

INNER JOIN, LEFT JOIN, RIGHT JOIN

Base de datos oficina

Empleados

	id	nombre	despacho	departamento	salario
	1	Juan	NULL	2	1200.00
	3	Sergio	NULL	1	1200.00
	5	Elena	DP02	4	1700.00
▶	7	Sara	NULL	4	1300.00
	8	Martín	DP03	1	1300.00
	10	Pedro	NULL	2	1200.00
	11	Leopoldo	DP04	5	1700.00
	12	Rafa	NULL	NULL	1200.00
	13	Willy	NULL	NULL	1200.00
✱	NULL	NULL	NULL	NULL	NULL

Departamentos

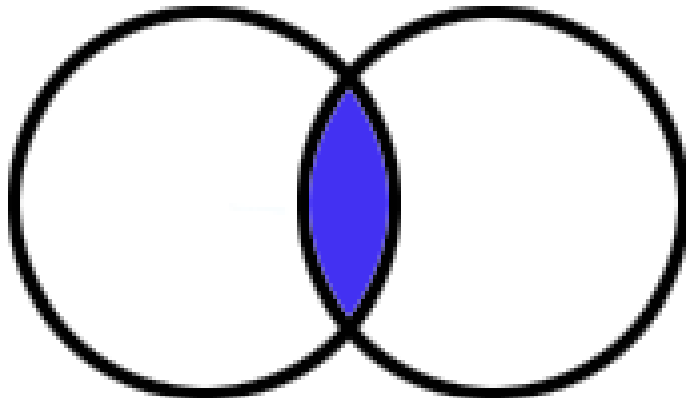
	id	nombre
▶	1	Matemáticas
	2	Inglés
	3	Informática
	4	Dibujo
	5	Filosofía
✱	NULL	NULL

Cláusula INNER JOIN

Lo más usual, lo primero que se suele aprender, es el uso de **INNER JOIN**, o generalmente abreviado como **JOIN**.

Esta cláusula busca coincidencias entre 2 tablas, en función a una columna que tienen en común. **Sólo la intersección se mostrará en los resultados.**

Tabla empleado Tabla departamento



Estos no aparecen porque no tienen departamento

12	Rafa	NULL	1200.00
13	Willy	NULL	1200.00

	id	nombre	despacho	departamento	salario	id	nombre
▶	3	Sergio	NULL	1	1200.00	1	Matemáticas
	8	Martín	DP03	1	1200.00	1	Matemáticas
	1	Juan	NULL	2	1200.00	2	Inglés
	10	Pedro	NULL	2	1200.00	2	Inglés
	5	Elena	DP02	4	1700.00	4	Dibujo
	7	Sara	NULL	4	1300.00	4	Dibujo
	11	Leopoldo	DP04	5	1700.00	5	Filosofía

```
SELECT * FROM empleado e INNER JOIN departamento d ON e.departamento = d.id
```

Cláusula LEFT JOIN

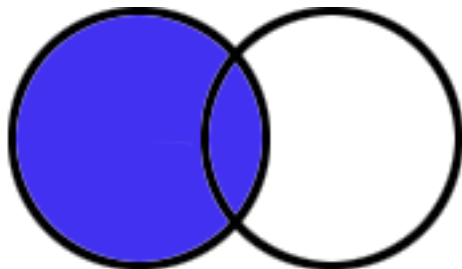
A diferencia de un **INNER JOIN**, donde se busca una intersección respetada por ambas tablas, con **LEFT JOIN** damos prioridad a la tabla de la izquierda, y buscamos en la tabla derecha.

Si no existe ninguna coincidencia para alguna de las filas de la tabla de la izquierda, de igual forma **todos los resultados de la primera tabla se muestran**.

```
SELECT * FROM empleado e LEFT JOIN departamento d on e.departamento = d.Id;
```

Tabla empleado

Tabla departamento



	id	nombre	despacho	departamento	salario	id	nombre
▶	1	Juan	NULL	2	1200.00	2	Inglés
	3	Sergio	NULL	1	1200.00	1	Matemáticas
	5	Elena	DP02	4	1700.00	4	Dibujo
	7	Sara	NULL	4	1300.00	4	Dibujo
	8	Martín	DP03	1	1300.00	1	Matemáticas
	10	Pedro	NULL	2	1200.00	2	Inglés
	11	Leopoldo	DP04	5	1700.00	5	Filosofía
	12	Rafa	NULL	NULL	1200.00	NULL	NULL
	13	Willy	NULL	NULL	1200.00	NULL	NULL

En este caso aparecen los dos últimos registros aunque no tengan departamento

Cláusula *RIGHT JOIN*

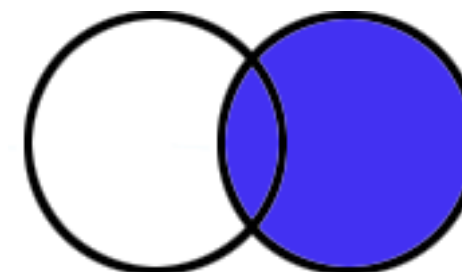
En el caso de **RIGHT JOIN** la situación es muy similar, pero aquí se da prioridad a la tabla de la derecha.

```
SELECT * FROM empleado e RIGHT JOIN departamento d on e.departamento = d.id;
```

	id	nombre	despacho	departamento	salario	id	nombre
▶	3	Sergio	NULL	1	1200.00	1	Matemáticas
	8	Martín	DP03	1	1300.00	1	Matemáticas
	1	Juan	NULL	2	1200.00	2	Inglés
	10	Pedro	NULL	2	1200.00	2	Inglés
	NULL	NULL	NULL	NULL	NULL	3	Informática
	5	Elena	DP02	4	1700.00	4	Dibujo
	7	Sara	NULL	4	1300.00	4	Dibujo
	11	Leopoldo	DP04	5	1700.00	5	Filosofía

Tabla empleado

Tabla departamento



En este caso aparecen los departamentos que no tengan empleados