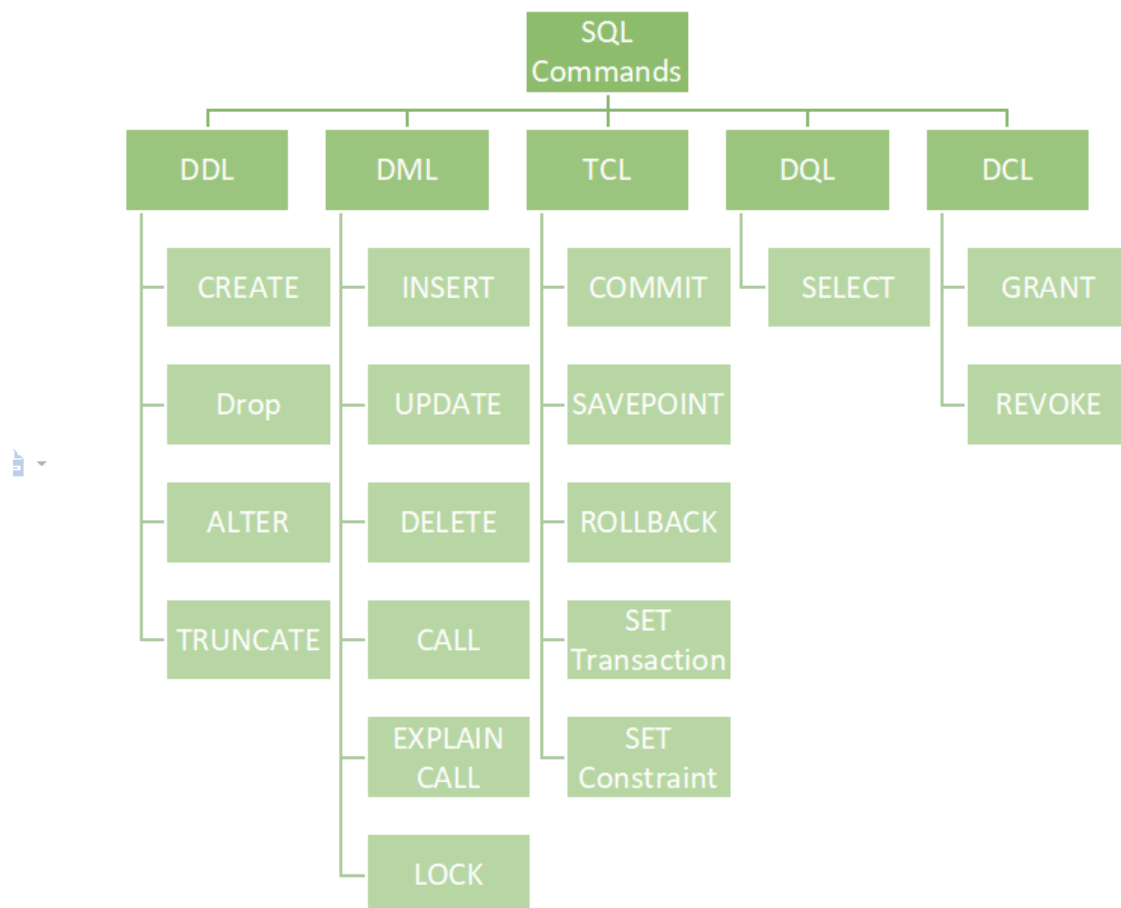


Comandos Básicos SQL

Introdução

Tipos de Comandos



Comandos **CREATE** e **USE**

Como já vimos antes, enquanto o comando **CREATE** é usado para criar bancos de dados, tabelas e usuários, o **USE** declara que todo código que vier em seguida será feito em um determinado banco de dados. Por exemplo:

```
CREATE DATABASE IF NOT EXISTS cinema;
USE cinema;

CREATE TABLE IF NOT EXISTS Filme (
    id INT AUTO_INCREMENT,
    nomeFilme VARCHAR(100) NOT NULL,
    genero VARCHAR(50) NOT NULL,
    tempo INT NOT NULL,
    restricaoIdade INT NOT NULL,
    PRIMARY KEY (id)
);
```

Constraints **ON DELETE** e **ON UPDATE**

Os *constraints* **ON UPDATE** e **ON DELETE** definem o que deverá ser feito ao se excluir ou modificar um registro em outra tabela relacionada pela chave estrangeira (*foreign key*). Nestes casos, podemos atribuir os seguintes atributos:

- **NO ACTION** ou **RESTRICT** → Nada acontece. Feijoadá.
- **SET NULL** → O valor da chave estrangeira passará a ser nulo.
- **CASCADE** → A alteração na tabela inicial se estende ao registro da tabela com a chave estrangeira. Ou seja, acontece alteração nas duas.

Segue um exemplo abaixo:

```
CREATE TABLE IF NOT EXISTS Ator (
    id INT AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    dataNasc DATE NOT NULL,
    sexo VARCHAR(20) NOT NULL,
    idFilme INT,
    PRIMARY KEY (id),
    CONSTRAINT fk_ator_filme
        FOREIGN KEY (idFilme)
        REFERENCES Filme(id)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);
```

Comando **DROP**

O comando **DROP** é usado para apagar um banco de dados ou tabelas dentro dele. Pode ser usado também para apagar colunas, quando usado com em conjunto com o comando **ALTER**, conforme mostrado no próximo tópico.

```
CREATE TABLE IF NOT EXISTS PC (  
    id INT NOT NULL AUTO_INCREMENT,  
    modelo VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);  
  
DROP TABLE PC;  
DROP TABLE IF EXISTS PC;  
# DROP DATABASE IF EXISTS cinema;
```

Comando **ALTER**

O comando **ALTER** usado para alterar colunas de uma tabela.

- **ADD** → Adiciona uma coluna nova na tabela.
- **DROP** → Apaga a coluna da tabela.
- **CHANGE** → Altera a coluna da tabela, como nome, tipo, *constraints*...

```
ALTER TABLE Filme CHANGE nomeFilme titulo VARCHAR(50) NOT NULL;  
ALTER TABLE Filme ADD diretor VARCHAR(100);  
ALTER TABLE Filme DROP restricaoIdade;
```

Exercício

```
# DROP DATABASE IF EXISTS cinema;  
CREATE DATABASE IF NOT EXISTS cinema;  
USE cinema;  
  
CREATE TABLE IF NOT EXISTS Filme (  
    id INT AUTO_INCREMENT,  
    nomeFilme VARCHAR(100) NOT NULL,  
    genero VARCHAR(50) NOT NULL,  
    tempo INT NOT NULL,  
    restricaoIdade INT NOT NULL,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE IF NOT EXISTS Ator (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    dataNasc DATE NOT NULL,  
    sexo VARCHAR(20) NOT NULL,  
    idFilme INT,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_ator_filme  
        FOREIGN KEY (idFilme)  
        REFERENCES Filme(id)
```

```

ON DELETE SET NULL
ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS PC (
    id INT NOT NULL AUTO_INCREMENT,
    modelo VARCHAR(100) NOT NULL,
    PRIMARY KEY (id)
);

DROP TABLE IF EXISTS PC;

ALTER TABLE Filme CHANGE nomeFilme titulo VARCHAR(50) NOT NULL;
ALTER TABLE Filme ADD diretor VARCHAR(100);
ALTER TABLE Filme DROP restricaoIdade;

```

1. Adicione as colunas `pais` e `idioma` na tabela `Ator`. Ambos `VARCHAR` com tamanho limite de 50 caracteres.
2. Altere a coluna `dataNasc` para `idade`, do tipo `INT` e `NOT NULL` na tabela `Ator`.
3. Remova a coluna `sexo` da tabela `Ator`.
4. Entre com os dados abaixo nas respectivas tabelas:

Filme				Ator					
id	titulo	genero	tempo	id	nome	idade	idFilme	pais	idioma
1	Jogos Vorazes	ação	143	1	Jennifer Lawrence	30	1	EUA	ingles
2	O destino de Júpiter	ficção científica	127	2	Josh Hutcherson	28	1	EUA	ingles
				3	Bae Doona	41	2	Coréia do Sul	coreano
				4	Mila Kunis	35	2	Ucrânia	ucraniano
				5	Eddie Redmayne	38	2	Reino Unido	ingles

5. Altere a `idade` da atriz Mila Kunis para 37.
6. Apague o registro do ator de `id` igual a 5.
7. Altere o `id` para 43 do `Filme` com `id` igual a 1.
8. Apague o `Filme` de `id` igual a 2.
9. Altere apenas o `nome` da coluna `pais` para `paisDeOrigem`.
Mantenha o mesmo tipo de dado a ser guardado e `NOT NULL`.
10. Remova a coluna `idade` da tabela `Ator`.

Respostas

```

USE cinema;

# Exercício 1
ALTER TABLE Ator ADD pais VARCHAR(100), ADD idioma VARCHAR(100);

# Exercício 2
ALTER TABLE ATOR CHANGE dataNasc idade INT NOT NULL;

# Exercício 3
ALTER TABLE Ator DROP sexo;

# Exercício 4
INSERT INTO Filme (titulo, genero, tempo) VALUES
    ("Jogos Vorazes", "ação", 143),
    ("O Destino de Júpiter", "ficção científica", 127);
INSERT INTO ATOR (nome, idade, idFilme, pais, idioma) VALUES
    ("Jennifer Lawrence", 30, 1, "EUA", "inglês"),
    ("Josh Hutcherson", 28, 1, "EUA", "inglês"),
    ("Bae Doona", 41, 2, "Coreia do Sul", "coreano"),
    ("Mila Kunis", 35, 2, "Ucrânia", "ucraniano"),
    ("Eddie Redmayne", 38, 2, "Reino Unido", "inglês");

# Exercício 5
UPDATE Ator SET idade = 37 WHERE nome = "Mila Kunis";

# Exercício 6
DELETE FROM Ator WHERE id = 5;

# Exercício 7
UPDATE Filme SET id = 43 WHERE id = 1;

# Exercício 8
DELETE FROM Filme WHERE id = 2;

# Exercício 9
ALTER TABLE Ator CHANGE pais paisDeOrigem VARCHAR(100) NOT NULL;

# Exercício 10
ALTER TABLE Ator DROP idade;

SELECT * FROM Filme;
SELECT * FROM Ator;

```