparecerá en la consulta.

```
SELECT Autores.Nombre, Libros.Titulo  
FROM Autores  
LEFT JOIN Libros ON Autores.AutorID = Libros.AutorID;
```

Resultado:

Nombre	Titulo
Isabel Allende	La Casa de los Espíritus
Julio Cortázar	Rayuela
Julio Cortázar	Bestiario
Isabel Allende	NULL

### c) RIGHT JOIN

Es similar al LEFT, pero devuelve todos los registros de la tabla derecha.

En muchos motores de BD (como MySQL) se usa menos, porque suele alcanzarnos con LEFT JOIN reorganizando el orden de las tablas.

## 3. Subconsultas

Una **subconsulta** es una consulta dentro de otra. Sirve para calcular un dato intermedio y usarlo como condición o como columna.

Ejemplo: listar los libros cuyo precio sea mayor al promedio.

```
SELECT Titulo, Precio
FROM Libros
WHERE Precio > (
    SELECT AVG(Precio) FROM Libros
);
```

**Resultado (suponiendo datos de ejemplo):**

Titulo	Precio
Rayuela	600
Cien Años de Soledad	700

La subconsulta calcula el promedio de precios, y la consulta principal filtra con ese valor.

## 4. Vistas (VIEW)

Las vistas son como “consultas guardadas” que se pueden reutilizar como si fueran tablas.

Sirven para simplificar consultas complejas y dar acceso seguro a ciertos datos.

Ejemplo: crear una vista con el total de libros por autor.

```
CREATE VIEW LibrosPorAutor AS
SELECT AutorID, COUNT(*) AS CantidadLibros
FROM Libros
GROUP BY AutorID;
```

Ahora podemos usarla como si fuera una tabla:

```
SELECT * FROM LibrosPorAutor;
```

**Resultado:**

AutorID	CantidadLibros
1	1
2	2

## 5. Ejercicio

1. Hacer un INNER JOIN entre **Autores** y **Libros** para mostrar el nombre del autor y el título de cada libro.
2. Hacer un LEFT JOIN que muestre todos los autores, incluso los que no tienen libros.
3. Crear una subconsulta que muestre los libros más caros que el promedio de precios.
4. Crear una vista llamada **AutoresConLibros** que muestre el nombre del autor y la cantidad de libros que tiene.