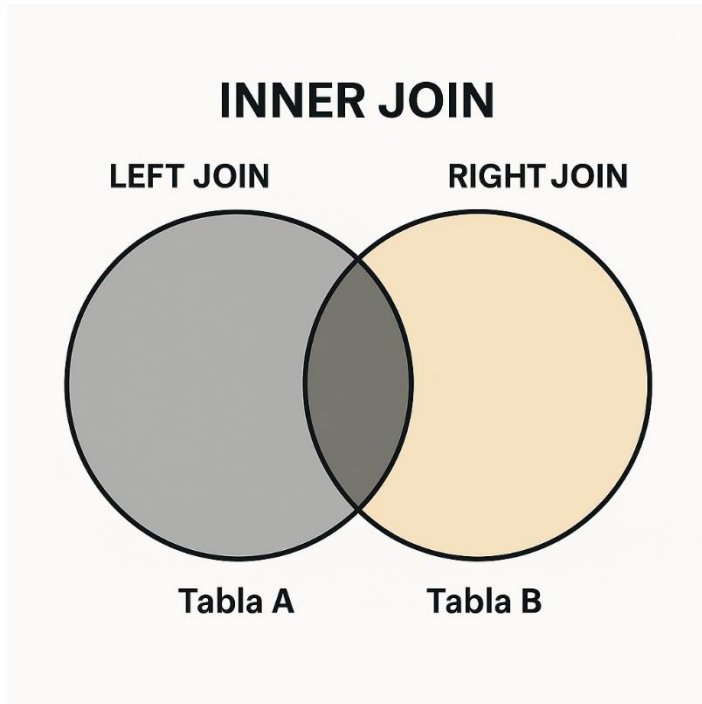


# TIPOS DE JOIN en MYSQL



## TABLAS DE EJEMPLO:

Dos tablas muy comunes, fáciles de entender:

### Clientes

id_cliente	nombre
1	Ana
2	Luis
3	Marta

### Pedidos

id_pedido	id_cliente	producto
101	1	Teclado
102	2	Ratón
103	4	Monitor

## 1. INNER JOIN

Devuelve solo las filas que tienen coincidencias en ambas tablas.

```
SELECT c.nombre, p.producto
FROM Clientes c
INNER JOIN Pedidos p ON c.id_cliente = p.id_cliente;
```

Resultado:

nombre	producto
Ana	Teclado
Luis	Ratón

**Contexto:** Cuando queremos ver solo los pedidos realizados por clientes registrados.

## 2. LEFT JOIN (LEFT OUTER JOIN)

Devuelve **todos los clientes**, y si tienen pedidos, también se muestra.

```
SELECT c.nombre, p.producto
FROM Clientes c
LEFT JOIN Pedidos p ON c.id_cliente = p.id_cliente;
```

Resultado:

nombre	producto
Ana	Teclado
Luis	Ratón
Marta	NULL

**Contexto:** Ideal para mostrar todos los clientes, aunque no hayan hecho pedidos.

Cuando usamos un **LEFT JOIN**, estamos diciendo: “muéstrame **todos los registros de la tabla de la izquierda**, aunque no tengan correspondencia en la tabla de la derecha”.

## 3. RIGHT JOIN (RIGHT OUTER JOIN)

Devuelve **todos los pedidos**, incluso si no están vinculados a un cliente.

```
SELECT c.nombre, p.producto
FROM Clientes c
RIGHT JOIN Pedidos p ON c.id_cliente = p.id_cliente;
```

Resultado:

nombre	producto
Ana	Teclado

Luis	Ratón
NULL	Monitor

**Contexto:** Cuando queremos ver todos los pedidos, incluso los de clientes no registrados (ej. importados o antiguos).

Cuando usamos un **RIGHT JOIN** decimos: “muéstrame **todos los registros de la tabla de la derecha**, aunque no tengan correspondencia en la tabla de la izquierda”.

#### 4. FULL JOIN (NO DISPONIBLE DIRECTAMENTE EN MYSQL)

No está soportado de forma nativa en MySQL, pero se puede simular con UNION:

```
SELECT c.nombre, p.producto
FROM Clientes c
LEFT JOIN Pedidos p ON c.id_cliente = p.id_cliente
UNION
SELECT c.nombre, p.producto
FROM Clientes c
RIGHT JOIN Pedidos p ON c.id_cliente = p.id_cliente;
```

Resultado:

nombre	producto
Ana	Teclado
Luis	Ratón
Marta	NULL
NULL	Monitor

**Contexto:** Para informes completos que incluyan todos los clientes y todos los pedidos, coincidan o no.

### EJERCICIO: JOIN entre Profesores y Cursos

**Tabla 1: Profesores**

id_prof	nombre
1	Marta
2	Javier
3	Lucía

**Tabla 2: Cursos**

id_curso	id_prof	curso
101	1	Redes
102	4	Seguridad
103	2	Bases de Datos

## PREGUNTAS

1. Realiza un **INNER JOIN** entre `Profesores` y `Cursos`, y explica qué profesores aparecen y por qué.
2. Haz un **LEFT JOIN** desde `Profesores` y observa qué aparece para Lucía.
3. Realiza un **RIGHT JOIN** desde `Cursos` y explica qué ocurre con el curso "Seguridad".
4. ¿Qué valor aparece como `NULL` y en qué consulta? ¿Qué significa?

## SOLUCIONES Y EXPLICACIONES

### 1. INNER JOIN

```
SELECT p.nombre, c.curso
FROM Profesores p
INNER JOIN Cursos c ON p.id_prof = c.id_prof;
```

**Resultado:**

nombre	curso
Marta	Redes
Javier	Bases de Datos

**Explicación:**

- Solo aparecen los profesores que tienen cursos asignados con coincidencia exacta (`id_prof`).
- Lucía no tiene ningún curso → no aparece.
- El curso "Seguridad" fue asignado a `id_prof = 4`, que no existe → no aparece.

### 2. LEFT JOIN

```
SELECT p.nombre, c.curso
FROM Profesores p
LEFT JOIN Cursos c ON p.id_prof = c.id_prof;
```

**Resultado:**

nombre	curso
Marta	Redes
Javier	Bases de Datos
Lucía	NULL

**Explicación:**

- Muestra todos los profesores.
- Lucía aparece con `curso = NULL` porque no tiene cursos asignados.

- El curso "Seguridad" sigue sin mostrarse porque no hay profesor con `id_prof = 4`.

### 3. RIGHT JOIN

```
SELECT p.nombre, c.curso
FROM Profesores p
RIGHT JOIN Cursos c ON p.id_prof = c.id_prof;
```

#### Resultado:

nombre	curso
Marta	Redes
Javier	Bases de Datos
NULL	Seguridad

#### Explicación:

- Muestra todos los cursos.
- El curso "Seguridad" aparece aunque no tenga un profesor asociado.
- `nombre = NULL` indica que no hay profesor en la tabla `Profesores` con `id_prof = 4`.

### 4. ¿Qué valor aparece como NULL y qué significa?

- En el **LEFT JOIN**, `curso = NULL` para Lucía → no tiene curso asignado.
- En el **RIGHT JOIN**, `nombre = NULL` para el curso "Seguridad" → no hay profesor registrado para ese curso.
- Los `NULL` señalan **falta de coincidencia** en la tabla contraria al tipo de JOIN.