

1. Introdução à DDL: Definindo a Estrutura dos Dados

A **DDL (Data Definition Language)** é um subconjunto de comandos SQL responsável por definir e gerenciar a estrutura dos objetos de um banco de dados. Em vez de manipular os dados em si, a DDL trabalha com o "esquema" do banco de dados, ou seja, a sua arquitetura.

As principais tarefas realizadas com comandos DDL incluem:

- **Criação de objetos:** Tabelas, views (visões), índices, etc.
- **Alteração de objetos:** Modificação da estrutura de tabelas existentes.
- **Exclusão de objetos:** Remoção de tabelas e outros elementos do banco de dados.

Os comandos DDL mais importantes no MariaDB são:

- **CREATE:** Utilizado para criar novos objetos.
- **ALTER:** Usado para modificar a estrutura de objetos existentes.
- **DROP:** Empregado para excluir objetos.
- **TRUNCATE:** Remove todos os registros de uma tabela rapidamente, mantendo a estrutura.

2. Tipos de Dados no MariaDB

A escolha correta do tipo de dado otimiza o armazenamento e garante a integridade dos dados.

Categorias de Tipos de Dados:

- **Numéricos:**
 - INT ou INTEGER: Números inteiros.
 - TINYINT: Inteiro muito pequeno.
 - DECIMAL(p, s): Números decimais com precisão fixa.
 - FLOAT, DOUBLE: Números de ponto flutuante.
- **Texto (String):**
 - VARCHAR(n): String de tamanho variável.
 - CHAR(n): String de tamanho fixo.
 - TEXT: Para textos longos.
- **Data e Hora:**
 - DATE: Armazena uma data ('AAAA-MM-DD').

- TIME: Armazena uma hora ('HH:MM:SS').
- DATETIME: Armazena data e hora.
- TIMESTAMP: Similar ao DATETIME, mas com fuso horário.
- YEAR: Armazena um ano.

3. Constraints (Restrições): Garantindo a Integridade dos Dados

Constraints são regras aplicadas às colunas para garantir a precisão e a confiabilidade dos dados.

- **NOT NULL:** Garante que uma coluna não pode ter um valor nulo.
- **UNIQUE:** Assegura que todos os valores em uma coluna são diferentes.
- **PRIMARY KEY:** Uma combinação de NOT NULL e UNIQUE. Identifica unicamente cada registro.
- **FOREIGN KEY:** Garante a integridade referencial entre duas tabelas.
- **CHECK:** Verifica se os valores satisfazem uma condição.
- **DEFAULT:** Define um valor padrão para uma coluna.
- **AUTO_INCREMENT:** Gera automaticamente um número sequencial único.

4. O Comando CREATE: Criando Estruturas

Criando um Banco de Dados

```
CREATE DATABASE IF NOT EXISTS livraria;

USE livraria;
```

Criando Tabelas (CREATE TABLE)

Exemplo 1: Tabela de Autores

```
CREATE TABLE autores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    nacionalidade VARCHAR(50) DEFAULT 'Brasileira',
    data_nascimento DATE
);
```

Exemplo 2: Tabela de Livros com Chave Estrangeira

```
CREATE TABLE livros (  
    isbn VARCHAR(13) PRIMARY KEY,  
    titulo VARCHAR(255) NOT NULL,  
    ano_publicacao INT,  
    preco DECIMAL(10, 2),  
    autor_id INT,  
    FOREIGN KEY (autor_id) REFERENCES autores(id)  
);
```

5. O Comando ALTER: Modificando Estruturas Existentes

5.1. Adicionando uma Coluna

```
ALTER TABLE livros ADD COLUMN genero VARCHAR(50);
```

5.2. Modificando uma Coluna

```
ALTER TABLE livros MODIFY COLUMN titulo VARCHAR(300) NOT NULL;
```

5.3. Renomeando uma Coluna

```
ALTER TABLE livros CHANGE COLUMN preco valor DECIMAL(10, 2);
```

5.4. Removendo uma Coluna

```
ALTER TABLE livros DROP COLUMN genero;
```

5.5. Adicionando uma Constraint

```
ALTER TABLE livros ADD CONSTRAINT uk_titulo UNIQUE (titulo);
```

6. O Comando DROP: Excluindo Estruturas

Atenção: Esta operação é irreversível.

6.1. Excluindo uma Tabela

```
DROP TABLE IF EXISTS tabela_temporaria;
```

6.2. Excluindo um Banco de Dados

```
DROP DATABASE IF EXISTS livraria_backup;
```

7. O Comando TRUNCATE: Limpando Tabelas

Remove todos os registros de uma tabela de forma rápida, mantendo sua estrutura.

```
TRUNCATE TABLE livros;
```