JOIN

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

¿Por qué usar JOIN?

Además de realizar consultas dentro de una tabla y de haber empleado **table reference** para consultas en múltiples tablas, existe la herramienta **JOIN** que nos permite hacer consultas a distintas tablas y unir los resultados.

Ventajas del uso de **JOIN**:

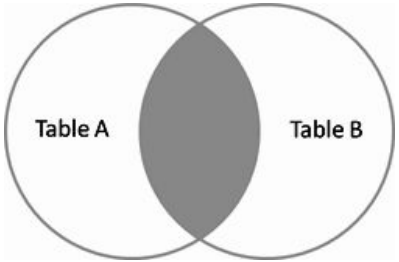
- Su sintaxis es mucho más comprensible.
- Presentan una mejor performance.
- Proveen de ciertas flexibilidades.

INNER JOIN

El **INNER JOIN** es la opción predeterminada y nos devuelve **todos los registros** donde se **cruzan dos o más tablas**. Por ejemplo, si tenemos una tabla cliente y otra factura, al cruzarlas con **INNER JOIN**, nos devuelve aquellos registros o filas donde haya un valor coincidente en ambas tablas.

cliente		
id	**nombre**	**apellido**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

INNER JOIN



factura		
id	**cliente_id**	**fecha**
1	2	12/03/2019
2	2	22/08/2019
3	1	04/09/2019

Consulta a múltiples tablas

Antes con **table reference** escribíamos:

```sql
SELECT cliente.id, cliente.nombre, factura.fecha
FROM cliente, factura;
```

Ahora con **INNER JOIN** escribimos:

```sql
SELECT cliente.id, cliente.nombre, factura.fecha
FROM cliente
INNER JOIN factura;
```

Si bien ya dimos el primer paso (que es **cruzar** ambas tablas), aún nos falta aclarar **dónde** está ese cruce.

Es decir, qué **clave primaria (PK)** se cruzará con qué **clave foránea (FK)**.

Definiendo el INNER JOIN

Para definir el **INNER JOIN** tenemos que indicar el filtro por el cual se **evaluará** el **cruce**. Para esto, debemos utilizar la palabra reservada **ON**. Es decir, que lo que antes escribíamos en el **WHERE** de table reference, ahora lo escribiremos en el **ON** de INNER JOIN.

SQL

```sql
SELECT cliente.id, cliente.nombre, factura.fecha
FROM cliente
INNER JOIN factura
ON cliente.id = factura.cliente_id;
```


DigitalHouse>
Coding School