Tipos de JOIN

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Temas

1 INNER

2 LEFT

3 RIGHT

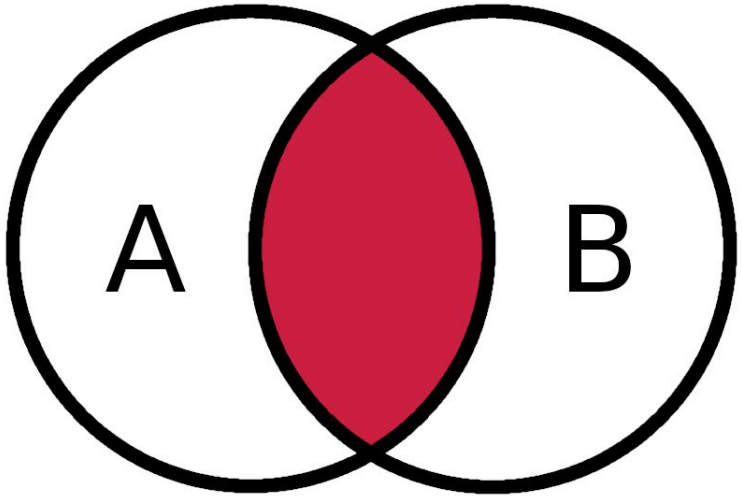
1 | INNER

INNER JOIN

O **INNER JOIN** entre duas tabelas retorna apenas os registros que atendem à condição indicada na cláusula **ON.**

SQL

```sql
SELECT coluna1, coluna2, ...
FROM tabela A
INNER JOIN tabela B
ON condicao
```

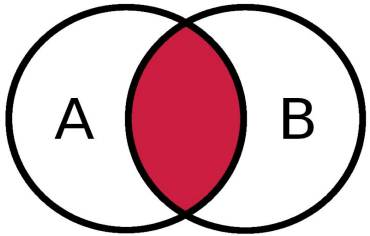


INNER JOIN

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 clientes e outra faturas, ao cruzá-las com **INNER JOIN**, retorna aqueles registros ou linhas onde há um valor correspondente em ambas as tabelas.

clientes		
id	**nome**	**sobrenome**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

INNER JOIN



faturas		
id	**cliente_id**	**data**
11	2	12/09/2019
12	**null**	20/09/2019
13	1	24/09/2019

INNER JOIN

O exemplo anterior pode ser construído da seguinte forma:

```sql
SELECT faturas.id AS nr_fatura, sobrenome, nome, data
FROM clientes
INNER JOIN faturas
ON clientes.id = faturas.cliente_id;
```

Os dados obtidos são apresentados a seguir:

nr_fatura	sobrenome	nome	data
11	Sanchez	Clara	12/09/2019
13	Perez	Juan	24/09/2019

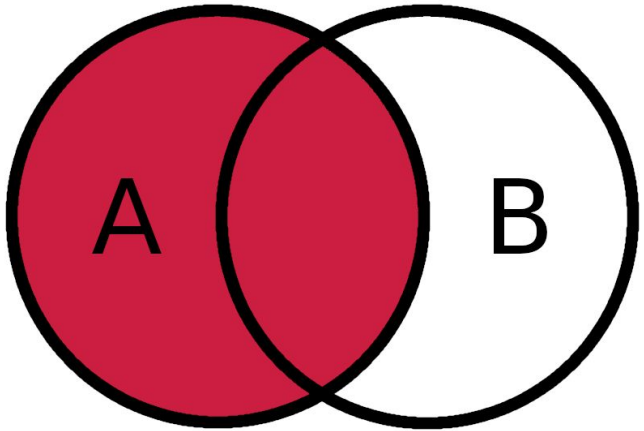
2 | LEFT

LEFT JOIN

O **LEFT JOIN** entre duas tabelas retorna todos os registros da primeira tabela (neste caso seria a tabela A), mesmo quando os registros não atendam à condição indicada na cláusula **ON**.

SQL

```
SELECT coluna1, coluna2, ...
FROM tabela A
LEFT JOIN tabela B
ON condicao
```

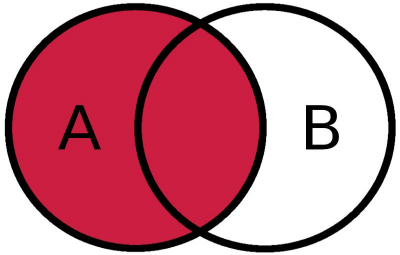


LEFT JOIN

Portanto, **LEFT JOIN** retorna todos os registros onde duas ou mais **tabelas** se **cruzam**. Mesmo se os registros de uma primeira tabela (A) **não atendam** à condição indicada na cláusula **ON**. Por exemplo, se tivermos uma tabela de clientes e outra faturas, ao cruzá-las, ele retorna aqueles registros onde há um valor de casamento entre as duas, mais os registros daqueles clientes que não possuem uma fatura atribuída.

clientes		
id	**nome**	**sobrenome**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

LEFT JOIN



faturas		
id	**cliente_id**	**data**
11	2	12/09/2019
12	**null**	20/09/2019
13	1	24/09/2019

LEFT JOIN

O exemplo anterior pode ser construído da seguinte forma:

```sql
SELECT faturas.id AS nr_factura, sobrenome, nome, data
FROM clientes
LEFT JOIN faturas
ON clientes.id = faturas.cliente_id;
```

Os dados obtidos são apresentados a seguir:

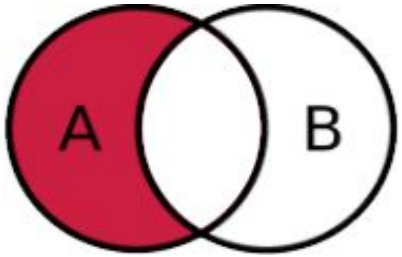
nr_fatura	sobrenome	nome	data
11	Sanchez	Clara	12/09/2019
13	Perez	Juan	24/09/2019
null	García	Marta	**null**

LEFT Excluding JOIN

Este tipo de **LEFT JOIN** retorna apenas os registros de uma primeira tabela (A), excluindo os registros que atendem à condição indicada na cláusula **ON**. Por exemplo, se tivermos uma tabela de clientes e outra faturas, ao cruzá-las, ele retorna apenas os registros de **clientes** que não possuem **nota fiscal** atribuída.

clientes		
id	**nome**	**sorenome**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

LEFT Excluding JOIN



faturas		
id	**cliente_id**	**data**
11	2	12/09/2019
12	**null**	20/09/2019
13	1	24/09/2019

LEFT Excluding JOIN

Continuando com o exemplo, ele poderia ser construído da seguinte forma:

```sql
SELECT faturas.id AS nr_fatura, sobrenome, nome, data
FROM clientes
LEFT JOIN faturas
ON clientes.id = faturas.cliente_id
WHERE ISNULL(faturas.id);
```

Os dados obtidos são apresentados a seguir:

nr_fatura	sobrenome	nome	data
null	García	Marta	**null**

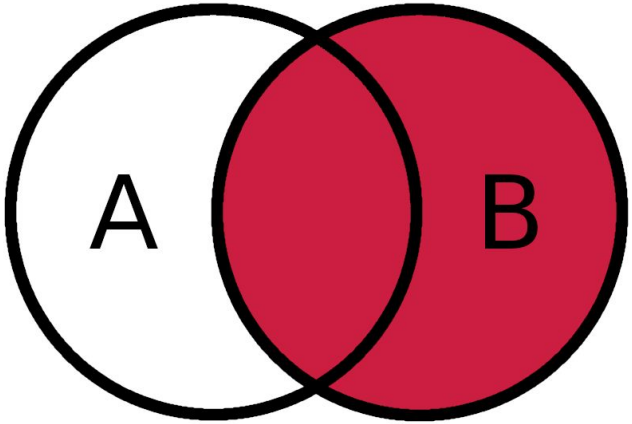
3 | RIGHT

RIGHT JOIN

O **RIGHT JOIN** entre duas tabelas retorna todos os registros da segunda tabela, mesmo quando os registros não atendem à condição indicada na cláusula **ON**.

SQL

```
SELECT coluna1, coluna2, ...
FROM tabela A
RIGHT JOIN tabela B
ON condicao
```

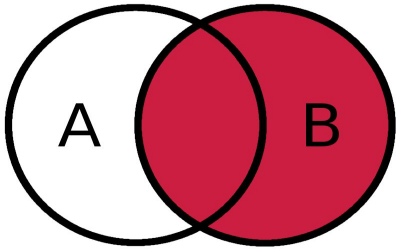


RIGHT JOIN

Portanto, **RIGHT JOIN** retorna todos os registros onde duas ou mais tabelas se **cruzam**. Mesmo se os registros de uma segunda tabela (B) não atendam à condição indicada na cláusula **ON**. Por exemplo, se tivermos uma tabela de clientes e outra faturas, ao cruzá-las, retorna aqueles registros onde há um valor de casamento entre as duas, mais os registros daquelas faturas que não têm um cliente atribuído.

clientes		
id	**nome**	**sobrenome**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

RIGHT JOIN



faturas		
id	**cliente_id**	**data**
11	2	12/09/2019
12	**null**	20/09/2019
13	1	24/09/2019

RIGHT JOIN

O exemplo anterior pode ser construído da seguinte forma:

```sql
SELECT faturas.id AS nr_fatura, sobrenome, nome, data
FROM clientes
RIGHT JOIN faturas
ON clientes.id = faturas.cliente_id;
```

Os dados obtidos são apresentados a seguir:

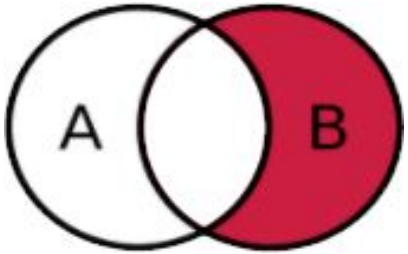
nr_fatura	sobrenome	nome	data
11	Sanchez	Clara	12/09/2019
12	**null**	**null**	20/09/2019
13	Perez	Juan	24/09/2019

RIGHT Excluding JOIN

Este tipo de **RIGTH JOIN** retorna apenas os registros de uma segunda tabela (B), excluindo os registros que atendem à condição indicada na cláusula **ON**. Por exemplo, se tivermos uma tabela de clientes e outra faturas, ao cruzá-las, ele retorna apenas aqueles registros de nota fiscal que **não têm** um cliente atribuído.

clientes		
id	**nome**	**sobrenome**
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

RIGHT Excluding JOIN



faturas		
id	**cliente_id**	**data**
11	2	12/09/2019
12	**null**	20/09/2019
13	1	24/09/2019

RIGHT Excluding JOIN

Continuando com o exemplo, ele poderia ser construído da seguinte forma:

```sql
SELECT faturas.id AS nr_fatura, sobrenome, nome, data
FROM clientes
RIGHT JOIN faturas
ON clientes.id = faturas.cliente_id
WHERE ISNULL(clientes.id);
```

Os dados obtidos são apresentados a seguir:

nr_fatura	sobrenome	nome	data
12	**null**	**null**	20/09/2019


DigitalHouse>
Coding School