

# Atividade Prática – Laboratório E/R

## Objetivo:

Praticar a criação de Modelos Entidade-Relacionamento (E/R) e o mapeamento desses modelos para esquemas de banco de dados relacionais utilizando SQL.

## Instruções:

### 1. Modele o Diagrama Entidade-Relacionamento (E/R):

- Identifique as entidades principais.
- Identifique os atributos de cada entidade, classificando-os (simples, composto, multivalorado, chave, etc.).
- Identifique os relacionamentos entre as entidades, especificando suas cardinalidades (mínima e máxima).
- Utilize o conceito de especialização/generalização onde for apropriado para representar os diferentes tipos de pessoas, alunos e servidores.  
Preste atenção à possibilidade de sobreposição e totalidade/parcialidade da especialização.

### 2. Mapeie o Modelo E/R para SQL:

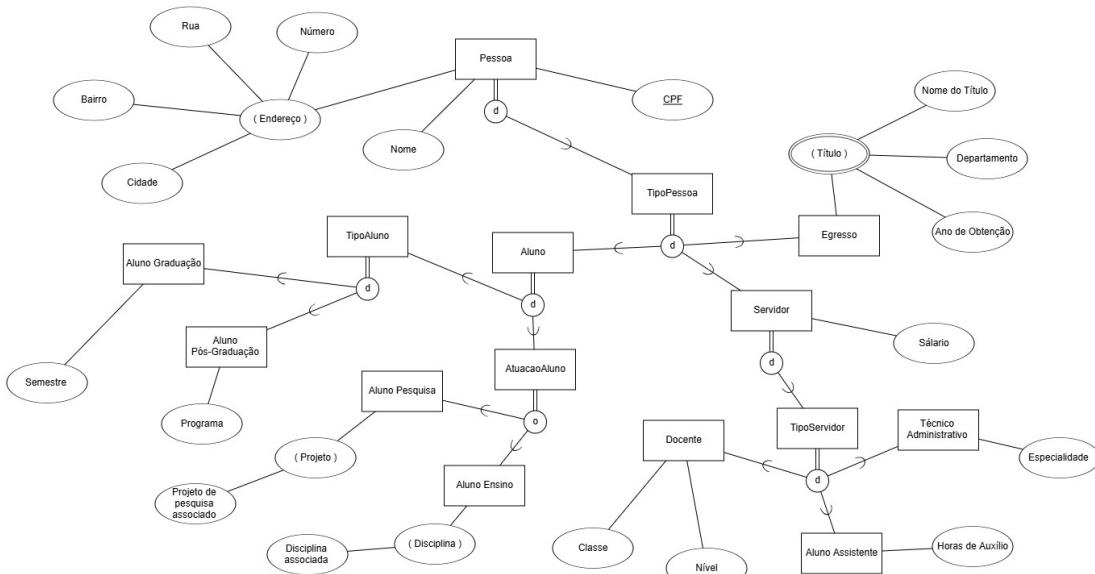
- Converta o seu diagrama E/R em um conjunto de comandos SQL CREATE TABLE.
- Para cada tabela, defina:
  - As colunas correspondentes aos atributos, escolhendo tipos de dados SQL apropriados (ex: VARCHAR, INT, etc.).
  - A chave primária (PRIMARY KEY).
  - Outras restrições que julgar necessárias (NOT NULL, UNIQUE, CHECK).

## Descrição do Sistema:

1. Uma pessoa possui CPF, nome e endereço.
2. Pessoas podem ser servidores, alunos ou egressos, ou mais de um tipo ao mesmo tempo.
  - Servidores possuem salário.

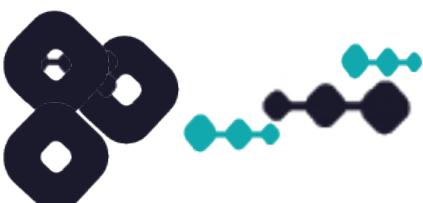


- Alunos possuem o identificador do departamento ao qual está matriculado.
- Egressos possuem um ou mais títulos acadêmicos; cada atributo da titulação é composto pelo nome do título, departamento e ano de obtenção.
- 3. Alunos podem ser de pós-graduação ou graduação.
  - Alunos de pós-graduação possuem um atributo identificando o programa (mestrado, doutorado, especialização, etc).
  - Alunos de graduação possuem a informação do semestre em que se encontram desde o ingresso no curso.
- 4. Um servidor pode ser técnico administrativo (TA), docente ou um aluno assistente.
  - TAs possuem uma especialidade.
  - Docentes possuem classe e nível.
  - Alunos assistentes possuem a quantidade de horas dedicada a atividades de auxílio à docência.
- 5. Finalmente, um aluno pode trabalhar com pesquisa ou ensino.
  - Aluno em pesquisa possui um atributo identificando o projeto de pesquisa ao qual está vinculado.
  - Aluno em ensino possui um atributo identificando a disciplina à qual está associado.



### Comandos SQL a serem utilizados:

- SELECT
- FROM
- WHERE
- ORDER BY
- LIMIT



-- Cria tabela Pessoa

CREATE TABLE Pessoa (

    CPF VARCHAR(11) PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Rua VARCHAR(100),  
    Numero VARCHAR(10),  
    Bairro VARCHAR(50),  
    Cidade VARCHAR(50)

);

-- Cria tabela Servidor

CREATE TABLE Servidor (

    CPF VARCHAR(11) PRIMARY KEY,  
    Salario DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)

);

-- Cria tabela Aluno

CREATE TABLE Aluno (

    CPF VARCHAR(11) PRIMARY KEY,  
    Departamento VARCHAR(50) NOT NULL,  
    FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)

);

-- Cria tabela Egresso

CREATE TABLE Egresso (

    CPF VARCHAR(11) PRIMARY KEY,  
    FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)

);

-- Cria tabela Titulo (multivalorado para Egresso)

CREATE TABLE Titulo (

    ID\_Titulo INTEGER PRIMARY KEY,  
    CPF\_Egresso VARCHAR(11),  
    Nome\_Titulo VARCHAR(100) NOT NULL,  
    Departamento VARCHAR(50) NOT NULL,  
    Ano\_Obtencao INT NOT NULL,  
    FOREIGN KEY (CPF\_Egresso) REFERENCES Egresso(CPF)

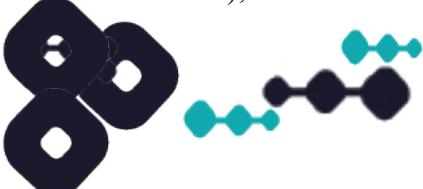
);

-- Cria tabela Tecnico\_Administrativo

CREATE TABLE Tecnico\_Administrativo (

    CPF VARCHAR(11) PRIMARY KEY,  
    Especialidade VARCHAR(100) NOT NULL,  
    FOREIGN KEY (CPF) REFERENCES Servidor(CPF)

);



-- Cria tabela Docente

```
CREATE TABLE Docente (
    CPF VARCHAR(11) PRIMARY KEY,
    Classe VARCHAR(10) NOT NULL,
    Nivel INT NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Servidor(CPF)
);
```

-- Cria tabela Aluno\_Assistente

```
CREATE TABLE Aluno_Assistente (
    CPF VARCHAR(11) PRIMARY KEY,
    Horas_Auxilio INT NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Servidor(CPF)
);
```

-- Cria tabela Aluno\_Graduacao

```
CREATE TABLE Aluno_Graduacao (
    CPF VARCHAR(11) PRIMARY KEY,
    Semestre INT NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Aluno(CPF)
);
```

-- Cria tabela Aluno\_Pos\_Graduacao

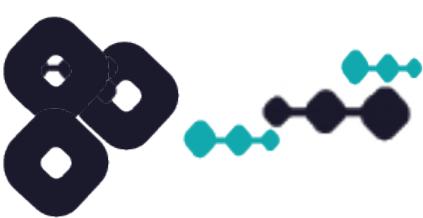
```
CREATE TABLE Aluno_Pos_Graduacao (
    CPF VARCHAR(11) PRIMARY KEY,
    Programa VARCHAR(100) NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Aluno(CPF)
);
```

-- Cria tabela Aluno\_Pesquisa

```
CREATE TABLE Aluno_Pesquisa (
    CPF VARCHAR(11) PRIMARY KEY,
    Projeto_Pesquisa VARCHAR(100) NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Aluno(CPF)
);
```

-- Cria tabela Aluno\_Espresso

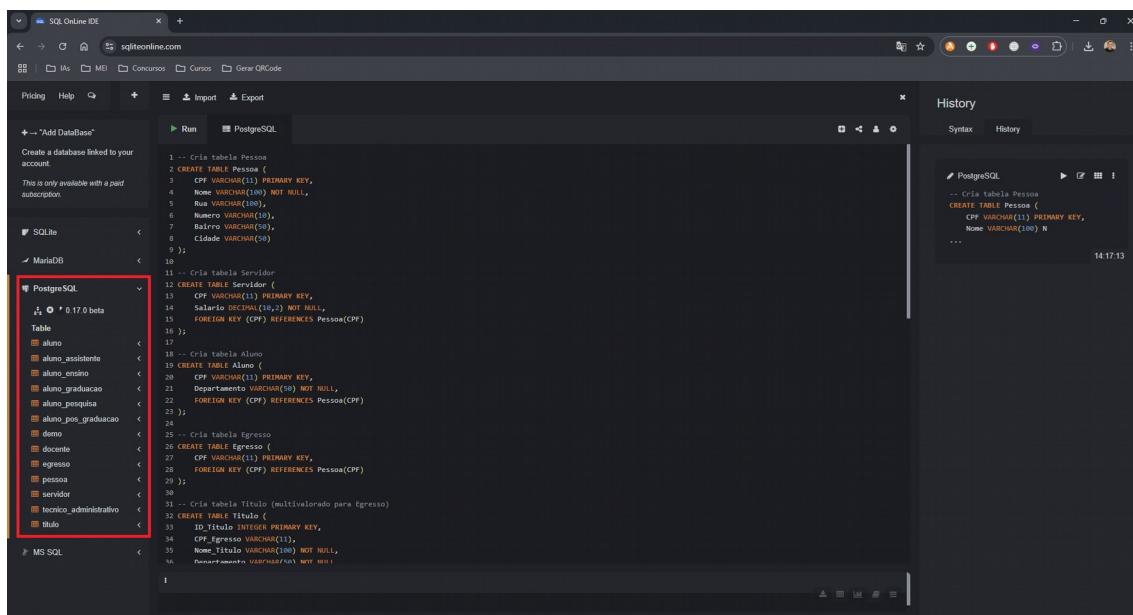
```
CREATE TABLE Aluno_Espresso (
    CPF VARCHAR(11) PRIMARY KEY,
    Disciplina_Associada VARCHAR(100) NOT NULL,
    FOREIGN KEY (CPF) REFERENCES Aluno(CPF)
);
```



# Banco de Dados

## Willgnner Ferreira Santos - 2025200271

Testado os comandos SQL em: <https://sqliteonline.com/>



The screenshot shows the SQLiteOnline IDE interface. On the left, there's a sidebar with options like 'Pricing', 'Help', 'Import', 'Export', and a 'Run' button. Below these are sections for 'Add DataBase', 'SQLite', 'MariaDB', and 'PostgreSQL'. The 'PostgreSQL' section is highlighted with a red box and contains a tree view of tables: 'aluno', 'aluno\_acidente', 'aluno\_ensino', 'aluno\_graduacao', 'aluno\_pesquisa', 'aluno\_pos\_graduacao', 'demo', 'docente', 'egresso', 'pessoa', 'servidor', 'tecnico\_administrativo', and 'titulo'. The main area displays a large block of SQL code for creating tables and defining primary keys and foreign keys. The code includes:

```
1 -- Cria tabela Pessoa
2 CREATE TABLE Pessoa (
3     CPF VARCHAR(11) PRIMARY KEY,
4     Nome VARCHAR(100) NOT NULL,
5     RG VARCHAR(10),
6     Numero VARCHAR(10),
7     Bairro VARCHAR(50),
8     Cidade VARCHAR(50)
9 );
10
11 -- Cria tabela Servidor
12 CREATE TABLE Servidor (
13     CPF VARCHAR(11) PRIMARY KEY,
14     Salario DECIMAL(10,2) NOT NULL,
15     FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)
16 );
17
18 -- Cria tabela Aluno
19 CREATE TABLE Aluno (
20     CPF VARCHAR(11) PRIMARY KEY,
21     Departamento VARCHAR(50) NOT NULL,
22     FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)
23 );
24
25 -- Cria tabela Egresso
26 CREATE TABLE Egresso (
27     CPF VARCHAR(11) PRIMARY KEY,
28     FOREIGN KEY (CPF) REFERENCES Pessoa(CPF)
29 );
30
31 -- Cria tabela Titulo (multivaleorado para Egresso)
32 CREATE TABLE Titulo (
33     CPF_Egresso VARCHAR(11) PRIMARY KEY,
34     CPF_Egresso VARCHAR(11),
35     Nome_Titulo VARCHAR(100) NOT NULL,
36     Data_Emissao VARCHAR(10) NOT NULL
37 );
```

