



# JOIN



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



## Por que usar JOIN?

Além de fazer consultas dentro de uma tabela e ter usado uma **referência de tabela** para consultas em várias tabelas, existe a ferramenta **JOIN** que nos permite consultar diferentes tabelas e juntar os resultados.

Vantagens de usar **JOIN**:

- Sua sintaxe é muito mais compreensível.
- Apresentam um melhor desempenho.
- Fornecem certas flexibilidades.



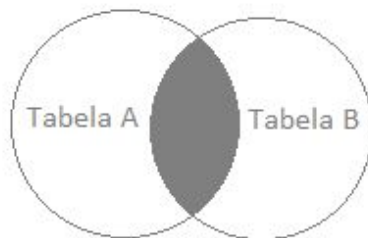


# INNER JOIN

O **INNER JOIN** é a opção padrão e retorna todos os registros onde duas ou mais tabelas se cruzam. Por exemplo, se temos uma tabela de cliente e outra fatura, ao cruzá-las com **INNER JOIN**, retorna aqueles registros ou linhas onde há um valor correspondente em **ambas** as tabelas.

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

## INNER JOIN



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



# Consulta a múltiplas tabelas

Antes, com **referência à tabela**, escrevíamos:

```
SQL SELECT cliente.id, cliente.nome, fatura.data  
FROM cliente, fatura;
```

Agora, com **INNER JOIN** escrevemos:

```
SQL SELECT cliente.id, cliente.nome, fatura.data  
FROM cliente  
INNER JOIN fatura;
```

“

Embora já tenhamos dado o primeiro passo (que é cruzar duas tabelas), ainda precisamos esclarecer onde fica esse **cruzamento**.

Ou seja, qual **chave primária** (PK) será cruzada com qual **chave estrangeira** (FK).



”



## Definindo o INNER JOIN

Para definir o **INNER JOIN**, temos que indicar o filtro pelo qual a junção será avaliada. Para isso, devemos usar a palavra reservada **ON**. Ou seja, o que escrevíamos no **WHERE** da referência de tabela, agora vamos escrever no **ON** do **INNER JOIN**.

SQL

```
SELECT cliente.id, cliente.nome, fatura.data  
FROM cliente  
INNER JOIN fatura  
ON cliente.id = fatura.cliente_id;
```

DigitalHouse>  
Coding School