

# Sistemas Operacionais III

*SQL básico no MariaDB*

Anderson Alves de Albuquerque

# Sumário

- Introdução;
- Criar banco de dados;
- Criar tabelas;
- Inserir registros na tabela;
- Alterar registros da tabela;
- Alterar colunas da tabela;
- Removendo registros;
- Consultando os dados;
- Restrições;
- Fim.

# Mãos na massa 1 de

Introdução ao cenário utilizado nos exemplos



# Cenário

- Os comandos SQL serão utilizados em uma tabela chamada cidades contendo as colunas Matricula, sigla e DDD;
- Depois existiram outras tabelas bem simples para utilização do comandos SQL, como, por exemplo: Alunos, Estudantes, Discentes, Vendas e Compras;
- As observações importantes estarão com o fundo da letra em amarelo claro.
- Quando aparecer o caractere especial tralha (ou jogo da velha) significa que a digitação ocorre no Linux e com a conta do super usuário **root**;
- Quando aparecer “MariaDB [(none)]> ” indica que a digitação ocorre dentro do SGBD MariaDB;

# Mãos na massa 1 de 12

Criar banco de dados



# Entrar no SGBD MariaDB

- Pela conta **root** conecte no SGBD MariaDB:

**# mysql**

MariaDB [(none)]>

# Testando o SGBD



- Dentro do SGBD MariaDB crie o banco de dados escola.

MariaDB [(none)]> `create database escola;`

- Permite ver os bancos de dados existentes:

MariaDB [(none)]> `show databases;`

**Obs.: localize o banco de dados escola.**

# Mãos na massa 2 de 12

Remover banco de dados





# Remover o banco de dados escola

- Se você não estiver no SGBD MariaDB, então pela conta **root** conecte nesse SGBD da seguinte forma:

**# mysql**

MariaDB [(none)]>

- Permite ver os bancos de dados existentes antes do “*drop*”:

MariaDB [(none)]> **show databases;**

- Remover o banco de dados escola:

MariaDB [(none)]> **drop database escola;**

MariaDB [(none)]>

- Ver os bancos de dados existentes depois do “*drop*”:

MariaDB [(none)]> **show databases;**

**Obs.: verifique se o banco de dados escola foi removido.**

# Testando o SGBD



- Agora veja as tabelas do SGBD MariaDB, observe que o Banco de Dados escola apareceu na listagem da tabela bancos de dados existentes.

MariaDB [(none)]> **show databases;**

- Para sair do MariaDB e retornar para o Linux:

MariaDB [(none)]> **exit**

root@campus:/home/aluno#



```
root@campus:/home/aluno# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| escola   |
| information_schema |
| mysql     |
| performance_schema |
| test      |
+-----+
5 rows in set (0.000 sec)

MariaDB [(none)]>
```

# Mãos na massa 3 de 12

Criar e remover tabelas



# Entrar no SGBD MariaDB



- Se você não estiver no SGBD MariaDB, então pela conta **root** conecte nesse SGBD da seguinte forma:

**# mysql**

MariaDB [(none)]>

# Testando o SGBD

- Dentro do MariaDB, realize os procedimentos abaixo para ver se o SGBD está funcionando perfeitamente.

MariaDB [(none)]> `create database escola;`



Criando o banco de dados escola

MariaDB [(none)]> `use escola`



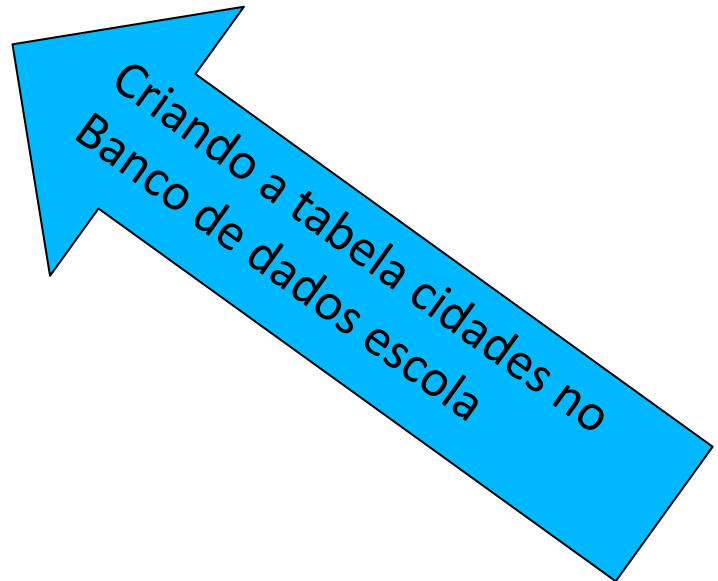
O “use escola” serve para entrar no banco de dados escola

# Testando o SGBD

- Criar a tabela:

```
MariaDB [(none)]> CREATE TABLE cidades ( nome varchar(80), sigla varchar(2), DDD varchar(3) );
```

```
MariaDB [(none)]> CREATE TABLE bairros ( nome varchar(80) );
```



# Veja as tabelas que existem no banco de dados escola

- Ver as tabelas existentes:

```
MariaDB [(none)]> show tables;
```

**Obs.: veja que existem 2 tabelas, cidades e bairros.**

# Remover uma tabela do banco de dados

- Remover uma tabela “Venda”:

```
MariaDB [(none)]> drop table bairros;
```

```
MariaDB [(none)]> show tables;
```

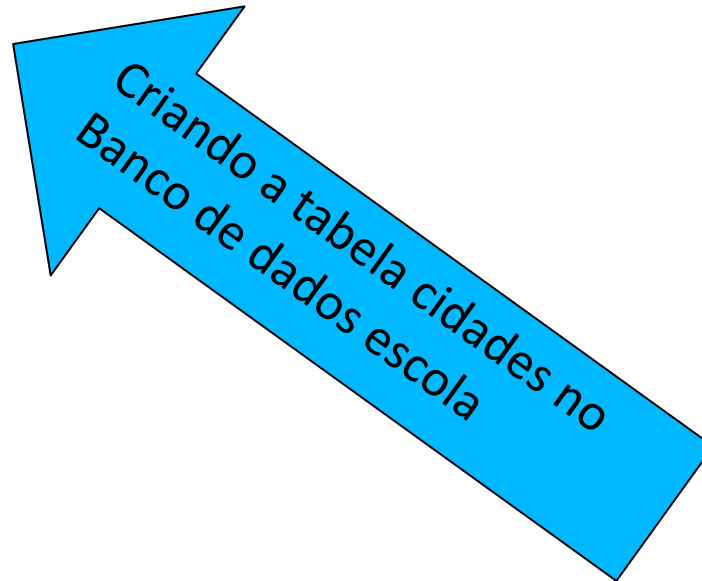
**Obs.: veja que agora existe apenas 1 tabela, pois a tabela bairros foi excluída.**



# Testando o SGBD

- Criar novamente a tabela:

MariaDB [(none)]> **CREATE TABLE cidades ( nome varchar(80), sigla varchar(2), DDD varchar(3) );**



# Mãos na massa 4 de 12

Inserir registros na tabela



# Testando o SGBD

- Inserindo registros na tabela cidades:

MariaDB [(none)]> INSERT INTO cidades VALUES ('Rio de Janeiro', 'RJ', '021');

MariaDB [(none)]> INSERT INTO cidades VALUES ('Sao Paulo', 'SP', '011');

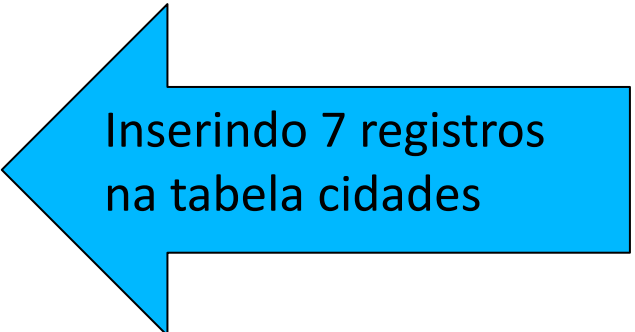
MariaDB [(none)]> INSERT INTO cidades VALUES ('Minas Gerais', 'MG', '031');

MariaDB [(none)]> INSERT INTO cidades VALUES ('Minha cidade', 'RJ', '0XX');

MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade01', 'RJ', '0x');

MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade02', 'RJ', '0xx');

MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade03', 'RJ', '0XX');



Inserindo 7 registros  
na tabela cidades

# Testando o SGBD

- Verificando com “*select*” se os registros foram inseridos na tabela cidades:

```
MariaDB [(none)]> SELECT * FROM cidades;
```

# Mãos na massa 5 de 12

Consultando os dados



# Entrar no SGBD MariaDB



- Se você não estiver no SGBD MariaDB, então pela conta **root** conecte nesse SGBD e acesse o banco de dados escola da seguinte forma:

**# mysql**

```
MariaDB [(none)]>
```

```
MariaDB [(none)]> use escola
```

```
MariaDB [(none)]>
```

# Testando o SGBD

- Verificando com “*select*” os registros que foram inseridos na tabela cidades:

MariaDB [(none)]> *SELECT \* FROM cidades;*

- Mostrar a quantidade de registros de uma tabela:

MariaDB [(none)]> *select count(\*) from cidades;*

- Mostrar o registro com DDD igual a “031”:

MariaDB [(none)]> *select \* from cidades Where DD=“031”;*

# Remover registros de uma tabela

- Remover registro de uma tabela “Venda”:

MariaDB [(none)]> `delete from cidades;`



# Testando o SGBD

- Verifique se agora existe algum registro na tabela ou se todos os registros foram removidos.

MariaDB [(none)]> **SELECT \* FROM cidades;**

# Testando o SGBD

- Inserindo **NOVAMENTE** registros na tabela cidades:

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('Rio de Janeiro', 'RJ', '021');
```

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('Sao Paulo', 'SP', '011');
```

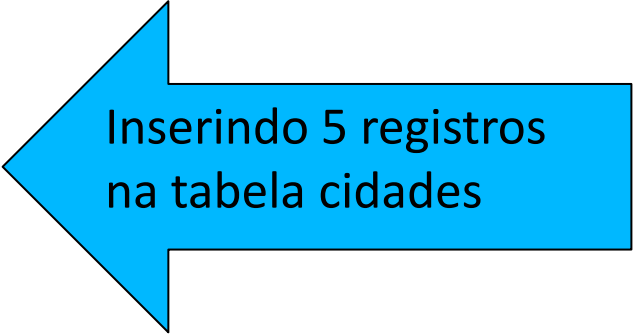
```
MariaDB [(none)]> INSERT INTO cidades VALUES ('Minas Gerais', 'MG', '031');
```

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('Minha cidade', 'RJ', '0XX');
```

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade01', 'RJ', '0x');
```

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade02', 'RJ', '0xx');
```

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('cidade03', 'RJ', '0XX');
```



Inserindo 5 registros  
na tabela cidades

# Remover registros de uma tabela

- Remover uma tabela “Venda” contendo ddd igual a '00':

MariaDB [(none)]> delete from cidades Where DDD='00';

**OBS: a clausula Where serve para especificar o que será removido.**

# Testando o SGBD

- Verifique se a cidade com DDD igual a '00' foi removida:

```
MariaDB [(none)]> SELECT * FROM cidades;
```

# Mãos na massa 6 de 12

Alterar registros da tabela



# Entrar no SGBD MariaDB



- Se você não estiver no SGBD MariaDB, então pela conta **root** conecte nesse SGBD e acesse o banco de dados escola da seguinte forma:

**# mysql**

```
MariaDB [(none)]>
```

```
MariaDB [(none)]> use escola
```

```
MariaDB [(none)]>
```

# Alterar registro

- Mudar os registros contando DDD igual a “0XX” para DDD igual a “9”:

MariaDB [(none)]> **UPDATE** cidades SET DDD = "9" WHERE DDD="0XX";

Valor inicial

Valor após a mudança

```
aluno@campus: ~  
MariaDB [escola]>  
MariaDB [escola]>  
MariaDB [escola]> select * from cidades;  
+-----+-----+-----+-----+  
| nome          | sigla | DDD  | ano  |  
+-----+-----+-----+-----+  
| Rio de Janeiro | RJ    | 021  | NULL |  
| Sao Paulo     | SP    | 011  | NULL |  
| Minas Gerais  | MG    | 031  | NULL |  
| Minha cidade  | RJ    | 0XX  | NULL |  
+-----+-----+-----+-----+  
4 rows in set (0.000 sec)  
  
MariaDB [escola]> UPDATE cidades SET DDD = "9" WHERE DDD="0XX";  
Query OK, 1 row affected (0.001 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
MariaDB [escola]> select * from cidades;  
+-----+-----+-----+-----+  
| nome          | sigla | DDD  | ano  |  
+-----+-----+-----+-----+  
| Rio de Janeiro | RJ    | 021  | NULL |  
| Sao Paulo     | SP    | 011  | NULL |  
| Minas Gerais  | MG    | 031  | NULL |  
| Minha cidade  | RJ    | 9    | NULL |  
+-----+-----+-----+-----+  
4 rows in set (0.000 sec)  
  
MariaDB [escola]>
```

Utilização do Update  
para trocar o valor da  
coluna DDD quando  
DDD = “0XX”

# Mãos na massa 7 de 12

Alterar colunas da tabela





# Alterando colunas da tabela

- **Adicionar a coluna capital:**

```
MariaDB [(none)]> desc cidades;  
MariaDB [(none)]> ALTER TABLE cidades ADD capital varchar(50);  
MariaDB [(none)]> desc cidades;
```

- **Mudar o tipo da coluna capital:**

```
MariaDB [(none)]> desc cidades;  
MariaDB [(none)]> ALTER TABLE cidades modify COLUMN capital int;  
MariaDB [(none)]> desc cidades;
```

- **Trocar o nome da coluna “capital” para o nome “ano” com o tipo inteiro (int):**

```
MariaDB [(none)]> desc cidades;  
MariaDB [(none)]> ALTER TABLE cidades change column capital ano int;  
MariaDB [(none)]> desc cidades;
```

- **Adicionando uma coluna e depois removendo-a:**

```
MariaDB [(none)]> desc cidades;  
MariaDB [(none)]> ALTER TABLE cidades ADD telefone varchar(50);  
MariaDB [(none)]> desc cidades;  
MariaDB [(none)]> ALTER TABLE cidades DROP telefone;  
MariaDB [(none)]> desc cidades;
```

# Alterando colunas da tabela

- **Modificando o tipo da coluna da tabela:**

MariaDB [(none)]> **desc cidades;**

MariaDB [(none)]> **ALTER TABLE cidades ADD codigo varchar(50);**

MariaDB [(none)]> **desc cidades;**

MariaDB [(none)]> **ALTER TABLE cidades MODIFY codigo int;**

MariaDB [(none)]> **desc cidades;**

# Mãos na massa 8 de 12

Removendo registros



# Removendo registros

- Veja que clausula *Where* especifica o que será removido.

```
MariaDB [(none)]> INSERT INTO cidades VALUES ('Sergipe', 'SE', '0x');
```

```
MariaDB [(none)]> DELETE FROM cidades WHERE DDD = '0x';
```

**Obs.: se não utilizar a condição *Where*, todos os registros são removidos.**

# Mãos na massa 9 de 12

Chave primária



# Criando tabela com chave primária

- Criando tabela com a chave primária “Matricula”:

MariaDB [(none)]> **CREATE TABLE alunos (Matricula INTEGER PRIMARY KEY, nome TEXT, nascimento int);**

MariaDB [(none)]> **desc alunos;**

```
aluno@campus: ~  
  
MariaDB [escola]> CREATE TABLE alunos (Matricula INTEGER PRIMARY KEY, nome TEXT, nascimento int);  
Query OK, 0 rows affected (0.012 sec)  
  
MariaDB [escola]> desc alunos;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type   | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Matricula  | int(11)| NO   | PRI | NULL    |       |  
| nome       | text   | YES  |     | NULL    |       |  
| nascimento | int(11)| YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.001 sec)  
  
MariaDB [escola]> █
```

Indicação de chave primária.

# Inserir registros

- **Inserir registros seguindo a ordem dos campos digitados no “Create tabela”:**

```
MariaDB [(none)]> INSERT INTO alunos (Matricula, nome, nascimento) VALUES (1,“Fulano”, 2002);  
MariaDB [(none)]> INSERT INTO alunos (Matricula, nome, nascimento) VALUES (25,“Ciclano”, 1999);  
MariaDB [(none)]> INSERT INTO alunos VALUES (11,“Maria”, 1991);  
MariaDB [(none)]> INSERT INTO alunos VALUES (4,“Maria”, 2002);  
MariaDB [(none)]> INSERT INTO alunos VALUES (41,“Maria”, 1992);  
MariaDB [(none)]> INSERT INTO alunos VALUES (42,“Maria”, 1999);  
MariaDB [(none)]> INSERT INTO alunos VALUES (8,“Ana”, 2002);  
MariaDB [(none)]> INSERT INTO alunos VALUES (82,“Ana”, 1992);  
MariaDB [(none)]> INSERT INTO alunos VALUES (81,“Andre”, 2002);
```

- **Veja os dados inseridos:**

```
MariaDB [(none)]> Select * from alunos;
```

# Ordenação utilizando o “Select”

- **Mostrar apenas as colunas “matricula” e “nome” ordenando pelo campo “matricula”:**

MariaDB [(none)]> `SELECT matricula, nome FROM alunos ORDER BY matricula;`

- **Mostrando as colunas “matricula”, “nome” e “Nascimento” ordenando pelo campo “matricula”:**

MariaDB [(none)]> `SELECT matricula, nome FROM alunos ORDER BY matricula;`

- **Agora utilize ordenação pela coluna “nome” para mostrar apenas as colunas “matricula” e “nome”:**

MariaDB [(none)]> `SELECT matricula, nome FROM alunos ORDER BY nome;`



# Ordenação utilizando o “Select”

- Utilize ordenação pela coluna “Nascimento” e no caso de registros com campo “Nascimento” igual, a ordenação ocorre pelo campo “nome”:

```
MariaDB [(none)]> SELECT * FROM alunos ORDER BY matricula,nome;
```

**Obs.: observe que os anos 1992, 1999 e 2002 possuem registros com os nomes dos alunos diferentes.**

- Agora utilize ordenação pela coluna “nome” e no caso de registros com campo “nome” igual, a ordenação ocorre pelo campo “nascimento”:

```
MariaDB [(none)]> SELECT * FROM alunos ORDER BY nome e matricula;
```

**Obs.: observe o caso da Maria que possui 4 registros com datas diferentes. Também existe o caso da Ana com 2 registros com datas diferentes.**

# Inserir matricula repetida

- Tente inserir um aluno com a mesma matricula do aluno chamado Ciclano:

MariaDB [(none)]> **INSERT INTO alunos VALUES (1, "Joao", 2001);**

**Obs.: veja a mensagem de erro.**

- Veja que o Joao não foi inserido:

MariaDB [(none)]> **Select \* from alunos;**

# Erro por causa da falta de especificação das colunas

- Veja o que ocorre se não colocar a Matricula e também se não colocar a especificação das colunas “Matricula”, “nome” e “Nascimento”:

```
MariaDB [(none)]> INSERT INTO alunos VALUES (“Joao da Silva”, 2002);
```

**Obs.: veja a mensagem de erro.**

```
MariaDB [(none)]> Select * from alunos;
```

- Veja o que ocorre se não colocar a Matricula e colocar apenas a especificação das colunas “nome” e “Nascimento”:

```
MariaDB [(none)]> INSERT INTO alunos (nome, nascimento) VALUES (“Andre”, 2001);
```

```
MariaDB [(none)]> Select * from alunos;
```

# Mãos na massa 10 de 12

Chave primária com auto incremento



# Utilizando chave primária (PK) com “*AUTOINCREMENT*”

- Criar a tabela com *autoincrement*:

```
MariaDB [(none)]> CREATE TABLE estudantes (Matricula INTEGER PRIMARY KEY AUTO_INCREMENT, nome TEXT, nascimento int);
```

- Inserir os registros especificando as colunas “nome” e “Nascimento”:

```
MariaDB [(none)]> INSERT INTO estudantes (nome, nascimento) VALUES (“Fulano”, 1990);
```

```
MariaDB [(none)]> INSERT INTO estudantes (nome, nascimento) VALUES (“Ciclano”, 1999);
```

```
MariaDB [(none)]> INSERT INTO estudantes (nome, nascimento) VALUES (“Beltrano”, 1999);
```

```
MariaDB [(none)]> Select * from alunos;
```

# Utilizando chave primária (PK) com “*AUTOINCREMENT*”

- **Agora tente inserir registros sem a descrição das colunas:**

MariaDB [(none)]> `INSERT INTO estudantes VALUES ("Bob", 1995);`

**Obs.:** veja o erro que irá aparecer, para consertar utilize `"INSERT INTO estudantes (nome, nascimento) VALUES ("Bob", 1995);"`.

- **Tente inserir um aluno com a mesma matricula do Cilano:**

MariaDB [(none)]> `INSERT INTO estudantes (Matricula, nome, nascimento) VALUES (1, "Joao", 2001);`

**Obs.:** veja o erro `"Duplicate entry '1' for key 'PRIMARY'"`.

- **Veja que o Joao não foi inserido:**

MariaDB [(none)]> `Select * from estudantes;`

# Forçar a inserção da chave primária em “Autoincrement”

- Criar a tabela com *autoincrement*:

```
MariaDB [(none)]> CREATE TABLE discentes (Matricula INTEGER PRIMARY KEY AUTO_INCREMENT, nome TEXT, nascimento int);
```

- Agora tente inserir os registro impondo o valor da chave primária (PK):

```
MariaDB [(none)]> INSERT INTO discentes (Matricula, nome, nascimento) VALUES (1,“Fulano”, 1990);
```

```
MariaDB [(none)]> INSERT INTO discentes (Matricula, nome, nascimento) VALUES (2,“Ciclano”, 1999);
```

```
MariaDB [(none)]> Select * from discentes;
```

- Agora tente inserir fora da ordem decrescente da chave primária (PK):

```
MariaDB [(none)]> INSERT INTO discentes (Matricula, nome, nascimento) VALUES (9,“Alice”, 1997);
```

```
MariaDB [(none)]> INSERT INTO discentes (Matricula, nome, nascimento) VALUES (5,“Bernado”, 1998);
```

```
MariaDB [(none)]> Select * from discentes;
```

# Mãos na massa 11 de 12

Tipos e Formatos de dados e restrições  
(*Constraint*)





# Formatando dados nos campos da tabela



- Observe que “Valor\_compra” receberá valores com 2 casas decimais:

```
CREATE TABLE vendas (  
    Produto varchar(20),  
   Codigo_Compra INTEGER,  
    Valor_Compra NUMERIC(10,2)  
);
```

```
MariaDB [(none)]> desc vendas;
```

```
MariaDB [(none)]> Insert Into vendas values (“Mesa”, 1, 100);
```

```
MariaDB [(none)]> Insert Into vendas values (“Cadeira”, 2, 200.12);
```

```
MariaDB [(none)]> Insert Into vendas values (“Porta”, 4, 2000.11);
```

```
MariaDB [(none)]> select * from vendas;
```

# Utilizando restrições para inserir na tabela

- Observe que “Valor\_compra” receberá a restrição de valores maiores do que 1 mil.

```
CREATE TABLE compras (  
    Produto varchar(20),  
    Codigo_Compra INTEGER,  
    Valor_Compra NUMERIC(10,2) CHECK(Valor_Compra >= 1000);  
)
```

```
MariaDB [(none)]> desc compras;
```

```
MariaDB [(none)]> Insert Into compras values (“Mesa”, 1, 1000);
```

```
MariaDB [(none)]> Insert Into compras values (“Cadeira”, 2, 1000.12);
```

```
MariaDB [(none)]> Insert Into compras values (“Porta”, 4, 2000.11);
```

```
MariaDB [(none)]> select * from compras;
```

- Agora tente inserir valores menores que 1 mil, veja os erros:

```
MariaDB [(none)]> Insert Into compras values (“Caneta”, 5, 20.11);
```

```
MariaDB [(none)]> Insert Into compras values (“Papel A4”, 6, 5);
```

```
MariaDB [(none)]> select * from compras;
```

# Mãos na massa 12 de 12

Alterando o nome da tabela



# Alterar nome da tabela

- **Veja que clausula *Where* especifica o que será removido.**

MariaDB [(none)]> desc cidades;

MariaDB [(none)]> ALTER TABLE cidades rename to city;

- **Veja que a tabela “cidades” não existe mais.**

MariaDB [(none)]> desc cidades;

- **Agora o que existe é a tabela “city”.**

MariaDB [(none)]> desc city;

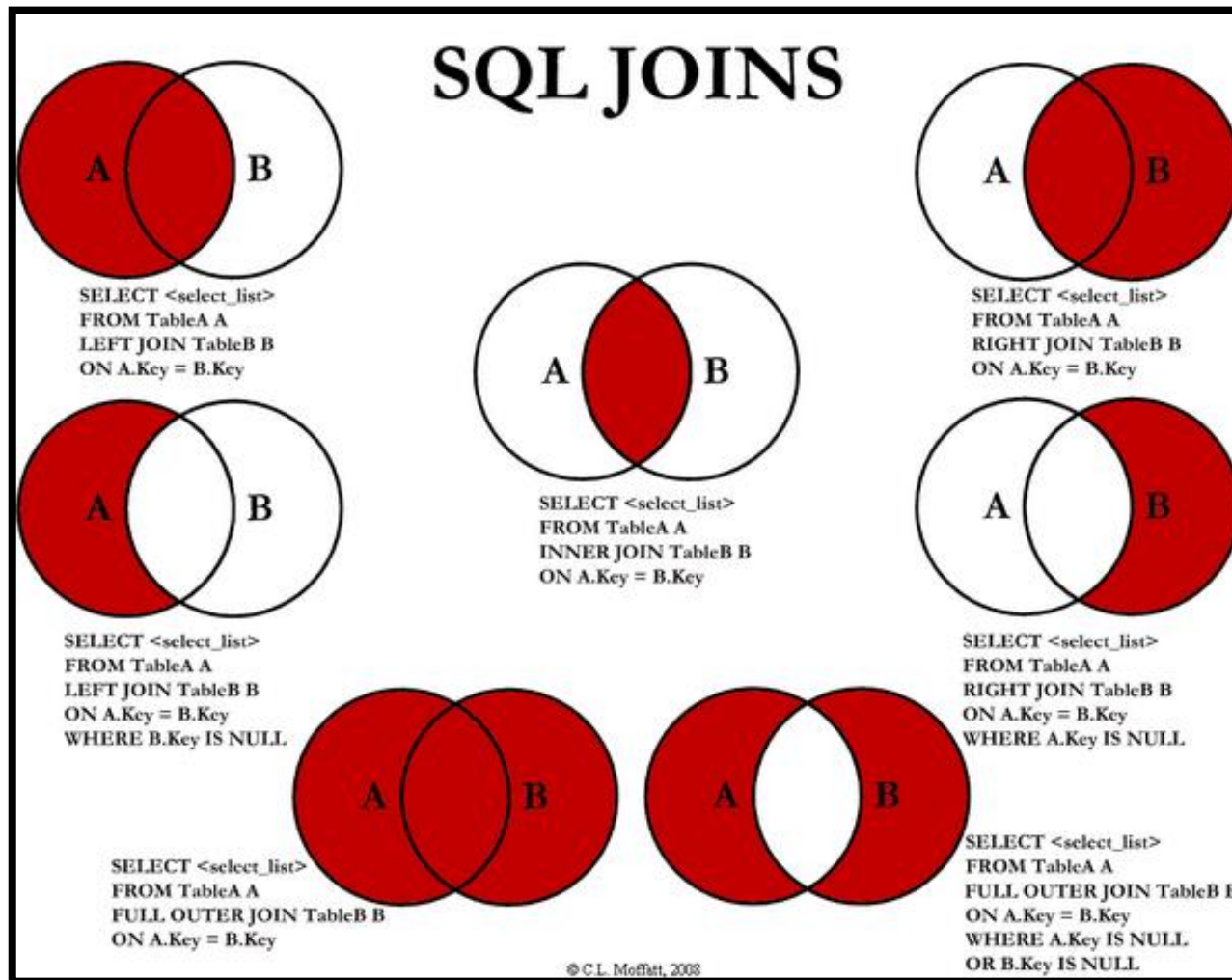
# Mãos na massa Extra 01

Tutorial de *Joins* em SQL



# Tutorial de *Joins* em SQL

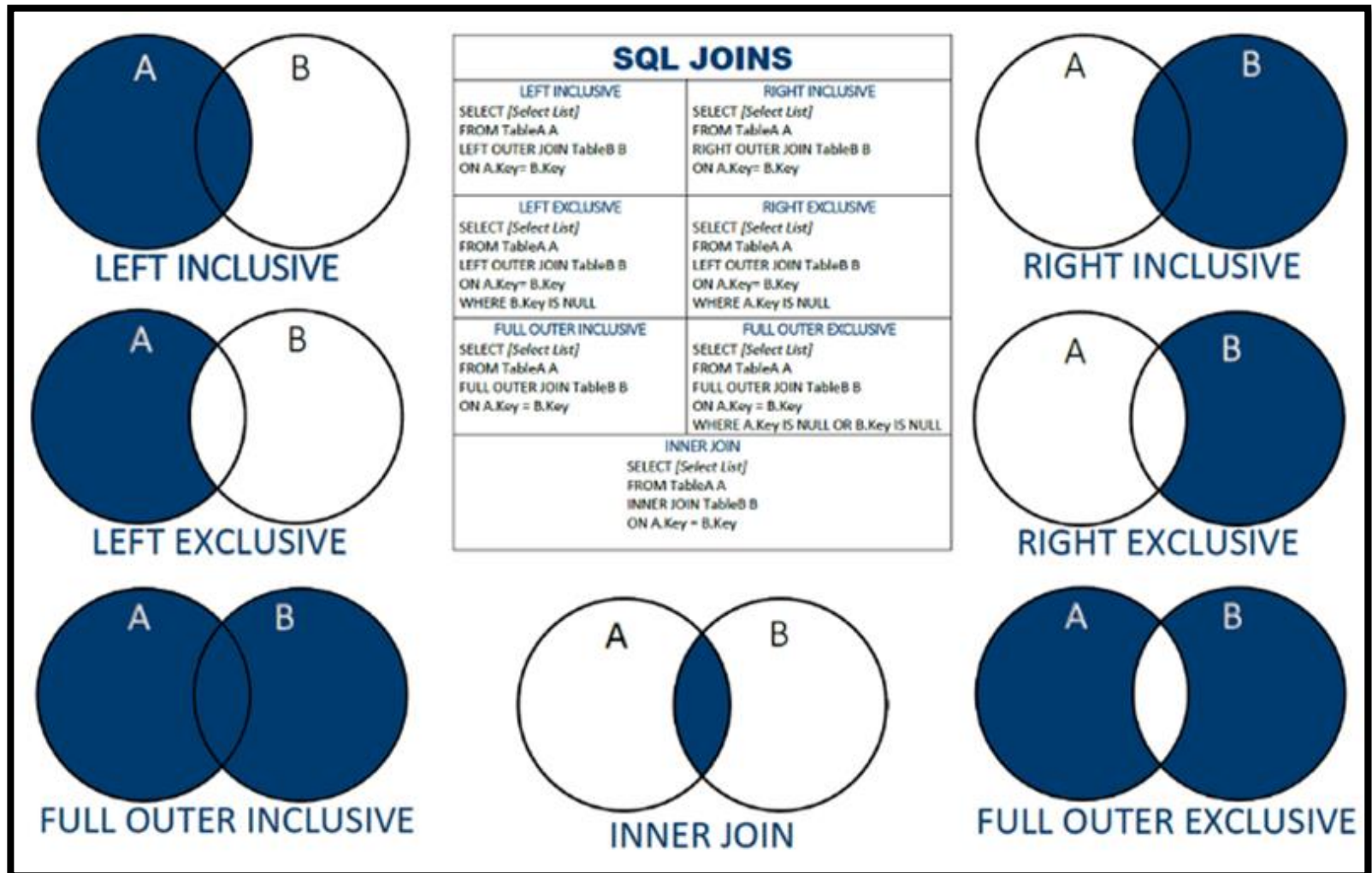
- Tutorial de Joins em SQL: Cross Join, Full Outer Join, Inner Join, Left Join e Right Join:  
<https://www.freecodecamp.org/portuguese/news/tutorial-de-joins-em-sql-cross-join-full-outer-join-inner-join-left-join-e-right-join/>





# Tutorial de *Joins* em SQL

- Tutorial de Joins em SQL: Cross Join, Full Outer Join, Inner Join, Left Join e Right Join:  
<https://www.freecodecamp.org/portuguese/news/tutorial-de-joins-em-sql-cross-join-full-outer-join-inner-join-left-join-e-right-join/>



# Tópicos para pesquisa





# Tópicos extras para pesquisar

- **Indicações para pesquisas:**
  - Conhecendo *Subquery*;
  - *Subquery* utilizando IN e NOT IN;
  - Gerando uma nova coluna com *Subquery*;
  - Selecionando dados de uma *Subquery*;
  - *Subqueries* utilizando mais de uma tabela;
- SQL: Seleção Múltiplas tabelas;
- Ordenando os registros;
- Funções de agregação;
- Agrupamentos de dados;
- Junções de tabelas;
- União entre consultas;
- Operadores e Funções;
- *Trigger* no SQL;
- *Views*;
- *Backup / Dump*;
- *Script*.

Ref.: [http://www.regilan.com.br/wp-content/uploads/2015/03/banco\\_de\\_dados\\_aula\\_05-Relacionamento-entre-TabelasPK-e-FK.pdf](http://www.regilan.com.br/wp-content/uploads/2015/03/banco_de_dados_aula_05-Relacionamento-entre-TabelasPK-e-FK.pdf)  
e <http://18.229.168.129:8080/publicacoesArquivos/publicacoesNovo.filepart>



*The End*

