

ESCOLA E FACULDADE DE TECNOLOGIA SENAI GASPAR RICARDO JÚNIOR
DESENVOLVIMENTO DE SISTEMAS

ANDRESSA ALMEIDA BARTOLOMEU CUSTÓDIO
MARCELA MORAES MULATO
NATÁLIA NOGUEIRA
RODRIGO FERREIRA SOARES

BIBLIOTECA

Projeto final de banco de dados

PROFESSOR RESPONSÁVEL: ANDRÉ SOUZA

DISCIPLINA: BANCO DE DADOS

SOROCABA

20/06/2025

SUMÁRIO

1	INTRODUÇÃO.....	3
2	MODELAGEM CONCEITUAL.....	4
2.1	ENTIDADES, ATRIBUTOS E RELACIONAMENTOS.....	4
2.2	DIAGRAMA ENTIDADE-RELACIONAMENTO	5
3	MODELAGEM LÓGICA	7
3.1	NORMALIZAÇÃO	7
4	ESTRUTURA DO BANCO DE DADOS.....	8
4.1	<i>DATA DEFINITION LANGUAGE (DDL)</i>	8
4.2	TABELAS, ATRIBUTOS E RELACIONAMENTOS	13
4.2.1	Tabela principal de livros	13
4.2.2	Tabela de autores	13
4.2.3	Tabela associativa entre livros e autores	13
4.2.4	Tabela de gêneros	13
4.2.5	Tabela associativa entre livros e gêneros	13
4.2.6	Tabela de cargos.....	14
4.2.7	Tabela Pessoa	14
4.2.8	Tabela especialização de Pessoa - Funcionário	14
4.2.9	Tabela especialização de Pessoa - Usuário	14
4.2.10	Tabela de endereço	14
4.2.11	Tabela de telefone.....	14
4.2.12	Tabela de email	14
4.2.13	Tabela de empréstimos	15
4.2.14	Tabela de histórico de <i>status</i> do empréstimo	15
4.2.15	Tabela de livros emprestados.....	15
4.2.16	Tabela de histórico de <i>status</i> dos livros emprestados.....	15
5	MANIPULAÇÃO DE DADOS	16
5.1	<i>DATA MANIPULATION LANGUAGE (DML)</i>	16
5.2	<i>DATA QUERY LANGUAGE (DQL)</i>	25
6	CONTROLE DE ACESSO	29
6.1	<i>DATA CONTROL LANGUAGE (DCL)</i>	29
7	CONTROLE DE TRANSAÇÕES.....	39

7.1	<i>DATA TRANSACTION LANGUAGE (DTL)</i>	39
8	CONCLUSÃO	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

O tema desse projeto é um sistema de gerenciamento de uma biblioteca comunitária de porte médio, que foi implementada visando a aplicação dos conceitos estudados pela equipe durante as aulas de Banco de Dados. O banco de dados Biblioteca permite a visualização e manipulação dos dados que fazem parte de um sistema literário, gerenciando dados de empréstimos, funcionários e leitores.

O projeto foi implementado no sistema de banco de dados relacional PostgreSQL, com sua modelagem realizada na plataforma digital do Mermaid Chart Inc., utilizando sua ferramenta baseada na linguagem JavaScript para a criação de um diagrama mais limpo e rápido.

2 MODELAGEM CONCEITUAL

O banco de dados é uma coleção em que são armazenados os dados utilizados por um sistema. A biblioteca teve sua modelagem pensada nas funcionalidades de uma biblioteca real, considerando a sua utilização por seus funcionários e leitores.

A modelagem conceitual é a representação do banco de dados em formato de diagrama, feito a partir da análise das necessidades do sistema.

2.1 ENTIDADES, ATRIBUTOS E RELACIONAMENTOS

A *database* Biblioteca possui dezesseis entidades. "Pessoa" é a primeira entidade do banco, e possui duas especializações: "Pessoa_Usuario" e "Pessoa_Funcionario". "Pessoa" possui os atributos gerais de qualquer pessoa, seja ela um usuário ou um funcionário, como nome, Cadastro de Pessoa Física (CPF) e data de nascimento. O usuário comum possui adicionalmente a data de cadastro na biblioteca, e o funcionário possui a identificação do seu cargo e data de admissão, a entidade "Cargo" é exclusivamente relacionada ao funcionário, e possui dados como o nome do cargo, o salário e a carga horária que deve ser cumprida pelo funcionário.

"Pessoa" se relaciona também com as entidades "Endereco", com atributos como bairro, número, cidade, rua, Código de Endereçamento Postal (CEP) e estado, "Telefone", com Discagem Direta à Distância (DDD), número do telefone e tipo (se é residencial ou celular), e "Email", que salva os e-mails da pessoa.

Ambos - usuários e funcionários - estão relacionados à entidade "Emprestimo", em que são armazenados quem solicitou o empréstimo do livro e quem o registrou no sistema. "Empréstimo", por sua vez, se relaciona com a entidade "Historico_Emprestimo", que revela a ordem dos livros emprestados com datas e descrições, e com a entidade "Livro_Emprestado", que registra a qual empréstimo o livro foi associado e qual livro foi emprestado, de tal modo, em um único empréstimo podem ser adicionados vários livros.

Já "Livro_Emprestado" se conecta com "Historico_Livro_Emprestado", com informações como ordem, descrição do empréstimo e *status* desse empréstimo, e com "Livro", que possui as informações do objeto do empréstimo, com título, data de lançamento, editora, país de origem, edição, idioma e International Standart Book Number (ISBN, código de identificação de livros internacional), assim, cada livro tem seu próprio *status*, como "Disponível", "Emprestado" ou "Reservado".

Um livro pode ser escrito por várias pessoas diferentes, então "Livro" se relaciona com "Autor", que possui os dados do escritor como nome e nacionalidade, e com a tabela associativa "Livro_Autor", já que é um relacionamento múltiplo em ambas as vias. O mesmo ocorre com a entidade "Genero" e sua tabela associativa "Livro_Genero".

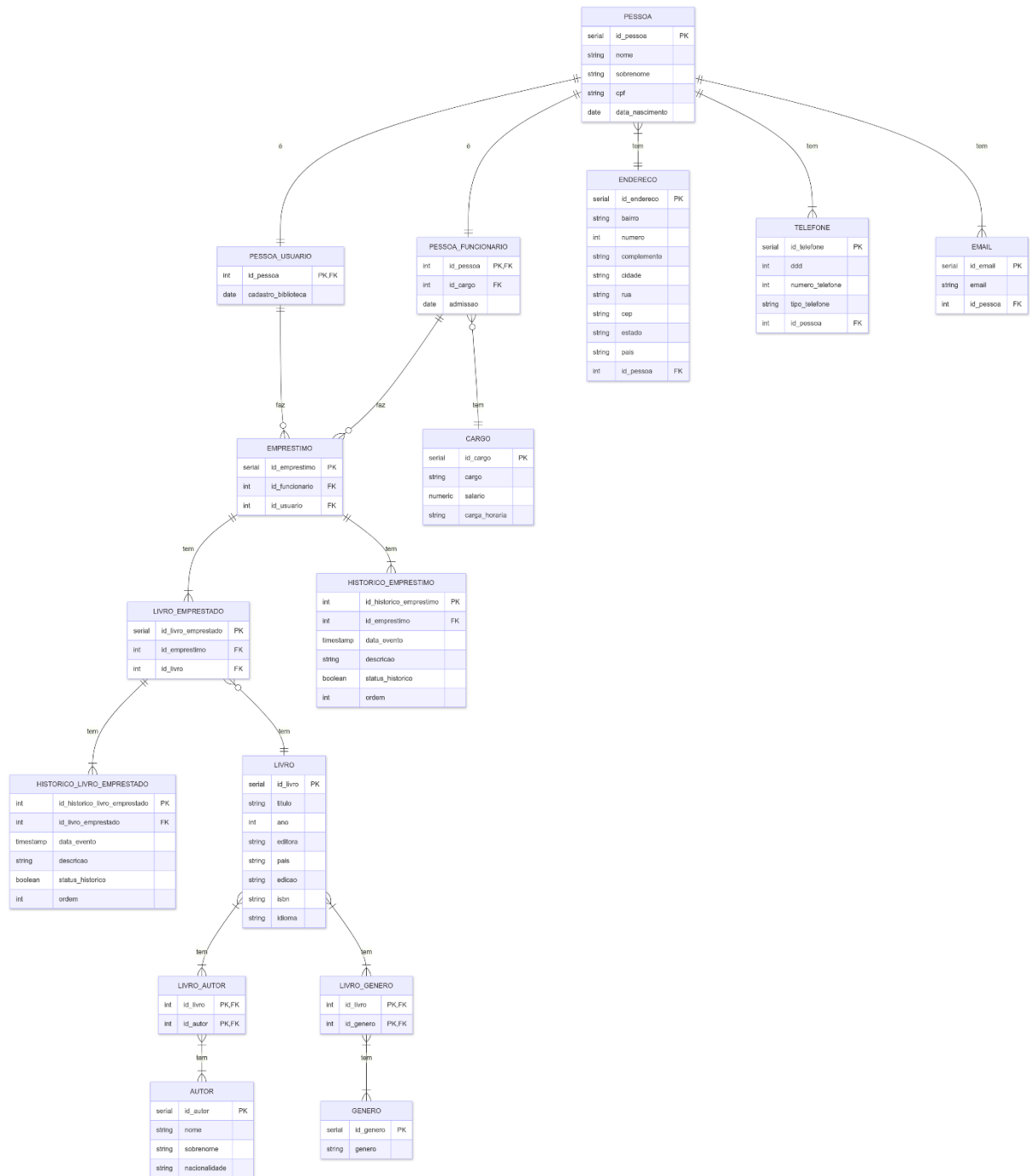
2.2 DIAGRAMA ENTIDADE-RELACIONAMENTO

O diagrama de entidade-relacionamento é a representação visual do banco de dados, especificando como as diferentes entidades se relacionam, facilitando a manipulação desses dados dentro do sistema. A equipe optou por utilizar a nomenclatura do Mermaid para uma organização digital mais clara e acessível.

O diagrama abaixo pode ser acessado também no endereço eletrônico:

< <https://www.mermaidchart.com/app/projects/44ff6337-dfa6-4d0a-8977-d1aa44be2a2b/diagrams/48611d71-ff6e-4917-916a-ea95d4f1a3ab/share/invite/eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkb2N1bWVudEEljojNDg2MTFkNzEtZmY2ZS00OTE3LTkxNmEtZWE5NWQ0ZjFhM2FiliwiYWVudEEljojRWRpdCI6MTc1MDM1OTEzOH0.utjg4IsOKwnblg7HjZEI0KRKLejXavsLbUYU-BElyE>> Acesso em: 10 jun. 2025.

Figura 1 - Diagrama Entidade-Relacionamento



Fonte: própria (2025).

3 MODELAGEM LÓGICA

Durante a modelagem do banco de dados, o grupo realizou diversas reuniões para aperfeiçoar o modelo do banco de dados, conforme as aulas progrediam, sempre aplicando os novos conhecimentos ao modelo. Durante essas reuniões, as tabelas foram iniciadas de maneira mais resumida, sem as tabelas associativas e especializações e foram sendo progressivamente aperfeiçoadas conforme a normalização ocorria, o que acarretou a criação de novas chaves estrangeiras e atributos.

3.1 NORMALIZAÇÃO

A normalização é o processo em que o banco de dados é organizado, relacionando as tabelas do banco de maneira a proteger os dados e tornar a base de dados mais flexível, eliminando inconsistências e redundâncias.

O banco de dados Biblioteca atende à primeira forma normal, que estabelece a utilização de valores atômicos e chaves primárias singulares, à segunda forma normal, em que as tabelas não possuem dependências parciais, e à terceira forma normal, em que as relações não possuem dependências transitivas. A fim de manter o banco normalizado, mas também funcional e não redundante, foi realizada a especialização da tabela pessoa em pessoa_usuario e pessoa_funcionário. A especialização mantém os atributos comuns na tabela pessoa, enquanto as suas especializações possuem apenas os atributos que se diferem.

4 ESTRUTURA DO BANCO DE DADOS

Depois da modelagem, o banco Biblioteca passou a ser implementado e testado no PostgreSQL.

4.1 DATA DEFINITION LANGUAGE (DDL)

A criação do *script* de DDL é quem construiu o banco, definindo as tabelas e atributos, como pode ser observado abaixo:

```
CREATE DATABASE biblioteca;
```

```
-- Tabela principal de livros
```

```
CREATE TABLE livro (  
    id_livro SERIAL PRIMARY KEY,  
    titulo VARCHAR(200) NOT NULL,  
    ano INTEGER,  
    editora VARCHAR(100) NOT NULL,  
    pais VARCHAR(50),  
    edicao VARCHAR(50),  
    isbn VARCHAR(13) NOT NULL,  
    idioma VARCHAR(50)  
);
```

```
-- Tabela de autores
```

```
CREATE TABLE autor (  
    id_autor SERIAL PRIMARY KEY,  
    nome VARCHAR(200) NOT NULL,  
    sobrenome VARCHAR(200) NOT NULL,  
    nacionalidade VARCHAR(100)  
);
```

```
-- Tabela associativa entre livros e autores
```

```
CREATE TABLE livro_autor (  
    id_livro INTEGER NOT NULL,
```

```

        id_autor INTEGER NOT NULL,
        PRIMARY KEY (id_livro, id_autor),
        FOREIGN KEY (id_livro) REFERENCES livro(id_livro) ON DELETE
CASCADE,
        FOREIGN KEY (id_autor) REFERENCES autor(id_autor) ON DELETE
CASCADE
    );

```

-- Tabela de gêneros

```

CREATE TABLE genero (
    id_genero SERIAL PRIMARY KEY,
    genero VARCHAR(100) NOT NULL
);

```

-- Tabela associativa entre livros e gêneros

```

CREATE TABLE livro_genero (
    id_livro INTEGER NOT NULL,
    id_genero INTEGER NOT NULL,
    PRIMARY KEY (id_livro, id_genero),
    FOREIGN KEY (id_livro) REFERENCES livro(id_livro) ON DELETE
CASCADE,
    FOREIGN KEY (id_genero) REFERENCES genero(id_genero) ON DELETE
CASCADE
);

```

-- Tabela de cargos

```

CREATE TABLE cargo (
    id_cargo SERIAL PRIMARY KEY,
    cargo VARCHAR(100) NOT NULL,
    carga_horaria VARCHAR(100) NOT NULL,
    salario numeric NOT NULL
);

```

-- Tabela Pessoa

```
CREATE TABLE pessoa (
    id_pessoa SERIAL PRIMARY KEY,
    nome VARCHAR(200) NOT NULL,
    sobrenome VARCHAR(200) NOT NULL,
    cpf VARCHAR(14) NOT NULL UNIQUE,
    data_nascimento DATE NOT NULL
);
```

-- Tabela especialização de Pessoa - Funcionário

```
CREATE TABLE pessoa_funcionario (
    id_pessoa INTEGER PRIMARY KEY,
    id_cargo INTEGER NOT NULL,
    admissao DATE NOT NULL,
    FOREIGN KEY (id_cargo) REFERENCES cargo(id_cargo),
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa) ON DELETE
CASCADE
);
```

-- Tabela especialização de Pessoa - Usuário

```
CREATE TABLE pessoa_usuario (
    id_pessoa INTEGER PRIMARY KEY,
    cadastro_biblioteca DATE NOT NULL,
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa) ON DELETE
CASCADE
);
```

-- Tabela de endereço

```
CREATE TABLE endereco (
    id_endereco SERIAL PRIMARY KEY,
    bairro VARCHAR(100) NOT NULL,
    numero INTEGER,
    complemento VARCHAR(100),
    cidade VARCHAR(100) NOT NULL,
    rua VARCHAR(100) NOT NULL,
```

```

        cep VARCHAR(20) NOT NULL,
        estado VARCHAR(50) NOT NULL,
        pais VARCHAR(50) NOT NULL,
        id_pessoa INTEGER NOT NULL,
        FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa) ON DELETE
CASCADE
    );

```

```

-- Tabela de telefone genérica
CREATE TABLE telefone (
    id_telefone SERIAL PRIMARY KEY,
    ddd VARCHAR(5) NOT NULL,
    numero_telefone VARCHAR(20) NOT NULL,
    tipo_telefone VARCHAR(20) NOT NULL DEFAULT 'Celular'
        CHECK (tipo_telefone IN ('Celular', 'Fixo', 'Recado')),
    id_pessoa INTEGER NOT NULL,
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa) ON DELETE
CASCADE
    );

```

```

-- Tabela de email genérica
CREATE TABLE email (
    id_email SERIAL PRIMARY KEY,
    email VARCHAR(255) NOT NULL,
    id_pessoa INTEGER NOT NULL,
    FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa) ON DELETE
CASCADE
    );

```

```

-- Empréstimos
CREATE TABLE emprestimo (
    id_emprestimo SERIAL PRIMARY KEY,
    id_funcionario INTEGER NOT NULL,
    id_usuario INTEGER NOT NULL,

```

```

        data_devolucao DATE NOT NULL,
        FOREIGN          KEY          (id_funcionario)          REFERENCES
        pessoa_funcionario(id_pessoa) ON DELETE CASCADE,
        FOREIGN KEY (id_usuario) REFERENCES pessoa_usuario(id_pessoa) ON
        DELETE CASCADE
    );

```

```

-- Histórico de status do empréstimo
CREATE TABLE historico_emprestimo (
    id_historico_emprestimo SERIAL PRIMARY KEY,
    id_emprestimo INTEGER NOT NULL,
    data_evento TIMESTAMP NOT NULL,
    descricao VARCHAR(255) NOT NULL,
    ordem INTEGER NOT NULL,
    FOREIGN          KEY          (id_emprestimo)          REFERENCES
    emprestimo(id_emprestimo) ON DELETE CASCADE
);

```

```

-- Livros emprestados
CREATE TABLE livro_emprestado (
    id_livro_emprestado SERIAL PRIMARY KEY,
    id_emprestimo INTEGER NOT NULL,
    id_livro INTEGER NOT NULL,
    FOREIGN KEY (id_livro) REFERENCES livro(id_livro) ON DELETE
    CASCADE,
    FOREIGN          KEY          (id_emprestimo)          REFERENCES
    emprestimo(id_emprestimo) ON DELETE CASCADE
);

```

```

-- Histórico de status dos livros emprestados
CREATE TABLE historico_livro_emprestado (
    id_historico_livro_emprestado SERIAL PRIMARY KEY,
    id_livro_emprestado INTEGER NOT NULL,
    data_evento TIMESTAMP NOT NULL,

```

```

        descricao VARCHAR(255) NOT NULL,
        status_historico VARCHAR(50) NOT NULL,
        ordem INTEGER NOT NULL,
        FOREIGN KEY (id_livro_emprestado) REFERENCES
livro_emprestado(id_livro_emprestado) ON DELETE CASCADE
);

```

4.2 TABELAS, ATRIBUTOS E RELACIONAMENTOS

Abaixo é possível visualizar a descrição de cada uma das tabelas.

4.2.1 Tabela principal de livros

Livro possui as informações de cada livro possuído pela biblioteca. Livro possui uma relação N:N com Autor e uma relação N:N com gênero, uma vez que um livro pode ter sido escrito por várias pessoas e possuir mais de um gênero.

4.2.2 Tabela de autores

Autor possui as informações básicas sobre os autores dos livros da biblioteca. Autor possui uma relação N:N com Livro.

4.2.3 Tabela associativa entre livros e autores

Livro_Autor é a tabela que surgiu a partir da relação N:N entre Livro e Autor, e possui ambas suas chaves primárias como meio de ligação.

4.2.4 Tabela de gêneros

Gênero possui o nome e a identificação de cada um dos gêneros presentes nos livros da biblioteca. Gênero possui uma ligação N:N com Livro.

4.2.5 Tabela associativa entre livros e gêneros

Assim como Livro_Autor, Livro_Gênero é a tabela que surgiu a partir da relação N:N entre Livro e Gênero.

4.2.6 Tabela de cargos

Cargo é a tabela que possui as informações referentes a determinado cargo, como o título e salário. Cargo possui uma relação N:1 com tabela Pessoa_Funcionário.

4.2.7 Tabela Pessoa

Pessoa possui as informações sobre qualquer pessoa envolvida com a biblioteca. Pessoa possui duas especializações: Funcionário e Usuário, uma relação 1:N com Endereço, N:1 com Telefone e N:1 com Email.

4.2.8 Tabela especialização de Pessoa - Funcionário

Pessoa_Funcionário possui as informações adicionais de um funcionário da biblioteca como cargo e data de admissão.

4.2.9 Tabela especialização de Pessoa - Usuário

Pessoa_Usuário possui as informações adicionais de um leitor da biblioteca como a data do cadastro inicial na biblioteca.

4.2.10 Tabela de endereço

Endereço possui as informações de um endereço como CEP e especializações sobre o local como rua, número e bairro. Endereço possui uma relação N:1 com Pessoa.

4.2.11 Tabela de telefone

Telefone possui as informações de um telefone: DDD, o número e o tipo de número (fixo ou celular). Telefone possui uma relação 1:N com Pessoa.

4.2.12 Tabela de email

Email possui contém o e-mail da pessoa. Email possui uma relação 1:N com Pessoa.

4.2.13 Tabela de empréstimos

Empréstimo contém o id do funcionário que realizou o empréstimo e do usuário que solicitou o empréstimo. Empréstimo possui uma relação 1:N com Pessoa_Usuário.

4.2.14 Tabela de histórico de *status* do empréstimo

Histórico_Empréstimo possui as informações sobre o empréstimo como data do empréstimo e descrição. Histórico_Empréstimo possui uma relação 1:N com Empréstimo.

4.2.15 Tabela de livros emprestados

Livro_Emprestado possui as informações sobre o livro emprestado, como a data e a descrição. Livro_Emprestado possui uma relação 1:N com Empréstimo, N:1 com Histórico_Livro_Emprestado, e 1:N com Livro.

4.2.16 Tabela de histórico de *status* dos livros emprestados

Histórico_Livro_Emprestado possui as informações sobre o histórico de Livro_Emprestado. Histórico_Livro_Emprestado possui uma relação 1:N com Livro_Emprestado.

5 MANIPULAÇÃO DE DADOS

A manipulação de dados é ação de manipular os dados conforme necessário. Essa manipulação inclui inserção, atualização e exclusão de dados, assim como a leitura desses dados como for necessário, visualizando dados específicos de tabelas diferentes, por exemplo.

5.1 DATA MANIPULATION LANGUAGE (DML)

Abaixo se pode observar o *script* do DML com o *Data Transaction Language (DTL)* aplicado.

```
BEGIN;
```

```
-- Inserir um novo gênero:
```

```
INSERT INTO genero (id_genero, genero) VALUES (1, 'Fantasia');
```

```
INSERT INTO genero (id_genero, genero) VALUES (2, 'Romance');
```

```
INSERT INTO genero (id_genero, genero) VALUES (3, 'Terror');
```

```
INSERT INTO genero (id_genero, genero) VALUES (4, 'Ficção Científica');
```

```
INSERT INTO genero (id_genero, genero) VALUES (5, 'Aventura');
```

```
-- Inserir um novo autor:
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (1,  
'J.K.', 'Rowling', 'Britânica');
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (2,  
'George', 'Orwell', 'Britânica');
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (3,  
'Machado', 'Assis', 'Brasileira');
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (4,  
'Stephen', 'King', 'Americana');
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (5,  
'Clarice', 'Lispector', 'Brasileira');
```

```
INSERT INTO autor (id_autor, nome, sobrenome, nacionalidade) VALUES (6,  
'Isaac', 'Asimov', 'Russo-americana');
```

-- Inserir um novo livro:

```
INSERT INTO livro (id_livro, titulo, ano, editora, pais, edicao, isbn, idioma)
VALUES (1, 'Harry Potter e a Pedra Filosofal', 1997, 'Rocco', 'Reino Unido', '1ª',
'9788532530783', 'Português');

INSERT INTO livro (id_livro, titulo, ano, editora, pais, edicao, isbn, idioma)
VALUES (2, '1984', 1949, 'Companhia das Letras', 'Reino Unido', '1ª',
'9788571644040', 'Português');

INSERT INTO livro (id_livro, titulo, ano, editora, pais, edicao, isbn, idioma)
VALUES (3, 'Dom Casmurro', 1899, 'Saraiva', 'Brasil', '3ª', '9788502117203',
'Português');

INSERT INTO livro (id_livro, titulo, ano, editora, pais, edicao, isbn, idioma)
VALUES (4, 'It: A Coisa', 1986, 'Suma', 'EUA', '1ª', '9788556510780',
'Português');

INSERT INTO livro (id_livro, titulo, ano, editora, pais, edicao, isbn, idioma)
VALUES (5, 'A Hora da Estrela', 1977, 'Rocco', 'Brasil', '1ª', '9788532525437',
'Português');
```

-- Associação entre livro e autor:

```
INSERT INTO livro_autor (id_livro, id_autor) VALUES (1, 1);
INSERT INTO livro_autor (id_livro, id_autor) VALUES (2, 2);
INSERT INTO livro_autor (id_livro, id_autor) VALUES (3, 3);
INSERT INTO livro_autor (id_livro, id_autor) VALUES (4, 4);
INSERT INTO livro_autor (id_livro, id_autor) VALUES (5, 5);
INSERT INTO livro_autor (id_livro, id_autor) VALUES (4, 6);
```

-- Associação entre livro e gênero:

```
INSERT INTO livro_genero (id_livro, id_genero) VALUES (1, 1);
INSERT INTO livro_genero (id_livro, id_genero) VALUES (2, 2);
INSERT INTO livro_genero (id_livro, id_genero) VALUES (3, 3);
INSERT INTO livro_genero (id_livro, id_genero) VALUES (4, 4);
INSERT INTO livro_genero (id_livro, id_genero) VALUES (5, 5);
```

-- Inserir cargo:

```
INSERT INTO cargo (id_cargo, cargo, carga_horaria, salario)
```

```
VALUES (1, 'Bibliotecário', '25 horas semanais', 2200);
INSERT INTO cargo (id_cargo, cargo, carga_horaria, salario)
VALUES (2, 'Atendente', '20 horas semanais', 1800);
INSERT INTO cargo (id_cargo, cargo, carga_horaria, salario)
VALUES (3, 'Auxiliar de Biblioteca', '30 horas semanais', 2000);
INSERT INTO cargo (id_cargo, cargo, carga_horaria, salario)
VALUES (4, 'Técnico de Informação', '40 horas semanais', 2600);
INSERT INTO cargo (id_cargo, cargo, carga_horaria, salario)
VALUES (5, 'Gerente', '40 horas semanais', 4000);
```

-- Inserir um novo usuário:

```
INSERT INTO pessoa (id_pessoa, nome, sobrenome, cpf, data_nascimento)
VALUES (1, 'Ana Clara', 'Silva', '12345678901', '2001-03-15');
INSERT INTO pessoa_usuario (id_pessoa, cadastro_biblioteca)
VALUES (1, '2023-03-15');
INSERT INTO pessoa (id_pessoa, nome, sobrenome, cpf, data_nascimento)
VALUES (2, 'João Pedro', 'Souza', '47565670001', '1985-11-09');
INSERT INTO pessoa_funcionario (id_pessoa, id_cargo, admissao)
VALUES (2, 5, '2005-01-15');
INSERT INTO pessoa (id_pessoa, nome, sobrenome, cpf, data_nascimento)
VALUES (3, 'Julia', 'Castanhas', '3465670001', '2006-10-06');
INSERT INTO pessoa_usuario (id_pessoa, cadastro_biblioteca)
VALUES (3, '2024-01-10');
INSERT INTO pessoa (id_pessoa, nome, sobrenome, cpf, data_nascimento)
VALUES (4, 'Rodrigo', 'Soares', '22565670001', '1990-07-19');
INSERT INTO pessoa_funcionario (id_pessoa, id_cargo, admissao)
INSERT INTO pessoa_funcionario (id_pessoa, id_cargo, admissao)
VALUES (4, 1, '2005-02-11');
INSERT INTO pessoa (id_pessoa, nome, sobrenome, cpf, data_nascimento)
VALUES (5, 'Maria', 'Barbosa', '74656704301', '1990-06-26');
INSERT INTO pessoa_usuario (id_pessoa, cadastro_biblioteca)
VALUES (5, '2025-04-12');
```

-- Inserir um novo e-mail:

```
INSERT INTO email (id_email, email, id_pessoa)VALUES (1, 'ana@gmail.com',
1);
```

```
INSERT INTO email (id_email, email, id_pessoa) VALUES (2,
'joao@email.com',2);
```

```
INSERT INTO email (id_email, email, id_pessoa) VALUES (3,
'julia@email.com',3);
```

```
INSERT INTO email (id_email, email, id_pessoa) VALUES (4,
'rodrigo@email.com', 4);
```

```
INSERT INTO email (id_email, email, id_pessoa) VALUES (5,
'maria@email.com', 5);
```

-- Inserir um novo endereço:

```
INSERT INTO endereco (id_endereco, bairro, numero, complemento, cidade,
rua, cep, estado, pais, id_pessoa)
```

```
VALUES (1, 'Andador 123', 42, 'Bloco 1', 'Jandira', 'Boas vistas', '17821321',
'São Paulo', 'Brasil', 1);
```

```
INSERT INTO endereco (id_endereco, bairro, numero, complemento, cidade,
rua, cep, estado, pais, id_pessoa)
```

```
VALUES (2, 'Centro', 100, 'Apto 1', 'São Paulo', 'Rua A', '01001000', 'SP',
'Brasil', 2);
```

```
INSERT INTO endereco (id_endereco, bairro, numero, complemento, cidade,
rua, cep, estado, pais, id_pessoa)
```

```
VALUES (3, 'Jardins', 200, 'Casa', 'Rio de Janeiro', 'Rua B', '22020000', 'RJ',
'Brasil', 3);
```

```
INSERT INTO endereco (id_endereco, bairro, numero, complemento, cidade,
rua, cep, estado, pais, id_pessoa)
```

```
VALUES (4, 'Savassi', 300, 'Apto 501', 'Belo Horizonte', 'Av. C', '30130000',
'MG', 'Brasil', 4);
```

```
INSERT INTO endereco (id_endereco, bairro, numero, complemento, cidade,
rua, cep, estado, pais, id_pessoa)
```

```
VALUES (5, 'Batel', 400, NULL, 'Curitiba', 'Rua D', '80240000', 'PR', 'Brasil', 5);
```

-- Inserir um novo telefone:

```
INSERT INTO telefone (id_telefone, ddd, numero_telefone, tipo_telefone,
id_pessoa)
```

```
VALUES (1, '11', '998622463', 'Celular', 1);
```

```
INSERT INTO telefone (id_telefone, ddd, numero_telefone, tipo_telefone,
id_pessoa)
```

```
VALUES (2, '11', '987654321', 'Celular', 2);
```

```
INSERT INTO telefone (id_telefone, ddd, numero_telefone, tipo_telefone,
id_pessoa)
```

```
VALUES (3, '21', '999888777', 'Fixo', 3);
```

```
INSERT INTO telefone (id_telefone, ddd, numero_telefone, tipo_telefone,
id_pessoa)
```

```
VALUES (4, '31', '912345678', 'Fixo', 4);
```

```
INSERT INTO telefone (id_telefone, ddd, numero_telefone, tipo_telefone,
id_pessoa)
```

```
VALUES (5, '41', '976543210', 'Celular', 5);
```

```
-- Inserir um novo empréstimo:
```

```
INSERT INTO emprestimo (id_emprestimo, id_funcionario, id_usuario,
data_devolucao)
```

```
VALUES (1, 2, 1, '2023-10-10');
```

```
INSERT INTO emprestimo (id_emprestimo, id_funcionario, id_usuario,
data_devolucao)
```

```
VALUES (2, 2, 1, '2023-10-12');
```

```
INSERT INTO emprestimo (id_emprestimo, id_funcionario, id_usuario,
data_devolucao)
```

```
VALUES (3, 2, 3, '2023-10-15');
```

```
INSERT INTO emprestimo (id_emprestimo, id_funcionario, id_usuario,
data_devolucao)
```

```
VALUES (4, 2, 5, '2023-10-18');
```

```
INSERT INTO emprestimo (id_emprestimo, id_funcionario, id_usuario,
data_devolucao)
```

```
VALUES (5, 2, 1, '2023-10-12');
```

```
-- Inserir novos status e históricos de empréstimo:
```

```

INSERT INTO historico_emprestimo (id_historico_emprestimo, id_emprestimo,
data_evento, descricao, ordem)
VALUES (1, 2, '2023-10-01', 'Empréstimo realizado', 1);
INSERT INTO historico_emprestimo (id_historico_emprestimo, id_emprestimo,
data_evento, descricao, ordem)
VALUES (2, 2, '2023-10-01', 'Empréstimo realizado', 2);
INSERT INTO historico_emprestimo (id_historico_emprestimo, id_emprestimo,
data_evento, descricao, ordem)
VALUES (3, 2, '2023-10-02', 'Empréstimo realizado', 3);
INSERT INTO historico_emprestimo (id_historico_emprestimo, id_emprestimo,
data_evento, descricao, ordem)
VALUES (4, 2, '2023-10-03', 'Empréstimo aumentado', 4);
INSERT INTO historico_emprestimo (id_historico_emprestimo, id_emprestimo,
data_evento, descricao, ordem)
VALUES (5, 2, '2023-10-04', 'Empréstimo finalizado', 5);

-- Inserir um novo livro emprestado:
INSERT INTO livro_emprestado (id_livro_emprestado, id_emprestimo, id_livro)
VALUES (1, 1, 2);
INSERT INTO livro_emprestado (id_livro_emprestado, id_emprestimo, id_livro)
VALUES (2, 2, 4);
INSERT INTO livro_emprestado (id_livro_emprestado, id_emprestimo, id_livro)
VALUES (3, 3, 1);
INSERT INTO livro_emprestado (id_livro_emprestado, id_emprestimo, id_livro)
VALUES (4, 4, 3);
INSERT INTO livro_emprestado (id_livro_emprestado, id_emprestimo, id_livro)
VALUES (5, 5, 5);

-- Inserir novos status e históricos de livros emprestados:
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)
VALUES (1, 1, '2023-10-01', 'Emprestado', 'Ativo', 1);
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)

```

```

VALUES (2, 2, '2023-10-01', 'Emprestado', 'Ativo', 1);
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)
VALUES (3, 3, '2023-10-02', 'Emprestado', 'Ativo', 1);
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)
VALUES (4, 4, '2023-10-03', 'Empréstimo aumentado', 'Ativo', 2);
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)
VALUES (5, 5, '2023-10-02', 'Emprestado', 'Inativo', 1);
INSERT INTO historico_livro_emprestado (id_historico_livro_emprestado,
id_livro_emprestado, data_evento, descricao, status_historico, ordem)
VALUES (6, 5, '2023-10-04', 'Disponível', 'Ativo', 2);

```

```
SAVEPOINT original;
```

```
-- Atualizar o status de um livro:
```

```

UPDATE livro
SET idioma = 'Francês'
WHERE id_livro = 1;

```

```

UPDATE livro
SET titulo = '1984 (Edição Especial)'
WHERE id_livro = 2;

```

```

UPDATE livro
SET edicao = '2ª'
WHERE id_livro = 3;

```

```

UPDATE livro
SET ano = 2020
WHERE id_livro = 4;

```

```
UPDATE livro
```

```
SET editora = 'Nova Fronteira'  
WHERE id_livro = 5;
```

-- Corrigir o nome de um autor:

```
UPDATE autor  
SET nome = 'Joanne'  
WHERE id_autor = 1;
```

```
UPDATE autor  
SET nacionalidade = 'Luso-Brasileira'  
WHERE id_autor = 3;
```

```
UPDATE autor  
SET nome = 'Stephen Edwin'  
WHERE id_autor = 4;
```

```
UPDATE autor  
SET nome = 'Clarice A.'  
WHERE id_autor = 5;
```

```
UPDATE autor  
SET nacionalidade = 'Americana'  
WHERE id_autor = 6;
```

-- Atualizar o e-mail de um usuário:

```
UPDATE email  
SET email = 'ana.clara.novo@email.com'  
WHERE id_email = 1;
```

```
UPDATE email  
SET email = 'joao.costa@email.com'  
WHERE id_email = 2;
```

```
UPDATE email
```



```
SET email = 'julia.oliveira@email.com'  
WHERE id_email = 3;
```

```
UPDATE email  
SET email = 'rodrigo.senai@email.com'  
WHERE id_email = 4;
```

```
UPDATE email  
SET email = 'msria.rosa@email.com'  
WHERE id_email = 5;
```

-- Atualizar a data de devolução do empréstimo:

```
UPDATE emprestimo  
SET data_devolucao = CURRENT_DATE + INTERVAL '7 days'  
WHERE id_emprestimo = 1;
```

```
UPDATE emprestimo  
SET data_devolucao = CURRENT_DATE + INTERVAL '7 days'  
WHERE id_emprestimo = 2;
```

```
UPDATE emprestimo  
SET data_devolucao = CURRENT_DATE + INTERVAL '7 days'  
WHERE id_emprestimo = 5;
```

```
SAVEPOINT updates;
```

-- Excluir um gênero:

```
DELETE FROM genero  
WHERE id_genero = 2;
```

```
DELETE FROM genero  
WHERE genero = 'Terror';
```

-- Excluir um empréstimo:

```
DELETE FROM emprestimo  
WHERE id_usuario = 3;
```

```
-- Excluir um livro:
```

```
-- Primeiro remover as relações
```

```
DELETE FROM livro_autor WHERE id_livro = 1;
```

```
DELETE FROM livro_autor WHERE id_livro = 2;
```

```
DELETE FROM livro_genero WHERE id_livro = 1;
```

```
DELETE FROM livro_genero WHERE id_livro = 2;
```

```
DELETE FROM emprestimo WHERE id_emprestimo = 1;
```

```
DELETE FROM emprestimo WHERE id_emprestimo = 2;
```

```
-- Depois remover o livro
```

```
DELETE FROM livro WHERE id_livro = 1;
```

```
DELETE FROM livro WHERE id_livro = 2;
```

```
ROLLBACK TO original;
```

```
COMMIT;
```

```
ORDER BY data_evento ASC, ordem ASC;
```

5.2 DATA QUERY LANGUAGE (DQL)

O Data Query Language (DQL) permite a visualização das tabelas. Abaixo é possível analisar seu *script* e alguns exemplos de sua visualização.

```
-- Consultas simples
```

```
SELECT * FROM cargo
```

```
ORDER BY id_cargo DESC;
```

```
SELECT * FROM pessoa
```

```
ORDER BY id_pessoa ASC;
```

```
SELECT * FROM pessoa_funcionario  
ORDER BY id_pessoa ASC;
```

```
SELECT * FROM pessoa_usuario  
ORDER BY id_pessoa ASC;
```

```
SELECT * FROM emprestimo  
ORDER BY id_emprestimo DESC;
```

```
SELECT * FROM historico_emprestimo  
ORDER BY data_evento DESC, ordem DESC;
```

```
SELECT * FROM livro_emprestado  
ORDER BY id_livro_emprestado DESC;
```

```
SELECT * FROM historico_livro_emprestado  
ORDER BY data_evento DESC, ordem DESC;
```

```
SELECT * FROM autor  
ORDER BY id_autor ASC;
```

```
SELECT * FROM livro  
ORDER BY id_livro ASC;
```

```
SELECT * FROM genero  
ORDER BY id_genero ASC;
```

```
SELECT * FROM livro_autor  
ORDER BY id_livro ASC;
```

```
SELECT * FROM livro_genero  
ORDER BY id_livro ASC;
```

```
SELECT * FROM endereco
```

```
ORDER BY cep ASC;
```

```
SELECT * FROM telefone  
ORDER BY ddd ASC;
```

```
SELECT * FROM email  
ORDER BY id_email ASC;
```

```
SELECT nome, sobrenome, cpf  
FROM pessoa  
WHERE nome LIKE 'J%';
```

```
SELECT titulo, ano  
FROM livro  
LIMIT 3;
```

```
-- Views
```

```
-- View para listar os livros emprestados e em qual empréstimos estão
```

```
CREATE VIEW vw_livros_emprestimos AS
```

```
SELECT
```

```
    e.id_emprestimo,
```

```
    l.titulo
```

```
FROM emprestimo e
```

```
JOIN livro_emprestado le ON e.id_emprestimo = le.id_emprestimo
```

```
JOIN livro l ON le.id_livro = l.id_livro;
```

```
SELECT * FROM vw_livros_emprestimos;
```

```
-- View para listar os titulos dos livros emprestados conforme o usuário e seu  
status atual
```

```
CREATE VIEW vw_emprestimos_livros_usuario AS
```

```
SELECT
```

```
    e.id_emprestimo,
```

```
e.id_usuario,  
l.titulo,  
h.descricao AS status_livro  
FROM emprestimo e  
JOIN livro_emprestado le ON e.id_emprestimo = le.id_emprestimo  
JOIN livro l ON le.id_livro = l.id_livro  
JOIN historico_livro_emprestado h ON le.id_livro_emprestado =  
h.id_livro_emprestado  
WHERE h.status_historico = TRUE  
GROUP BY e.id_emprestimo, e.id_usuario, l.titulo, h.descricao;  
  
SELECT * FROM vw_emprestimos_livros_usuario  
WHERE id_usuario = 1;
```

6 CONTROLE DE ACESSO

O controle de acesso de um banco de dados é realizado através do Data Control Language (DCL), que, assim como descrito em seu nome, controla quem acessa os dados. Ele gerencia quem acessa e o que esse usuário pode fazer com os dados acessados a partir dos comandos grant, revoke e deny.

6.1 DATA CONTROL LANGUAGE (DCL)

O *script* do DCL pode ser visto abaixo junto com imagens que comprovam seu funcionamento dentro do banco de dados Biblioteca.

-- Criação de usuários

-- Perfil do gerente

```
CREATE USER pessoa_admin WITH PASSWORD 'admin123';
```

-- Perfil de funcionários no geral

```
CREATE USER pessoa_funcionario WITH PASSWORD 'funcionario123';
```

-- Perfil de usuário comum

```
CREATE USER pessoa_usuario WITH PASSWORD 'usuario123';
```

-- DCL

-- Concede todos os privilégios ao gerente

```
GRANT ALL PRIVILEGES ON SCHEMA public TO pessoa_admin;
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO  
pessoa_admin;
```

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO  
pessoa_admin;
```

-- Concede todos os privilégios aos funcionários comuns

```
GRANT ALL PRIVILEGES ON SCHEMA public TO pessoa_funcionario;
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO
pessoa_funcionario;
```

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO
pessoa_funcionario;
```

```
-- Concede apenas permissão de leitura as pessoas comuns
```

```
GRANT USAGE ON SCHEMA public TO pessoa_usuario;
```

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO pessoa_usuario;
```

```
-- Garante que permissões sejam propagadas para futuras tabelas
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public
```

```
GRANT SELECT ON TABLES TO pessoa_usuario;
```

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public
```

```
GRANT SELECT ON TABLES TO pessoa_funcionario;
```

```
-- Revoga a permissão de inserção na tabela cargos para pessoa_funcionario
```

```
REVOKE INSERT ON public.cargo FROM pessoa_funcionario;
```

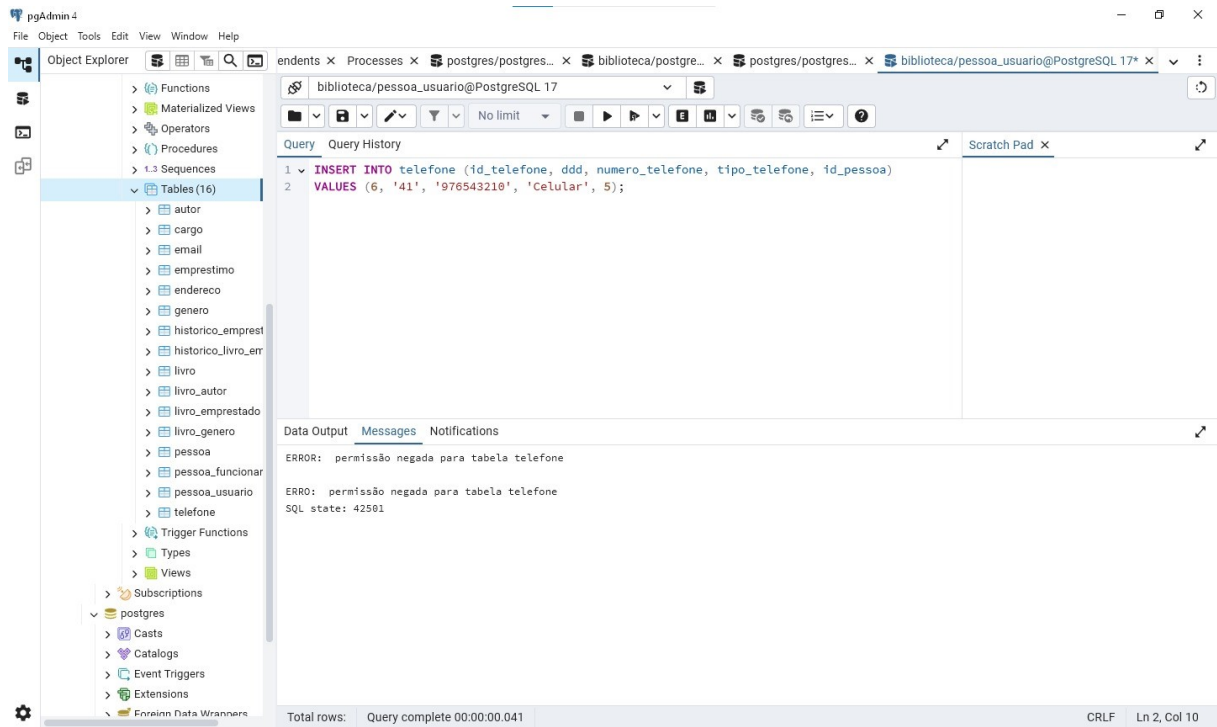
```
-- Revoga a permissão de atualização na tabela cargos para
pessoa_funcionario
```

```
REVOKE UPDATE ON public.cargo FROM pessoa_funcionario;
```

```
-- Revoga a permissão de exclusão na tabela cargos e pessoa para
pessoa_funcionario
```

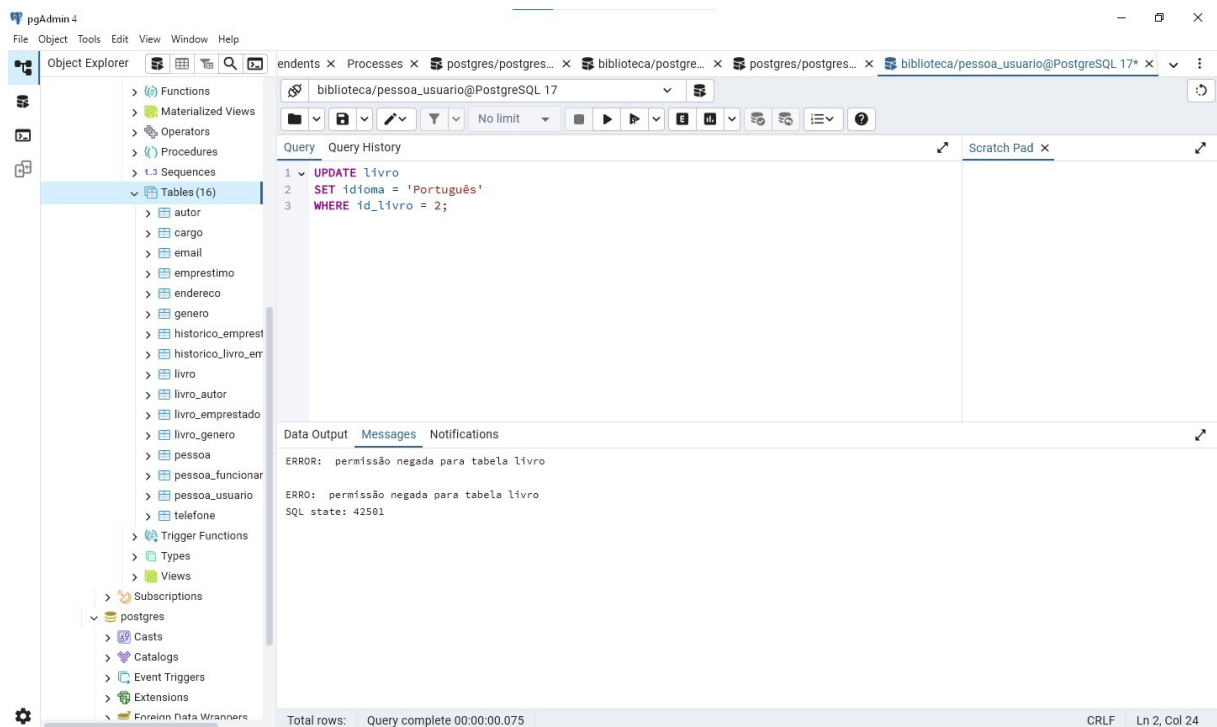
```
REVOKE DELETE ON public.cargo, public.pessoa FROM pessoa_funcionario;
```

Figura 2 - Usuário comum não tem permissão para realizar inserções



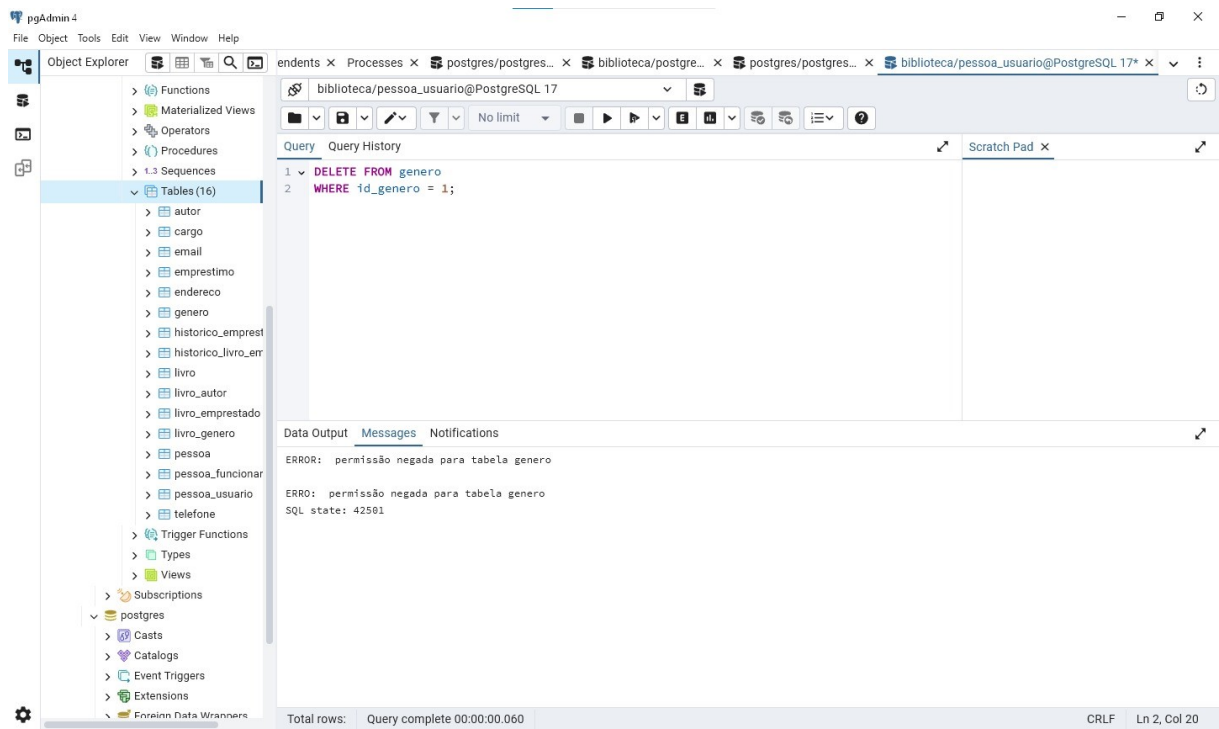
Fonte: própria (2025).

Figura 3 - Usuário comum não tem permissão para realizar atualizações



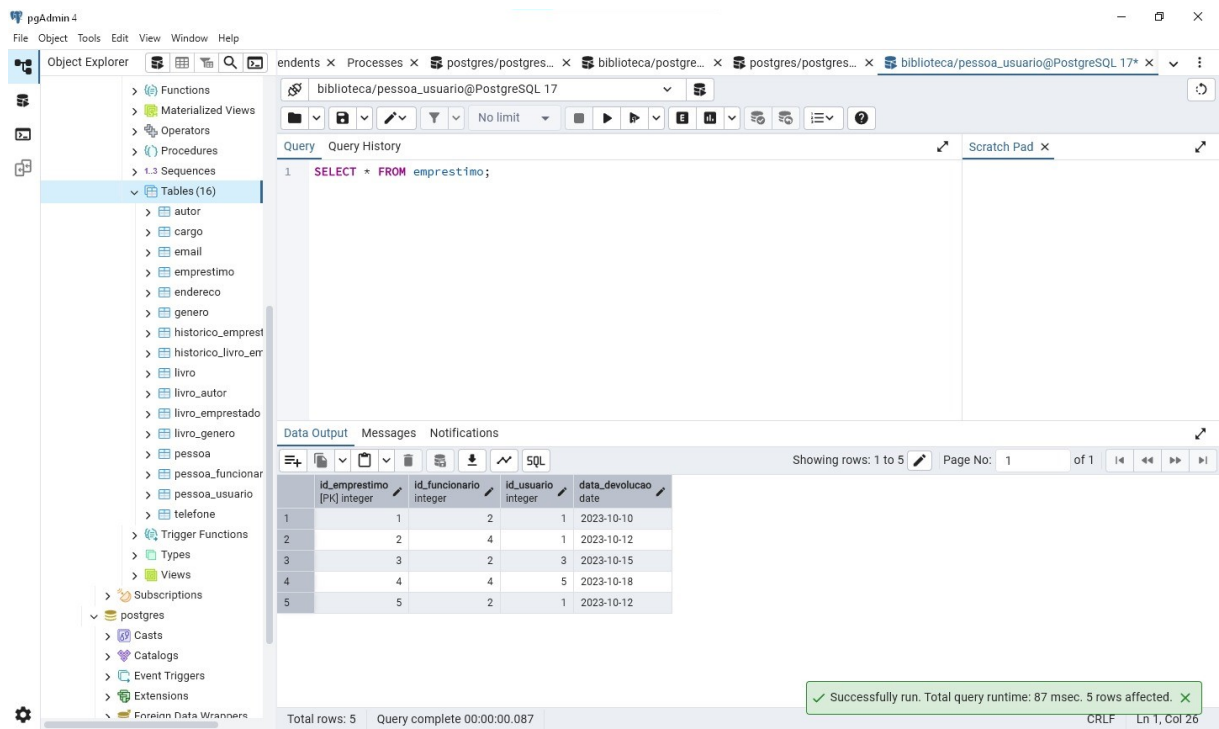
Fonte: própria (2025).

Figura 4 - Usuário comum não tem permissão para realizar exclusão



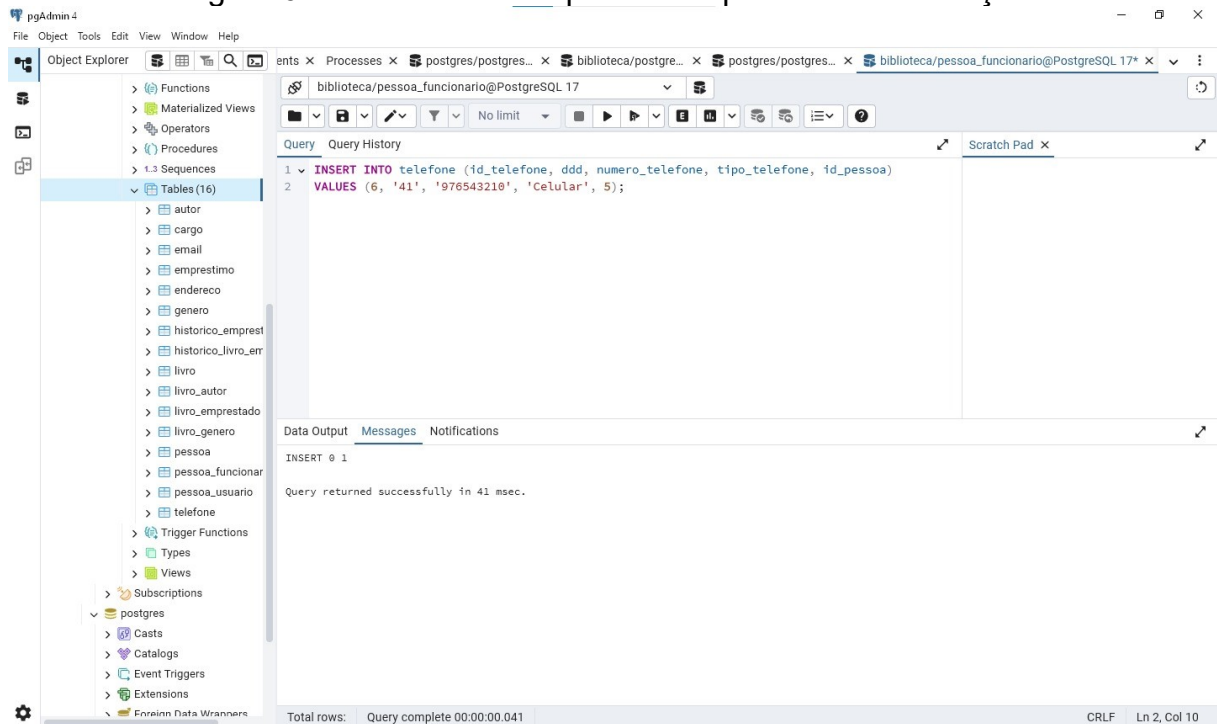
Fonte: própria (2025).

Figura 5 - Usuário comum tem permissão para realizar consultas



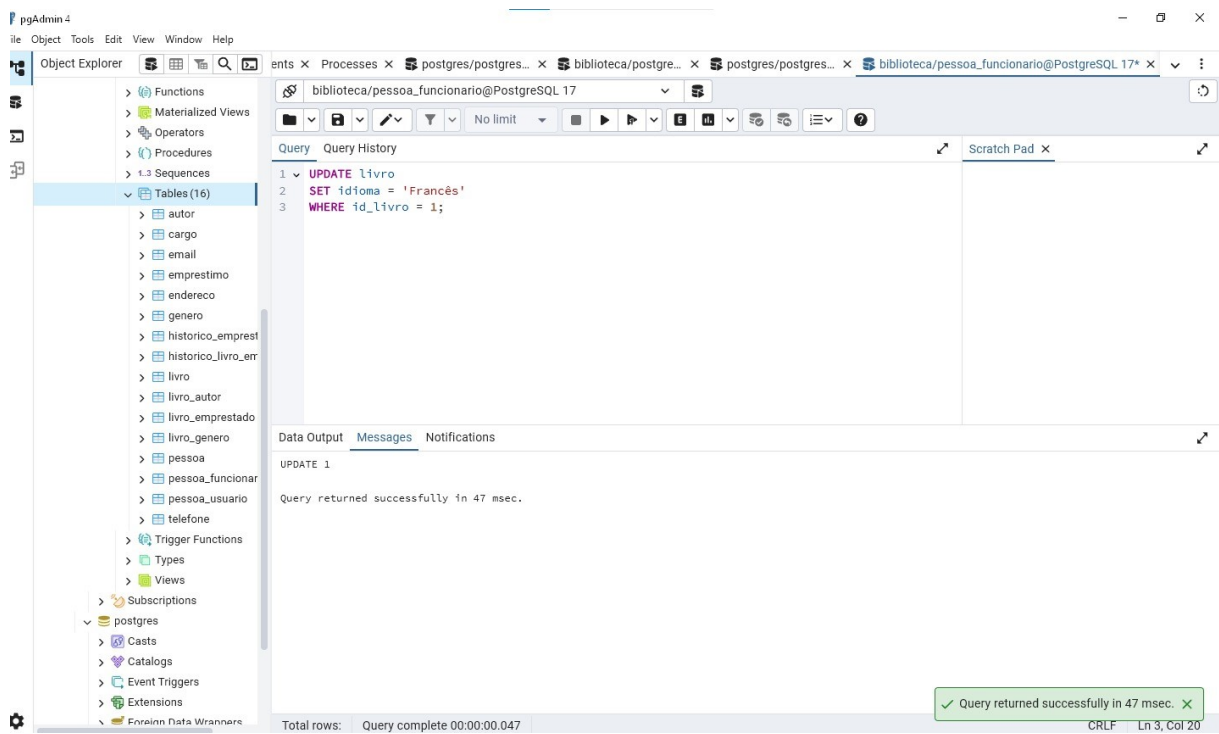
Fonte: própria (2025).

Figura 6 - Funcionário tem permissão para realizar inserções



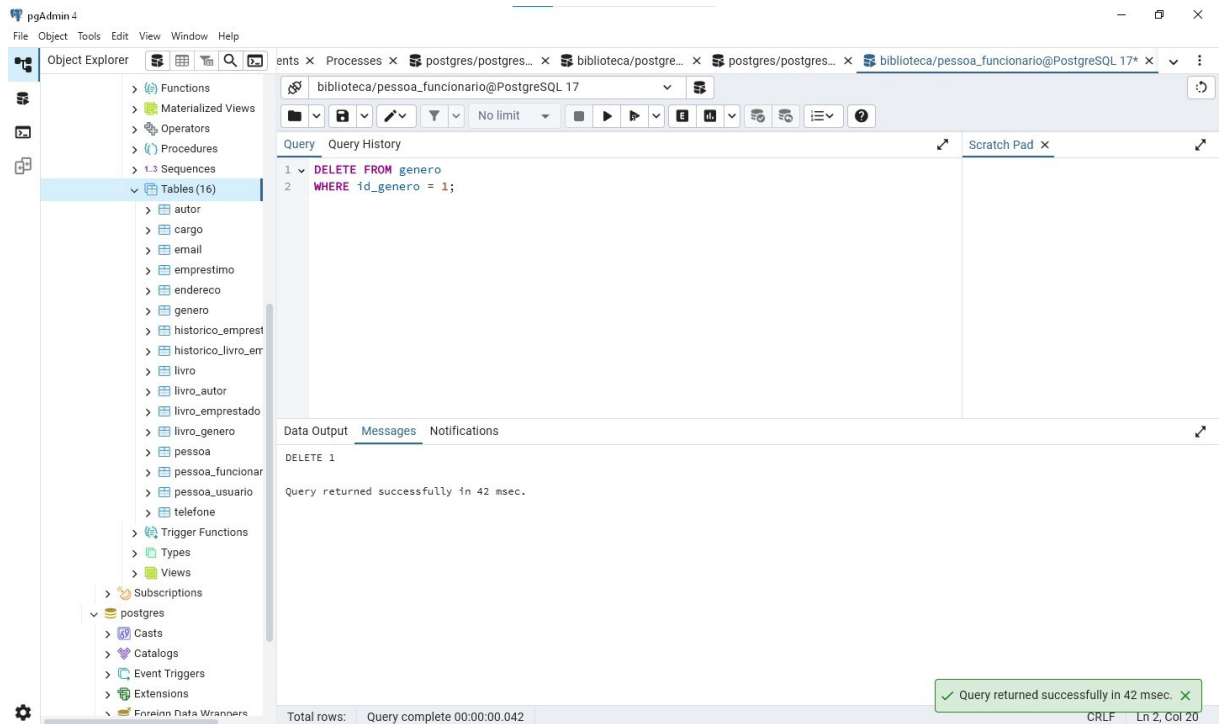
Fonte: própria (2025).

Figura 7 - Funcionário tem permissão para realizar atualizações



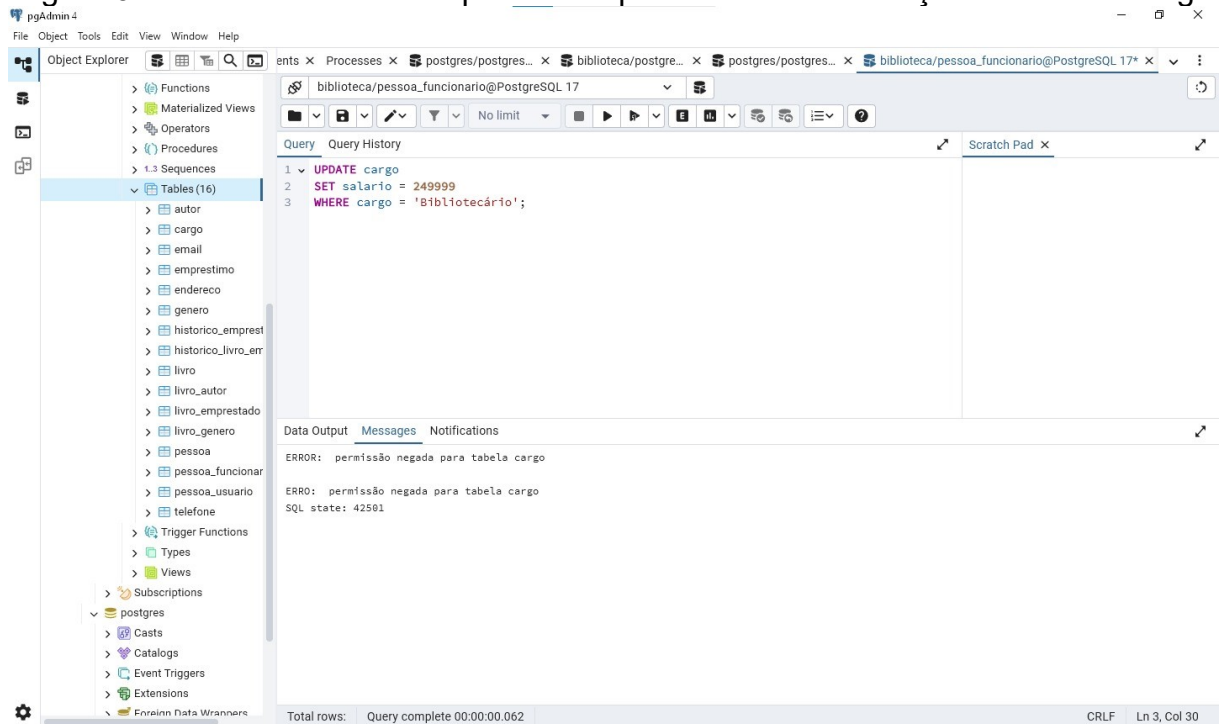
Fonte: própria (2025).

Figura 8 - Funcionário tem permissão para realizar exclusão



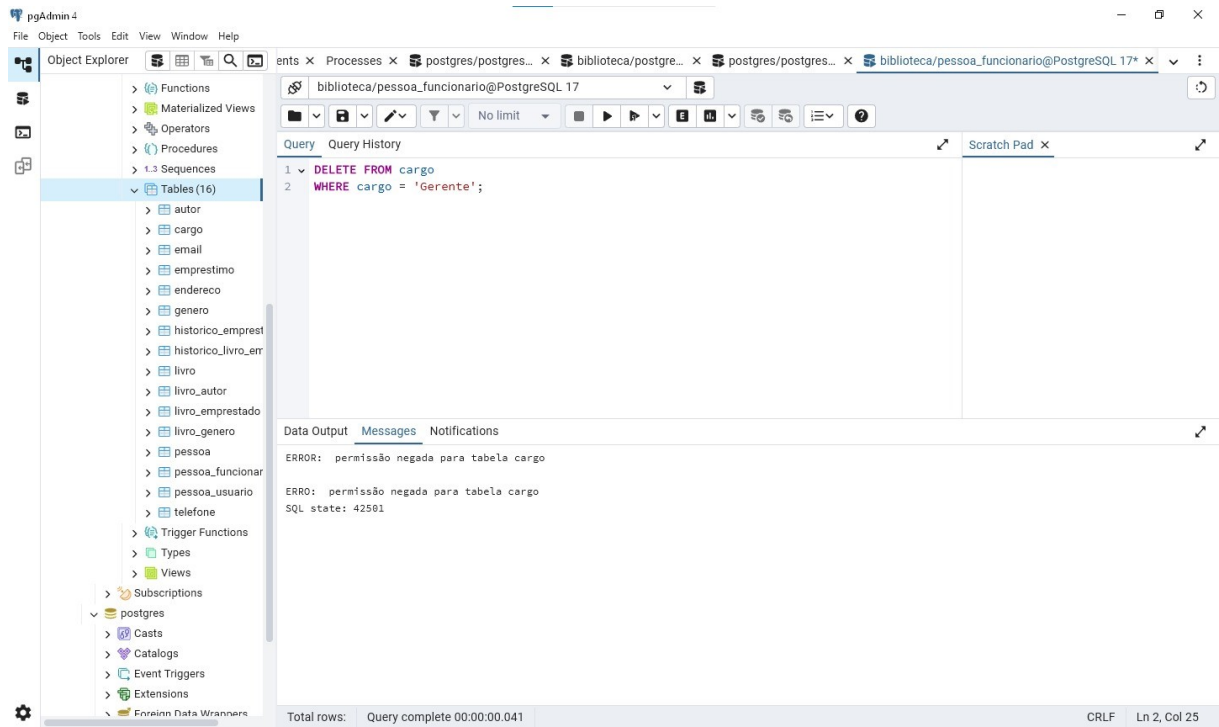
Fonte: própria (2025).

Figura 9 - Funcionário não tem permissão para realizar atualizações na tabela cargo



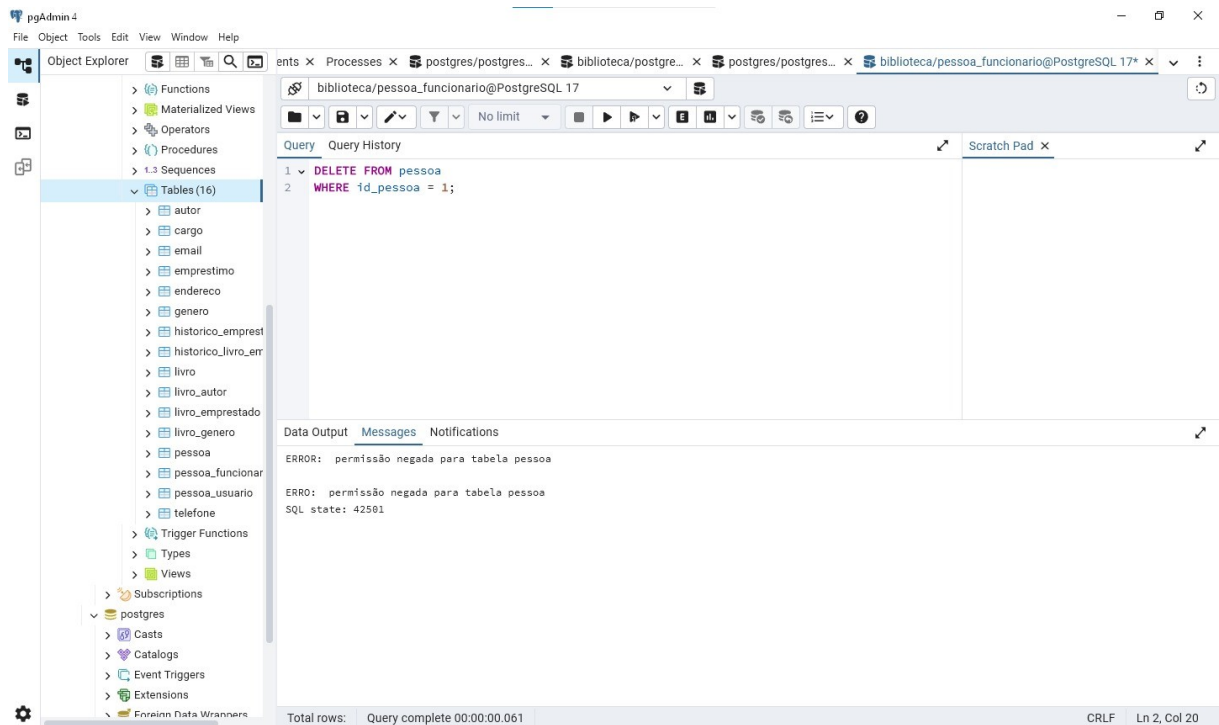
Fonte: própria (2025).

Figura 10 - não tem permissão para realizar exclusões na tabela cargo



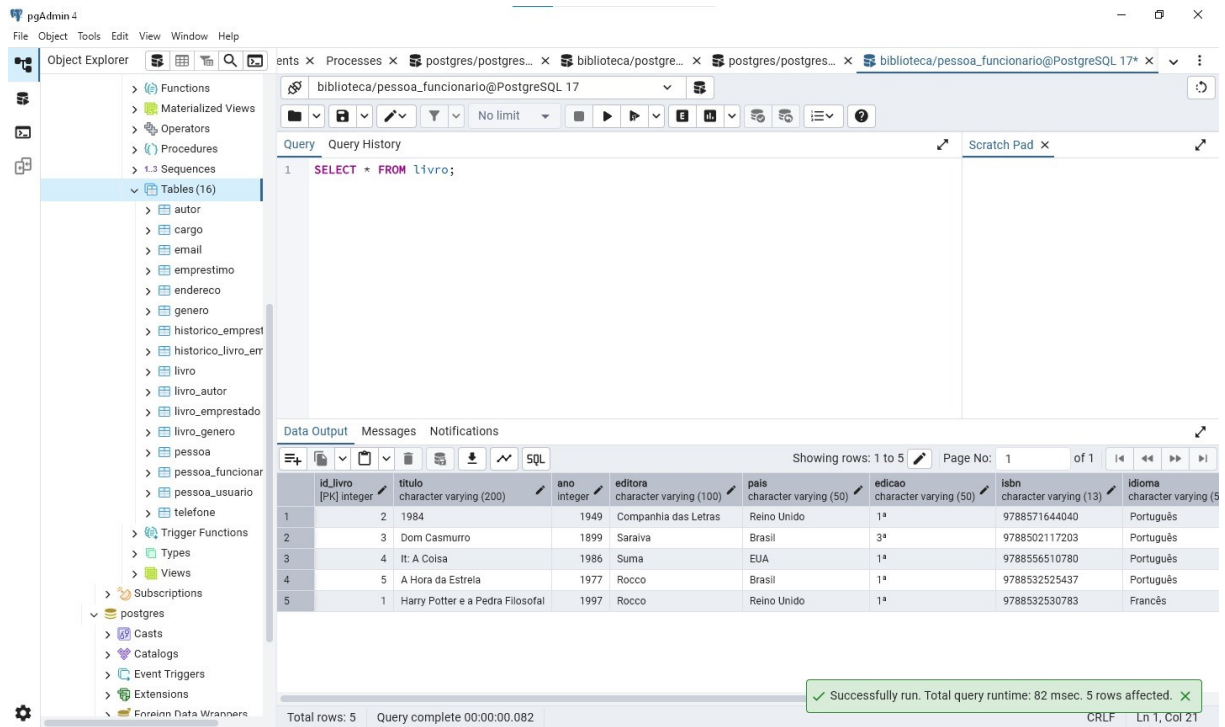
Fonte: própria (2025).

Figura 11 – Funcionário não tem permissão para realizar exclusões na tabela pessoa



Fonte: própria (2025).

Figura 12 - Funcionário tem permissão para realizar consultas

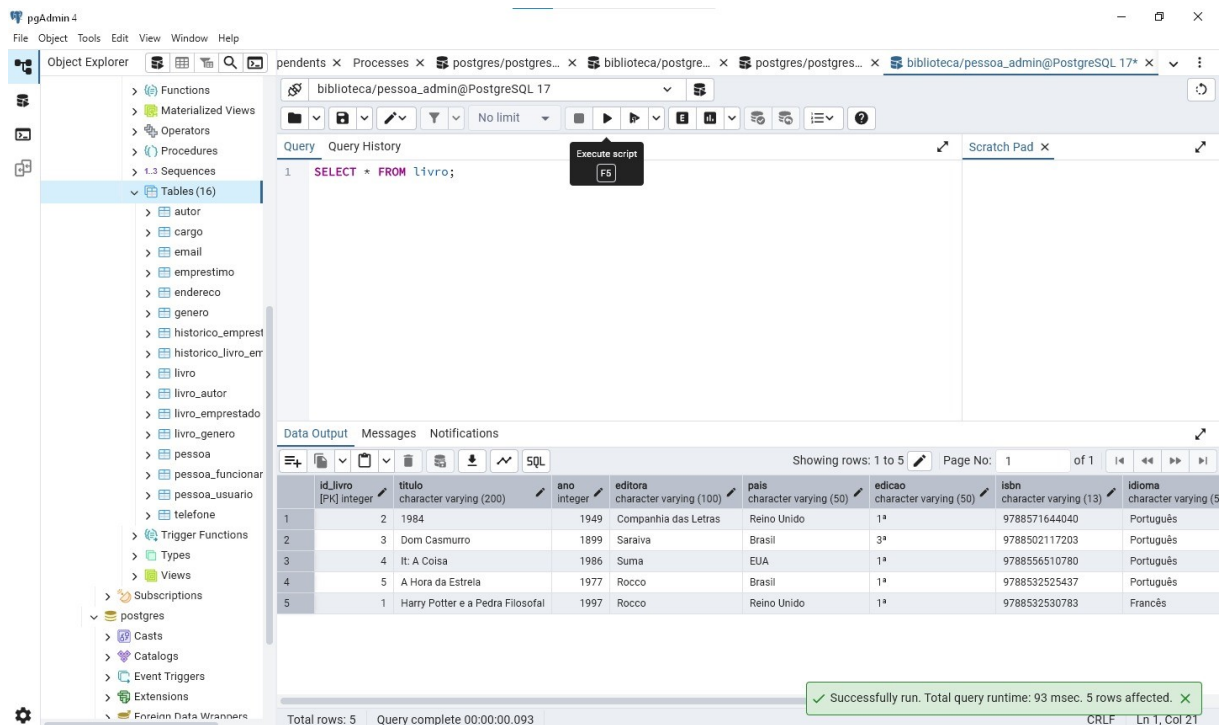


The screenshot shows the pgAdmin 4 interface with the 'biblioteca/pessoa_funcionario@PostgreSQL 17*' connection selected. The 'Tables (16)' folder is expanded in the left sidebar. The query editor contains the SQL statement: `SELECT * FROM livro;`. The 'Data Output' pane displays the results of the query, showing 5 rows of book information. The status bar at the bottom indicates: 'Total rows: 5 Query complete 00:00:00.082' and 'Successfully run. Total query runtime: 82 msec. 5 rows affected.'

	id_livro [PK] integer	titulo character varying (200)	ano integer	editora character varying (100)	pais character varying (50)	edicao character varying (50)	isbn character varying (13)	idioma character varying (5)
1	2	1984	1949	Companhia das Letras	Reino Unido	1ª	9788571644040	Português
2	3	Dom Casmurro	1899	Saraiva	Brasil	3ª	9788502117203	Português
3	4	It: A Coisa	1986	Suma	EUA	1ª	9788556510780	Português
4	5	A Hora da Estrela	1977	Rocco	Brasil	1ª	9788532525437	Português
5	1	Harry Potter e a Pedra Filosofal	1997	Rocco	Reino Unido	1ª	9788532530783	Francês

Fonte: própria (2025).

Figura 13 - Gerente tem permissão para realizar consultas

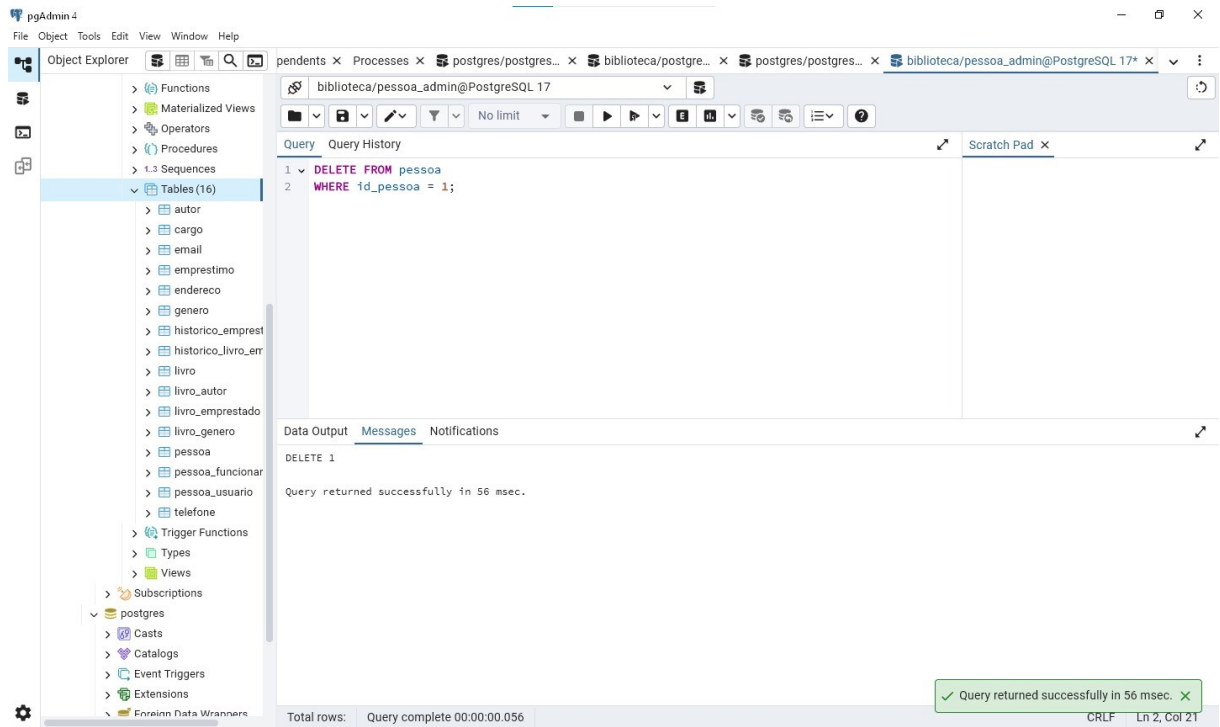


The screenshot shows the pgAdmin 4 interface with the 'biblioteca/pessoa_admin@PostgreSQL 17*' connection selected. The 'Tables (16)' folder is expanded in the left sidebar. The query editor contains the SQL statement: `SELECT * FROM livro;`. The 'Data Output' pane displays the results of the query, showing 5 rows of book information. The status bar at the bottom indicates: 'Total rows: 5 Query complete 00:00:00.093' and 'Successfully run. Total query runtime: 93 msec. 5 rows affected.'

	id_livro [PK] integer	titulo character varying (200)	ano integer	editora character varying (100)	pais character varying (50)	edicao character varying (50)	isbn character varying (13)	idioma character varying (5)
1	2	1984	1949	Companhia das Letras	Reino Unido	1ª	9788571644040	Português
2	3	Dom Casmurro	1899	Saraiva	Brasil	3ª	9788502117203	Português
3	4	It: A Coisa	1986	Suma	EUA	1ª	9788556510780	Português
4	5	A Hora da Estrela	1977	Rocco	Brasil	1ª	9788532525437	Português
5	1	Harry Potter e a Pedra Filosofal	1997	Rocco	Reino Unido	1ª	9788532530783	Francês

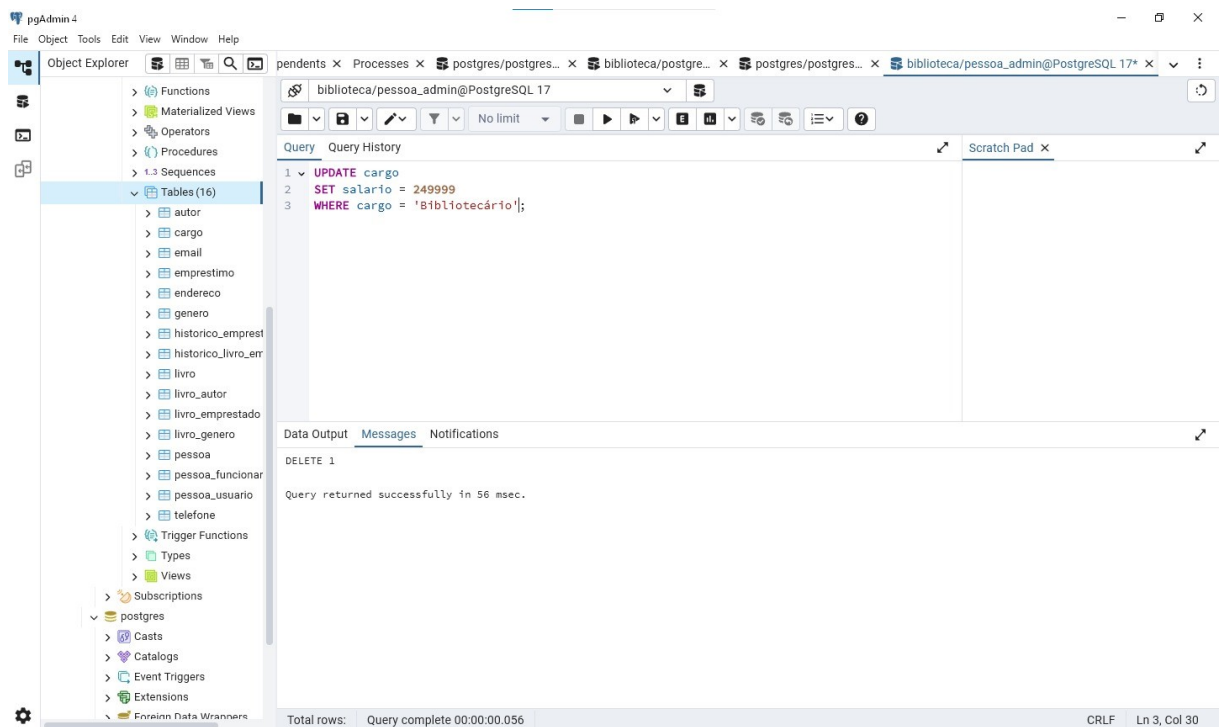
Fonte: própria (2025).

Figura 14 - Gerente tem permissão para realizar exclusões em todas as tabelas



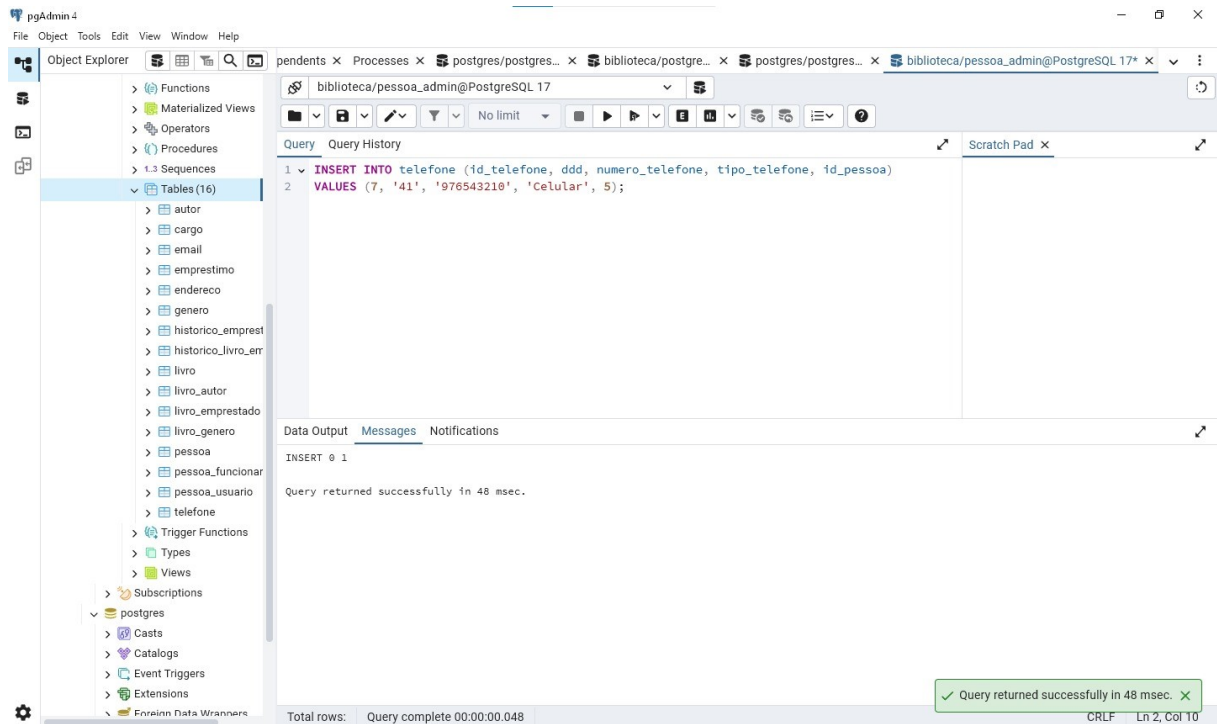
Fonte: própria (2025).

Figura 15 - Gerente tem permissão para realizar atualizações em todas as tabelas



Fonte: própria (2025).

Figura 16 - Gerente tem permissão para realizar inclusões em todas as tabelas



Fonte: própria (2025).

7 CONTROLE DE TRANSAÇÕES

O controle das transações é o controle da validade dos comandos realizados no banco de dados. Ele é feito para que os comandos possam ser testados de uma maneira mais segura, já que não há como reverter um comando em um banco de dados, possibilitando a realização dos testes e a sua aplicação propriamente dita depois que os testes forem aprovados.

7.1 DATA TRANSACTION LANGUAGE (DTL)

É possível conferir o DTL aplicado no *script* do DML no tópico 5: Manipulação de dados.

8 CONCLUSÃO

O desenvolvimento desse projeto foi feito ao longo das semanas disponibilizadas, iniciado com a modelagem, seguido dos *scripts* e testes e finalizado com a documentação. Trabalhar nesse projeto foi esclarecedor para a equipe em todos os tópicos aplicados, uma vez que foram colocados em prática sem auxílio do professor, o que também resultou em certo aprofundamento nas tecnologias utilizadas. Se o projeto fosse trabalhado por mais tempo, algumas melhorias a serem feitas seriam: a expansão do banco para uma biblioteca com diversas unidades e mais cargos, além de mais opções de restrições.

REFERÊNCIAS

GITHUB. *Material/Banco de Dados/*. profAndreSouza, c2025. Disponível em: <
<https://github.com/profAndreSouza/Material/tree/main/Banco%20de%20Dados>>

Acesso em: 10 jun. 2025.

MERMAID. *Entity Relationship Diagrams*. Mermaid, c2025. Disponível em:
<<https://mermaid.js.org/syntax/entityRelationshipDiagram.html>.> Acesso em: 10 jun.

2025.

POSTGRESQL. *Documentation*. Postgres, c2025. Disponível em: <
<https://www.postgresql.org/docs/>.> Acesso em: 10 jun. 2025.