

# Universidade Presbiteriana Mackenzie

Banco de Dados - Aula 10

## Linguagem SQL CREATE e DROP TABLE

**Profa. Elisângela Botelho Gracias**

Faculdade de Computação e Informática

# Roteiro da Apresentação

- Introdução
- CREATE TABLE
- DROP TABLE



# Roteiro da Apresentação

- Introdução
- CREATE TABLE
- DROP TABLE



# Introdução

- Os comandos da linguagem SQL podem ser divididos em três classes
  - Linguagem de Definição de Dados (DDL)
  - Linguagem de Manipulação de Dados (DML)
  - Linguagem de Controle de Dados (DCL)

# Introdução

- Linguagem de Definição de Dados (DDL)
  - inclui comandos para definir, alterar e remover tabelas e índices



# Introdução

- Os seguintes comandos pertencem à Linguagem de Definição de Dados (DDL):
  - CREATE TABLE: criação de tabelas
  - DROP TABLE: remoção de tabelas

# Roteiro da Apresentação

- Introdução
- **CREATE TABLE**
- DROP TABLE



# CREATE TABLE

- **CREATE TABLE**

- cria uma tabela, dando-lhe um nome e especificando seus atributos, chave primária e chave estrangeira (se for o caso)
- para cada atributo é definido um nome, um domínio e, se necessário, uma restrição



# CREATE TABLE

- Sintaxe do comando CREATE TABLE

```
CREATE TABLE nome_tabela  
(nome_atributo1    tipo    [NOT NULL] [UNIQUE] [CHECK( )],  
  nome_atributo2    tipo    [NOT NULL] [UNIQUE] [CHECK( )],  
  ...  
  nome_atributoN    tipo    [NOT NULL] [UNIQUE] [CHECK( )],  
  [PRIMARY KEY (atributos_compoem_chave_primaria)],  
  [FOREIGN KEY (nome_atributo) REFERENCES  
    nome_tabela_origem (nome_atributo_origem)]  
);
```

*Obs: tudo que está entre [ ] é opcional, mas se for utilizar tire-o*

# CREATE TABLE

- Onde:
  - nome\_tabela: indica o nome da tabela a ser criada
  - nome\_atributo: indica o nome do campo a ser criado na tabela
  - tipo: indica a definição do tipo de atributo (INTEGER, VARCHAR(N), CHAR(N), etc., sendo “N” o número de caracteres)
  - nome\_tabela\_origem: indica a tabela onde a chave estrangeira foi originalmente criada
  - nome\_atributo\_origem: indica o nome da chave primária na tabela em que foi criada

# CREATE TABLE

- Onde:
  - NOT NULL: não permite a inserção de valores nulos
  - UNIQUE: não permite que os valores de um atributo se repita. É possível ter vários atributos UNIQUE
  - CHECK(condição): permite validar os dados inseridos neste atributo, através de uma condição
  - PRIMARY KEY: explicada a seguir
  - FOREIGN KEY: explicada a seguir

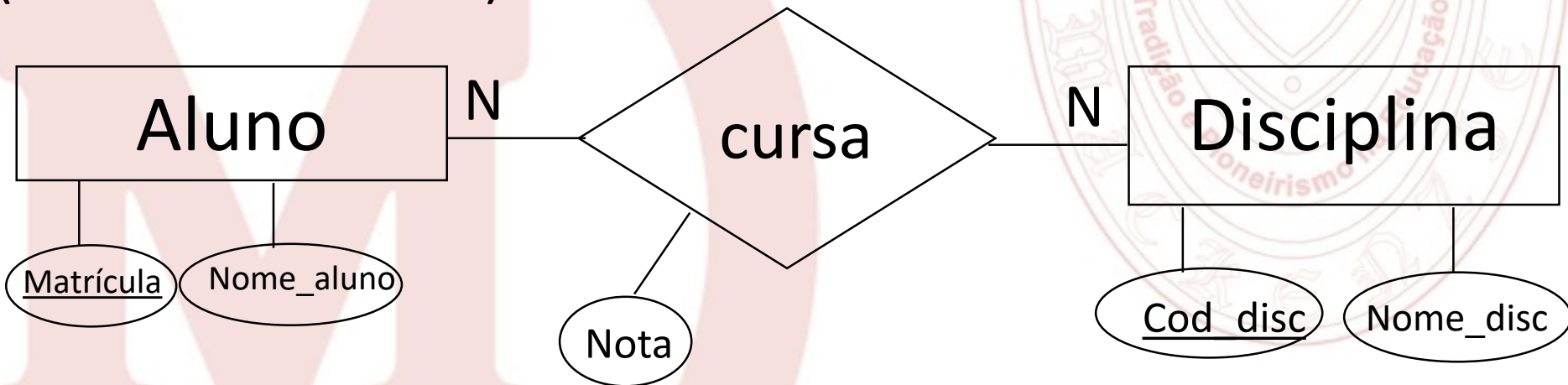
# CREATE TABLE

- Conceito de **PRIMARY KEY** e **FOREIGN KEY**:
  - **PRIMARY KEY** ou chave primária é o identificador único de cada instância (linha ou registro) de uma tabela. Cada tabela só pode ter uma única PRIMARY KEY, que pode ser formada por um ou vários atributos e não pode ser nula
  - **FOREIGN KEY** ou chave estrangeira é que faz o relacionamento entre as tabelas. Ela sempre referencia a PRIMARY KEY de uma tabela. Ela pode ser nula, mas não pode ter um valor que não exista na tabela que está referenciando. Uma tabela pode não ter FOREIGN KEY ou ter várias

# CREATE TABLE

**Exemplo de criação de tabelas do seguinte banco de dados:**

Considere parte de um Modelo Entidade-Relacionamento (modelo conceitual) de um banco de dados:





# CREATE TABLE

O Modelo Entidade-Relacionamento anterior gera o seguinte **Modelo Relacional (modelo lógico)**, ou seja, as tabelas que serão criadas no banco de dados:

Aluno = {Matricula, Nome\_aluno}

Disciplina = {Cod\_disc, Nome\_disc}

Aluno\_Disciplina = {Matricula, Cod\_disc, Nota}

- *Matricula é **chave estrangeira** que referencia Matricula da tabela Aluno*

- *Cod\_disc é **chave estrangeira** que referencia Cod\_disc da tabela Disciplina*

# CREATE TABLE

- Criação das tabelas do Banco de Dados mostrado anteriormente:

```
CREATE TABLE Aluno
```

```
(Matricula INTEGER CHECK(Matricula >= 1),
```

```
Nome_aluno VARCHAR(50) NOT NULL,
```

```
PRIMARY KEY (Matricula)
```

```
);
```

# CREATE TABLE

- Criação das tabelas do Banco de Dados mostrado anteriormente:

```
CREATE TABLE Disciplina
```

```
(Cod_disc INTEGER CHECK(Cod_disc >= 1),
```

```
Nome_disc VARCHAR(25) NOT NULL UNIQUE,
```

```
PRIMARY KEY (Cod_disc)
```

```
);
```

# CREATE TABLE

- Criação das tabelas do Banco de Dados mostrado anteriormente:

```
CREATE TABLE Aluno_Disciplina  
(Matricula INTEGER,  
Cod_disc INTEGER,  
Nota INTEGER CHECK(Nota BETWEEN 0 AND 10),  
PRIMARY KEY (Matricula, Cod_disc),  
FOREIGN KEY (Matricula) REFERENCES Aluno(Matricula),  
FOREIGN KEY (Cod_disc) REFERENCES Disciplina(Cod_disc)  
);
```

# Roteiro da Apresentação

- Introdução
- CREATE TABLE
- DROP TABLE





# DROP TABLE

- DROP TABLE
  - Remove uma tabela existente no banco de dados

# DROP TABLE

- Sintaxe do comando DROP TABLE

**DROP TABLE** nome\_tabela;

- Exemplo:

**DROP TABLE** Aluno\_Disciplina;



# DROP TABLE

- Nas tabelas que criamos anteriormente, **não seria possível eliminar a tabela Aluno ou a tabela Disciplina antes de eliminar a tabela Aluno\_Disciplina**, pois esta tem 2 chaves estrangeiras que referenciam as tabelas Aluno e Disciplina

# Obrigado

Profa. Elisângela Botelho Gracias  
[elisangela.botelho@mackenzie.br](mailto:elisangela.botelho@mackenzie.br)