JOINS

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

¿Por qué usar JOINS?

Además de hacer consultas dentro de una tabla o hacia muchas tablas a través de **table reference**, también es posible y necesario hacer consultas a **distintas tablas** y unir esos resultados con **JOINS**.

Si bien cumplen la misma función que **table reference**, los **JOINS:**

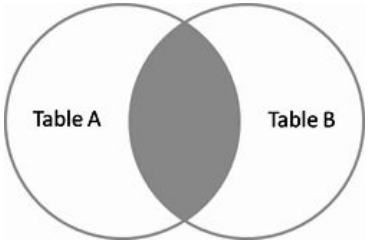
- Proveen ciertas flexibilidades adicionales.
- Su sintaxis es mucho más utiliza.
- Presentan una mejor performance.

INNER JOIN

El **INNER JOIN** hará una **cruza** entre dos tablas. Si cruzáramos las tablas de **clientes** y **ventas** y hubiese algún cliente **sin ventas**, el INNER JOIN **no traería** a ese cliente como resultado.

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

INNER JOIN



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

Creando un INNER JOIN

Antes *escribíamos*:

```SQL
SELECT clientes.id AS id, clientes.nombre, ventas.fecha
FROM clientes, ventas
```

Ahora *escribiremos*:

```SQL
SELECT clientes.id AS id, clientes.nombre, ventas.fecha
FROM clientes
INNER JOIN ventas
```

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)**.

Creando un INNER JOIN *(cont.)*

La sintaxis del JOIN **no utiliza** el **WHERE** si no que **requiere** la palabra **ON**. Es ahí en donde indicaremos el **filtro** a tener en cuenta para realizar el cruce.

Es decir, que lo que antes escribíamos en el **WHERE** ahora lo escribiremos en el **ON**.

SQL

```sql
SELECT clientes.id AS id, clientes.nombre, ventas.fecha
FROM clientes
INNER JOIN ventas
ON clientes.id = ventas.cliente_id
```


DigitalHouse>
Coding School