

CURSO

---

# Administrador de Banco de Dados

Disciplina III

**Implementação de Banco de Dados em SQL**



## AULA 02

---

- **Comandos DDL (Linguagem de Definição de Dados):  
CREATE TABLE**



# Introdução ao SQL

- Todo **Sistema Gerenciador de Banco de Dados** (SGBD) deve oferecer aos seus usuários e administradores meios de **criar** e **manipular dados armazenados** em seus banco de dados;
- A **linguagem SQL**, sigla em inglês para *Structured Query Language* ou *Linguagem de Consulta Estruturada* é a linguagem padrão adotada por diferentes SGBDs para criar e manipular banco de dados relacionais;
- A linguagem SQL possui uma **sintaxe única**, porém alguns SGBDs trazem **variações na sintaxe**. Porém a **lógica estrutural** da linguagem é sempre a mesma;

# Introdução ao SQL

- O SQL tem o objetivo de criar de forma **estruturada** e **declarativa** as tabelas de bancos de dados;
- Da mesma forma **cria** e **manipula** os registros inseridos nestas tabelas;
- A função do SQL é criar e manter banco de dados, possibilitando a **persistência** e **recuperação** destes registros com segurança;

# Introdução ao SQL

- O SQL é caracterizado pela utilização de comandos também chamados de instruções que podem ser **classificadas**, de acordo com sua função, normalmente em **dois tipos**:
  - **DDL – Linguagem de Definição de Dados;**
  - **DML – Linguagem de Manipulação de Dados;**

# DDL - Linguagem de Definição de Dados

- Os comandos DDL permite ao utilizador **criar novas tabelas**, **alterar a estrutura de uma tabela** ou **excluir uma tabela**;
- Os comandos DDL são:
  - **CREATE TABLE** (Criar Tabela)
  - **DROP TABLE** (Excluir Tabela)
  - **ALTER TABLE** (Alterar Tabela)

# DML - Linguagem de Manipulação de Dados

- Os comando DML permite ao utilizador **inserir** registros em uma tabela, **atualizar** registros inseridos em uma tabela, **deletar** registros em uma tabela e **selecionar** registros em uma tabela;
- Os comandos DML são:
  - **INSERT** (Inserir)
  - **UPDATE** (Atualizar)
  - **DELETE** (Deletar)
  - **SELECT** (Selecionar)

# Diferença entre os Comandos DDL e DML

- **LEMBRE-SE:**

- ✓ Os **comandos DDL** são utilizados para *criar, alterar e apagar tabelas*;
  - ✓ Os **comandos DML** são utilizados para *criar, alterar, apagar e selecionar registros*;
- 
- **Lembre-se** que uma tabela possui a mesma analogia que uma tabela no Microsoft Excel. Veja:



# Conceitos sobre Registros

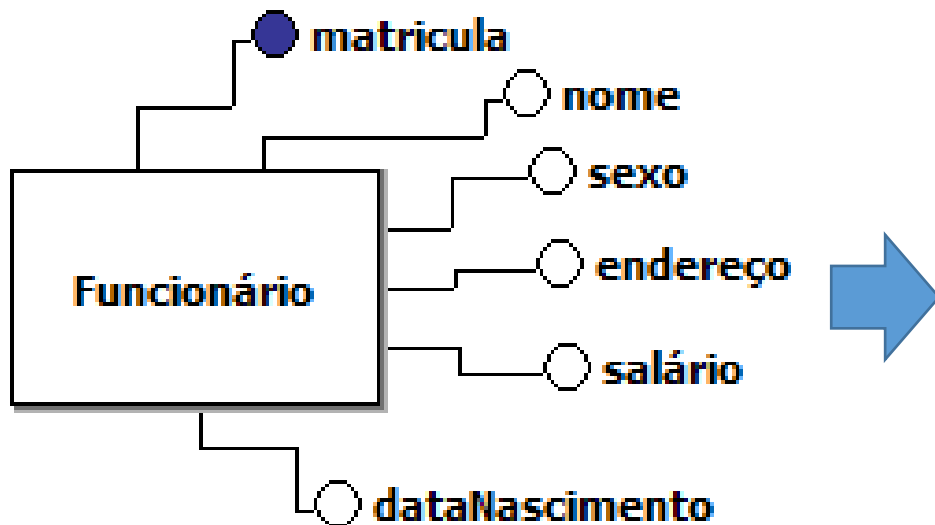
## COMANDOS DML

*Insert, Update, Delete e Select*



## COMANDOS DDL

*Create, Alter e Drop*



Matricula	Nome	Sexo	Endereço	Salário	dataNasc
1	Jackson Henrique	Masculino	Rua X...	5.000,00	30/06/1987
2	Hanna Karoline	Feminino	Rua Y...	6.000,00	23/10/1990
3	Jaqueline Leão	Feminino	Av. B	8.000,00	21/06/1993

# Comandos Básicos Iniciais

- **Criação de Banco de Dados**

- **Sintaxe:**

- *CREATE DATABASE nome\_do\_banco\_de\_dados;*

- **Exemplo:**

- *CREATE DATABASE locadora;*

# Comandos Básicos Iniciais

- **Usar o banco de dados**

- **Sintaxe:**

- *USE nome\_do\_banco\_de\_dados;*

- **Exemplo:**

- USE locadora;

# Create Table (Criar Tabela)

- **Sintaxe:**

➤ *CREATE TABLE nome\_da\_tabela (  
    nomeatributo1 tipodedado tipoatributo,  
    nomeatributo2 tipodedado,  
    nomeatributo3 tipodedado  
);*

# Sintaxe do Create Table (Criar Tabela)

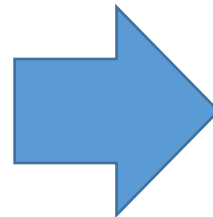
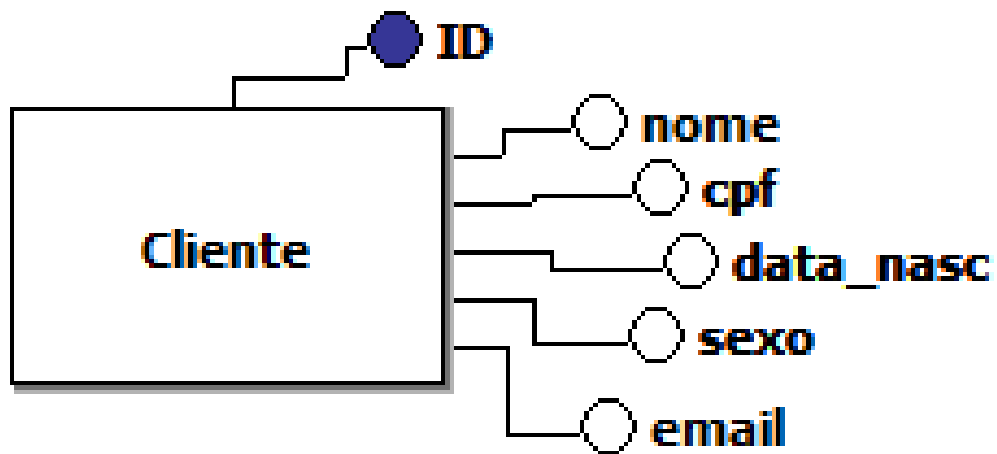
- **Tipo de Dados:**

- **VARCHAR (100)** -> *Textos*
- **INTEGER** -> *Números Inteiros*
- **FLOAT** -> *Números Fracionados*
- **DOUBLE** -> *Números Fracionados*
- **DATE** -> *Datas*
- **TIME** -> *Horas*

- **Tipos de Atributos**

- **PRIMARY KEY** -> Chave Primária
- **FOREIGN KEY** -> Chave Estrangeira
- **NOT NULL** -> Não Nulo
- **UNIQUE** -> Único
- **AUTO\_INCREMENT** -> Incremento Numérico Automático;

# Exemplo do Create Table



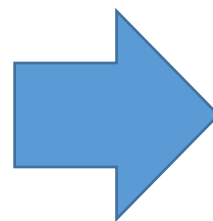
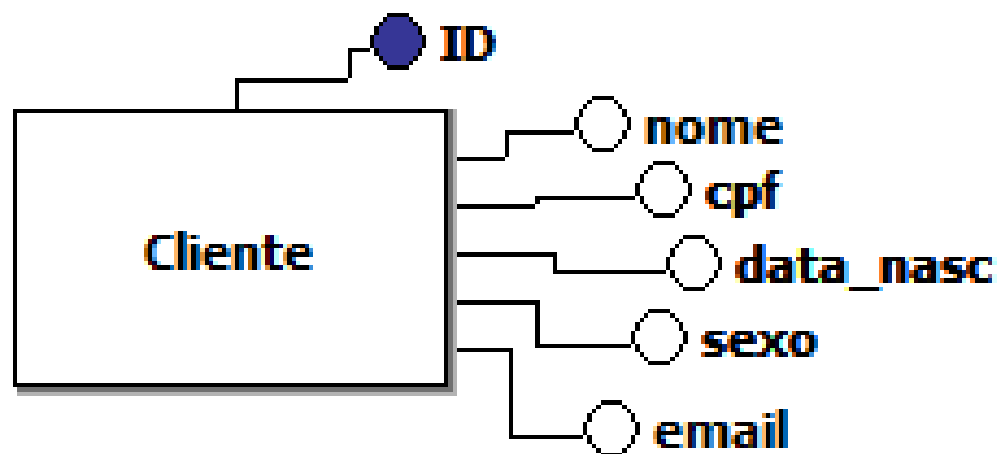
- **Exemplo**

➤ CREATE TABLE Cliente (  
id INTEGER PRIMARY KEY,  
nome VARCHAR (100),  
cpf VARCHAR(15),  
data\_nasc DATE,  
sexo VARCHAR (20),  
email VARCHAR (100)  
);

# Regra de Boa Prática de Programação

- Existe uma regra no CREATE TABLE sobre boa prática de programação sobre a **forma de nomear os atributos** da tabela no Modelo Físico:
- **Regra:**
  - **Todo atributo** deve possuir um **sufixo** ou **prefixo** que **identifique** a sua tabela de origem.
  - Os sufixo ou prefixo deve possuir de **3 a 5 letras**;
  - O sufixo ou prefixo devem ser separado por ***underline*** do nome do atributo;
  - Todo nome de atributo começa com a letra **minúscula**;
  - Todo nome de tabela começa com a letra **MAIÚSCULA**;
  - Não utilize acentuação nos nomes dos atributos e tabelas;

# Exemplo do Create Table com Sufixos



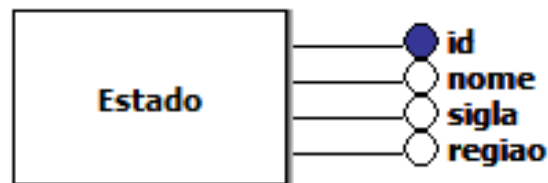
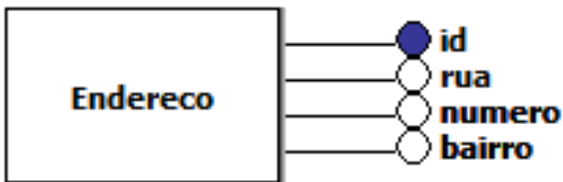
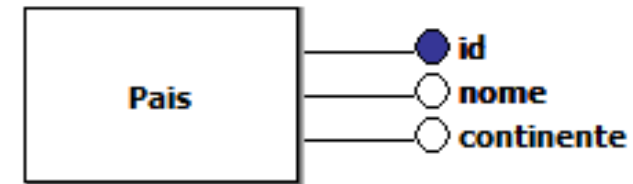
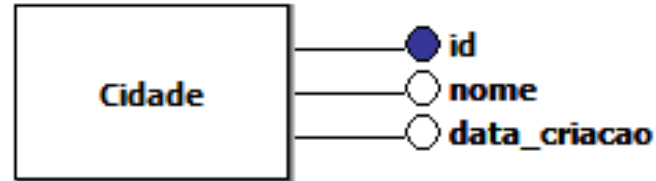
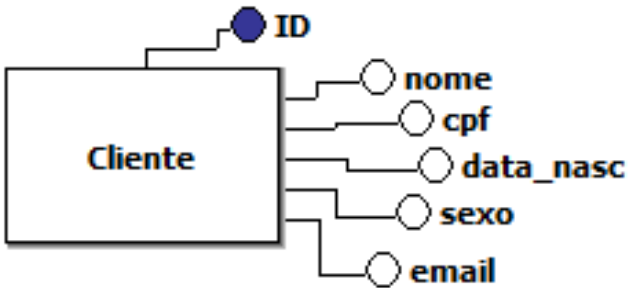
- **Exemplo**

➤ CREATE TABLE Cliente (  
id\_cli INTEGER PRIMARY KEY,  
nome\_cli VARCHAR (100),  
cpf\_cli VARCHAR(15),  
data\_nasc\_cli DATE,  
sexo\_cli VARCHAR (20),  
email\_cli VARCHAR (100)  
);



# Hora de Praticar 1

- Crie as tabelas seguir no seu MySQL. Salve o script com o nome *teste1* e crie um banco de dados chamado *pratica1*;



# Chaves Estrangeiras

- As chaves estrangeiras são atributos do tipo INTEGER e devem possuir um COMANDO que o identifica como FOREIGN KEY;
- Para criar uma Chave Estrangeira devemos aplicar **dois comandos**, veja as regras para isso:
  - **Regra nº 1:** Cria-se um atributo simples do tipo INTEGER que **SERÁ** a Chave Estrangeira;
  - **Regra nº 2:** O nome do atributo será o **MESMO** da Chave Primária a que ele se refere (Origem);
  - **Regra nº 3:** No final do atributo deve-se adicionar **\_fk** a fim de identificá-lo como **Chave Estrangeira** através do nome do atributo;
  - **Regra nº 4:** Na próxima linha cria-se o comando FOREIGN KEY que transformará o atributo **id\_cli\_fk** em uma Chave Estrangeira de fato;

# Chaves Estrangeiras

## ✓ EXEMPLO:

✓ Nome da Chave Primária de Origem: **id\_cli INTEGER**

✓ Nome do Chave Estrangeira: **id\_cli\_fk INTEGER**

## ➤ EXEMPLO:

✓ **FOREIGN KEY (id\_cli\_fk) REFERENCES Cliente (id\_cli);**

Diz para o SGBD qual atributo  
será a chave estrangeira

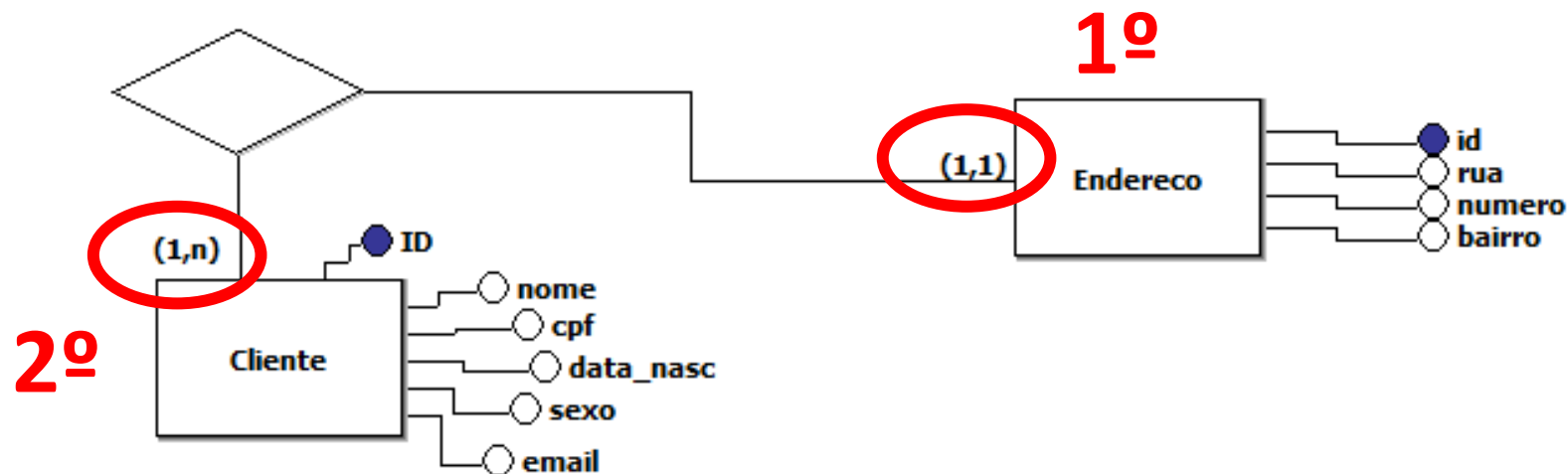
Faz a referencia entre  
as tabelas

Diz para o SGBD de qual  
TABELA e ATRIBUTO  
será a referencia

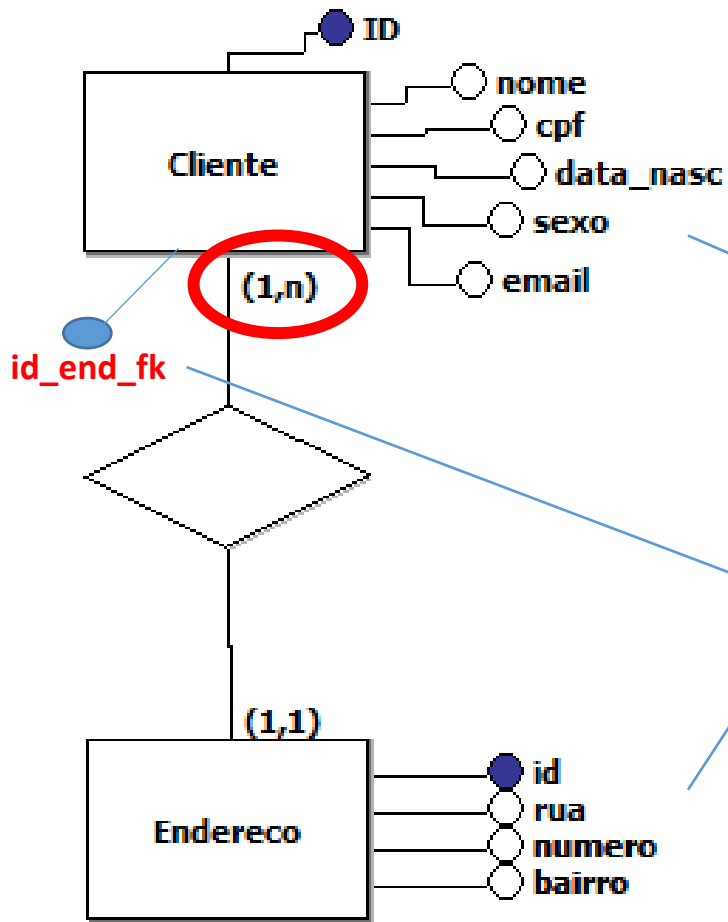
# Ordem de Criação das Tabelas

- **LEMBRE-SE:**

1. Na criação das Chaves Estrangeiras só **podemos fazer referencia a uma tabela que JÁ EXISTE**;
2. Assim, entre **duas tabelas com relação (cardinalidade máxima) N para 1**, criamos **primeiro** a tabela com a **cardinalidade 1** para **depois** criar a tabela com o **N**;



# Exemplo Prático



➤ CREATE TABLE Endereco (  
id\_end INTEGER PRIMARY KEY  
rua\_end VARCHAR (200) NOT NULL,  
numero\_end INTEGER,  
bairro\_end VARCHAR (100)  
);

➤ CREATE TABLE Cliente (  
id\_cli INTEGER PRIMARY KEY,  
nome\_cli VARCHAR (100),  
cpf\_cli VARCHAR(15),  
data\_nasc\_cli DATE,  
sexo\_cli VARCHAR (20),  
email\_cli VARCHAR (100) ,  
**id\_end\_fk INTEGER,**  
**FOREIGN KEY (id\_end\_fk) REFERENCES Endereco (id\_end)**  
);

# DDL – Drop (Apagar)

- Comando usado para deletar uma tabela no banco de dados;
- Síntese: DROP TABLE nome\_da\_tabela;
- Exemplo: DROP TABLE cliente;

# DDL – Drop (Apagar)

- Comando usado para **apagar** uma tabela no banco de dados;
- Síntese:

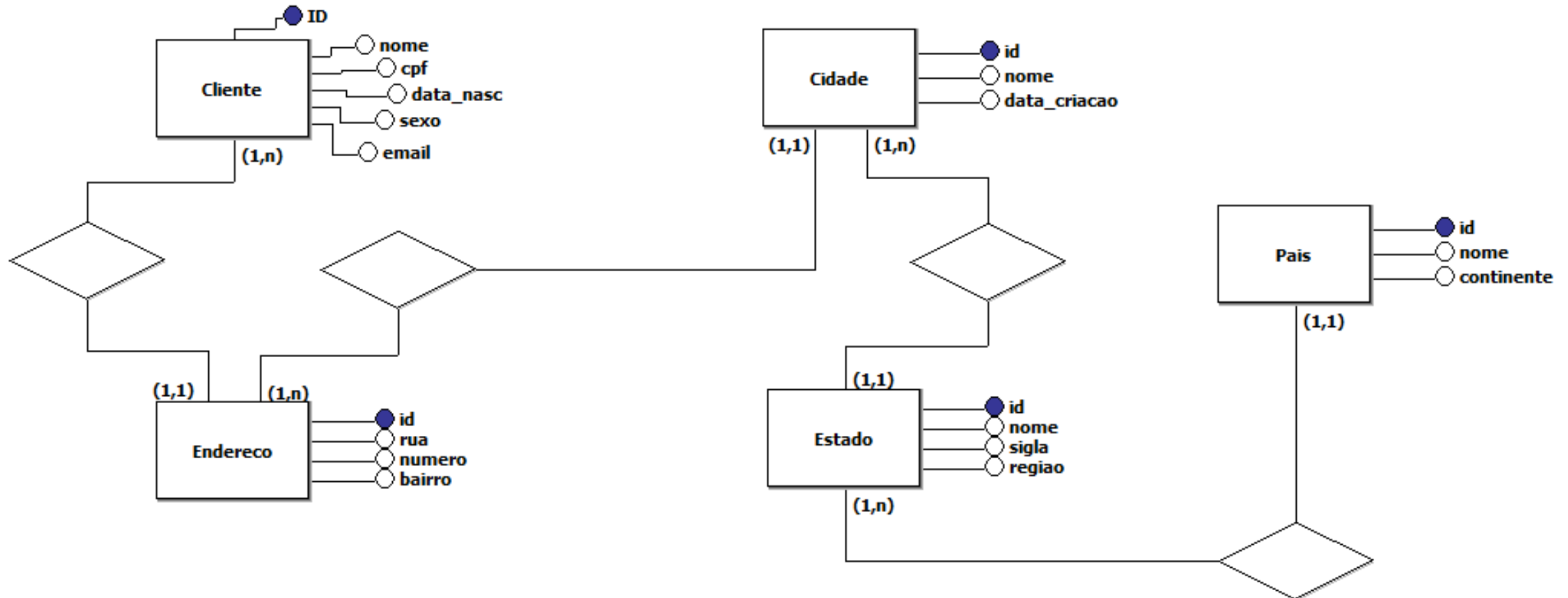
➤ ***DROP TABLE nome\_da\_tabela;***

- Exemplo:

➤ **DROP TABLE cliente;**

# Hora Praticar 2

- Crie um novo script chamado *teste2*. Crie um novo banco de dados chamado *pratica2*. Copie e cole o código criado na **Hora de Praticar 1**. Não execute ainda. Crie as chaves estrangeiras para fazer as relações do diagrama a seguir. Observe a ordem de criação das tabelas de acordo com as chaves estrangeiras.





# Jackson Henrique

Professor Formador

E-mail:

[Jackson.henrique@ifro.edu.br](mailto:Jackson.henrique@ifro.edu.br)

