

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	DESCRICAÇÃO	UF	DESCRICAÇÃO
SC	Santa Catarina	SP	São Paulo

SQL de CRIAÇÃO/INSERÇÃO:

```
CREATE TABLE ex_estado (uf VARCHAR2(2), descricao VARCHAR2(100));
INSERT INTO ex_estado VALUES ('SC', 'Santa Catarina');
INSERT INTO ex_estado VALUES ('SP', 'São Paulo');
COMMIT;
```

EX_CIDADE

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

SQL de CRIAÇÃO/INSERÇÃO:

```
CREATE TABLE ex_cidade (idcidade INTEGER, nome VARCHAR2(200), uf VARCHAR2(2));
INSERT INTO ex_cidade VALUES (1, 'Criciúma', 'SC');
INSERT INTO ex_cidade VALUES (2, 'Florianópolis', 'SC');
INSERT INTO ex_cidade VALUES (3, 'Curitiba', 'PR');
COMMIT;
```

INNER JOIN

image1646844726590.jpg

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
```

LEFT OUTER JOIN

image-1646845009495.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 CIB
```

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
```

RIGHT OUTER JOIN

image-1646845256600.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 | |
```

FULL OUTER JOIN

image-1646845502120.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 #2

Criado 9 March 2022 13:44:35 por Nicolly Andrielly

Atualizado 10 March 2022 14:23:40 por Nicolly Andrielly