

**INSTITUTO FEDERAL**

Goiano

Campus Morrinhos

# ***BANCO DE DADOS***

## ***SQL JOINS***

***PROFESSOR MARCEL MELO***

***MARCEL.MELO@IFGOIANO.EDU.BR***

O comando **SELECT** recupera os dados de uma ou mais tabelas, sendo um dos comandos mais simples e, ao mesmo tempo, mais extenso da SQL devido as suas funções, operandos, comandos, subcomandos e cláusulas não obrigatórias.

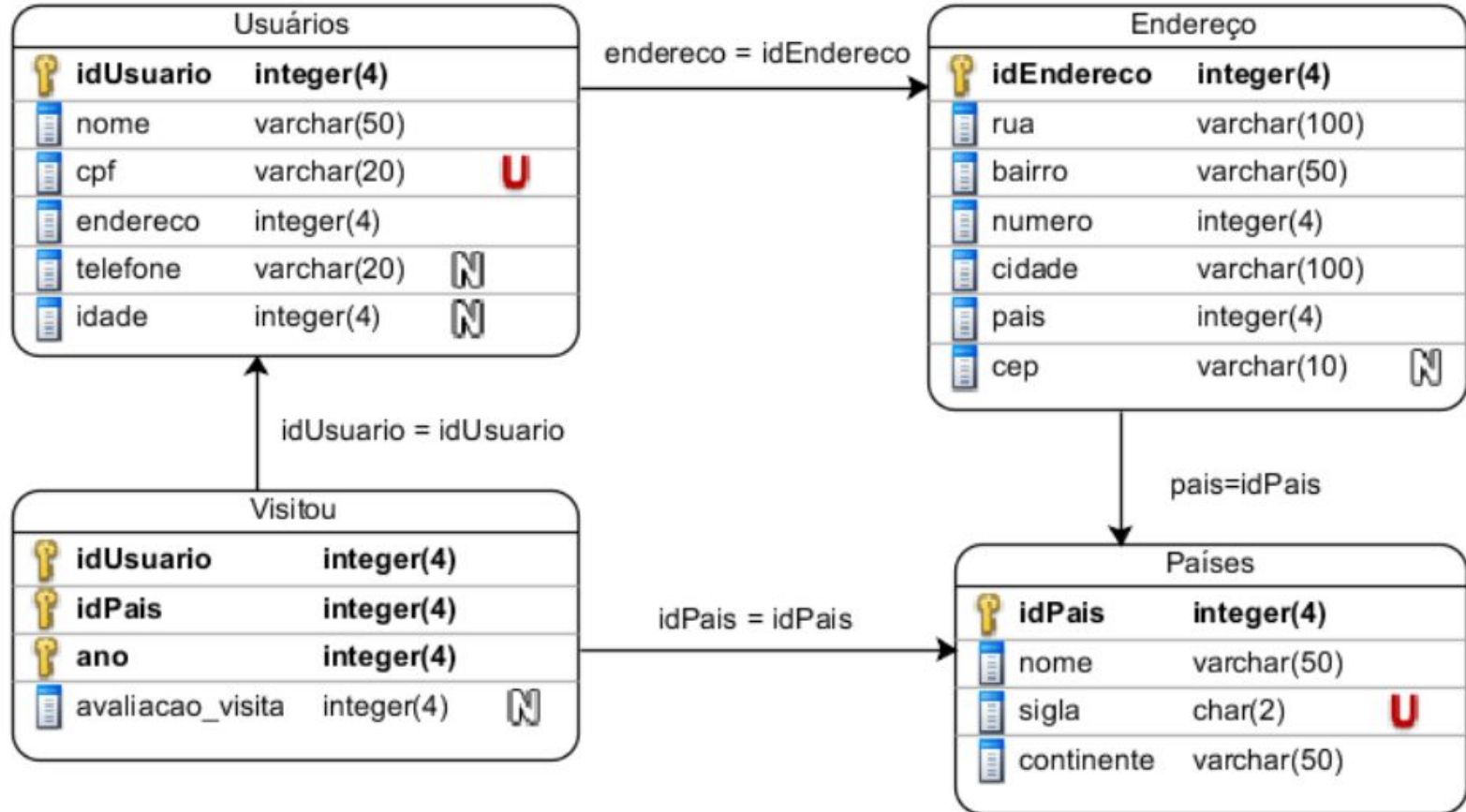
Para juntar duas ou mais tabelas, estávamos aplicando uma condição específica no **WHERE**. Esse tipo de junção tem o problema de sempre realizar o Produto Cartesiano em todas tabelas listadas no **FROM** antes de efetivamente fazer a filtragem das informações.

O JOIN é o comando mais indicado para realizar a junção de duas ou mais tabelas em uma consulta de dados. Como fizemos anteriormente, o JOIN também se baseia no relacionamento entre colunas destas tabelas (Chaves Estrangeiras).

## O JOIN pode ser de vários tipos:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

# EXEMPLO



A Junção básica de tabelas utiliza de uma condição realizada na cláusula WHERE e utiliza-se das chaves estrangeiras e primárias das tabelas envolvidas.

**Exemplo: Encontre todos os usuários que visitaram o país com sigla BR.**

```
SELECT u.nome, u.cpf
FROM Usuarios as u, Visitou as v, Países as p
WHERE u.idUsuario = v.idUsuario AND
v.idPaís = p.idPaís AND
p.sigla = 'BR'
```

***INNER JOIN***

Utilizando o operador **INNER JOIN** a condição para a junção das tabelas é realizada na própria cláusula **FROM**, impedindo assim que seja feito o Produto Cartesiano.

O **INNER JOIN** compara cada linha das tabelas T1 com as linhas da tabela T2 e retorna as linhas que satisfazem a condição de junção. Assim, se a condição for verdadeira é gerada uma nova linha combinando os dados da tabela T1 e T2.

- Geralmente através de uma chave estrangeira.

```
SELECT <colunas>
```

```
FROM tabela1 as t1 INNER JOIN tabela2 as t2
```

```
ON (t1.coluna = t2.coluna)
```

## EXEMPLO ANTERIOR COM JOIN

**Exemplo: Encontre todos os usuários que visitaram o país com sigla BR.**

```
SELECT u.nome, u.cpf
FROM Usuarios as u INNER JOIN Visitou as v
    ON (u.idUsuario = v.idUsuario)
    INNER JOIN Países as p ON (v.idPaís = p.idPaís)
WHERE p.sigla = 'BR'
```



É uma seleção que não requer que os registros de uma tabela possuam registros equivalentes em outra.

O registro é mantido mesmo se não existe outro registro que lhe corresponda.

Este tipo de junção se subdivide dependendo da tabela do qual admitiremos os registros que não possuem correspondência: a tabela esquerda, a direita ou ambas.

- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

***LEFT JOIN***

A cláusula LEFT JOIN ou LEFT OUTER JOIN permite obter não apenas os dados relacionados de duas tabelas, mais também os dados não relacionados encontrados na tabela à esquerda da cláusula JOIN.

```
SELECT <colunas>
```

```
FROM tabela1 as t1 LEFT JOIN tabela2 as t2
```

```
ON (t1.coluna = t2.coluna)
```



Para cada linha da tabela T1, a consulta a compara com todas as linhas da tabela T2. Se um par de linhas fizer com que a condição de junção seja avaliado como verdadeira, os valores dessas linhas serão combinados para formar uma nova linha que será incluída no conjunto de resultados.

Se uma linha da tabela “esquerda” T1 não possuir nenhuma linha correspondente da tabela “direita” T2, a consulta irá combinar os valores da linha da tabela “esquerda” T1 com **NULL** para cada valor da coluna da tabela da “direita” T2 que não satisfaça a condição de junção



## EXEMPLO LEFT JOIN

Exemplo: Recuperar os dados de todos usuários e, caso ele tenha visitado algum país, apresentar os dados da visita

```
SELECT u.nome, u.cpf  
FROM Usuarios as u LEFT JOIN Visitou as v  
ON (u.idUsuario = v.idUsuario)
```

# EXPLICAÇÃO EXEMPLO LEFT JOIN

Note que nesse caso os dados de todos usuários serão apresentados, independentemente se visitou uma país ou não. **O LEFT JOIN garante que todos os dados da tabela a esquerda serão retornados.**

No caso de um usuário não ter visitado nenhum país, os campos correspondente a tabela Visitou serão preenchidos com o valor nulo.

**SELECT** \*

**FROM** Usuarios as u **LEFT JOIN** Visitou as v

**ON** (u.idUsuario = v.idUsuario)

**LEFT JOIN** Países as p **ON** (p.idPaís = v.idPaís)

***RIGHT JOIN***

Ao contrário do LEFT JOIN, a cláusula **RIGHT JOIN** ou **RIGHT OUTER JOIN** permite obter não apenas os dados relacionados de duas tabelas, mais também os dados não relacionados encontrados na tabela à direita da cláusula JOIN.

Caso não existam dados associados entre as tabelas à esquerda e à direita de JOIN, serão retornados valores nulos no preenchimento das colunas da tabela a esquerda.

```
SELECT <colunas>  
FROM tabela1 as t1 RIGHT JOIN tabela2 as t2  
ON (t1.coluna = t2.coluna)
```



## EXEMPLO RIGHT JOIN

Exemplo: Recupera todos os países cadastrados e, caso algum usuário tenha o visitado, retornar os dados da visita e do usuário.

- No caso de ninguém tenha visitado o país, as colunas referentes as tabelas Visitou e Usuários serão preenchidas com o valor NULL

```
SELECT *  
FROM Usuarios as u RIGHT JOIN Visitou as v  
    ON (u.idUsuario = v.idUsuario)  
    RIGHT JOIN Países as p ON (p.idPaís = v.idPaís)
```

***FULL JOIN***

# ***FULL JOIN (NÃO SUPORTADO PELO MYSQL)***

A cláusula FULL JOIN ou FULL OUTER JOIN equivale a um LEFT JOIN e um RIGHT JOIN ao mesmo tempo. Todas as linhas de dados da tabela à esquerda e da tabela à direita de JOIN serão retornadas pela cláusula.

Caso uma linha de dados não esteja associada a qualquer linha da outra tabela, os valores das colunas a lista de seleção serão nulos. Caso contrário, os valores obtidos serão baseados nas tabelas usadas como referência.

```
SELECT <colunas>
```

```
FROM tabela1 as t1 FULL JOIN tabela2 as t2
```

```
ON (t1.coluna = t2.coluna)
```

***CROSS JOIN***

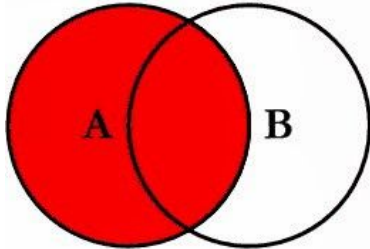
Todos os dados da tabela à esquerda de JOIN são cruzados com os dados da tabela à direita de JOIN por meio do CROSS JOIN, também conhecido como produto cartesiano. É possível cruzarmos informações de duas ou mais tabelas.

- Equivalente a somente listar a tabelas no FROM.

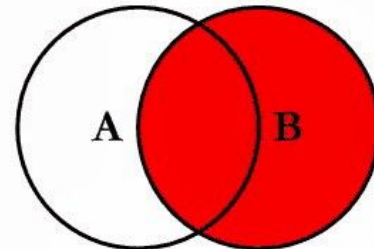
```
SELECT <colunas>
```

```
FROM tabela1 as t1 CROSS JOIN tabela2 as t2
```

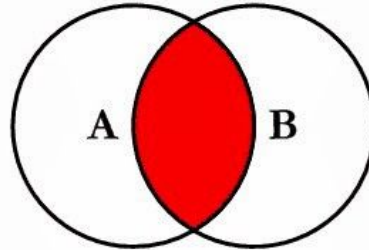
# SQL JOINS



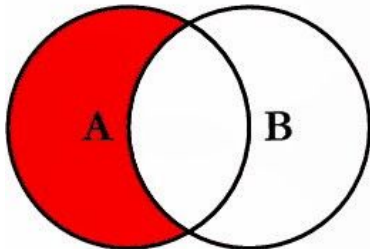
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



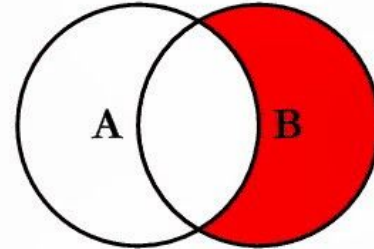
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



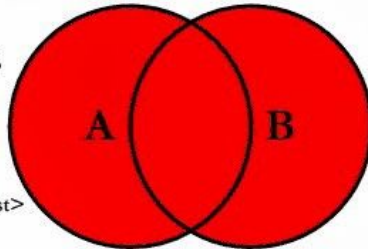
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



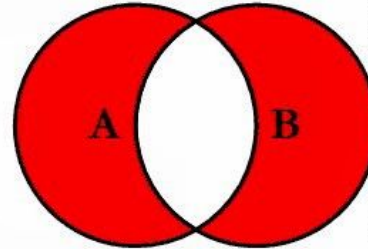
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```