

Banco de Dados – C07

Felipe Tagawa Reis

Comandos DDL

Linguagem SQL

SQL COMMANDS

1

DATA DEFINITION LANGUAGE (DDL)

- CREATE
- DROP
- ALTER
- TRUNCATE

2

DATA MANIPULATION LANGUAGE (DML)

- INSERT
- UPDATE
- DELETE

3

DATA CONTROL LANGUAGE (DCL)

- GRANT
- REVOKE

4

TRANSACTION CONTROL LANGUAGE (TCL)

- COMMIT
- ROLLBACK
- SAVEPOINT

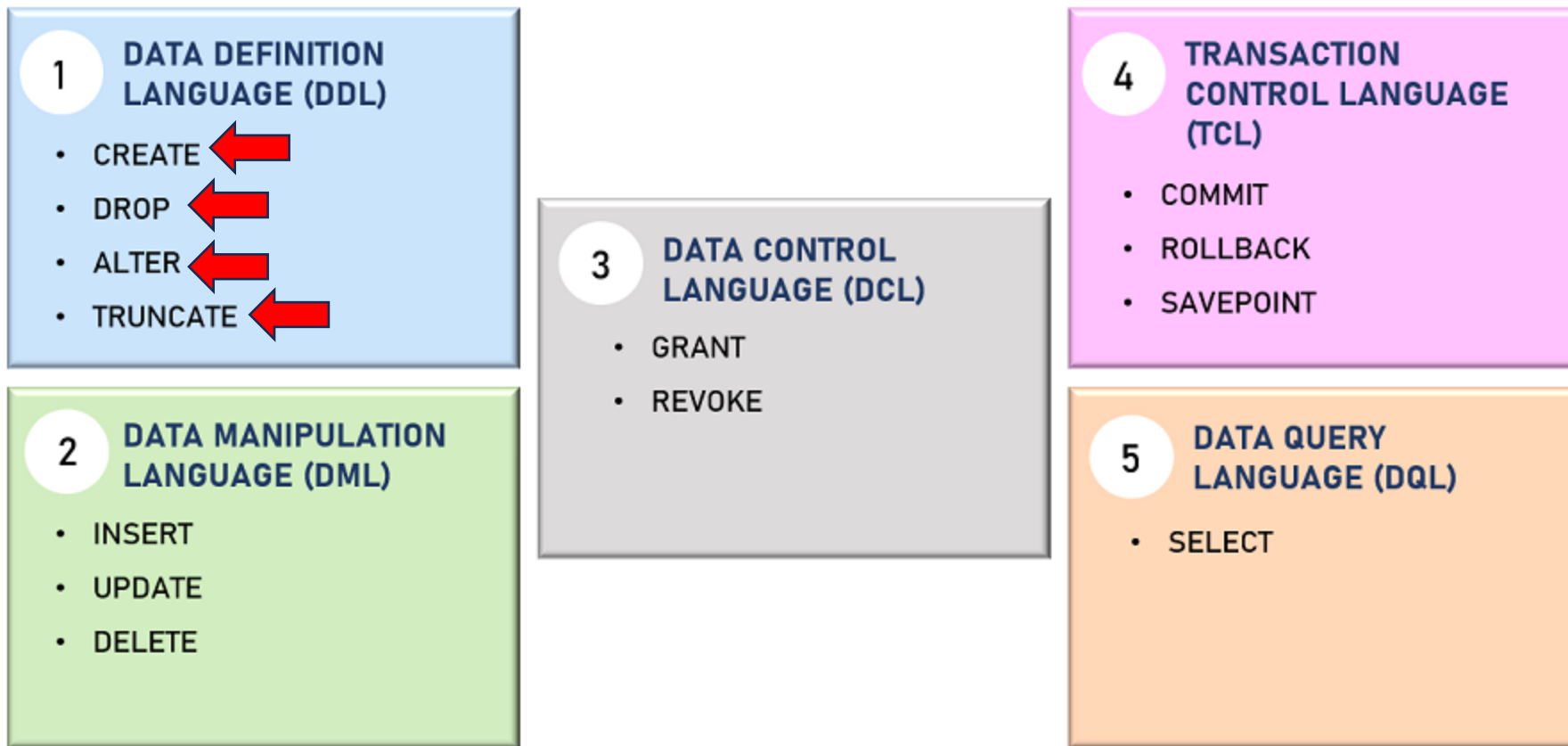
5

DATA QUERY LANGUAGE (DQL)

- SELECT

Aula de Hoje

SQL COMMANDS



CREATE DATABASE BD;

Comando CREATE

Objeto

CREATE USER 'Monitor' IDENTIFIED BY 'senha';

```
CREATE TABLE Cliente(  
    codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    telefone VARCHAR(15) NOT NULL  
);
```

codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY

Nome do
Atributo

Tipo do
Atributo

Indica que o
Atributo não
pode ser
nulo

Para Atributos
numéricos, esse
modificador incrementa
em 1 no último inserido,
sem a necessidade de
inserir manualmente

Indica que este
é um Atributo
chave para a
tabela em
questão

Comando DROP

Comando de exclusão de objetos presentes
no banco de dados

```
DROP DATABASE IF EXISTS BD;
```

Objeto

```
DROP USER 'Monitor'@'%';
```

Objeto

```
DROP TABLE pessoa;
```

Objeto

Comando ALTER

Comando de modificação de objetos presentes no banco de dados

ALTER **TABLE** pessoa **ADD** saldo **DOUBLE**;

Objeto

Adiciona algo novo ao objeto

ALTER **TABLE** pessoa **DROP COLUMN** telefone;

Exclui algo existente

ALTER **TABLE** pessoa **MODIFY** nome **VARCHAR(35)** **NOT NULL**;

Altera algo existente:

- Tipo de dado, capacidade do dado ou restrição

Comando SHOW

Comando para mostrar informações de Objetos presentes
no banco de dados

SHOW DATABASES;

Objetos

SHOW TABLES;

Objetos

SHOW GRANTS FOR 'Monitor';

Objetos

Comando TRUNCATE

Comando para remover dados de uma tabela sem a necessidade de deletar sua estrutura

```
TRUNCATE TABLE Pedido;
```

- Muito mais rápido que o DELETE;
- Ideal para casos de exclusão geral de dados de uma tabela, a fim de iniciar novos registros com novos valores de ID.

Modificadores Úteis

- IF EXISTS – completamente opcional.
- IF NOT EXISTS – fundamental para evitar erros relativos à duplicidade de cláusulas.

```
DROP DATABASE IF EXISTS BD;
```

```
DROP DATABASE BD;
```

➤ Comando USE

Instrui o SGBD a utilizar o banco de dados especificado para executar os comandos.

```
CREATE DATABASE teste;
```

```
USE teste;
```

```
CREATE TABLE IF NOT EXISTS `Pedido` (  
    `idPedido` INT AUTO_INCREMENT PRIMARY KEY,  
    `dataPedido` DATE NOT NULL,  
    `valorTotal` DECIMAL (10,2) NOT NULL  
);
```

```
CREATE USER IF NOT EXISTS 'Monitor'@'%'  
IDENTIFIED BY '1234';
```

Regras Úteis

➤ **Check** – Regra aplicada a colunas para garantir a integridade dos dados, rejeitando valores que não cumprem condições específicas.

➤ **UNIQUE** – Regra que impede valores duplicados em uma ou mais colunas.

```
CREATE TABLE `Pessoa` (  
    `senha` VARCHAR(40) PRIMARY KEY,  
    `nome` VARCHAR(30),  
    `cidade` VARCHAR(30) UNIQUE,  
    `idade` INT CHECK (`idade` > 18),  
    `Data` DATE  
);
```

Regras Úteis

➤ **Default** – Define um valor padrão quando nenhum valor é informado.

➤ **NOT NULL** – Impede valores nulos para atributos da tabela.

```
CREATE TABLE `Personagem` (  
    `idPersonagem` INT AUTO_INCREMENT PRIMARY KEY,  
    `nome` VARCHAR(30) NOT NULL,  
    `danoArma` INT DEFAULT 20,  
    `idade` INT,  
    `descricao` TEXT  
);
```

Regras Úteis

- INDEX – Melhora desempenho de consultas.

```
CREATE TABLE `Personagem` (  
  `idPersonagem` INT AUTO_INCREMENT PRIMARY KEY,  
  `nome` VARCHAR(30) NOT NULL,  
  `danoArma` INT DEFAULT 20,  
  `idade` INT,  
  `descricao` TEXT  
);
```

```
CREATE INDEX idx_personagem_nome  
ON Personagem(nome);
```

```
SELECT * FROM Personagem  
WHERE nome = 'Artorias';
```

Consulta

	idPersonagem	nome	danoArma	idade	descricao
▶	1	Artorias	30	NULL	Chefe

Tipo Útil

- ENUM – Tipo de dado que restringe uma coluna a um conjunto predefinido de valores.

```
CREATE TABLE `Pedido` (  
  `idPedido` INT AUTO_INCREMENT PRIMARY KEY,  
  `dataPedido` DATE NOT NULL,  
  `valorTotal` DECIMAL (10,2) NOT NULL,  
  `formaPagamento` ENUM('PIX', 'CARTAO', 'BOLETO') NOT NULL,  
  `statusPedido` ENUM('PENDENTE', 'PAGO', 'CANCELADO') DEFAULT 'PENDENTE'  
);
```

Exercício

Utilizando os conceitos abordados nesta aula, faça o que se pede:

1. Banco de Dados:

- Crie um banco de dados chamado 'streaming_db' verificando sua existência. Caso exista, exclua antes mesmo de criá-lo;
- Execute o comando para utilizar seu banco de dados;

2. Tabela:

- Crie uma nova tabela 'Filme', que deve conter um identificador único obrigatório e auto-incremental, um título (20 caracteres), um gênero (30 caracteres) e o ano de lançamento (obrigatório);

3. Manutenção:

- Inclua uma nova coluna para a classificação etária;
- Percebeu-se que 20 caracteres para o título são insuficientes, aumente para 60;

Exercício

Utilizando os conceitos abordados nesta aula, faça o que se pede:

4. Operação de Dados:

- Faça a inserção de 3 filmes;
- Desative a trava de segurança de updates;
- Atualize o gênero de um dos filmes inseridos;
- Delete apenas o filme que tiver o primeiro ID;
- Mostre todos os dados da tabela para validar as operações.

OBS: Não se esqueça de usar `SET SQL_SAFE_UPDATES = 0`.

5. Finalização:

- Use o comando `SHOW` para visualizar as tabelas;
- Use UM comando para remover todos os Registros da tabela.

AULA
CONCLUÍDA!