

Tipos de JOIN

O comando **JOIN** é usado para combinar linhas de duas ou mais tabelas com base em uma coluna relacionada entre elas. A forma como essa combinação é feita depende do tipo de **JOIN** que você usa.

- **INNER JOIN**: É o tipo de **JOIN** mais comum. Ele retorna apenas as linhas que têm correspondência **em ambas as tabelas**. Se uma linha em uma tabela não tiver uma correspondência na outra, ela simplesmente não será incluída no resultado.
 - **LEFT JOIN** (ou **LEFT OUTER JOIN**): Retorna **todas as linhas da tabela da esquerda**, e as linhas correspondentes da tabela da direita. Se não houver correspondência na tabela da direita, os valores para essa linha serão **NULL**. É útil quando você quer ver tudo de uma tabela, mesmo que não haja dados relacionados na outra.
 - **RIGHT JOIN** (ou **RIGHT OUTER JOIN**): Funciona como o **LEFT JOIN**, mas ao contrário. Retorna **todas as linhas da tabela da direita**, e as linhas correspondentes da tabela da esquerda. Se não houver correspondência na tabela da esquerda, os valores para essa linha serão **NULL**.
 - **FULL JOIN** (ou **FULL OUTER JOIN**): Retorna **todas as linhas de ambas as tabelas**. Se uma linha em uma tabela não tiver correspondência na outra, os valores correspondentes serão **NULL**. É o tipo de **JOIN** mais abrangente, unindo tudo o que existe nas duas tabelas.
-

Cheatsheet do Comando JOIN

Sintaxe Geral

SQL

SELECT colunas

FROM tabela_a

TIPO_DE_JOIN tabela_b ON tabela_a.coluna_em_comum = tabela_b.coluna_em_comum;

- **FROM tabela_a**: A tabela principal (ou "esquerda" nos joins).
- **TIPO_DE_JOIN tabela_b**: O tipo de **JOIN** que você está usando e a segunda tabela (a "direita").
- **ON tabela_a.coluna = tabela_b.coluna**: A condição que define a relação entre as tabelas, geralmente as chaves primárias e estrangeiras.

Exemplos Práticos com Cenário de E-commerce

Imagine que temos duas tabelas:

- **clientes** (colunas: **id**, **nome**)
- **pedidos** (colunas: **id**, **cliente_id**, **data**)

1. INNER JOIN

- **Missão:** Listar os clientes que fizeram pedidos.
- **Lógica:** Trazer apenas os clientes que têm um **id** correspondente na tabela **pedidos**.

SQL

```
SELECT c.nome, p.data  
FROM clientes c  
INNER JOIN pedidos p ON c.id = p.cliente_id;
```

- **Resultado:** Apenas clientes com pedidos aparecerão. Se um cliente não tiver feito um pedido, ele não estará na lista.

2. LEFT JOIN

- **Missão:** Listar todos os clientes, mostrando seus pedidos se existirem.
- **Lógica:** Trazer todos os clientes, independentemente de terem um pedido. Se não houver pedido, a data do pedido será **NULL**.

SQL

```
SELECT c.nome, p.data  
FROM clientes c  
LEFT JOIN pedidos p ON c.id = p.cliente_id;
```

- **Resultado:** Todos os clientes serão listados. Para os clientes sem pedidos, a coluna **data** mostrará o valor **NULL**.

3. RIGHT JOIN

- **Missão:** Listar todos os pedidos e seus respectivos clientes, mesmo que haja pedidos sem um cliente associado.
- **Lógica:** Trazer todos os pedidos, independentemente de terem um cliente. Se não houver cliente, o nome do cliente será **NULL**.

SQL

```
SELECT c.nome, p.data
```

```
FROM clientes c
RIGHT JOIN pedidos p ON c.id = p.cliente_id;
```

- **Resultado:** Todos os pedidos serão listados. Para pedidos sem um `cliente_id` válido, a coluna `nome` mostrará `NULL`.

4. FULL JOIN

- **Missão:** Listar todos os clientes e todos os pedidos, independentemente de terem uma correspondência.
- **Lógica:** Trazer a união completa das duas tabelas.

```
SQL
SELECT c.nome, p.data
FROM clientes c
FULL JOIN pedidos p ON c.id = p.cliente_id;
```

- **Resultado:** A lista incluirá clientes sem pedidos (com a coluna `data` como `NULL`), pedidos sem clientes (com a coluna `nome` como `NULL`), e as combinações de clientes com pedidos.

Cenário de JOIN com 3 Tabelas: Sistema de Biblioteca

Imagine que você quer responder a uma pergunta específica: **"Quais são os títulos dos livros que a Maria pegou emprestado?"**

Para responder a essa pergunta, você precisa de informações de três tabelas diferentes:

1. **membros:** Para encontrar o ID do membro chamado "Maria".
2. **emprestimos:** Para encontrar os IDs dos livros que o membro com o ID da Maria pegou.
3. **livros:** Para encontrar os títulos dos livros com base nos IDs que você encontrou na tabela de **emprestimos**.

Vamos montar a consulta passo a passo:

Passo 1: O JOIN inicial

Primeiro, você precisa conectar a tabela **membros** com a tabela **emprestimos**. A coluna em comum é o ID do membro, que aparece como **id** em **membros** e como **membro_id** em **emprestimos**.

```
SQL
SELECT *
FROM membros m
INNER JOIN emprestimos e ON m.id = e.membro_id;
```

- **m** e **e** são apelidos para as tabelas, o que torna a consulta mais curta e fácil de ler.

Passo 2: Adicionando a terceira tabela

Agora, você tem a conexão entre membros e seus empréstimos. O próximo passo é juntar a tabela **livros** para obter os títulos. A coluna em comum é o ID do livro, que aparece como **livro_id** em **emprestimos** e como **id** em **livros**.

Para isso, basta adicionar um segundo **INNER JOIN**:

```
SQL
SELECT *
FROM membros m
INNER JOIN emprestimos e ON m.id = e.membro_id
INNER JOIN livros l ON e.livro_id = l.id;
```

Com essa consulta, você já tem acesso a todas as colunas das três tabelas.

Passo 3: Adicionando o filtro e selecionando as colunas

Para responder à pergunta inicial, você precisa filtrar os resultados para o membro "Maria" e selecionar apenas as colunas relevantes (**nome** do membro e **título** do livro). Para isso, adicione a cláusula **WHERE** e especifique as colunas no **SELECT**.

```
SQL
SELECT m.nome, l.titulo
FROM membros m
INNER JOIN emprestimos e ON m.id = e.membro_id
INNER JOIN livros l ON e.livro_id = l.id
WHERE m.nome = 'Maria';
```

Exemplo de Aplicação Real

Aqui estão os dados de exemplo que você pode usar para testar a consulta:

Tabela **membros** | id | nome | |----|-----| | 1 | João | | 2 | Maria | | 3 | Pedro |

Tabela **livros** | id | titulo | |----|-----| | 10 | A Arte da Guerra | | 11 | O Senhor dos Anéis | | 12 | 1984 |

Tabela **emprestimos** | id | membro_id | livro_id | |----|-----|-----| | 1 | 2 | 10 | | 2 | 2 | 12 | | 3 | 1 | 11 |

Quando você executar a consulta completa com a cláusula **WHERE**, o resultado será:

nome	titulo
Maria	A Arte da Guerra
Maria	1984