

**SENAI “GASPAR RICARDO JÚNIOR”**

**Ian Lucas**

**Juan Correa**

**Caroline Almeida**

**Matheus Pulcinelli**

**Banco de Dados Relacionais**

**Sorocaba**

**2025**

**Ian Lucas**

**Juan Correa**

**Caroline Almeida**

**Matheus Pulcinelli**

**Banco de Dados Relacionais**

Trabalho apresentado para a disciplina de Banco de Dados, do curso de Análise e Desenvolvimento de Sistemas da Escola SENAI ‘Gaspar Ricardo Júnior’.

Orientador: André Souza.

**Sorocaba**

**2025**

**SUMÁRIO**

1 INTRODUÇÃO....................................................................................................4

Modelagem Conceitual......................................................................................04

2.1 Descrição das Entidades, Atributos e Relacionamento.....................................04

2.2 Diagrama Entidade-Relacionamento (DER)......................................................08

3 Modelagem Lógica..............................................................................................08

3.1 Transformações do DER para a Estrutura do Banco de Dados.........................08

3.2 Discussão sobre a Normalização........................................................................11

4 Estrutura do Banco de Dados.............................................................................13

4.1 Definição das Tabelas – DDL (Data Definition Language) .................................13

4.2 Descrição das Tabelas e Atributos.......................................................................13

4.3 Relacionamento entre Tabelas.............................................................................17

5 Manipulação de Dados..........................................................................................17

5.1 Inserção, Atualização e Deleção de Dados (DML)...............................................18

5.2 Exemplos de Consultas SQL (DQL)............................................................... 21-22

6 Controle de Acesso (Bonificação – DCL)............................................................24

6.1 1 Introdução ao Controle de Acesso.....................................................................24

6.2 Criação de Usuários (exemplo)............................................................................25

6.3 Concessão de Permissões (GRANT)...................................................................25

6.4 Revogação de Permissões (REVOKE)................................................................26

7 CONCLUSÃO.........................................................................................................27

8 REFERÊNCIAS.......................................................................................................28

# **INTRODUÇÃO**

O presente projeto tem como tema o desenvolvimento de um Sistema de Gerenciamento de Biblioteca, elaborado como atividade final da disciplina de Banco de Dados Relacional. A proposta consiste em criar um sistema que permita o controle e a organização dos principais processos de uma biblioteca, como o cadastro de clientes, gestão de livros, controle de empréstimos, contratos, estoque e pedidos. Esse cenário foi escolhido por ser representativo e aplicável ao uso real de bancos de dados relacionais, promovendo uma abordagem prática dos conteúdos abordados em sala de aula.

# **2. Modelagem Conceitual**

**2.1** **Descrição das Entidades, Atributos e Relacionamentos**

A modelagem conceitual do sistema de gerenciamento de biblioteca foi elaborada com base na identificação dos principais elementos e processos envolvidos na operação de uma biblioteca. A seguir, são descritas as entidades, seus respectivos atributos e os relacionamentos estabelecidos entre elas.

# **Cliente**

• Representa os usuários cadastrados na biblioteca.

**Atributos principais:** ID do cliente, nome completo, CPF, data de nascimento, telefone, e-mail, endereço.

**Relacionamentos:**

• Pode realizar empréstimos

• Pode firmar contratos

• Pode fazer pedidos

# **Produto**

• Entidade genérica que representa qualquer item disponível na biblioteca.

**Atributos principais:** ID do produto, título, descrição, valor, tipo.

**Relacionamentos:**

• Pode estar associado a um ou mais registros de estoque

• Pode ser um livro

# **Livro (especialização de Produto)**

• Entidade especializada, representa os livros da biblioteca.

**Atributos principais**: ID do livro, autor, editora, ano de publicação, número de páginas.

**Relacionamentos:**

• Está associado a um gênero

• É um tipo de produto

# **Gênero**

• Classifica os livros por tema ou categoria.

**Atributos principais:** ID do gênero, nome, descrição.

**Relacionamentos:**

• Pode estar associado a vários livros

# **Estoque**

• Controla a quantidade e localização dos produtos disponíveis.

**Atributos principais:** ID do estoque, ID do produto, quantidade disponível, localização.

**Relacionamentos:**

• Está relacionado com produtos

# **Pedido**

• Histórico de solicitações de produtos feitas pelos clientes.

**Atributos principais:** ID do pedido, ID do cliente, data, status.

**Relacionamentos:**

• Relacionado a clientes

• Pode incluir um ou mais produtos

# **Empréstimo**

• Registro de itens emprestados pela biblioteca.

**Atributos principais**: ID do empréstimo, ID do cliente, ID do produto, data de retirada, data de devolução prevista, data de devolução real, status.

**Relacionamentos:**

Vinculado a clientes e produtos

# **Contrato**

• Formaliza um acordo entre o cliente e a biblioteca.

**Atributos principais:** ID do contrato, ID do cliente, tipo de contrato, data de início, data de término, termos.

**Relacionamentos:**

• Relacionado a clientes

# **Funcionário**

• Representa os colaboradores que trabalham na biblioteca**.**

**Atributos principais:** ID do funcionário, nome, CPF, cargo, telefone, e-mail.

**Relacionamentos:**

• Pode ser responsável por registros de empréstimos e pedidos (opcional)

# **2.2** **Diagrama Entidade-Relacionamento (DER)**

O Diagrama Entidade-Relacionamento (DER) representa graficamente a estrutura do banco de dados, destacando as entidades, atributos e os relacionamentos entre elas.

<https://miro.com/app/board/uXjVImkHCvA=/>

O DER evidencia as ligações entre as entidades e permite uma visualização clara da modelagem adotada, essencial para garantir a consistência e a integridade do banco de dados.

# **3. Modelagem Lógica**

**3.1** **Transformações do DER para a Estrutura do Banco de Dados**

A modelagem lógica representa a transição entre o modelo conceitual e a implementação física do banco de dados. Com base no Diagrama Entidade-Relacionamento (DER) apresentado anteriormente, foram realizadas as transformações necessárias para a definição das tabelas, chaves primárias, chaves estrangeiras e atributos correspondentes.

As entidades identificadas na modelagem conceitual foram convertidas em tabelas relacionais conforme descrito a seguir:

**Cliente**

A entidade Cliente originou a tabela cliente, contendo os atributos essenciais para identificação e contato dos usuários.

**Chave primária: id\_cliente**

**Atributos:** nome\_cliente, cpf\_cliente, email\_cliente, telefone\_cliente, endereco\_cliente, data\_cadastro

**Produto**

A entidade Produto corresponde à tabela produto, que representa os itens disponíveis na biblioteca.

**Chave primária: id\_produto**

**Atributos:** nome\_produto, descrição, preço, categoria

**Livro**

A tabela livro especializa a entidade Produto e mantém uma relação direta com a tabela gênero.

• Chave primária: id\_livro

• Chaves estrangeiras: id\_produto, id\_genero

• Atributos: nome\_livro, descrição

**Gênero**

A entidade Gênero tornou-se a tabela gênero, utilizada para classificar os livros.

Chave primária: id\_genero

**Atributo: descrição**

**Estoque**

A tabela estoque representa o controle da quantidade e localização dos produtos.

• Chave primária: id\_estoque

• Chave estrangeira: id\_produto

• Atributos: quantidade, localização

**Pedido**

A entidade Pedido deu origem à tabela pedido, associada aos clientes e funcionários responsáveis.

• Chave primária: id\_pedido

• Chaves estrangeiras: id\_cliente, id\_funcionario

• Atributos: data\_pedido, total

**Empréstimo**

A tabela empréstimo representa os registros de empréstimos realizados.

• Chave primária: id\_emprestimo

• Chaves estrangeiras: id\_cliente, id\_produto

• Atributos: data\_emprestimo, data\_devolucao

**Contrato**

A tabela contrato documenta os acordos firmados entre cliente e biblioteca.

• Chave primária: id\_contrato

• Chave estrangeira: id\_cliente

• Atributos: data\_inicio, data\_fim, valor

**Funcionário**

A entidade Funcionário foi convertida na tabela funcionário, que armazena os dados dos colaboradores da biblioteca.

**Chave primária: id\_funcionario**

**Atributos:** nome\_funcionario, cpf\_funcionario, cargo, telefone\_funcionario

**3.2** **Discussão sobre a Normalização**

Para garantir a integridade e a eficiência da estrutura do banco de dados, foi adotado um processo de normalização até a Terceira Forma Normal (3FN). A seguir, apresentam-se as etapas aplicadas e suas justificativas:

**Primeira Forma Normal (1FN)**

Nesta etapa, foram eliminados grupos repetitivos e garantido que todos os atributos fossem atômicos, ou seja, com valores únicos e indivisíveis. Por exemplo, os telefones e endereços foram tratados como atributos simples por cliente, evitando múltiplos dados no mesmo campo.

**Segunda Forma Normal (2FN)**

Foram evitadas dependências parciais, assegurando que todos os atributos não-chave dependessem da chave primária completa. A modelagem utilizou predominantemente chaves simples (autoincrementadas), o que facilitou a aplicação dessa forma normal.

**Terceira Forma Normal (3FN)**

Na 3FN, foram eliminadas as dependências transitivas. Todos os atributos não-chave dependem apenas da chave primária de sua respectiva tabela. Por exemplo, a descrição do gênero não está presente na tabela livro, mas sim corretamente associada via chave estrangeira com a tabela gênero.

**Justificativa da Aplicação das Formas Normais**

**A adoção da 3FN visa:**

• Redução da redundância de dados

• Aumento da clareza e organização das tabelas

• Facilidade na manutenção e atualização

• Garantia de consistência e integridade referencial

A estrutura resultante é eficiente para consultas e escalável para futuras expansões do sistema.

**4.** **Estrutura do Banco de Dados**

**4.1** **Definição das Tabelas – DDL (Data Definition Language)**

A estrutura do banco de dados foi criada com comandos da linguagem DDL (Data Definition Language) no SGBD PostgreSQL. O script sql/ddl.sql contém as instruções para criar as tabelas, chaves primárias, chaves estrangeiras, tipos de dados e restrições.

A seguir, são descritas as tabelas que compõem o sistema de gerenciamento de biblioteca, com seus respectivos atributos e relacionamentos.

**4.2** **Descrição das Tabelas e Atributos**

**Cliente**

• Armazena os dados cadastrais dos usuários.

• Id\_cliente: Identificador único do cliente (chave primária).

• nome\_cliente: Nome completo do cliente.

• cpf\_cliente: CPF do cliente.

• email\_cliente: Endereço de e-mail.

• telefone\_cliente: Telefone de contato.

• endereco\_cliente: Endereço completo.

• data\_cadastro: Data de cadastro no sistema.

**Produto**

Representa os itens disponíveis na biblioteca (ex.: livros, revistas).

• id\_produto: Identificador único do produto (chave primária).

• nome\_produto: Nome do item.

• descrição: Descrição detalhada do produto.

• preço: Valor do item.

• categoria: Categoria do produto (ex.: livro, periódico).

**Livro**

É uma especialização da tabela Produto.

• id\_livro: Identificador do livro (chave primária).

• nome\_livro: Título do livro.

• descrição: Breve descrição ou sinopse.

• id\_genero: Chave estrangeira para a tabela Genero.

**Gênero**

Define a classificação dos livros com base em seu conteúdo.

• id\_genero: Identificador único do gênero (chave primária).

• descrição: Nome ou tipo do gênero (ex.: Romance, Aventura).

**Estoque**

Controla a quantidade de produtos disponíveis.

• id\_estoque: Identificador do registro de estoque (chave primária).

• id\_produto: Chave estrangeira para Produto.

• quantidade: Quantidade disponível em estoque.

• localização: Local físico do item na biblioteca.

**Pedido**

Registra as solicitações feitas por clientes.

• id\_pedido: Identificador do pedido (chave primária).

• id\_cliente: Chave estrangeira para Cliente.

• data\_pedido: Data em que o pedido foi realizado.

• status: Situação atual do pedido.

• total: Valor total do pedido.

**Contrato**

Formaliza um acordo entre cliente e biblioteca.

• id\_contrato: Identificador do contrato (chave primária).

• id\_cliente: Chave estrangeira para Cliente.

• data\_inicio: Data de início do contrato.

• data\_fim: Data de término do contrato.

• valor: Valor acordado.

**Empréstimo**

Controla os itens emprestados pelos clientes.

• id\_emprestimo: Identificador do empréstimo (chave primária).

• id\_cliente: Chave estrangeira para Cliente.

• id\_produto: Chave estrangeira para Produto.

• data\_emprestimo: Data em que o item foi emprestado.

• data\_devolucao: Data prevista ou efetiva de devolução.

• status: Situação do empréstimo (ativo, devolvido, atrasado).

**Funcionário**

Armazena dados dos colaboradores da biblioteca.

• id\_funcionario: Identificador do funcionário (chave primária).

• nome\_funcionario: Nome completo do funcionário.

• cpf\_funcionario: CPF do funcionário (único).

• cargo: Função ou cargo ocupado.

• telefone\_funcionario: Telefone de contato.

**4.3** **Relacionamento entre Tabelas**

Os relacionamentos entre as tabelas foram definidos por meio de chaves estrangeiras, garantindo a integridade referencial. São eles:

• A tabela Pedido está relacionada à tabela Cliente através do campo id\_cliente.

• A tabela Estoque está ligada à tabela Produto por meio de id\_produto.

• A tabela Livro referência Gênero com o campo id\_genero.

• A tabela Empréstimo conecta Cliente e Produto, permitindo registrar qual produto foi emprestado por qual cliente.

• A tabela Contrato também depende de Cliente.

Essas ligações garantem que o sistema reflita corretamente a lógica da biblioteca e permita operações seguras e consistentes sobre os dados.

# **5. Manipulação de Dados**

**5.1** **Inserção, Atualização e Deleção de Dados (DML)**

A linguagem de manipulação de dados (DML – Data Manipulation Language) foi utilizada para simular operações de rotina no sistema da biblioteca, como cadastro de novos registros, atualização de informações e exclusão de dados obsoletos.

**Inserção de Dados (INSERT)**

O comando INSERT INTO foi utilizado para adicionar novos registros nas tabelas do banco de dados.

Exemplos:

• Inserção de um cliente:

sql

Copiar

Editar

INSERT INTO Cliente (

Id\_cliente, nome\_cliente, cpf\_cliente, email\_cliente,

telefone\_cliente, endereco\_cliente, data\_cadastro)

VALUES (

1, 'João da Silva', '12345678901', 'joao@email.com',

'11999999999', 'Rua A, 123', CURRENT\_DATE);

Inserção de um produto (livro):

sql

Copiar

Editar

INSERT INTO Produto (

id\_produto, nome\_produto, descrição, preço, categoria)

VALUES (

101, 'Dom Casmurro', 'Livro de Machado de Assis', 39.90, 'Livro');

Inserção de gênero e livro:

sql

Copiar

Editar

INSERT INTO Gênero (id\_genero, descrição)

VALUES (1, 'Romance');

INSERT INTO Livro (id\_livro, nome\_livro, descrição, id\_genero)

VALUES (201, 'Dom Casmurro', 'Clássico da literatura brasileira', 1);

b) Atualização de Dados (UPDATE)

O comando UPDATE permite modificar informações existentes no banco.

**Exemplos:**

Atualizar e-mail de um cliente:

sql

Copiar

Editar

UPDATE Cliente

SET email\_cliente = 'joao.silva@email.com'

WHERE Id\_cliente = 1;

Atualizar a quantidade de um produto em estoque:

sql

Copiar

Editar

UPDATE Estoque

SET quantidade = 5

WHERE id\_produto = 101;

c) Exclusão de Dados (DELETE)

O comando DELETE FROM remove registros de uma tabela com base em critérios definidos.

**Exemplos:**

• Excluir cliente:

sql

Copiar

Editar

DELETE FROM Cliente

WHERE Id\_cliente = 1;

Excluir produto:

sql

Copiar

Editar

DELETE FROM Produto

WHERE id\_produto = 101;

**5.2** **Exemplos de Consultas SQL (DQL)**

A linguagem de consulta de dados (DQL – Data Query Language) é utilizada para recuperar informações do banco de dados de acordo com critérios específicos.

Abaixo, seguem exemplos práticos de consultas relevantes para o sistema:

a) Listar todos os clientes cadastrados

sql

Copiar

Editar

SELECT \* FROM Cliente;

Essa consulta retorna todos os campos e registros da tabela de clientes.

• Exibir produtos disponíveis em estoque (quantidade maior que 0)

sql

Copiar

Editar

SELECT p.nome\_produto, e.quantidade

FROM Produto p

JOIN Estoque e ON p.id\_produto = e.id\_produto

WHERE e.quantidade > 0;

Consulta útil para verificar quais itens estão disponíveis na biblioteca.

• Consultar livros e seus respectivos gêneros

sql

Copiar

Editar

SELECT l.nome\_livro, g.descricao AS gênero

FROM Livro l

JOIN Genero g ON l.id\_genero = g.id\_genero;

Relaciona os livros cadastrados com a descrição de seu gênero.

• Ver empréstimos em andamento

sql

Copiar

Editar

SELECT e.id\_emprestimo, c.nome\_cliente, p.nome\_produto,

e.data\_emprestimo, e.data\_devolucao, e.status

FROM Emprestimo e

JOIN Cliente c ON e.id\_cliente = c.Id\_cliente

JOIN Produto p ON e.id\_produto = p.id\_produto

WHERE e.status = 'Em andamento';

• Permite acompanhar os produtos que ainda não foram devolvidos.

• Total de contratos por cliente

sql

Copiar

Editar

SELECT c.nome\_cliente, COUNT(con.id\_contrato) AS total\_contratos

FROM Contrato con

JOIN Cliente c ON con.id\_cliente = c.Id\_cliente

GROUP BY c.nome\_cliente;

Apresenta um resumo do número de contratos firmados por cliente.

Essa seção demonstra a aplicação prática das operações básicas de um banco relacional, contribuindo para a validação da modelagem e funcionamento do sistema desenvolvido.

**6.** **Controle de Acesso (Bonificação – DCL)**

**6.****1 Introdução ao Controle de Acesso**

O controle de acesso em um banco de dados garante que cada usuário tenha apenas as permissões necessárias para executar suas funções, seguindo o princípio do menor privilégio. Isso melhora a segurança, organização e auditoria do sistema.

A linguagem DCL (Data Control Language) é responsável pela gestão dessas permissões, por meio dos comandos GRANT (concessão de privilégios) e REVOKE (revogação de privilégios).

**6.2** **Criação de Usuários (exemplo)**

Antes de aplicar permissões, é necessário criar usuários no SGBD. No PostgreSQL, isso é feito com o comando:

sql

Copiar

Editar

-- Criação de usuários fictícios para o sistema

CREATE USER colaborador\_biblioteca WITH PASSWORD 'senha123';

CREATE USER visitante\_biblioteca WITH PASSWORD 'senha123';

**6.3** **Concessão de Permissões (GRANT)**

O comando GRANT permite liberar o acesso a objetos do banco (como tabelas) para determinados usuários, com permissões específicas.

• Permissões para o usuário colaborador\_biblioteca

Esse usuário precisa acessar e modificar registros das tabelas principais.

sql

Copiar

Editar

-- Permitir leitura e inserção nas tabelas Cliente e Produto

GRANT SELECT, INSERT, UPDATE ON Cliente TO colaborador\_biblioteca;

GRANT SELECT, INSERT, UPDATE ON Produto TO colaborador\_biblioteca;

-- Permitir empréstimos e registros de contratos

GRANT SELECT, INSERT, UPDATE ON Emprestimo TO colaborador\_biblioteca;

GRANT SELECT, INSERT, UPDATE ON Contrato TO colaborador\_biblioteca;

• Permissões para o usuário visitante\_biblioteca

Esse usuário deve ter apenas permissão de leitura sobre determinadas tabelas.

sql

Copiar

Editar

-- Permitir apenas leitura das tabelas de consulta pública

GRANT SELECT ON Produto, Livro, Gênero TO visitante\_biblioteca;

**6.4** **Revogação de Permissões (REVOKE)**

Caso seja necessário restringir o acesso de algum usuário, utiliza-se o comando REVOKE.

**Exemplo:**

sql

Copiar

Editar

-- Revogar permissão de escrita no Produto do colaborador

REVOKE INSERT, UPDATE ON Produto FROM colaborador\_biblioteca;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**7. Conclusão**

O desenvolvimento deste projeto final permitiu consolidar os conhecimentos teóricos e práticos adquiridos durante a disciplina de Banco de Dados Relacional. O processo envolveu desde a modelagem cuidadosa das entidades, atributos e relacionamentos até a implementação das tabelas e regras de integridade utilizando comandos DDL, DML e DQL.

A normalização do banco de dados até a Terceira Forma Normal (3FN) mostrou-se essencial para garantir a integridade e eficiência no armazenamento das informações, minimizando redundâncias e facilitando futuras manutenções. Além disso, a criação de consultas específicas atendeu aos requisitos funcionais do sistema, como controle de empréstimos, gestão de estoque e acompanhamento dos contratos.

Entre as lições aprendidas, destacam-se a importância da modelagem inicial bem estruturada e o domínio das transações para manter a consistência dos dados em operações simultâneas. Também foi possível compreender melhor as práticas de documentação e organização dos scripts SQL, facilitando a manutenção e evolução do sistema.

Para futuras melhorias, sugere-se a implementação de procedimentos armazenados e triggers para automatizar regras de negócio, bem como a criação de uma interface gráfica para facilitar o uso do sistema por usuários finais. Além disso, o projeto pode ser ampliado com recursos de autenticação e controle de acesso para garantir maior segurança.

Este projeto não apenas demonstra a aplicabilidade dos conceitos de banco de dados relacionais, mas também serve como base sólida para o desenvolvimento de sistemas mais complexos e robustos no futuro.

1. **REFERÊNCIAS**

DATE, C. J. Introdução a Sistemas de Bancos de Dados. 8. ed. Rio de Janeiro: Elsevier, 2004.

ELMAZRI, Ramez; NAVATHE, Shamkant. Sistemas de Banco de Dados. 7. ed. São Paulo: Pearson, 2011.

CORONEL, Carlos; MORRIS, Steven. Database Systems: Design, Implementation, and Management. Boston: Cengage Learning, 2015.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. Documentação oficial do PostgreSQL. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 21 jun. 2025.

ANSI. SQL – Structured Query Language: ISO/IEC 9075. International Organization for Standardization, 2016. Disponível em: https://standards.iso.org/iso-iec/9075/. Acesso em: 21 jun. 2025.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson Addison Wesley, 2013.

PEREIRA, Letícia. Linguagens SQL: DDL, DML, DCL e DQL explicadas. Blog Alura, 2022. Disponível em: https://www.alura.com.br/artigos/sql-ddl-dml-dcl-dql. Acesso em: 21 jun. 2025.

PANDAS DEVELOPMENT TEAM. Pandas Documentation. Disponível em: https://pandas.pydata.org/docs/. Acesso em: 21 jun. 2025.

SCIPY DEVELOPMENT TEAM. Scipy Documentation. Disponível em: <https://docs.scipy.org/doc/scipy/>. Acesso em: 21 jun. 2025.

HUNTER, J. D. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, v. 9, n. 3, p. 90-95, 2007. Disponível em: <https://matplotlib.org/stable/index.html>. Acesso em: 21 jun. 2025.