

Disciplina: *Sistemas de Banco de Dados 1*

Turma: A - 2º/2019

Professor: *Vandor Roberto Vilardi Rissoli*

Aluno: *Rogério S. dos Santos Júnior*

Matrícula: 17/0021751

## **AULA 9 – LINGUAGEM SQL / RESTRIÇÕES - EXERCÍCIO EXTRA 1**

### *Junções (JOINS) em SQL*

O comando *JOIN* (**Junção**) é uma instrução *SQL* que combinam dois relacionamentos que possuam uma coluna que seja comum às mesmas, ou não (*CROSS JOIN*). Esse comando gera um novo relacionamento que possui o grau de relacionamento que é a soma das tabelas que a geraram. Pode ser visto como um produto cartesiano seguido de uma operação de *SELECT*.

A principal **vantagem** do uso de **junção** em *SQL* é a rapidez para ser executada: pela indexação e nomenclatura das colunas e a otimização da *ENGINE* do banco de dados, o tempo de recuperação dos dados vai quase sempre ser menor que o uso de subconsultas (consultas aninhadas).

As principais **desvantagens** são relacionadas a complexidade desse tipo de comando:

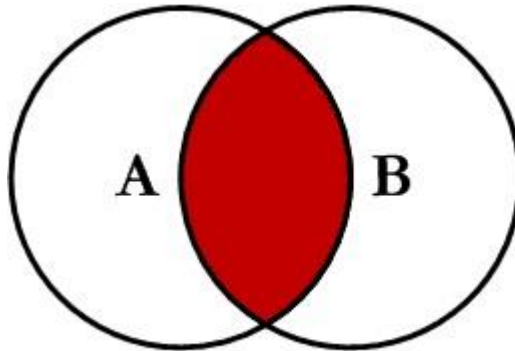
- Dificuldade de leitura em comparação à subconsultas;
- Confusão em qual o tipo apropriado de junção a ser realizada para poder obter o resultado esperado.

Os principais tipos de **junções** são:

*\* As tabelas usadas nos exemplos estão na última página*

### 1) Inner Join

Retorna as tuplas que estejam em ambos relacionamentos, como mostrado na figura abaixo.



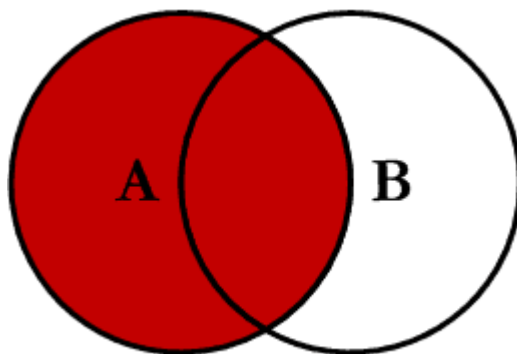
O comando a seguir retornaria a tabela abaixo:

```
1 SELECT A.nome, B.nomeCurso
2   FROM ALUNO as A
3   INNER JOIN CURSO as B
4   on A.idAluno = B.idAluno
```

nome	nomeCurso
Fabiana Ribas	Moda
Youssef Muhamad	Design Gráfico

## 2) Left Join

Retorna as tuplas da tabela esquerda (A) e as que estejam em ambos relacionamentos, como mostrado na figura abaixo.



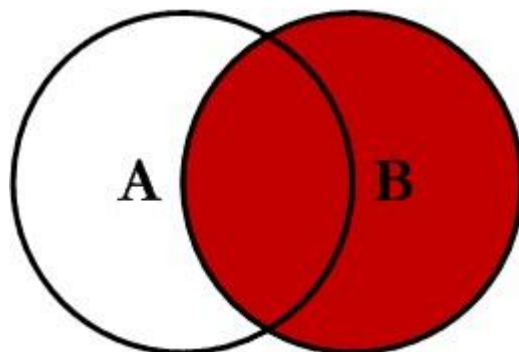
O comando a seguir retornaria a tabela abaixo:

```
1 SELECT A.nome, B.nomeCurso
2   FROM ALUNO as A
3   LEFT JOIN CURSO as B
4     on A.idAluno = B.idAluno
```

nome	nomeCurso
Rogério Júnior	NULL
Fabiana Ribas	Moda
Youssef Muhamad	Design Gráfico

### 3) Right Join

Retorna as tuplas da tabela direita (B) e as que estejam em ambos relacionamentos, como mostrado na figura abaixo.



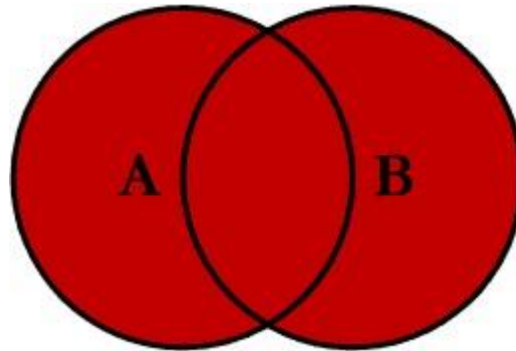
O comando a seguir retornaria a tabela abaixo:

```
1 SELECT A.nome, B.nomeCurso
2   FROM ALUNO as A
3   RIGHT JOIN CURSO as B
4     on A.idAluno = B.idAluno
```

nome	nomeCurso
Fabiana Ribas	Moda
Youssef Muhamad	Design Gráfico
NULL	Geografia Geral
NULL	Língua Portuguesa I

#### 4) Full Join

Retorna as tuplas de ambas as tabelas, como mostrado na figura abaixo.



Ainda que não exista o comando FULL JOIN no MySQL, o comando executado a seguir tem o mesmo propósito e retornaria a tabela abaixo:

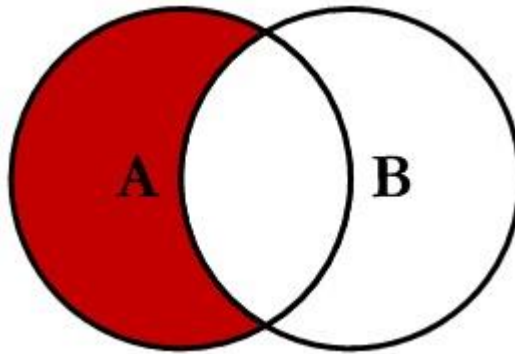
```

1  SELECT A.nome, B.nomeCurso
2      FROM ALUNO as A
3      LEFT JOIN CURSO as B
4          on A.idAluno = B.idAluno
5 UNION
6  SELECT A.nome, B.nomeCurso
7      FROM ALUNO as A
8      RIGHT JOIN CURSO as B
9          on A.idAluno = B.idAluno;
    
```

nome	nomeCurso
Rogério Júnior	NULL
Fabiana Ribas	Moda
Youssef Muhamad	Design Gráfico
NULL	Geografia Geral
NULL	Língua Portuguesa I

## 5) Left Excluding Join

Retorna somente as tuplas da tabela da esquerda (A), como mostrado na figura abaixo.



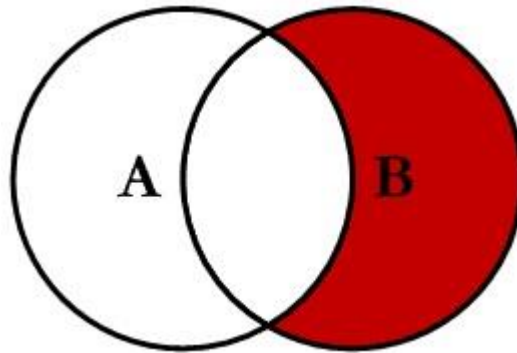
O comando a seguir retornaria a tabela abaixo:

```
1 SELECT A.nome, B.nomeCurso
2   FROM ALUNO as A
3  LEFT JOIN CURSO as B
4      on A.idAluno = B.idAluno
5      WHERE B.idAluno is NULL
```

nome	nomeCurso
Rogério Júnior	NULL

## 6) Right Ecluding Join

Retorna somente as tuplas da tabela da direita (B), como mostrado na figura abaixo.



O comando a seguir retornaria a tabela abaixo:

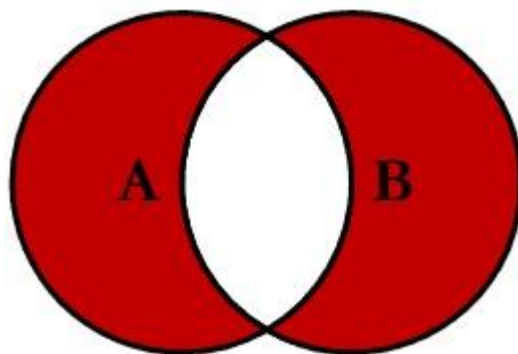
```

1 SELECT A.nome, B.nomeCurso
2   FROM ALUNO as A
3  RIGHT JOIN CURSO as B
4     on A.idAluno = B.idAluno
5     WHERE A.idAluno is NULL
    
```

nome	nomeCurso
NULL	Geografia Geral
NULL	Língua Portuguesa I

## 7) Outer Excluding Join

Retorna somente as tuplas que não são comuns à ambas as tabelas, como mostrado na figura abaixo.



Ainda que não exista o comando FULL JOIN no MySQL, o comando executado a seguir tem o mesmo propósito e retornaria a tabela abaixo

```

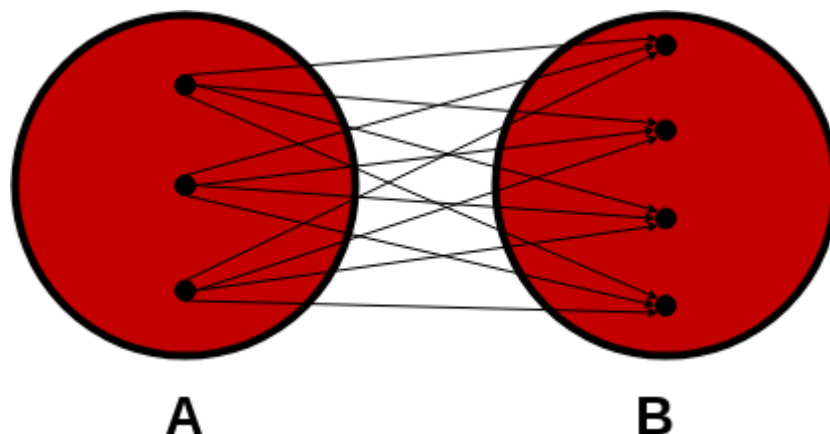
1  SELECT A.nome, B.nomeCurso
2  FROM ALUNO as A
3  LEFT JOIN CURSO as B
4      on A.idAluno = B.idAluno
5      WHERE B.idAluno is NULL
6 UNION
7  SELECT A.nome, B.nomeCurso
8  FROM ALUNO as A
9  RIGHT JOIN CURSO as B
10     on A.idAluno = B.idAluno
11     WHERE A.idAluno is NULL
    
```

nome	nomeCurso
Rogério Júnior	NULL
NULL	Geografia Geral
NULL	Língua Portuguesa I



## 8) Cross Join

Retorna o produto cartesiano das tuplas e o relacionamento resultante tem a quantidade das tuplas de A x tuplas de B.



O comando a seguir retornaria a tabela abaixo:

```
1 SELECT *  
2 FROM ALUNO  
3 CROSS JOIN CURSO
```

idAluno	nome	genero	nomeCurso	idAluno	creditos
1	Rogério Junior	M	Moda	2	6
2	Fabiana Ribas	F	Design Gráfico	3	8
3	Youssef Muhamad	M	Geografia Geral	6	4
1	Rogério Junior	M	Língua Portuguesa I	11	10
2	Fabiana Ribas	F	Moda	2	6
3	Youssef Muhamad	M	Design Gráfico	3	8
1	Rogério Junior	M	Geografia Geral	6	4
2	Fabiana Ribas	F	Língua Portuguesa I	11	10
3	Youssef Muhamad	M	Moda	2	6
1	Rogério Junior	M	Design Gráfico	3	8
2	Fabiana Ribas	F	Geografia Geral	6	4
3	Youssef Muhamad	M	Língua Portuguesa I	11	10



## TABELAS

### ALUNO

idAluno	nome	genero
1	Rogério Junior	M
2	Fabiana Ribas	F
3	Youssef Muhamad	M

### CURSO

nomeCurso	idAluno	creditos
Moda	2	6
Design Gráfico	3	8
Geografia Geral	6	4
Língua Portuguesa I	11	10

## REFERÊNCIAS

CROVETT, Lora. **What Are the Advantages & Disadvantages of a Join or a Subquery?**. [S. l.], 2019. Disponível em: <https://www.techwalla.com/articles/what-is-relational-database-schema>. Acesso em: 28 out. 2019.

ELMASRI, Ramez; NAVATCHE, Shamkant B. **Sistemas de Bancos de Dados**. 4. ed. São Paulo: Pearson Education do Brasil, 2005. 724 p. ISBN 85-88639-17-3.

SALLAI, Fernanda. **SQL Join**: Entenda como funciona o retorno dos dados. [S. l.], 2019. Disponível em: <https://www.devmedia.com.br/sql-join-entenda-como-funciona-o-retorno-dos-dados/31006>. Acesso em: 28 out. 2019.