

# Introdução ao Banco de Dados

André Luyde da Silva Souza

Aula 01: Introdução

# Linguagem SQL

- SQL (Structured Query Language) é uma linguagem comercial de criação e manipulação de bancos de dados relacionais.
- Tem como base a álgebra relacional e cálculo relacional.
- Reconhecido por todos os SGBDs relacionais existentes.

# Linguagem SQL

- DDL (Data Definition Language): linguagem de definição de dados.
  - Define esquemas, tabelas, chaves primárias e estrangeiras, exclusão de esquemas, tabelas e colunas, alteração.
- DML (Data Manipulation Language): linguagem de manipulação de dados.
  - Consulta, inserção dos dados do BD, exclusão de dados, alteração de dados.
- DCL (Data Control Language): linguagem de controle de dados.
  - Define permissões

# Linguagem SQL

- SQL = DDL + DML + DCL
- Principais comandos:
  - DDL (Linguagem de definição de dados):
    - CREATE TABLE
    - DROP TABLE
    - ALTER TABLE
  - DML (Linguagem de manipulação de dados):
    - SELECT, INSERT, UPDATE, DELETE
- Conceitos:
  - Table = Relação (tabela)
  - Row = tupla (linha)
  - Column = atributo (coluna)

# Linguagem SQL

- A DDL permite não só a especificação de um conjunto de relações, como também informações acerca de cada uma das relações, incluindo:
  - O esquema de cada relação;
  - O domínio dos valores associados a cada atributo;
  - As regras de integridade (chave, entidade, referencial);
  - Informações acerca de cada uma das relações:
    - O conjunto de índices para manutenção de cada relação
    - Informações sobre segurança e autoridade sobre cada relação
    - A estrutura de armazenamento físico de cada relação no disco

# DDL: CREATE DATABASE/ESQUEMA

- Em SQL uma base de dados (ou esquema) é identificada através de um nome. Os elementos do esquema incluem tabelas, restrições, etc.

## Sintaxe

```
CREATE SCHEMA <NOME_ESQUEMA>
```

- Exemplos:
  - CREATE SCHEMA Empresa;
  - CREATE SCHEMA Universidade;
  - CREATE SCHEMA Hospital;

# DDL: DROP DATABASE

- Para excluir uma base de dados (ou um esquema) existente, é necessário usar o comando DROP.

## Sintaxe

```
DROP <NOME_ESQUEMA>;
```

- Exemplos:
  - DROP Empresa;
  - DROP Universidade;
  - DROP Hospital;
- Cuidado ao remover uma base de dados. TODAS as tabelas, relacionamentos e dados contidos nela serão permanentemente removidos.

# DDL: TIPOS DE DOMÍNIOS

## ■ Numéricos:

- **INT** : é um inteiro, abreviação da palavra, integer (em inglês).
- **NUMERIC** ( $p$ ,  $d$ ): é um numero de ponto fixo cuja precisão é definida pelo usuário. O número consiste de  $p$  dígitos (mais o sinal), sendo que  $d$  dos  $p$  dígitos estão à direita do ponto decimal.
  - Ex.: `numeric(6,2)`: 3000,00
- **DOUBLE, FLOAT**: números reais e números de ponto flutuante.



# DDL: TIPOS DE DOMÍNIOS

- **Caracter/String:**

- **CHAR(n):** é uma cadeia de caracteres de tamanho fixo, com o tamanho n definido pelo DBA. Abreviação de character (em inglês).
- **VARCHAR(N):** é uma cadeia de caracteres de tamanho variável, com o tamanho n máximo definido pelo DBA. Abreviação de character varying (em inglês).

- **Booleano:**

- **BOOL, BOOLEAN:** assume dois valores, falso e verdadeiro (*true & false*).

# DDL: TIPOS DE DOMÍNIOS

## ■ Data/Tempo:

- **DATE**: tipo de dado que contém um ano (com 4 dígitos) mês e dia do mês.
- **TIME**: representa horário, em horas, minutos e segundos.
- OBS: Alguns SGBDs oferecem o domínio DATETIME que contém a data (ano, mês e dia) e horário (horas, minutos, segundos e milisegundos).

# DDL: TIPOS DE DOMÍNIOS

## ■ OBSERVAÇÕES

- Normalmente utiliza-se a comparação entre valores de domínios compatíveis;
- O valor nulo é um membro de todos os domínios;
- A SQL permite que a declaração de domínio de um atributo inclua a especificação de **not null**, proibindo, assim, a inserção de um valor nulo para esse tipo de atributo.

# DDL: CREATE TABLE (Cria uma tabela)

- SQL oferece instruções para definição do esquema da base de dados:
  - Define a estrutura de uma tabela, suas restrições de integridade e cria a tabela vazia

## Sintaxe

```
CREATE TABLE <nome_tabela> (...);
```

- Ex: CREATE TABLE empregado (...);

## DDL: DROP TABLE (Remove tabela)

- Remove todos os dados e a própria relação (estando vazia ou não).

### Sintaxe

```
DROP TABLE <nome_tabela>;
```

- Ex.: DROP TABLE empregado;

## DDL: CREATE TABLE - SINTAXE

- `CREATE TABLE FUNCIONARIO (Matric INT NOT NULL, Nome VARCHAR(30) NOT NULL, Salario DOUBLE NOT NULL, Cargo VARCHAR(15) DEFAULT 'Analista', Estado CHAR(2) NOT NULL, Idade INT NOT NULL, Cod_Depto INT NOT NULL, CONSTRAINT chavePrimFunc PRIMARY KEY (Matric), CONSTRAINT chaveEstrFuncDepto FOREIGN KEY (Cod_Depto) references DEPARTAMENTO );`
- `CREATE TABLE DEPARTAMENTO ( Cod_Depto INT NOT NULL, Desc_Depto VARCHAR(30), CONSTRAINT chavePrimDep PRIMARY KEY (Cod_Depto) );`

## DDL: CREATE TABLE - SINTAXE

- Criação de um campo que gere códigos automáticos não é padrão SQL.
- No MySQL basta colocar a palavra `AUTO_INCREMENT` na criação do campo.
- Exemplo:
  - `CREATE TABLE cidade ( cod_cidade INT NOT NULL AUTO_INCREMENT, nome_cidade VARCHAR(100) NOT NULL, CONSTRAINT cidade_pk PRIMARY KEY (cod_cidade) );`

# DDL: ALTER TABLE

- SQL oferece instruções para definição do esquema da base de dados:
- ALTER TABLE:
- Usado para alterar o esquema da relação.
- permite modificar a definição de uma tabela
  - Inserir, Alterar e Excluir
  - Atributos e restrições de integridade atributos chave não podem ser removidos;

## Sintaxe

```
ALTER TABLE <NOME_DA_TABELA> ...;
```



# DDL: ALTER TABLE

- Sintaxe básica para adicionar uma coluna ou restrição:
  - ALTER TABLE <nome\_tabela> ADD <coluna ou restrição>;
  - **Ex.:** ALTER TABLE empregado ADD identidade varchar(10);
- Sintaxe básica para remover uma coluna ou restrição:
  - ALTER TABLE <nome\_tabela> DROP <coluna ou restrição>;
  - **Ex.:** ALTER TABLE empregado DROP COLUMN ultimo\_nome;
- Sintaxe básica para modificar uma coluna:
  - ALTER TABLE <nome\_tabela> MODIFY <nome\_coluna> <código de criação> ;
  - **Ex.:** ALTER TABLE empregado MODIFY Sexo Genero;

# DDL: ALTER TABLE

- Observe-se que:
  - A instrução adiciona uma nova coluna com o valor vazio para todas as linhas, isto é, sem nenhum valor armazenado.
  - O mesmo acontece quando há a criação de uma tabela... A princípio ela não está “povoada”, está vazia, ausente de valores.
  - Os valores para as diversas linhas devem ser adicionadas através de instruções da DML (INSERT INTO...).

## DDL: ALTER TABLE - CHAVES

- ALTER TABLE permite incluir ou excluir restrições de chave
  - **Ex.:** ALTER TABLE Representantes DROP PRIMARY KEY
- Para excluir uma chave estrangeira é necessário que ela tenha recebido um nome na sua definição (CONSTRAINT)
  - **Ex:** ALTER TABLE Representantes ADD CONSTRAINT rep\_filial\_fk FOREIGN KEY (codigo\_filial) REFERENCES Filial (cod\_filial)
  - ALTER TABLE Representantes DROP CONSTRAINT rep\_filial\_fk

# DDL: Atributos, Primary Key, Foreign Key

## Chave primária

- PRIMARY KEY (PK)

## Restrições de atributos

- NOT NULL (não nulo) - NN
- DEFAULT <value>
- UNIQUE – UQ
- AUTO INCREMENTO – AI

## Chave estrangeira

- FOREIGN KEY (FK)
- Cláusulas:
- ON DELETE
- ON UPDATE
  - CASCADE
  - SET NULL
  - SET DEFAULT

## DDL: CHAVE ESTRANGEIRA (FK)

- Caso nada seja especificado na definição de chave estrangeira, uma linha que contenha uma chave primária referenciada na chave estrangeira não pode ser excluída (regra RESTRICT)
  - RESTRIÇÃO DE INTEGRIDADE REFERENCIAL.

# DDL:FK - ON DELETE

## ■ CASCADE

- Pode ser definida a propagação da exclusão da linha para as linhas que a referenciam.

### Sintaxe:

- ... FOREIGN KEY ... REFERENCES ... ON DELETE CASCADE

# DDL:FK - ON DELETE

- RESTRICT (padrão)
  - Especifica que uma tupla (linha) não pode ser excluída caso existam chaves estrangeiras que a referenciem

## Sintaxe:

- ... FOREIGN KEY ... REFERENCES ... ON DELETE RESTRICT

## DDL:FK - ON DELETE

- Ao excluir uma tabela, a utilização do CASCADE (padrão) informa se a exclusão deve ser propagada ou não para valores definidos com base na tabela (referencias).

### Exemplo:

- DROP TABLE embarque RESTRICT
- Exclui a tabela somente se não existirem valores definidos com base na tabela.



# DDL:FK - ON UPDATE

## ■ CASCADE

- Alteração da chave primária é propagada para as chaves estrangeiras que a referenciam.

### Sintaxe:

- FOREIGN KEY ... REFERENCES ... ON UPDATE CASCADE

# DDL: CREATE TABLE - SINTAXE

## ■ RESTRICT

- Uma chave primária não pode ser alterada, caso existam chaves estrangeiras que a referenciem.

### Sintaxe:

- FOREIGN KEY ... REFERENCES ... ON UPDATE RESTRICT

# PRATICANDO...

- CRIE A DDL PARA A RELAÇÃO (TABELA) ABAIXO - NÃO SE ESQUEÇA DE CRIAR UMA BASE DE DADOS ANTES DA CRIAÇÃO DA TABELA:
- Nome da tabela: **Peca**

Cod_Peca	Nome_Peca	Preco	Qte
56	Porca	23,90	10
99	Parafuso	56,99	5
200	Prego	80,00	0

- NA TABELA CRIADA, ADICIONE UMA NOVA COLUNA CHAMADA "TIPO", SENDO O DOMÍNIO VARCHAR(25).

## PRATICANDO ... RESPOSTA

- 1 CREATE SCHEMA LOJAPECAS;
- 2 CREATE TABLE peca ( cod\_pecas INT NOT NULL,  
nome\_pecas VARCHAR(20) NOT NULL,  
preco DOUBLE,  
quantidade INT,  
CONSTRAINT peca\_pk PRIMARY KEY (cod\_pecas));
- 3 ALTER TABLE peca ADD tipo varchar(25);

# Dúvidas



Utilizem o fórum da disciplina que responderei a dúvida de todos.