



Projeto Biblioteca

[db_biblioteca.sql](#)

Sistema de Gerenciamento de Banco de Dados - Biblioteca

1. Visão Geral

O banco de dados "Biblioteca" tem como objetivo armazenar informações relacionadas a uma coleção de livros, autores, membros e empréstimos de forma eficiente e organizada. Ele permite o gerenciamento de dados, como cadastros, consultas e controle de empréstimos.

2. Requisitos Principais do Sistema

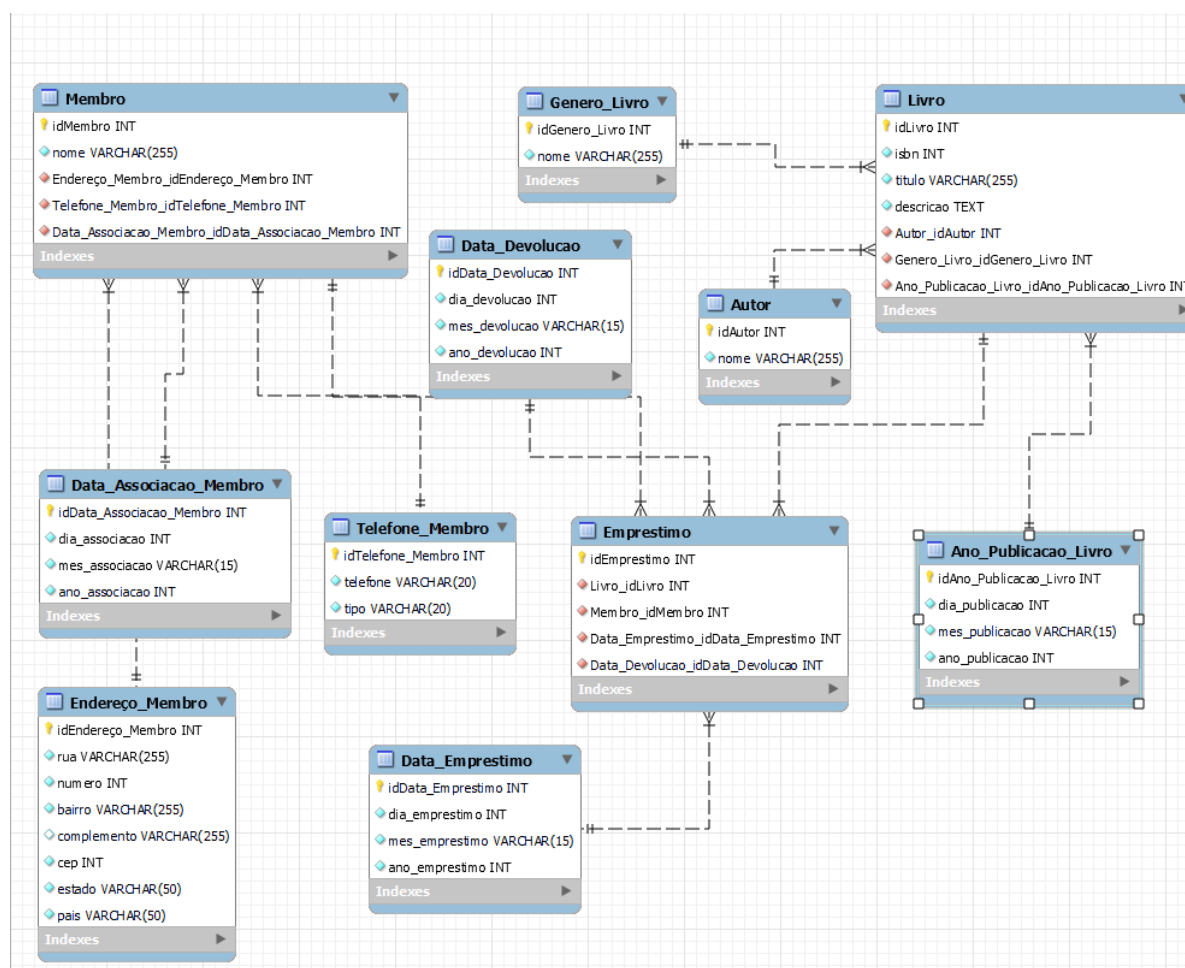
- **Gerenciamento de livros:** Registro de informações como título, autor, gênero, e número de exemplares.
- **Cadastro de usuários:** Registro dos membros da biblioteca, com informações como nome, endereço, e data de inscrição.
- **Controle de empréstimos:** Monitoramento de quais livros foram emprestados, para quem e em que data, além da data de devolução.

- **Relatórios:** Geração de relatórios para monitoramento de livros mais emprestados e membros mais ativos.

3. Estrutura do Banco de Dados

O banco de dados foi estruturado para armazenar as informações de forma otimizada, dividida em tabelas que se relacionam entre si. Aqui estão as principais tabelas:

- **Tabela Livros:** Armazena os dados sobre os livros (ID, título, autor, gênero, ano de publicação).
- **Tabela Autores:** Armazena as informações dos autores (ID, nome, nacionalidade).
- **Tabela Membros:** Contém os dados dos usuários (ID, nome, endereço, data de cadastro).
- **Tabela Empréstimos:** Registra os empréstimos realizados (ID, ID do livro, ID do membro, data de empréstimo, data de devolução).



4. Normalização do Banco de Dados

A normalização foi realizada para eliminar redundâncias e assegurar a consistência dos dados. As principais formas normais aplicadas foram:

- **Primeira Forma Normal (1FN):** Garantiu que todas as colunas contenham valores atômicos (sem valores repetidos ou conjuntos de valores).
Exemplo: cada livro tem apenas um título, autor, e informações unívocas em uma linha da tabela.
- **Segunda Forma Normal (2FN):** Cada tabela foi projetada para garantir que todas as colunas dependessem diretamente da chave primária. Exemplo: a tabela Empréstimos contém um ID que se relaciona diretamente com o ID do livro e do membro.
- **Terceira Forma Normal (3FN):** Evitamos dependências transitivas, ou seja, as colunas que não são chave primária dependem unicamente da chave primária. Exemplo: as informações sobre os autores estão na tabela Autores e não na tabela Livros.

5. Restrições de Integridade

Para garantir a integridade dos dados e evitar erros, aplicamos as seguintes restrições:

- **Chave Primária (Primary Key):** Cada tabela possui uma chave primária única, como o ID do Livro, ID do Autor, e ID do Membro, para garantir a unicidade de cada registro.
- **Chave Estrangeira (Foreign Key):** As tabelas Empréstimos usam chaves estrangeiras (ID do Livro e ID do Membro) para garantir a referência correta entre as tabelas.
- **Integridade Referencial:** A chave estrangeira garante que um empréstimo só pode ser registrado se o livro e o membro existirem no sistema. Isso evita, por exemplo, o empréstimo de livros inexistentes.
- **Restrição de Unicidade:** Garantimos que os membros e livros cadastrados tenham registros únicos em suas respectivas tabelas.
- **Not Null:** Certas colunas, como o título do livro ou o nome do membro, não podem ser deixadas em branco, o que garante que informações cruciais sempre estejam presentes.

```

-- Criação do banco de dados para o projeto Biblioteca Avalia
CREATE DATABASE IF NOT EXISTS Biblioteca;

-- Escolher o banco de dados para uso
USE Biblioteca;

-- Criação da tabela de autores
CREATE TABLE Autor (
    idAutor INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    nome VARCHAR(255) NOT NULL -- Nome do autor com até 255 ca
);

-- Criação da tabela de gêneros de livros
CREATE TABLE Genero_Livro (
    idGenero_Livro INT AUTO_INCREMENT PRIMARY KEY, -- Chave pr
    nome VARCHAR(255) NOT NULL -- Nome do gênero com até 255 c
);

-- Criação da tabela de anos de publicação dos livros
CREATE TABLE Ano_Publicacao_Livro (
    idAno_Publicacao_Livro INT AUTO_INCREMENT PRIMARY KEY, --
    dia_publicacao INT NOT NULL, -- Dia de publicação, campo o
    mes_publicacao VARCHAR(15) NOT NULL, -- Mês de publicação
    ano_publicacao INT NOT NULL -- Ano de publicação, campo ob
);

-- Criação da tabela de livros
CREATE TABLE Livro (
    idLivro INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    isbn INT NOT NULL, -- ISBN (identificação do livro), campo
    titulo VARCHAR(255) NOT NULL, -- Título do livro, campo ob
    descricao TEXT NOT NULL, -- Descrição do livro, campo obri
    Autor_idAutor INT NOT NULL, -- Referência ao ID do autor,
    Genero_Livro_idGenero_Livro INT NOT NULL, -- Referência ao
    Ano_Publicacao_Livro_idAno_Publicacao_Livro INT NOT NULL,
    CONSTRAINT fk_Livro_Autor FOREIGN KEY (Autor_idAutor) REFER
    CONSTRAINT fk_Livro_Genero_Livro FOREIGN KEY (Genero_Livro_
    CONSTRAINT fk_Livro_Ano_Publicacao_Livro FOREIGN KEY (Ano_P

```

```

);

-- Criação da tabela de endereços dos membros
CREATE TABLE Endereço_Membro (
    idEndereço_Membro INT AUTO_INCREMENT PRIMARY KEY, -- Chave
    rua VARCHAR(255) NOT NULL, -- Nome da rua, campo obrigatório
    numero INT NOT NULL, -- Número do endereço, campo obrigatório
    bairro VARCHAR(255) NOT NULL, -- Bairro, campo obrigatório
    complemento VARCHAR(255), -- Complemento (opcional)
    cep INT NOT NULL, -- Código postal (CEP), campo obrigatório
    estado VARCHAR(50) NOT NULL, -- Estado, campo obrigatório
    pais VARCHAR(50) NOT NULL -- País, campo obrigatório
);

-- Criação da tabela de telefones dos membros
CREATE TABLE Telefone_Membro (
    idTelefone_Membro INT AUTO_INCREMENT PRIMARY KEY, -- Chave
    telefone VARCHAR(20) NOT NULL, -- Número de telefone, campo obrigatório
    tipo VARCHAR(20) NOT NULL -- Tipo de telefone (Celular, Fixo)
);

-- Criação da tabela de datas de associação dos membros
CREATE TABLE Data_Associacao_Membro (
    idData_Associacao_Membro INT AUTO_INCREMENT PRIMARY KEY, -- Chave
    dia_associacao INT NOT NULL, -- Dia da associação, campo obrigatório
    mes_associacao VARCHAR(15) NOT NULL, -- Mês da associação, campo obrigatório
    ano_associacao INT NOT NULL -- Ano da associação, campo obrigatório
);

-- Criação da tabela de membros
CREATE TABLE Membro (
    idMembro INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    nome VARCHAR(255) NOT NULL, -- Nome do membro, campo obrigatório
    Endereço_Membro_idEndereço_Membro INT NOT NULL, -- Referência para Endereço_Membro
    Telefone_Membro_idTelefone_Membro INT NOT NULL, -- Referência para Telefone_Membro
    Data_Associacao_Membro_idData_Associacao_Membro INT NOT NULL, -- Referência para Data_Associacao_Membro
    CONSTRAINT fk_Membro_Endereço_Membro FOREIGN KEY (Endereço_Membro_idEndereço_Membro) REFERENCES Endereço_Membro (idEndereço_Membro),
    CONSTRAINT fk_Membro_Telefone_Membro FOREIGN KEY (Telefone_Membro_idTelefone_Membro) REFERENCES Telefone_Membro (idTelefone_Membro),
    CONSTRAINT fk_Membro_Data_Associacao_Membro FOREIGN KEY (Data_Associacao_Membro_idData_Associacao_Membro) REFERENCES Data_Associacao_Membro (idData_Associacao_Membro)
);

```

```

        CONSTRAINT fk_Membro_Data_Associacao_Membro FOREIGN KEY (Data_Emprestimo_idData_Emprestimo)
    );

-- Criação da tabela de datas de empréstimo
CREATE TABLE Data_Emprestimo (
    idData_Emprestimo INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    dia_emprestimo INT NOT NULL, -- Dia do empréstimo, campo obrigatório
    mes_emprestimo VARCHAR(15) NOT NULL, -- Mês do empréstimo, campo obrigatório
    ano_emprestimo INT NOT NULL -- Ano do empréstimo, campo obrigatório
);

-- Criação da tabela de datas de devolução
CREATE TABLE Data_Devolucao (
    idData_Devolucao INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    dia_devolucao INT NOT NULL, -- Dia da devolução, campo obrigatório
    mes_devolucao VARCHAR(15) NOT NULL, -- Mês da devolução, campo obrigatório
    ano_devolucao INT NOT NULL -- Ano da devolução, campo obrigatório
);

-- Criação da tabela de empréstimos
CREATE TABLE Emprestimo (
    idEmprestimo INT AUTO_INCREMENT PRIMARY KEY, -- Chave primária
    Livro_idLivro INT NOT NULL, -- Referência ao ID do livro, campo obrigatório
    Membro_idMembro INT NOT NULL, -- Referência ao ID do membro, campo obrigatório
    Data_Emprestimo_idData_Emprestimo INT NOT NULL, -- Referência ao ID da data de empréstimo, campo obrigatório
    Data_Devolucao_idData_Devolucao INT NOT NULL, -- Referência ao ID da data de devolução, campo obrigatório
    CONSTRAINT fk_Emprestimo_Livro FOREIGN KEY (Livro_idLivro) REFERENCES Livro (idLivro),
    CONSTRAINT fk_Emprestimo_Membro FOREIGN KEY (Membro_idMembro) REFERENCES Membro (idMembro),
    CONSTRAINT fk_Emprestimo_Data_Emprestimo FOREIGN KEY (Data_Emprestimo_idData_Emprestimo) REFERENCES Data_Emprestimo (idData_Emprestimo),
    CONSTRAINT fk_Emprestimo_Data_Devolucao FOREIGN KEY (Data_Devolucao_idData_Devolucao) REFERENCES Data_Devolucao (idData_Devolucao)
);

-- Inserção de autores
INSERT INTO Autor (nome) VALUES ('Autor 1'), ('Autor 2');

-- Inserção de gêneros de livros
INSERT INTO Genero_Livro (nome) VALUES ('Ficção'), ('Não-ficção');

```

```
-- Inserção de anos de publicação
INSERT INTO Ano_Publicacao_Livro (dia_publicacao, mes_publicacao, ano_publicacao, livro_idLivro)
VALUES (1, 1, 2023, 1),
(1, 2, 2023, 2),
(1, 3, 2023, 3),
(1, 4, 2023, 4),
(1, 5, 2023, 5),
(1, 6, 2023, 6),
(1, 7, 2023, 7),
(1, 8, 2023, 8),
(1, 9, 2023, 9),
(1, 10, 2023, 10),
(1, 11, 2023, 11),
(1, 12, 2023, 12),
(2, 1, 2024, 1),
(2, 2, 2024, 2),
(2, 3, 2024, 3),
(2, 4, 2024, 4),
(2, 5, 2024, 5),
(2, 6, 2024, 6),
(2, 7, 2024, 7),
(2, 8, 2024, 8),
(2, 9, 2024, 9),
(2, 10, 2024, 10),
(2, 11, 2024, 11),
(2, 12, 2024, 12),
(3, 1, 2025, 1),
(3, 2, 2025, 2),
(3, 3, 2025, 3),
(3, 4, 2025, 4),
(3, 5, 2025, 5),
(3, 6, 2025, 6),
(3, 7, 2025, 7),
(3, 8, 2025, 8),
(3, 9, 2025, 9),
(3, 10, 2025, 10),
(3, 11, 2025, 11),
(3, 12, 2025, 12),
(4, 1, 2026, 1),
(4, 2, 2026, 2),
(4, 3, 2026, 3),
(4, 4, 2026, 4),
(4, 5, 2026, 5),
(4, 6, 2026, 6),
(4, 7, 2026, 7),
(4, 8, 2026, 8),
(4, 9, 2026, 9),
(4, 10, 2026, 10),
(4, 11, 2026, 11),
(4, 12, 2026, 12),
(5, 1, 2027, 1),
(5, 2, 2027, 2),
(5, 3, 2027, 3),
(5, 4, 2027, 4),
(5, 5, 2027, 5),
(5, 6, 2027, 6),
(5, 7, 2027, 7),
(5, 8, 2027, 8),
(5, 9, 2027, 9),
(5, 10, 2027, 10),
(5, 11, 2027, 11),
(5, 12, 2027, 12),
(6, 1, 2028, 1),
(6, 2, 2028, 2),
(6, 3, 2028, 3),
(6, 4, 2028, 4),
(6, 5, 2028, 5),
(6, 6, 2028, 6),
(6, 7, 2028, 7),
(6, 8, 2028, 8),
(6, 9, 2028, 9),
(6, 10, 2028, 10),
(6, 11, 2028, 11),
(6, 12, 2028, 12),
(7, 1, 2029, 1),
(7, 2, 2029, 2),
(7, 3, 2029, 3),
(7, 4, 2029, 4),
(7, 5, 2029, 5),
(7, 6, 2029, 6),
(7, 7, 2029, 7),
(7, 8, 2029, 8),
(7, 9, 2029, 9),
(7, 10, 2029, 10),
(7, 11, 2029, 11),
(7, 12, 2029, 12),
(8, 1, 2030, 1),
(8, 2, 2030, 2),
(8, 3, 2030, 3),
(8, 4, 2030, 4),
(8, 5, 2030, 5),
(8, 6, 2030, 6),
(8, 7, 2030, 7),
(8, 8, 2030, 8),
(8, 9, 2030, 9),
(8, 10, 2030, 10),
(8, 11, 2030, 11),
(8, 12, 2030, 12),
(9, 1, 2031, 1),
(9, 2, 2031, 2),
(9, 3, 2031, 3),
(9, 4, 2031, 4),
(9, 5, 2031, 5),
(9, 6, 2031, 6),
(9, 7, 2031, 7),
(9, 8, 2031, 8),
(9, 9, 2031, 9),
(9, 10, 2031, 10),
(9, 11, 2031, 11),
(9, 12, 2031, 12),
(10, 1, 2032, 1),
(10, 2, 2032, 2),
(10, 3, 2032, 3),
(10, 4, 2032, 4),
(10, 5, 2032, 5),
(10, 6, 2032, 6),
(10, 7, 2032, 7),
(10, 8, 2032, 8),
(10, 9, 2032, 9),
(10, 10, 2032, 10),
(10, 11, 2032, 11),
(10, 12, 2032, 12),
(11, 1, 2033, 1),
(11, 2, 2033, 2),
(11, 3, 2033, 3),
(11, 4, 2033, 4),
(11, 5, 2033, 5),
(11, 6, 2033, 6),
(11, 7, 2033, 7),
(11, 8, 2033, 8),
(11, 9, 2033, 9),
(11, 10, 2033, 10),
(11, 11, 2033, 11),
(11, 12, 2033, 12),
(12, 1, 2034, 1),
(12, 2, 2034, 2),
(12, 3, 2034, 3),
(12, 4, 2034, 4),
(12, 5, 2034, 5),
(12, 6, 2034, 6),
(12, 7, 2034, 7),
(12, 8, 2034, 8),
(12, 9, 2034, 9),
(12, 10, 2034, 10),
(12, 11, 2034, 11),
(12, 12, 2034, 12),
(13, 1, 2035, 1),
(13, 2, 2035, 2),
(13, 3, 2035, 3),
(13, 4, 2035, 4),
(13, 5, 2035, 5),
(13, 6, 2035, 6),
(13, 7, 2035, 7),
(13, 8, 2035, 8),
(13, 9, 2035, 9),
(13, 10, 2035, 10),
(13, 11, 2035, 11),
(13, 12, 2035, 12),
(14, 1, 2036, 1),
(14, 2, 2036, 2),
(14, 3, 2036, 3),
(14, 4, 2036, 4),
(14, 5, 2036, 5),
(14, 6, 2036, 6),
(14, 7, 2036, 7),
(14, 8, 2036, 8),
(14, 9, 2036, 9),
(14, 10, 2036, 10),
(14, 11, 2036, 11),
(14, 12, 2036, 12),
(15, 1, 2037, 1),
(15, 2, 2037, 2),
(15, 3, 2037, 3),
(15, 4, 2037, 4),
(15, 5, 2037, 5),
(15, 6, 2037, 6),
(15, 7, 2037, 7),
(15, 8, 2037, 8),
(15, 9, 2037, 9),
(15, 10, 2037, 10),
(15, 11, 2037, 11),
(15, 12, 2037, 12),
(16, 1, 2038, 1),
(16, 2, 2038, 2),
(16, 3, 2038, 3),
(16, 4, 2038, 4),
(16, 5, 2038, 5),
(16, 6, 2038, 6),
(16, 7, 2038, 7),
(16, 8, 2038, 8),
(16, 9, 2038, 9),
(16, 10, 2038, 10),
(16, 11, 2038, 11),
(16, 12, 2038, 12),
(17, 1, 2039, 1),
(17, 2, 2039, 2),
(17, 3, 2039, 3),
(17, 4, 2039, 4),
(17, 5, 2039, 5),
(17, 6, 2039, 6),
(17, 7, 2039, 7),
(17, 8, 2039, 8),
(17, 9, 2039, 9),
(17, 10, 2039, 10),
(17, 11, 2039, 11),
(17, 12, 2039, 12),
(18, 1, 2040, 1),
(18, 2, 2040, 2),
(18, 3, 2040, 3),
(18, 4, 2040, 4),
(18, 5, 2040, 5),
(18, 6, 2040, 6),
(18, 7, 2040, 7),
(18, 8, 2040, 8),
(18, 9, 2040, 9),
(18, 10, 2040, 10),
(18, 11, 2040, 11),
(18, 12, 2040, 12),
(19, 1, 2041, 1),
(19, 2, 2041, 2),
(19, 3, 2041, 3),
(19, 4, 2041, 4),
(19, 5, 2041, 5),
(19, 6, 2041, 6),
(19, 7, 2041, 7),
(19, 8, 2041, 8),
(19, 9, 2041, 9),
(19, 10, 2041, 10),
(19, 11, 2041, 11),
(19, 12, 2041, 12),
(20, 1, 2042, 1),
(20, 2, 2042, 2),
(20, 3, 2042, 3),
(20, 4, 2042, 4),
(20, 5, 2042, 5),
(20, 6, 2042, 6),
(20, 7, 2042, 7),
(20, 8, 2042, 8),
(20, 9, 2042, 9),
(20, 10, 2042, 10),
(20,
```

```

    Autor.nome AS 'Autor', -- Seleciona o nome do autor
    Genero_Livro.nome AS 'Gênero', -- Seleciona o nome do gênero
    CONCAT(Ano_Publicacao_Livro.dia_publicacao, ' ', Ano_Publicacao_Livro.mes_publicacao) AS 'Data de Publicação',
FROM Livro
JOIN Autor ON Livro.Autor_idAutor = Autor.idAutor -- Relação Livro-Autor
JOIN Genero_Livro ON Livro.Genero_Livro_idGenero_Livro = Genero_Livro.idGenero_Livro -- Relação Livro-Genero
JOIN Ano_Publicacao_Livro ON Livro.Ano_Publicacao_Livro_idAno_Publicacao_Livro = Ano_Publicacao_Livro.idAno_Publicacao_Livro -- Relação Livro-Ano_Publicacao_Livro

-- Consultar todos os membros e seus respectivos endereços, telefones e data de associação
SELECT
    Membro.nome AS 'Nome do Membro', -- Seleciona o nome do membro
    CONCAT(Endereco_Membro.rua, ' ', Endereco_Membro.numero, ' ', Endereco_Membro.cidade) AS 'Endereço',
    Telefone_Membro.telefone AS 'Telefone', -- Seleciona o número de telefone
    CONCAT(Data_Associacao_Membro.dia_associacao, ' ', Data_Associacao_Membro.mes_associacao) AS 'Data de Associação',
FROM Membro
JOIN Endereco_Membro ON Membro.Endereco_Membro_idEndereco_Membro = Endereco_Membro.idEndereco_Membro -- Relação Membro-Endereco_Membro
JOIN Telefone_Membro ON Membro.Telefone_Membro_idTelefone_Membro = Telefone_Membro.idTelefone_Membro -- Relação Membro-Telefone_Membro
JOIN Data_Associacao_Membro ON Membro.Data_Associacao_Membro_idData_Associacao_Membro = Data_Associacao_Membro.idData_Associacao_Membro -- Relação Membro-Data_Associacao_Membro

-- Consultar todos os empréstimos e seus respectivos livros, membros e datas de empréstimo e devolução
SELECT
    Empréstimo.idEmpréstimo AS 'ID do Empréstimo', -- Seleciona o ID do empréstimo
    Livro.titulo AS 'Título do Livro', -- Seleciona o título do livro
    Membro.nome AS 'Nome do Membro', -- Seleciona o nome do membro
    CONCAT(Data_Empréstimo.dia_emprestimo, ' ', Data_Empréstimo.mes_emprestimo) AS 'Data de Empréstimo',
    CONCAT(Data_Devolucao.dia_devolucao, ' ', Data_Devolucao.mes_devolucao) AS 'Data de Devolução',
FROM Empréstimo
JOIN Livro ON Empréstimo.Livro_idLivro = Livro.idLivro -- Relação Empréstimo-Livro
JOIN Membro ON Empréstimo.Membro_idMembro = Membro.idMembro -- Relação Empréstimo-Membro
JOIN Data_Empréstimo ON Empréstimo.Data_Empréstimo_idData_Empréstimo = Data_Empréstimo.idData_Empréstimo -- Relação Empréstimo-Data_Empréstimo
JOIN Data_Devolucao ON Empréstimo.Data_Devolucao_idData_Devolucao = Data_Devolucao.idData_Devolucao -- Relação Empréstimo-Data_Devolucao

```

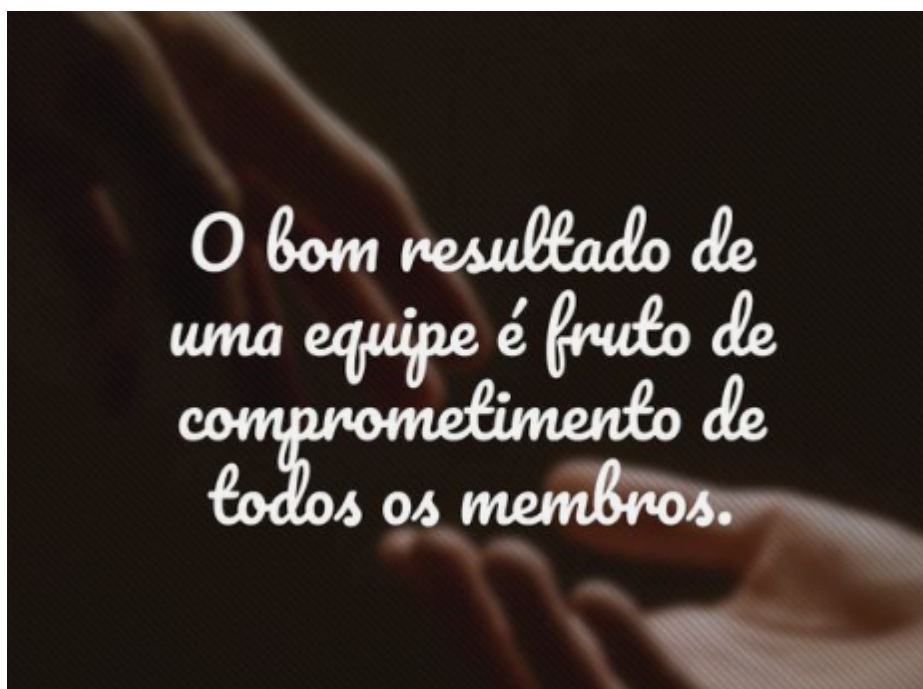
6. Considerações Finais

Este sistema de banco de dados para a Biblioteca foi projetado pensando em uma estrutura eficiente, segura e escalável, que facilita a gestão de livros e empréstimos. Com as regras de normalização e restrições de integridade

aplicadas, evitamos redundâncias e asseguramos a consistência dos dados ao longo do tempo.

7. Próximos Passos

- **Implementação:** Implementar o sistema utilizando ferramentas de gerenciamento de banco de dados como MySQL ou PostgreSQL.
- **Testes:** Realizar testes de inserção, atualização e exclusão de dados para verificar a funcionalidade e integridade do sistema.
- **Expansão:** Pensar em novos módulos, como uma interface de consulta online para usuários ou integração com plataformas digitais de livros.



Este projeto foi desenvolvido por **Maria Fernanda Ribeiro Corrales e Guido Fernandes da Guarda**. Trata-se da PARTE 2 da segunda avaliação do curso de Administrador de Banco de Dados, oferecido pelo SENAI - Taguatinga - DF em parceria com o Programa DF INOVATECH. Mentorado pela Professora Mirka Juliet.