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](#_Toc201264488)

[2 Modelagem conceitual 4](#_Toc201264489)

[2.1 Entidades, atributos e relacionamentos 4](#_Toc201264490)

[2.2 Diagrama entidade-relacionamento 5](#_Toc201264491)

[3 Modelagem lógica 7](#_Toc201264492)

[3.1 Normalização 7](#_Toc201264493)

[4 Estrutura do banco de dados 8](#_Toc201264494)

[*4.1* *Data Definition Language (DDL)* 8](#_Toc201264495)

[4.2 Tabelas, atributos e relacionamentos 13](#_Toc201264496)

[4.2.1 Tabela principal de livros 13](#_Toc201264497)

[4.2.2 Tabela de autores 13](#_Toc201264498)

[4.2.3 Tabela associativa entre livros e autores 13](#_Toc201264499)

[4.2.4 Tabela de gêneros 13](#_Toc201264500)

[4.2.5 Tabela associativa entre livros e gêneros 13](#_Toc201264501)

[4.2.6 Tabela de cargos 14](#_Toc201264502)

[4.2.7 Tabela Pessoa 14](#_Toc201264503)

[4.2.8 Tabela especialização de Pessoa - Funcionário 14](#_Toc201264504)

[4.2.9 Tabela especialização de Pessoa - Usuário 14](#_Toc201264505)

[4.2.10 Tabela de endereço 14](#_Toc201264506)

[4.2.11 Tabela de telefone 14](#_Toc201264507)

[4.2.12 Tabela de email 14](#_Toc201264508)

[4.2.13 Tabela de empréstimos 15](#_Toc201264509)

[4.2.14 Tabela de histórico de *status* do empréstimo 15](#_Toc201264510)

[4.2.15 Tabela de livros emprestados 15](#_Toc201264511)

[4.2.16 Tabela de histórico de *status* dos livros emprestados 15](#_Toc201264512)

[5 Manipulação de dados 16](#_Toc201264513)

[*5.1* *Data Manipulation Language (DML)* 16](#_Toc201264514)

[*5.2* *Data Query Language (DQL)* 25](#_Toc201264515)

[6 Controle de acesso 29](#_Toc201264516)

[*6.1* *Data Control Language (DCL)* 29](#_Toc201264517)

[7 Controle de transações 39](#_Toc201264518)

[*7.1* *Data Transaction Language (DTL)* 39](#_Toc201264519)

[8 CONCLUSÃO 40](#_Toc201264520)

[REFERÊNCIAS 41](#_Toc201264521)

# 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.

# 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.

## 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".

## 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.eyJkb2N1bWVudElEIjoiNDg2MTFkNzEtZmY2ZS00OTE3LTkxNmEtZWE5NWQ0ZjFhM2FiIiwiYWNjZXNzIjoiRWRpdCIsImlhdCI6MTc1MDM1OTEzOH0.utjg4IsOKwnblg7HjZEI0KRKLejXavsLbUYU-_BElyE>> Acesso em: 10 jun. 2025.

Figura 1 - Diagrama Entidade-Relacionamento

*Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.*

Fonte: própria (2025).

# 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.

## 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\_usuário e pessoa\_funcionário. A especialização mantém os atributos comuns na tabela pessoa, enquanto as suas especializações possuem apenas os atributos que se diferem.

# Estrutura do banco de dados

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

## *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

);

## Tabelas, atributos e relacionamentos

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

### 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.

### Tabela de autores

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

### 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.

### 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.

### 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.

### 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.

### 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.

### Tabela especialização de Pessoa - Funcionário

Pessoa\_Funcionário possui as informações adicionais de um funcionário da biblioteca como cargo e data de admissão.

### Tabela especialização de Pessoa - Usuário

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

### 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.

### 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.

### Tabela de email

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

### 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.

### 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.

### 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.

### 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.

# 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.

## *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;

## *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;

# 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.

## *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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 3 - Usuário comum não tem permissão para realizar atualizações

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 4 - Usuário comum não tem permissão para realizar exclusão

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 5 - Usuário comum tem permissão para realizar consultas

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 6 - Funcionário tem permissão para realizar inserções Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 7 - Funcionário tem permissão para realizar atualizações

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 8 - Funcionário tem permissão para realizar exclusão

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 9 - Funcionário não tem permissão para realizar atualizações na tabela cargoInterface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 12 - Funcionário tem permissão para realizar consultas

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

Figura 13 - Gerente tem permissão para realizar consultas

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

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

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Fonte: própria (2025).

# 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.

## *Data Transaction Language (DTL)*

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

# 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.