

## MARIA DB ESSENTIALS

O que é Maria DB?

É um banco de dados relacional: todas as informações são relacionadas em torno de um problema que deve ser resolvido; relacionamento 1...1, relacionamento 1...N e relacionamento de N... M.

Transacional: uma sequência de processos, que quando executadas formam apenas uma ação; ACID

1. Atomicidade: todo processo deve ter um fim,
2. Consistência: todos os processos devem ser executados obedecendo todas as regras e restrições impostas,
3. Isolamento: nenhuma transação pode afetar outra em andamento,
4. Durabilidade: [ou persistência]: toda informação escrita no repositório só pode ser desfeita/refeita por outra transação.

Normalizado: nas 5 formas normais tende a ser extremamente dinâmico, porém com perda de desempenho.

### CARACTERÍSTICA DO MARIA DB

Implementação C e C++,

Multiplataforma,

Open Source,

Aceita várias linguagens de programação,

Comparação mariaDB e MySQL,

Repositório

## ESCOLHA SEU STAGE

### TIPOS DE AMBIENTE

Produção - Serviço do MariaDB no cPanel,

Homologação (sandbox) - Localhost,

Serviço dedicado.

### INSTALAÇÃO NO UBUNTU

Instalação feita através do terminal seguindo o passo a passo do site: [Produtos e Ferramentas MariaDB baixa | MariaDB](#)

Após selecionar o SO, clicar no link MariaDB Package repository

Abra o terminal e utilize as linhas de comando da página do MariaDB.

Após instalação, execute o comando: `mysql -u root -p`

CREATE DATABASE teste;

SHOW DATABASES;

USE teste;

## ESCOLHA SEU APLICATIVO



## PARAMETRIZANDO ESTAÇÃO

TESTANDO CONEXÃO - telnet nome do domínio 3306

Teste conexão: telnet localhost 3306 -> Deve ser habilitado no Windows adicionar ou remover componentes do Windows.

## COMPREENDENDO O PROCESSO

EXECUTANDO IMPORT

EXECUTANDO EXPORT

Marcar Personalizado – estrutura

## TABELAS CAMPOS E ATRIBUTOS

### TIPOS DE DADOS INTEIROS

### TIPO DE DADOS [NÚMEROS REAIS]

### TIPO DE DADOS [TEXTOS]

### TIPOS DE DADOS [DATAS E HORAS]

TIPOS DE DADOS [OTHERS] -> tipos geométricos de dados, autoincremento e null.

## OVERVIEW DDL

Linguagem de definição de dados (Data Definition language)

**TRABALHANDO COM CREATE [DATABASE] -> CRIANDO BANCO DE DADOS!**

```
CREATE DATABASE mod_essentials DEFAULT CHARACTER SET 'utf8' DEFAULT COLLATE = 'utf8_general_ci';
```

```
CREATE DATABASE IF NOT EXISTS mod_essentials DEFAULT CHARACTER SET 'utf8' DEFAULT COLLATE = 'utf8_general_ci'; ----->>> O "IF NOT EXISTS" verifica se o banco já existe
```

USE mod\_essentials; --->>> Podemos utilizar o BD

**TRABALHANDO COM CREATE [TABLE]**

```
CREATE TABLE teste(
```

```
    teste_id INT(11) NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY COMMENT 'Campo para armazenar o ID',
```

```
    teste_nome VARCHAR(255) NOT NULL,
```

```
    teste_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
```

```
)
```

**MODIFICANDO COM ALTER**

```
ALTER DATABASE mod_essentials DEFAULT COLLATE = ' ';
```

```
ALTER TABLE teste MODIFY COLUMN teste_name VARCHAR(100) NOT NULL;
```

```
ALTER TABLE teste ADD COLUMN teste_descricao TEXT AFTER teste_nome; ----->>> Acrescentou uma coluna após a coluna teste_nome.
```

```
ALTER TABLE teste ADD COLUMN teste_title TEXT FIRST teste_nome; ----->> Inseriu uma coluna antes da coluna teste_nome.
```

```
ALTER TABLE teste DROP COLUMN IF EXISTS teste_titulo; ----->>> Excluiu a coluna teste_titulo da tabela.
```

**DELETANDO COM TRUNCATE**

```
TRUNCATE TABLE teste; --->>> Apaga todos os dados da tabela
```

```
DELETE FROM teste; ----->>> Apaga todos os dados da tabela
```

## RENAME DE TABELA

```
RENAME TABLE teste TO teste_geral;
```

```
RENAME TABLE teste_geral TO usuarios, teste2 TO produtos; -->>> Renomeando duas tabelas
```

## REMOVENDO ESTRUTURA COM DROP

```
DROP TABLE usuario; -->> Apaga a tabela do banco de dados
```

## OVERVIEW DML (CRUD) - DATA MANIPULATION LANGUAGE (LINGUAGEM DE MANIPULAÇÃO DE DADOS)

### EXECUTANDO INSERT

```
INSERT INTO tbl_users ( user_name, user_lastname, user_created, user_updated, user_years_old, user_type, user_birth) VALUES ('Fábio', 'Freitas', DEFAULT, DEFAULT, '39', 'customer', 1982-06-26);
```

**INSERT INTO tbl\_users SET user\_name = 'Fábio1', user\_lastname = 'Freitas1'; --> alterando somente nome e sobre nome**

### SELECIONANDO REGISTRO COM SELECT/WHERE

SELECT \* FROM tbl\_users; -> mostra a tabela inteira.

SELECT user\_id, user\_name, user\_lastname FROM tbl\_users; -> Trans informações específicas.

SELECT

user\_id,

UPPER(user\_name),

user\_lastname,

DATE\_FORMAT(user\_created, '%d/%m/%Y %H:%i:%s') user\_created\_br

FROM tbl\_users

-- JOIN

WHERE user\_years\_old IS NOT NULL; -> trazendo as informações com a data em formato BR e filtrando os resultados que esteja com a idade nula.

**WHERE user\_years\_old BETWEEN 23 AND 25;** -> filtra as informações com idade entre 23 e 25 anos.

```
INSERT INTO tbl_usuarios (usuario_name, usuario_idade) SELECT
```

```
UPPER(user_name),
```

```
user_years_old
```

FROM tbl\_users

WHERE user\_years\_old > 23; -> Inserir trabalhando com select para migrar informação de uma tabela para outra.

#### ATUALIZANDO COM UPDATE

UPDATE tbl\_users SET user\_name = 'Robson', user\_lastname = 'V. Leite'

WHERE user\_id = 1 ---->>> Atualizando informações do usuário de ID = 1.

UPDATE tbl\_users SET user\_name = 'Robson', user\_lastname = 'V. Leite'

WHERE user\_id = 1 --->>> Alterando id do usuário

UPDATE tbl\_users u, tbl\_usuarios u2

SET

u.user\_name = 'Robson',

u2.usuario\_nome = 'Robson'

WHERE u.user\_id = u2.usuario\_id

AND u.user\_id = 1; ----->>> unindo o registro de duas tabelas através do id do usuário.

#### EXCLUINDO COM DELETE

DELETE FROM tbl\_users WHERE user\_type IS NULL; -->>> Deletando usuário com campos setado como NULL.

### OVERVIEW TCL - TOOL COMMAND LANGUAGE (LINGUAGEM DE COMANDOS E FERRAMENTAS)

#### BEGIN, COMMIT E ROLLBACK

#### EXEMPLOS PRÁTICOS DE TRANSAÇÃO

BEGIN; -> abre uma nova transação

COMMIT; -> escreve a transação dentro do BD e encerra a transação.

Exemplo:

BEGIN; Inicia uma transação

COMMIT; certo, pode salvar as informações no BD

UPDATE tbl\_users SET user\_name = 'Teste'; -> Alterou o nome dos usuários da tabela para teste

SELECT \* FROM tbl\_users; -> mostrou o resultado.

### **EXEMPLO PRÁTICO COM ROLLBACK**

Descartando as alterações realizada na transação anterior!

BEGIN;

COMMIT;

ROLLBACK; -> Encerra a transação

UPDATE tbl\_users SET user\_name = 'Fábio' WHERE user\_id = 1;

SELECT \* FROM tbl\_users;