

CURSO

# Administrador de Banco de Dados

Disciplina III

**Implementação de Banco de Dados em SQL**



## AULA 05

---

- **Comandos DDL (Linguagem de Definição de Dados):  
ALTER E DROP TABLE**



# Alter Table (Alterar Tabela)

- Usando o comando **ALTER** é possível realizar as alterações na estrutura de uma tabela, tais como:
  - **Adicionar Atributos;**
  - **Excluir Atributos;**
  - **Alterar o tipo e o nome de um atributo já existente;**
- Para cada **tipo de alteração**, existe um **operador específico** a ser utilizado no comando;

# Alter Table (Alterar Tabela)

- **Sintaxe do Comando:**

- *ALTER TABLE nome\_tabela **OPERADOR** nome\_atributo;*

- Para cada **tipo de alteração**, existe um **operador específico** a ser utilizado no comando;

- **Tipos de Operadores:**

- Adicionar: **ADD**

- Excluir: **DROP**

- Alterar: **CHANGE**

# Adicionar Atributo – Operador ADD

- O operador **ADD** é utilizado no comando ALTER TABLE para **adicionar** um novo atributo em uma **tabela já existente**;
- **Síntese:**
  - *ALTER TABLE nome\_da\_tabela ADD atributo tipodedado;*
- **Exemplos:**
  - ALTER TABLE cliente ADD endereco\_cli VARCHAR (100);
  - ALTER TABLE cliente ADD renda\_cli FLOAT;

# Adicionar Atributo – Operador ADD

- **Exemplo Prático:** Considere uma tabela chamada **Aluno** com os atributos *id*, *nome* e *cpf*.
- Suponhamos que você queira adicionar um novo atributo chamado *data de nascimento*.
- Assim irá executar o comando Alter Table: **ALTER TABLE Aluno ADD data\_nascimento\_cli DATE;**
- Após executar este comando, o atributo será criado **sempre** no FINAL da lista de atributos. Veja a seguir:

# Adicionar Atributo – Operador ADD

## Antes

- Create table Aluno (
- id\_alu integer primary key,
- nome\_alu varchar (100),
- cpf\_alu varchar (20)
- );

## Depois

- Create table Aluno (
- id\_alu integer primary key,
- nome\_alu varchar (100),
- cpf\_alu varchar (20) ,
- **data\_nasc\_alu date**
- );

# Adicionar Atributo – Operador ADD

- É possível utilizar **palavras chaves** para **posicionar** este novo atributo na tabela.
- As palavras chaves **FIRST** (primeiro) e **AFTER** (após) podem ser utilizadas para **posicionar** o novo atributo na posição que desejar na tabela.
- Veja os **exemplos seguir**, utilizando com base a **adição** de um atributo chamado *renda familiar* na tabela Aluno;



Exemplo:

ALTER TABLE Aluno ADD renda\_familiar\_alu FLOAT **AFTER nome\_alu;**

### Antes

- Create table Aluno (
- id\_alu integer primary key,
- nome\_alu varchar (100),
- cpf\_alu varchar (20) ,
- data\_nasc\_alu date
- );

### Depois

- Create table Aluno (
- id\_alu integer primary key,
- nome\_alu varchar (100),
- **renda\_familiar\_alu float,**
- cpf\_alu varchar (20) ,
- data\_nasc\_alu date
- );

Exemplo:

ALTER TABLE Aluno ADD renda\_familiar\_alu FLOAT **FIRST;**

### Antes

- Create table Aluno (
- id\_alu integer primary key,
- nome\_alu varchar (100),
- cpf\_alu varchar (20) ,
- data\_nasc\_alu date
- );

### Depois

- Create table Aluno (
- **renda\_familiar\_alu float,**
- id\_alu integer primary key,
- nome\_alu varchar (100),
- cpf\_alu varchar (20) ,
- data\_nasc\_alu date
- );

# Excluir Atributo – Operador DROP

- O operador **DROP** é utilizado no comando ALTER TABLE para **excluir** um atributo em uma **tabela já existente**;
- **Sintaxe:**
  - *ALTER TABLE nome\_da\_tabela DROP atributo;*
- **Exemplos:**
  - ALTER TABLE cliente DROP email\_cli;
  - ALTER TABLE cliente DROP nome\_cli;

# Alterar Atributo – Operador CHANGE

- O operador **CHANGE** é utilizado no comando ALTER TABLE para **alterar** um atributo em uma **tabela já existente**.
- Com ele podemos mudar apenas o **nome**, o **tipo** ou o **nome e o tipo** ao mesmo tempo;
- **Sintaxe:**

➤ *ALTER TABLE nome\_da\_tabela CHANGE nome\_do\_atributo  
novo\_nome\_do\_atributo tipodedado;*

# Alterar Atributo – Operador CHANGE

- Exemplo para alterar apenas o NOME do atributo:

➤ ALTER TABLE cliente CHANGE **data\_nasc\_cli** **data\_nascimento\_cli** DATE;

- Exemplo para alterar apenas o TIPO do atributo:

➤ ALTER TABLE cliente CHANGE **data\_nascimento\_cli** **data\_nascimento\_cli** VARCHAR (20);

- Exemplo para alterar o NOME e o TIPO do atributo:

➤ ALTER TABLE cliente CHANGE **data\_nascimento\_cli** **nascimento\_cli** DATE;

# Comandos Básicos

- Para ver melhor a **alteração do TIPO de dado** do atributo podemos utilizar o comando **DESC** que **mostra a estrutura física da tabela**, incluindo o nome dos atributos e os seus tipos de dados;
- **Síntese:**
  - DESC nome\_da\_tabela;
- **Exemplo:**
  - DESC cliente;

# DDL – Drop (Apagar)

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

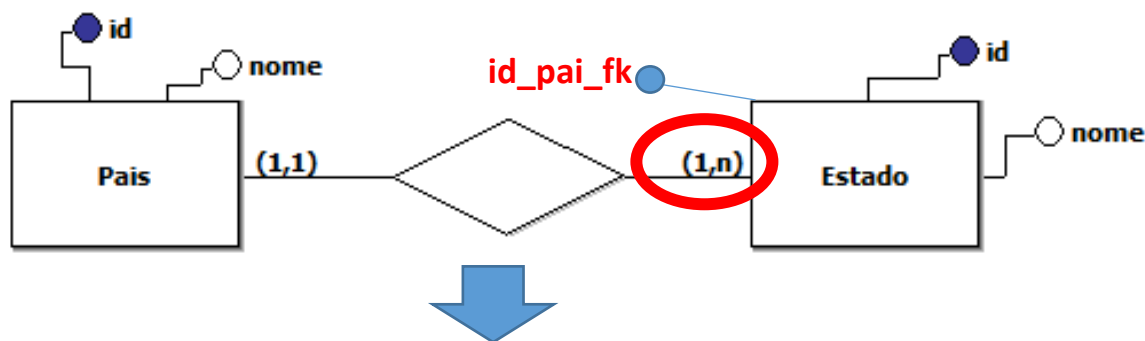
➤ ***DROP TABLE nome\_da\_tabela;***

- Exemplo:

➤ **DROP TABLE cliente;**

# DDL – Drop (Apagar)

- **LEMBRE-SE:** Não é possível excluir uma tabela que possua **registros dependentes** de registros de **outra tabela**;
- **Exemplo:**



Pais			Estado		
id_pai	nome_pai		id_est	nome_est	id_pai_fk
1	Brasil		1	São Paulo	1
2	Chile		2	Rio de Janeiro	1

**Pergunta:** É possível **excluir** a tabela PAÍS?

**Resposta:** **NÃO**, porque o registro com id nº 1 está vinculado com um ou mais registros na tabela Estado. Ou seja, para que os registros nº 1 e 2 da tabela Estado exista eles dependem de um registro (FK) de Pais.

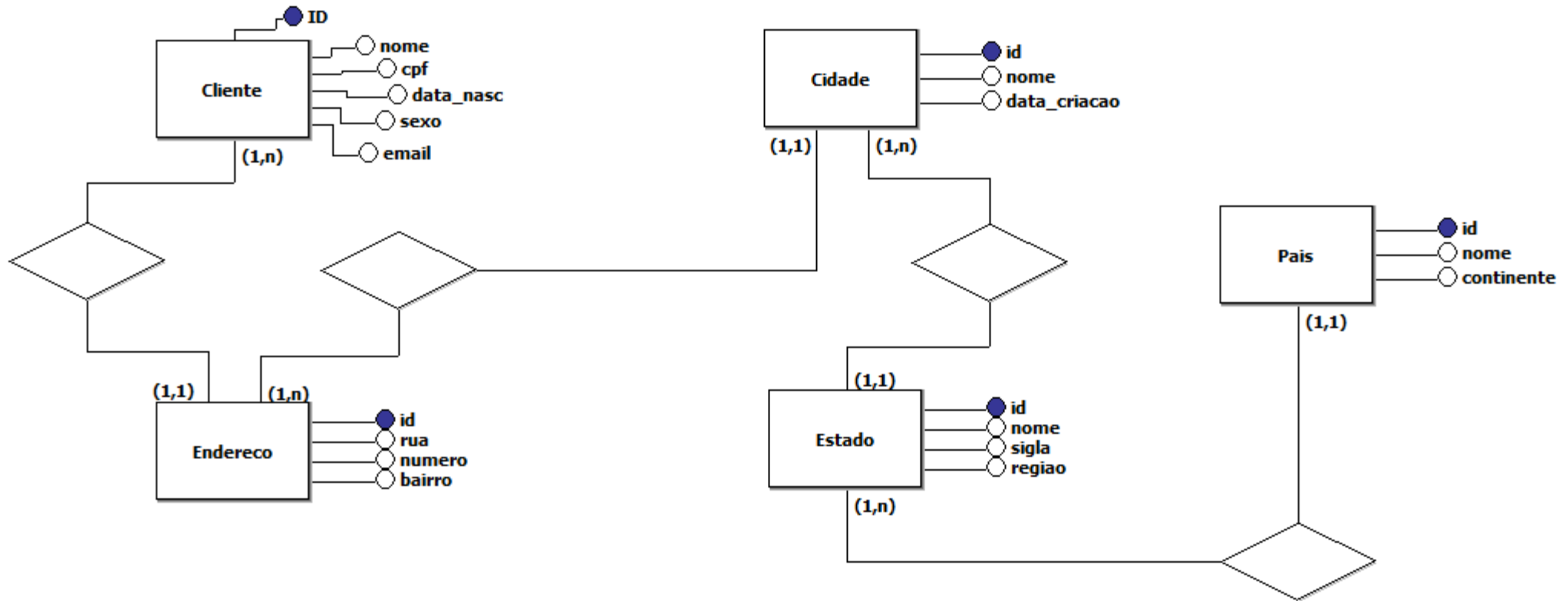
**Pergunta:** É possível **excluir** a tabela Estado?

**Resposta:** **SIM**, porque nenhum registro da tabela Estado possui o id vinculado a um ou mais registros da tabela Pais;



# Hora de Praticar 1

- Abra no seu MySQL o **SCRIPT SQL** do diagrama a seguir criado na Aula 02. Execute o mesmo para criar o banco de dados.



# Hora de Praticar 1

- Faça as ações a seguir no banco de dados:
  1. Adicione o atributo RG na tabela Cliente;
  2. Adicione o atributo TELEFONE na tabela Cliente;
  3. Adicione o atributo REFERENCIA na tabela Endereco;
  4. Exclua o atributo CONTINENTE na tabela Pais;
  5. Altere o nome do atributo SEXO para GENERO;
  6. Altere o tipo de dados do atributo DATA\_NASC de DATE para VARCHAR em Cliente;
  7. Crie uma a sequencia de comandos para excluir todas as tabelas, seguindo a lógica de dependência entre as tabelas;

# Jackson Henrique

Professor Formador

E-mail:

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

