

# JOINS

O JOIN é uma operação utilizada em bancos de dados relacionais para combinar dados de duas ou mais tabelas com base em uma ou mais colunas em comum. A utilização de joins permite que informações de diferentes tabelas sejam combinadas em uma única consulta, tornando possível a realização de consultas mais complexas e sofisticadas.

Existem vários tipos de JOIN disponíveis em SQL, sendo os mais comuns:

## INNER JOIN

O INNER JOIN retorna apenas os registros que possuem correspondência nas duas tabelas envolvidas. Em outras palavras, o INNER JOIN retorna apenas as linhas onde há uma correspondência entre as chaves primárias e estrangeiras das tabelas.



Fonte: Autoria própria

Exemplo:

Suponha que temos duas tabelas "Pedidos" e "Clientes", e queremos encontrar todos os pedidos feitos pelos clientes da cidade de São Paulo:

```
SELECT  p.Pedido,
        c.Nome
FROM    Pedidos p
        INNER JOIN Clientes c
        ON p.ClienteID = c.ClienteID
WHERE   c.Cidade = 'São Paulo';
```

Poderia retornar, por exemplo:

| Pedido | Nome | |-----:|-----:| | 1 | Vinicius | | 2 | Maria |

## LEFT JOIN

O LEFT JOIN retorna todos os registros da tabela à esquerda do JOIN (a tabela que aparece antes do JOIN na consulta) e os registros correspondentes da tabela à direita do JOIN. Se não houver correspondência, os campos da tabela à direita serão nulos.



Fonte: Autoria própria

Exemplo:

Suponha que temos duas tabelas "Pedidos" e "Clientes", e queremos listar todos os pedidos, mesmo que não tenham clientes associados:

```
SELECT  p.Pedido,
        c.Nome
FROM    Pedidos p
        LEFT JOIN Clientes c
        ON p.ClienteID = c.ClienteID
```

Retornaria, por exemplo:

| Pedido | Nome | | :-----: | :-----: | | 1 | Vinicius | | 2 | Maria | | 3 | NULL |

## RIGHT JOIN

O RIGHT JOIN retorna todos os registros da tabela à direita do JOIN (a tabela que aparece depois do JOIN na consulta) e os registros correspondentes da tabela à esquerda do JOIN. Se não houver correspondência, os campos da tabela à esquerda serão nulos.



Fonte: Autoria própria

Exemplo:

Suponha que temos duas tabelas "Pedidos" e "Clientes", e queremos listar todos os clientes, mesmo que não tenham pedidos associados:

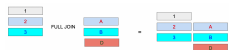
```
SELECT  p.Pedido,
        c.Nome
FROM    Pedidos p
        RIGHT JOIN Clientes c
        ON p.ClienteID = c.ClienteID
```

Essa consulta, retornaria, por exemplo:

| Pedido | Nome | |:-----:|:-----:| | 1 | Vinicius | | 2 | Maria | | *NULL* | Marcos Pitter |

## FULL OUTER JOIN

O FULL OUTER JOIN retorna todos os registros de ambas as tabelas envolvidas no JOIN, incluindo os registros que não possuem correspondência.



Exemplo:

Suponha que temos duas tabelas "Pedidos" e "Clientes", e queremos listar todos os pedidos e todos os clientes, mesmo que não haja correspondência:

```
SELECT  p.Pedido,
        c.Nome
FROM    Pedidos p
        FULL JOIN Clientes c
        ON p.ClienteID = c.ClienteID
```

A qual retornaria:

| Pedido | Nome | |:-----:|:-----:| | 1 | Vinicius | | 2 | Maria | | 3 | *NULL* | | *NULL* | Marcos Pitter |

Os joins são uma ferramenta poderosa para combinar dados de diferentes tabelas em consultas SQL, permitindo a realização de consultas mais complexas e sofisticadas. Cada tipo de JOIN tem suas próprias características e pode ser escolhido de acordo com as necessidades específicas da consulta em questão.

## Referências e materiais complementares

<https://www.postgresql.org/docs/current/tutorial-join.html>

Próximo Tópico