

Banco de Dados

JOINS

Profa. Tatiane Coreixas de Moraes

JOIN

Utilizamos o JOIN para associar dados de duas ou mais tabelas, utilizamos juntamente com o select

O Objetivo de associar uma ou mais tabelas seria de obter dados de tabelas diferentes, sendo que estas apresentam valores comuns existentes em colunas correspondentes.

JOIN

Tipos de JOINS

- **INNER JOIN** ou **JOIN** – quando queremos que o resultado seja por igualdade, ou seja, que o valor de um campo na tabela 1 seja correspondente na tabela 2.

- Sintaxe:

```
SELECT tabela1.campo1, tabela1.campoN,  
tabela2.campo1, tabela2.campoN
```

```
FROM tabela1
```

```
INNER JOIN tabela2
```

```
ON tabela1.campo = tabela2.campo;
```

JOIN

Cod_Pai	Nome_Pai	Idade_Pai
1	Daniel de Souza Leão Sobrinho	62
2	João Carlos da Silva	38
3	Fernando de Oliveira	36

FK

Cod_Filho	Cod_Pai	Nome_Filho	Sexo_Filho
1	1	Renata de Oliveira Leão	F
2	1	Fernando de Oliveira Leão	M
3	1	Roberta de Oliveira Leão	F
4	1	Jairo de Oliveira Leão	M
5	2	Giovanna da Silva	F
6	3	Lucas Ribeiro Oliveira	M
7	3	Helder Ribeiro Oliveira	M

JOIN

Ex.: INNER JOIN

Suponha que você precise obter o nome de todos os pais e o nome e o sexo de todos os filhos que cada pai possui.

```
SELECT Pai.Nome_Pai, Filho.Nome_Filho, Filho.Sexo_Filho  
FROM Pai INNER JOIN Filho  
ON Pai.Cod_Pai = Filho.Cod_Pai
```

JOIN

Tipos de JOINS

- **OUTER JOIN** – retorna os registros de ambas as tabelas mesmo quando a combinação apresentar diferenças.
- É dividido em: Right Join, Left Join e Full Join

JOIN

Tipos de JOINS

- **RIGHT OUTER JOIN** ou **OUTER JOIN** – retorna os registros de ambas as tabelas mesmo quando a combinação apresentar diferenças referente a tabela que está a direita do Join.

Sintaxe:

```
SELECT tabela1.campo1, tabela1.campoN,  
tabela2.campo1, tabela2.campoN  
  
FROM tabela1  
  
RIGHT JOIN tabela2  
  
ON tabela1.campo = tabela2.campo;
```

JOIN

Tipos de JOINS

- **RIGHT OUTER JOIN ou OUTER JOIN**

- **Ex.:**

Se precisasse obter como resposta os dados dos filhos e seus pais, mostrando inclusive os filhos órfãos, você escreveria o seguinte comando:

```
SELECT Pai.Nome_Pai, Filho.Nome_Filho, Filho.Sexo_Filho  
FROM Pai RIGHT OUTER JOIN Filho  
ON Pai.Cod_Pai = Filho.Cod_Pai
```


JOIN

Tipos de JOINS

- **LEFT OUTER JOIN** ou **LEFT JOIN** – retorna os registros de ambas as tabelas mesmo quando a combinação apresentar diferenças referente a tabela que está a esquerda do Join.

Sintaxe:

```
SELECT tabela1.campo1, tabela1.campoN,  
tabela2.campo1, tabela2.campoN
```

```
FROM tabela1
```

```
LEFT JOIN tabela2
```

```
ON tabela1.campo = tabela2.campo;
```

JOIN

Tipos de JOINS

- **LEFT OUTER JOIN ou LEFT JOIN:**
- **Ex.:** Escreva uma consulta que exiba os dados de todos os pais relacionando esses dados com os respectivos filhos, mostrando também os dados dos pais que ainda estão sem filhos registrados.

```
SELECT Pai.Nome_Pai, Filho.Nome_Filho, Filho.Sexo_Filho  
FROM Pai Left Outer Join Filho  
ON Pai.Cod_Pai = Filho.Cod_Pai
```

JOIN

Tipos de JOINS

- **FULL OUTER JOIN** ou **FULL JOIN** – retorna os registros de ambas as tabelas mesmo quando a combinação apresentar diferenças referente as tabelas.

Sintaxe:

```
SELECT tabela1.campo1, tabela1.campoN,  
tabela2.campo1, tabela2.campoN  
  
FROM tabela1  
  
FULL JOIN tabela2  
  
ON tabela1.campo = tabela2.campo;
```

JOIN

Tipos de JOINS

- **CROSS JOIN** – retorna um produto cartesiano.

Sintaxe:

```
SELECT tabela1.campo1, tabela1.campoN,  
tabela2.campo1, tabela2.campoN
```

```
FROM tabela1
```

```
CROSS JOIN tabela2;
```

JOIN

Para conhecimento

- Podemos utilizar o WHERE juntamente com os JOINS, sendo que o WHERE ficará no final e será utilizado como condição de restrição.

```
SELECT Pai.Nome_Pai, Filho.Nome_Filho,  
FROM Pai INNER JOIN Filho  
ON Pai.Cod_Pai = Filho.Cod_Pai  
WHERE Pai.Nome like 'João%';
```

JOIN

Para conhecimento

- A sintaxe abaixo é uma estrutura antiga, mas ainda permitida no banco de dados (essa mudança na sintaxe pode ser utilizada para todos os tipos de JOINS vistos anteriormente.):

```
SELECT tabela1.campo1, tabela1.campoN,  
       tabela2.campo1, tabela2.campoN  
  
FROM tabela1, tabela2  
  
WHERE tabela1.campo = tabela2.campo;
```