



# Introdução a SQL

☰ Conhecimentos	3. Linguagem SQL: utilização de comandos para consulta alteração inclusão exclusão ordenação e filtros d
📅 Data da aula	@8 de janeiro de 2024
☰ Tipo	Atividade em Classe Aula Expositiva

## 1. Introdução

SQL (Structured Query Language – Linguagem Estruturada de Consulta), é uma linguagem declarativa desenvolvida para os bancos de dados relacionais, que devido sua simplicidade e facilidade de uso tornou-se um padrão para banco de dados relacionais. O grande diferencial da SQL em relação a outras linguagens de consulta está no seu paradigma, pois a SQL é uma linguagem declarativa e não uma linguagem procedural.

Um projeto de um banco de dados é finalizado com a escolha do banco de dados que será usado e com a criação do modelo físico. O modelo físico está diretamente relacionado ao banco de dados escolhido, uma vez que o mesmo representa a forma como os dados serão armazenados fisicamente. Nesta aula, será utilizado o banco de dados PostgreSQL e veremos como criar o modelo físico baseado no mesmo.

## 1.2 Grupos

Os comandos da SQL são tradicionalmente separados em quatro grupos:

- **DDL (Data Definition Language – Linguagem de Definição de Dados):** Subconjunto utilizado para criar, alterar e excluir tabelas e elementos associados; esse é o grupo que mais muda de um fabricante para outro.
- **DML (Data Manipulation Language – Linguagem de Manipulação de Dados):** Subconjunto dos comandos usado para inserir, atualizar e apagar dados.
- **DCL (Data Control Language – Linguagem de Controle de Dados):** Subconjunto de comandos que controla o acesso dos usuários aos dados.
- **DQL (Data Query Language – Linguagem de Consulta de Dados):** Com apenas um único comando – SELECT – e suas várias cláusulas e opções – nem sempre obrigatórias – permite recuperar os dados de uma ou mais tabelas através de consultas elaboradas como uma descrição do resultado desejado.

Além desses grupos de comandos a SQL tem operadores lógicos, operadores relacionais e funções de agregação que, assim como na DDL, podem mudar de um fabricante para outro.

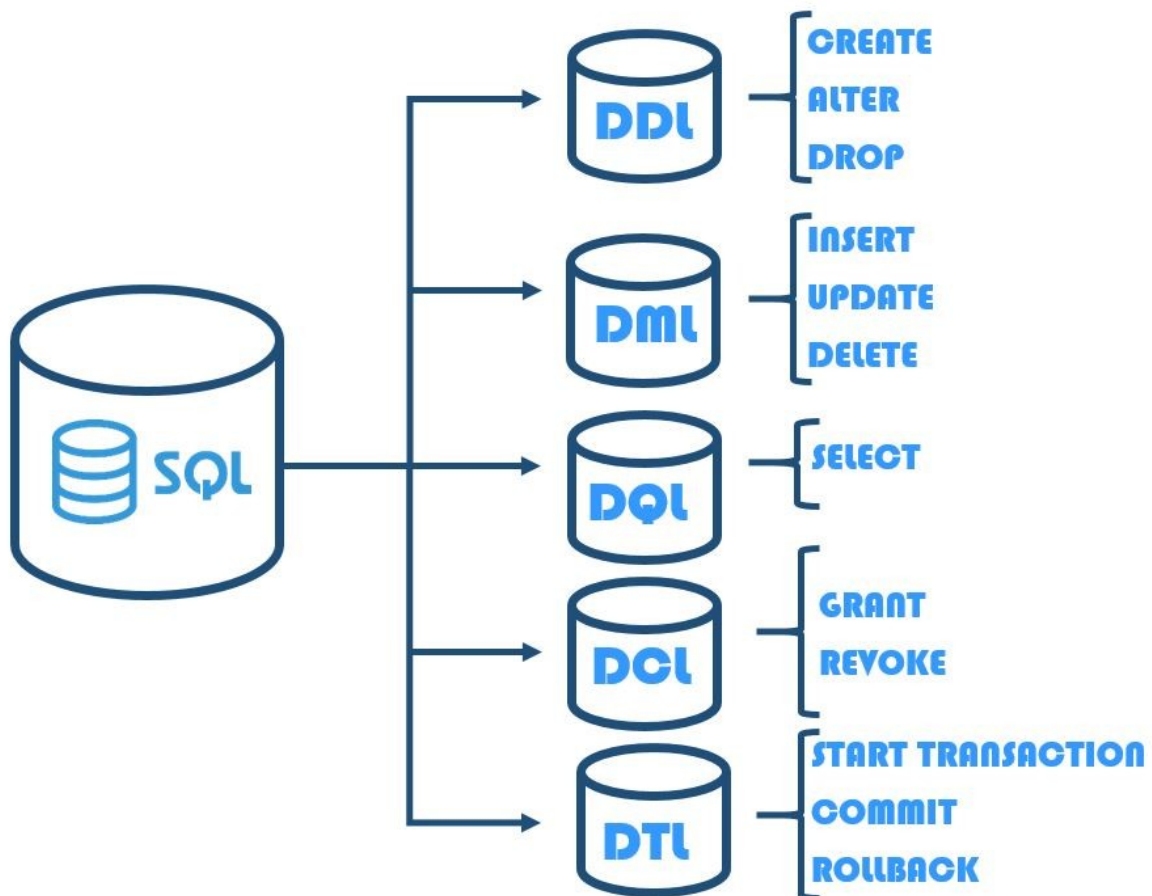


Diagrama dos Grupos de comandos da linguagem SQL

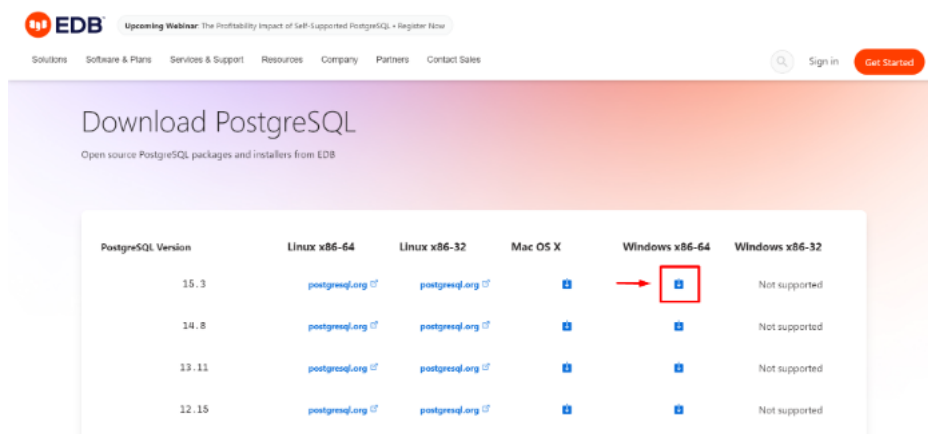
## 2. PostgreSQL

O PostgreSQL é um SGBD proveniente do POSTGRES que foi escrito na Universidade da Califórnia em Berkely. A primeira versão de demonstração do POSTGRES tornou-se operacional em 1987, em 1994 passou a se chamar Postgres95 e em 1996 recebeu o nome PostgreSQL, sendo o nome usado até hoje. O PostgreSQL é um projeto open source coordenado pelo PostgreSQL Global Development Group, tendo seu desenvolvimento sido feito por um grupo de desenvolvedores distribuídos pelo mundo, em sua maioria, voluntários. É considerado “atualmente o mais avançado banco de dados de código aberto disponível em qualquer lugar.”

### 2.1 Instalando o PostgreSQL

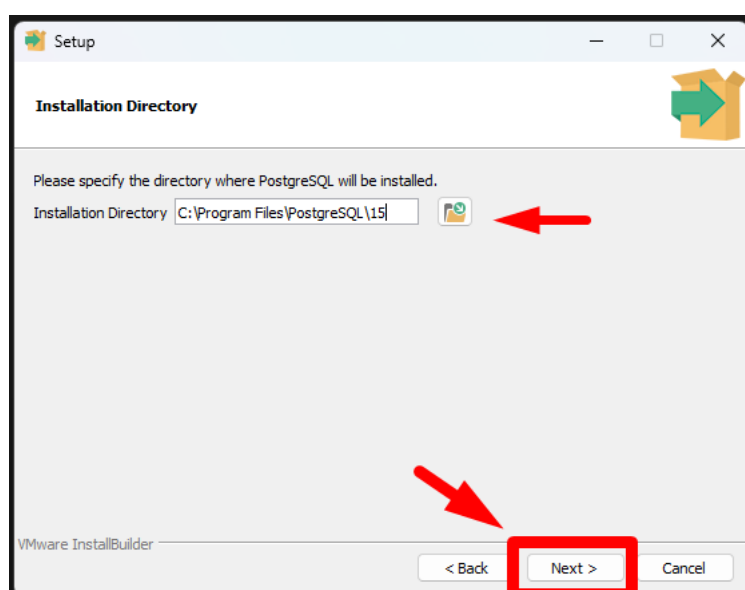
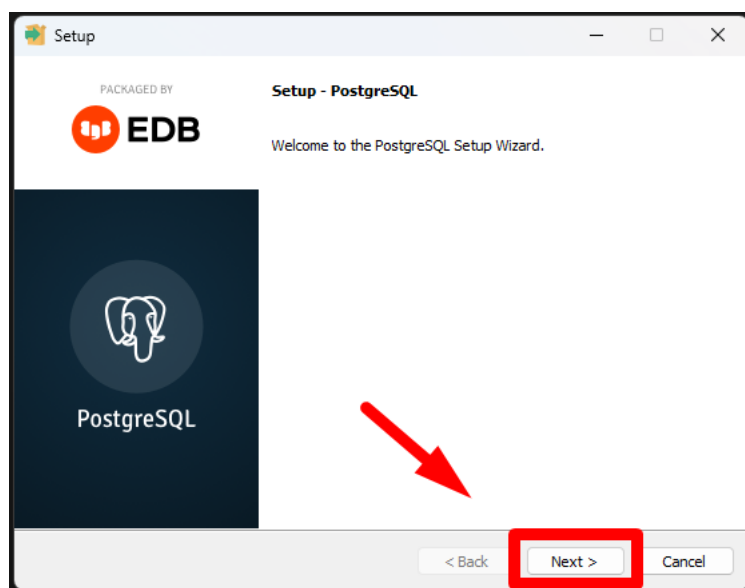
A instalação é bastante simples, são dois passos importantes, instalar o banco e o gerenciador (pgAdmin ou qualquer outro de sua preferência) eles são instalados de fontes diferentes.

No caso vamos utilizar o instalador de 64bits para o Postgres:

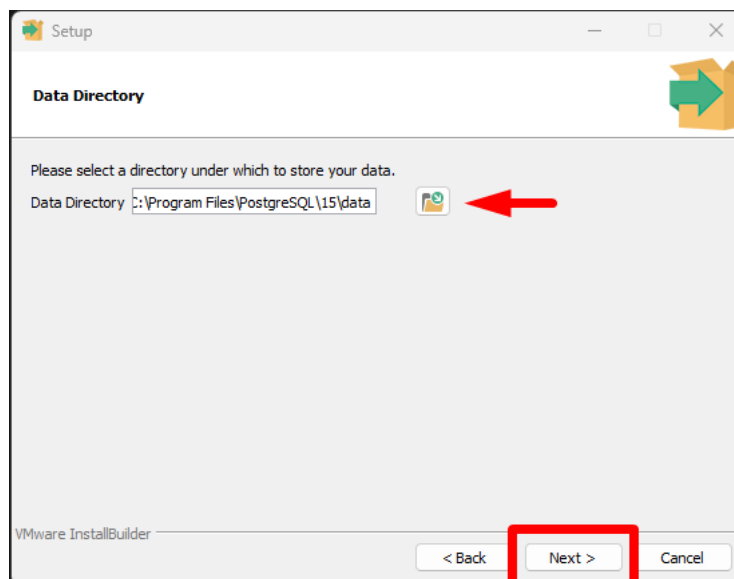
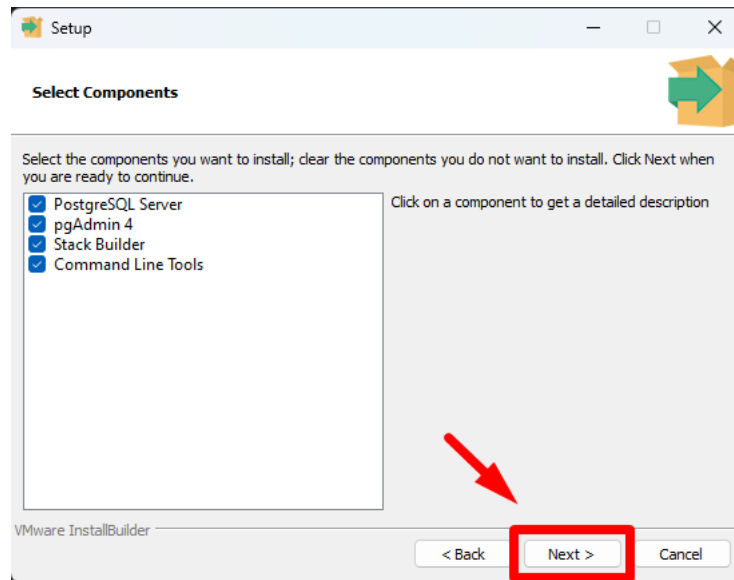


Basta clicar no ícone de download e assim que for concluído execute o arquivo baixado.

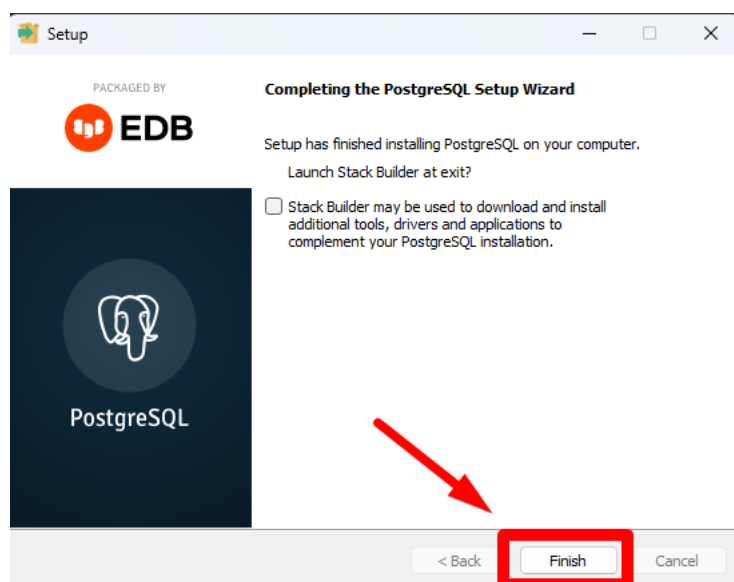
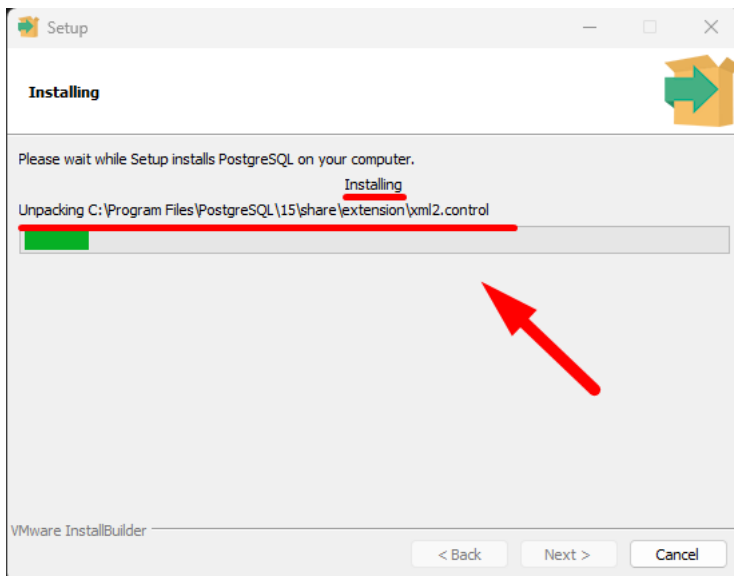
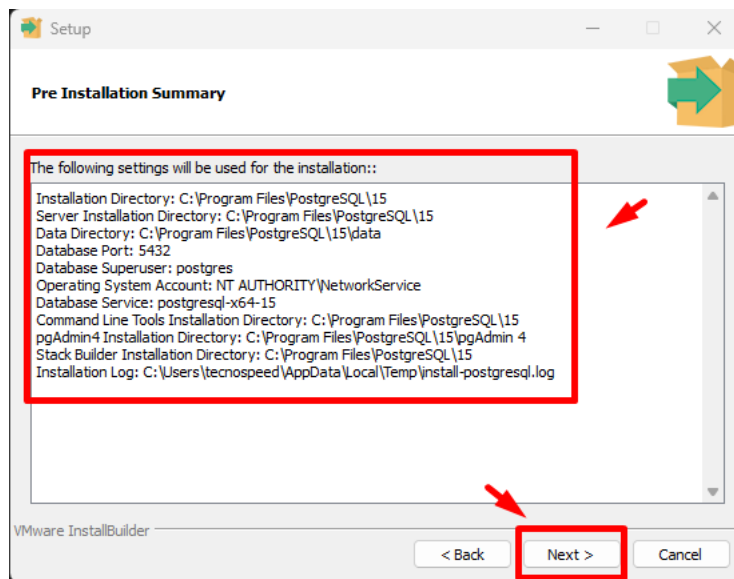
Então após abrir o arquivo, será exibida a tela de instalação e após avançar você deve escolher o diretório de instalação:



Basta clicar em "Next" para avançar após escolher o diretório de "data storage" que é onde os seus dados serão armazenados.



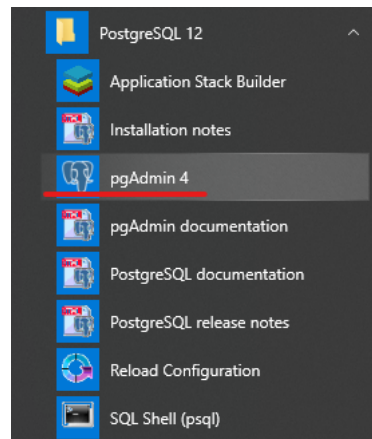
Então será exibida a tela com todas as informações do que será feito pelo instalador:



Então basta deixar essa caixa desmarcada e clicar em "Finish"

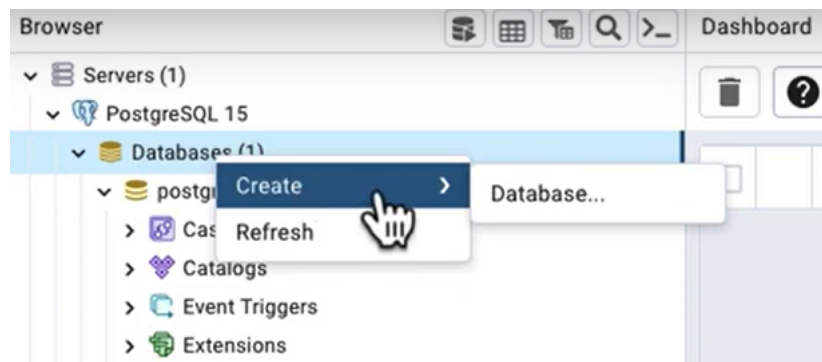
## 2.2 Criando Primeiro Banco de Dados

Para criar nosso primeiro Banco de Dados, precisamos iniciar a interface gráfica do PostgreSQL. Ela se chama pgAdmin.o pgAdmin inicia um servidor Web em nosso localhost. Você pode encontrá-lo no Menu Iniciar:



O servidor será aberto em nosso navegador. Vamos expandir a opção Servers > PostgreSQL 15.

Nesse menu, iremos clicar com o botão direito na opção "Databases" e clicar em Create > Database...



Iremos dar o nome "<insira-qualquer-nome>" para nosso Banco de Dados e clicar em "Save".

Após ter nosso Banco de Dados criado, vamos criar nossa primeira tabela. Para isso, vamos clicar em cima do Banco “<0-  
nome-do-banco-escolhido>” e então clicar no botão “Query Tool”, que fica ao lado da palavra “Browser”.

## 3. Grupos de Comandos SQL

Os comandos são divididos em categorias, cada qual com funcionalidades específicas. Dependendo da literatura utilizada, possuímos quatro ou cinco categorias.

### 3.1 Comandos DDL (Linguagem de Definição de Dados)

São comandos utilizados para construção dos modelos de dados. Por exemplo, se se deseja criar uma tabela (entidade), deve-se utilizar o comando CREATE TABLE para construí-la. Deve ficar claro que os comandos DDL são utilizados somente para criação de estruturas e não inserção de registros de dados.

Entre os comandos DDL temos:

#### 3.1.1 Comando CREATE

Utilizado para criação de tabelas, índices, relacionamentos, entre outros.

Como exemplo tem-se:

```
CREATE TABLE tb_cliente
(
  cli_codigo INTEGER NOT NULL,
  cli_nome VARCHAR2(60) NOT NULL,
  cli_rg VARCHAR2(15) NOT NULL,
  cli_cpf VARCHAR2(15) NOT NULL,
  cli_sexo CHAR(1) NOT NULL,
  PRIMARY KEY (cli_codigo)
);
```

Este exemplo apresenta a criação da entidade tb\_cliente, com cinco atributos, em que são definidos os tipos de dados e a chave primária (cli\_codigo) no final da instrução. O que indica o final da instrução corresponde ao (;) ponto e vírgula.

Para criação do índice único (não permite cadastrar dois clientes com o mesmo nome e com o mesmo CPF), um comando DDL deve ser utilizado, conforme o exemplo:

```
CREATE UNIQUE INDEX uk_cliente ON tb_cliente (cli_nome, cli_cpf);
```

### 3.1.2 Comando ALTER

É utilizado para alterar a estrutura de uma tabela, por exemplo. Através dele pode-se:

- Adicionar colunas:

```
ALTER TABLE TB_CLIENTE ADD CLI_FONE VARCHAR2(14) NOT NULL;
```

- Modificar colunas:

```
ALTER TABLE TB_CLIENTE MODIFY CLI_FONE VARCHAR2(18) NOT NULL;
```

- Dropar colunas:

```
ALTER TABLE TB_CLIENTE DROP COLUMN CLI_FONE;
```

- Marcar colunas como não utilizadas:

```
ALTER TABLE TB_CLIENTE SET UNUSED COLUMN CLI_FONE;
```

- Renomear colunas:

```
ALTER TABLE TB_CLIENTE RENAME COLUMN CLI_FONE TO CLI_TELEFONE;
```

- Definir tabelas como somente leitura:

```
ALTER TABLE TB_CLIENTE READ ONLY;
```

### 3.1.3 Comando DROP

Comando utilizado para apagar tabelas, colunas, índices, entre outros:

```
DROP TABLE tb_cliente;
```

## 3.2 Comandos DML (Linguagem de Manipulação de Dados)

Os comandos de manipulação de dados servem para incluir, alterar, deletar ou selecionar registro das estruturas de dados criadas, utilizando os comandos DDL.

### 3.2.1 Comando INSERT

Utilizado para inserir registros em uma entidade:

```
INSERT INTO tb_bairro (bai_codigo, bai_descricao) VALUES (1, 'CENTRO');
```

### 3.2.2 Comando UPDATE

Utilizado para atualizar registros em uma entidade:

```
UPDATE tb_funcionario  
SET fun_salario = 2000  
WHERE fun_codigo = 10;
```

Perceba que este comando irá atualizar a tabela de funcionários, alterando o salário do funcionário de código = 10 para R\$ 2000,00.



### 3.2.3 Comando DELETE

Utilizado para apagar registro de uma entidade:

```
DELETE FROM tb_cliente WHERE cli_codigo = 5;
```

Neste caso, o comando apagaria da tabela tb\_cliente cujo código for 5.

### 3.2.4 Comando SELECT

Utilizado para selecionar registros em uma ou mais tabelas:

```
SELECT fun_codigo, fun_nome, fun_salario FROM tb_funcionario;
```

Este comando selecionaria, na tabela tb\_funcionario, todos os funcionários e seus respectivos códigos, nomes e salários.

## 3.3 Comando DCL (Linguagem de Controle de Dados)

Permitem controlar quais usuários tem permissão para quais atribuições no banco de dados. O Comando GRANT dá permissão, o comando REVOKE exclui as permissões concedidas.

```
GRANT CREATE_TABLE ON USER_JOSE;  
REVOKE CREATE_TABEL ON USER_JOSE;
```

Respectivamente, dá-se permissão de criação de tabelas ao usuário USER\_JOSE e retira-se a permissão concedida.

## 3.4 Comandos DTL (Linguagem de Transação de Dados)

Estes comandos são necessários para que os dados sejam gravados fisicamente no disco, quando se confirma uma transação como concluída. Desta forma, o comando COMMIT, quando executado, grava toda a transação no disco, sem a possibilidade de retorno em uma situação anterior, a não ser através de mecanismos de backup. Quando se deseja desfazer uma transação, executa-se o comando ROLLBACK, e o sistema desfaz automaticamente tudo o que não foi “comitado” até o momento.

## Atividade

Construir o modelo de dados utilizando comandos SQL, e inserir em cada entidade pelo menos um registro:

