

# Operações Join

Usamos as operações JOIN para relacionar dados de duas ou mais tabelas em uma consulta, utilizando igualdade de colunas em comum ou não. Será mostrado um exemplo de situação onde pode ser usado os diferentes tipos de JOIN e seus resultados entre duas tabelas.

## TABELAS EXEMPLO:

**EX\_ESTADO**

UF	DESCRICAO
SC	Santa Catarina
SP	São Paulo

### SQL de CRIAÇÃO/INSERÇÃO:

```
CREATE TABLE ex_estado (uf VARCHAR(2), descricao TEXT);
INSERT INTO ex_estado VALUES ('SC', 'Santa Catarina'),
                              ('SP', 'São Paulo');
COMMIT;
```

**EX\_CIDADE**

IDCIDADE	NOME	UF
1	Criciúma	SC
2	Florianópolis	SC
3	Curitiba	PR

SQL de CRIAÇÃO/INSERÇÃO:

```
CREATE TABLE ex_cidade (idcidade INTEGER, nome TEXT, uf VARCHAR(2));  
INSERT INTO ex_cidade VALUES (1, 'Criciúma', 'SC'),  
                                (2, 'Florianópolis', 'SC'),  
                                (3, 'Curitiba', 'PR');  
  
COMMIT;
```

# INNER JOIN

image-1646740796923.jpg

Image not found or type unknown

É usado para relacionar e mostrar os dados de ambas as tabelas quando houver igualdade nos campos em comum.

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES  
INNER JOIN EX_CIDADE CI ON ES.UF = CI.UF
```

Usos alternativos que retornam o mesmo resultado:

Não é necessário informar a operação **INNER**.

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES  
JOIN EX_CIDADE CI ON ES.UF = CI.UF
```

Utilizando com a operação **NATURAL**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES  
NATURAL JOIN EX_CIDADE CI
```

Utilizando com a cláusula **USING**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas

```
SELECT es.descricao as estado,  
       ci.nome as cidade  
FROM ex_estado es  
JOIN ex_cidade ci USING (uf)
```

Utilizando com a cláusula **WHERE**.

```
SELECT ES.DESCRICAO AS ESTADO,
```

```
CI. NOME AS CIDADE
FROM EX_ESTADO ES, EX_CIDADE CI
WHERE ES. UF = CI. UF
```

Retorno:

ESTADO	CIDADE
Santa Catarina	Criciúma
Santa Catarina	Florianópolis

## LEFT OUTER JOIN

[image-1646744385807.jpg](#)

Image not found or type unknown

É usado para relacionar e mostrar os dados de ambas as tabelas, preservando somente os dados da **primeira tabela** mesmo que não haja igualdade nos campos em comum.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
LEFT OUTER JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

### Usos alternativos que retornam o mesmo resultado:

Não é necessário informar a operação OUTER.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
LEFT JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

Utilizando com a operação **NATURAL**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES NATURAL LEFT JOIN EX_CIDADE CI
```

Utilizando com a cláusula **USING**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
LEFT JOIN EX_CIDADE CI USING (UF)
```

Retorno:

ESTADO	CIDADE
Santa Catarina	Criciúma
Santa Catarina	Florianópolis
São Paulo	(null)

## RIGHT OUTER JOIN

[image-1646744708047.jpg](#)

Image not found or type unknown

É usado para relacionar e mostrar os dados de ambas as tabelas, preservando somente os dados da **segunda tabela** mesmo que não haja igualdade nos campos em comum.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
RIGHT OUTER JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

### Usos alternativos que retornam o mesmo resultado:

Não é necessário informar a operação **OUTER**.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
RIGHT JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

Utilizando com a operação **NATURAL**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas.

```
SELECT ES. DESCRICAO AS ESTADO,
```

```
CI. NOME AS CIDADE
FROM EX_ESTADO ES NATURAL RIGHT JOIN EX_CIDADE CI
```

Utilizando com a cláusula **USING**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
RIGHT JOIN EX_CIDADE CI USING (UF)
```

Retorno:

ESTADO	CIDADE
Santa Catarina	Florianópolis
Santa Catarina	Criciúma
(null)	Curitiba

## FULL OUTER JOIN

[image-1646744999045.jpg](#)

Image not found or type unknown

É usado para relacionar e mostrar os dados de ambas as tabelas, preservando os dados mesmo que não haja igualdade nos campos em comum.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
FULL OUTER JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

### Usos alternativos que retornam o mesmo resultado:

Não é necessário informar a operação **OUTER**.

```
SELECT ES. DESCRICAO AS ESTADO,
       CI. NOME AS CIDADE
FROM EX_ESTADO ES
FULL JOIN EX_CIDADE CI ON ES. UF = CI. UF
```

Utilizando com a operação **NATURAL**. É necessário que as colunas em comum tenham o mesmo

nome em ambas as tabelas

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES NATURAL FULL JOIN EX_CIDADE CI
```

Utilizando com a cláusula **USING**. É necessário que as colunas em comum tenham o mesmo nome em ambas as tabelas.

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES FULL JOIN EX_CIDADE CI USING (UF)
```

Retorno:

ESTADO	CIDADE
Santa Catarina	Criciúma
Santa Catarina	Florianópolis
(null)	Curitiba
São Paulo	(null)

## CROSS JOIN

É usado para mostrar os dados de ambas as tabelas sem relacionar por campos em comum. Essa operação relaciona todos registros de uma tabela com todos registros da outra tabela.

```
SELECT ES.DESCRICAO AS ESTADO,  
       CI.NOME AS CIDADE  
FROM EX_ESTADO ES CROSS JOIN EX_CIDADE CI
```

Retorno:

ESTADO	CIDADE
Santa Catarina	Criciúma
Santa Catarina	Florianópolis
Santa Catarina	Curitiba
São Paulo	Criciúma
São Paulo	Florianópolis
São Paulo	Curitiba

Revisão #5

Criado 8 March 2022 08:52:55 por Nicolly Andrielly

Atualizado 26 October 2023 17:55:55 por Nicolly Andrielly