



UNIVERSIDADE PRIVADA DE ANGOLA
FACULDADE DE CIÊNCIA EXATAS
DEPARTAMENTO DE ENGENHARIA

Escopo e Documentação Técnica: Projeto de Base de Dados para Biblioteca

Luanda/2025

UNIVERSIDADE PRIVADA DE ANGOLA
FACULDADE DE CIÊNCIA EXATAS
DEPARTAMENTO DE ENGENHARIA

Escopo e Documentação Técnica: Projeto de Base de Dados para Biblioteca

DOCENTE

Luanda/2025

INTEGRANTES DO GRUPO

NOME	Nº ESTUDANTE	NOTA
Dinétria Camilo	1023339	
Igor Texeira	1023179	
Mateus Cazunga	1023037	

Sumário

1.1. Título do Projeto	5
1.2. Introdução	5
1.4. Entregáveis	5
1.5. Metodologia	5
1.6. Requisitos	5
Requisitos Funcionais:	5
Requisitos Não Funcionais:	5
1.7. Sprints	6
1.8. Critérios de Aceitação.....	6
2.1. Arquitetura do Banco de Dados.....	8
Modelo Conceitual:	8
Entidades principais:.....	8
Diagrama UML.....	8
Modelo Lógico:	9
Tabelas e relacionamentos:.....	9
2.2. Tecnologias e Ferramentas	12
2.3.Procedimentos Armazenados.....	13
2.4.Consultas com Joins	15
2.5.Criação de Views	17
2.6.Consultas com Order by	19
2.7.Consultas com Like	19
2.4.Between(Not in, In)	20
2.5. Plano de Testes	21
2.6. Plano de Manutenção.....	21
APENDICE: A.....	22

1. Escopo do Projeto

1.1. Título do Projeto

Desenvolvimento de uma Base de Dados para Gerenciamento de Biblioteca

1.2. Introdução

O projeto tem como objetivo criar um banco de dados funcional para uma biblioteca, permitindo o gerenciamento eficiente de livros, usuários, empréstimos e devoluções. O sistema será desenvolvido utilizando **MySQL Workbench** como banco de dados relacional e **Git** para controle de versão do código e scripts.

1.3. Objetivos

- Organizar e armazenar informações de livros, usuários e transações de empréstimos.
- Garantir integridade, segurança e acesso eficiente aos dados.
- Facilitar consultas e relatórios sobre operações da biblioteca.

1.4. Entregáveis

- Modelo conceitual, lógico e físico do banco de dados.
- Scripts para criação e população do banco.
- Procedimentos armazenados, consultas com Joins, Criação de Views, consultas com Order By(ID e Alfabético), consultas com Like, consultas com Between(Not in, In).
- Documentação do banco de dados e do código.

1.5. Metodologia

- Scrum

1.6. Requisitos

Requisitos Funcionais:

- Registrar livros com título, autor, gênero, ISBN e disponibilidade.
- Gerenciar usuários (cadastro, exclusão e atualização de dados).
- Registrar empréstimos e devoluções com datas e status.

Requisitos Não Funcionais:

- Utilização do banco de dados MySQL.
- Controle de versão com Git.
- Segurança para evitar duplicidade e corrupção de dados.

1.7. Sprints

Nº da Sprint	Data	Objetivos
1º	12/11/2024	Instalar o WorkBenchMySQL; Criar a base de dados; Criar tabelas com relacionamentos.
2º	19/11/2024	Melhorar a modelagem; Reconstruir a base de dados; Estudar sobre: Join, join left, join main, join Righ.
3º	26/11/2024	Estudar sobre procedimento armazenado; Inserir dados na base de dados; Salvar o script; Fazer consultas relacionais(Inner join) em todas tabelas que têm relacionamentos.
4º	03/12/2024	Criar vários procedimentos armazenados(Mínimo 4 procedimentos) com instruções robustas, consultas e inserir; Views.
5º	10/12/2024	Views. Consultas com ordey by; Ordey by id, do último ao primeiro; Ordey by organizar por ordem alfabética; Consultas com like.

1.8. Critérios de Aceitação

- Banco de dados funcional com scripts para criação e manutenção.
- Todas as funcionalidades previstas implementadas e testadas.
- Documentação clara e bem estruturada.

2. Documentação Técnica

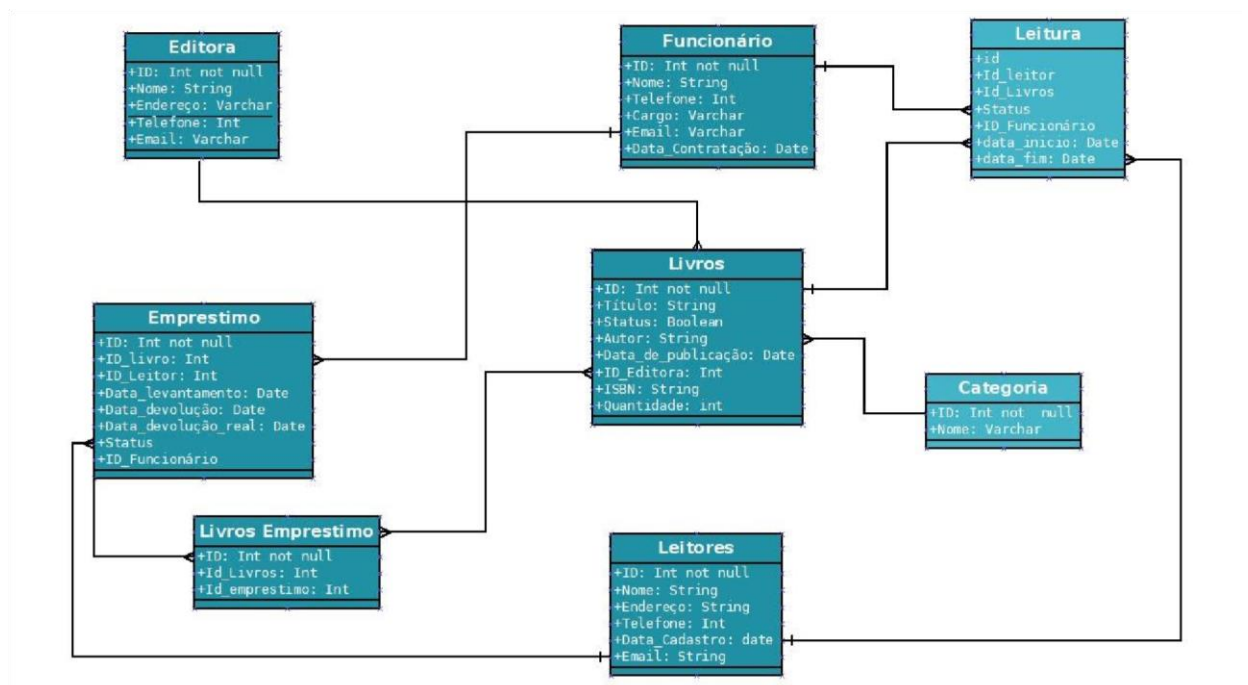
2.1. Arquitetura do Banco de Dados

Modelo Conceitual:

Entidades principais:

- **Categoria** (id_categoria, nome)
- **Livro** (id_livro, titulo, autor, ano_publicacao, quantidade, isbn, id_categoria, id_editora)
- **Editora** (id_editora, nome, endereço, email, telefone)
- **Funcionarios** (id_funcionario, nome, cargo, data_contratacao, telefone, email)
- **Leitores** (id_leitor, nome, email, telefone, endereço, data_cadastro)
- **Leitura** (id_leitura, id_leitor, id_livro, id_funcionario, data_inicio, data_fim, status)
- **Emprestimo** (id_emprestimo, ID_livro, ID_Leitor, data_emprestimo, data_devolucao, Data_devolução_real, id_funcionario)
- **Livro_Emprestimo** (id, id_livro, id_emprestimo)

Diagrama UML



Modelo Lógico:

Tabelas e relacionamentos:

Tabela: Categoria

Campo	Tipo	Descrição
id_categoria	int (pk, incremento_auto)	Identificador único da categoria.
nome	varchar	Nome da categoria (não nulo).

Tabela: Livros

Campo	Tipo	Descrição
id_livro	int (auto_increment, primary key)	Identificador único do livro.
título	varchar (not null)	Título do livro.
autor	varchar (not null)	Autor do livro.
ano_publicacao	year	Ano de publicação.
quantidade	int	Quantidade em estoque.
ISBN	varchar (unique)	Código ISBN (único).
id_categoria	int (fk)	Referência à tabela Categoria.
id_editor	int (fk)	Referência à tabela Editora.

Tabela: Editora

Campo	Tipo	Descrição
id_editora	int (pk)	Identificador único da editora.
nome	varchar (not null)	Nome da editora.
endereco	varchar	Endereço da editora.
e-mail	varchar	Email da editora.
telefone	varchar	Telefone da editora.

Tabela: Funcionários

Campo	Tipo	Descrição
id_funcionario	int (int, auto_increment, primary key)	Identificador único do funcionário.
nome	varchar (not null)	Nome do funcionário.
carga	varchar	Cargo do funcionário.
data_contratacao	date	Data da contratação.
telefone	varchar	Telefone do funcionário.
e-mail	varchar	E-mail do funcionário.

Tabela: Leitores

Campo	Tipo	Descrição
id_leitor	int (auto_increment, primary key)	Identificador único do leitor.
nome	varchar (not null)	Nome do leitor.
e-mail	varchar (unique)	E-mail único do leitor.
telefone	varchar	Telefone do leitor.
endereco	varchar	Endereço do leitor.
data_cadastro	date	Data de cadastro do leitor.

Tabela: Leitura

Campo	Tipo	Descrição
id_leitura	int (pk, incremento_auto)	Identificador único de leitura.
id_leitor	int (fk)	Referência à tabela Leitores.
id_livro	int (fk)	Referência à tabela Livros.
id_funcionario	int (fk)	Referência à tabela Funcionarios.
data_inicio	date	Data de início da leitura.
data_fim	date	Data prevista do fim da leitura.
status	enum ('disponível', 'emprestado', 'reservado')	Status da leitura.

Tabela: Emprestimo

Campo	Tipo	Descrição
id_emprestimo	int (auto_increment primary key)	Identificador único do empréstimo.
id_livro	int (fk)	Referência à tabela Livros.
id_leitor	int (fk)	Referência à tabela Leitores.
data_emprestimo	date (not null)	Data do empréstimo.
data_devolucao	date	Data prevista para devolução.
data_devolucao_real	date	Data reais da devolução.
id_funcionario	int (fk)	Referência à tabela Funcionarios.

Tabela: Livro_Emprestimo

Campo	Tipo	Descrição
id	int (pk, auto_increment, primary key)	Identificador único.
id_livro	int (fk)	Referência à tabela Livros.
id_emprestimo	int (fk)	Referência à tabela Emprestimo.

Relacionamentos:

- Categoria 1 :N Livros
- Editora 1 :N Livros
- Livros N :N Emprestimo (via Livro_Emprestimo)
- Leitores 1 :N Leitura
- Funcionários 1 :N Leitura, 1 :N Emprestimo

Modelo Físico:

- Tabelas e atributos serão implementados no MySQL conforme o modelo lógico derivado do conceito acima.

-Vide Apendice A

2.2. Tecnologias e Ferramentas

Banco de Dados: **MySQL**.

Controle de Versão: **Git**.

Ferramentas Adicionais: MySQL Workbench, GitHub ou GitLab.

2.3.Procedimentos Armazenados

```
303  -- CRIAÇÃO DE PROCESSOS ARMazenados
304
305  DELIMITER //
306
307  CREATE PROCEDURE InserirLivro(
308      IN titulo VARCHAR(255),
309      IN autor VARCHAR(255),
310      IN ano_publicacao YEAR,
311      IN quantidade INT,
312      IN isbn VARCHAR(20),
313      IN id_categoria INT,
314      IN id_editora INT
315  )
316  BEGIN
317      INSERT INTO Livros (titulo, autor, ano_publicacao, quantidade, isbn, id_categoria, id_editora)
318      VALUES (titulo, autor, ano_publicacao, quantidade, isbn, id_categoria, id_editora);
319  END //
320
321  DELIMITER ;
```

Procedimento para inserção de livros

```
337  CREATE PROCEDURE InserirEmprestimo(
338      IN id_livro INT,
339      IN id_leitor INT,
340      IN id_funcionario INT,
341      IN data_emprestimo DATE,
342      IN data_devolucao DATE,
343      IN status ENUM('Emprestado', 'Devolvido', 'Atrasado')
344  )
345  BEGIN
346      INSERT INTO Emprestimo (id_livro, id_leitor, id_funcionario, data_emprestimo, data_devolucao, status)
347      VALUES (id_livro, id_leitor, id_funcionario, data_emprestimo, data_devolucao, status);
348  END //
349
350  DELIMITER ;
351
352  CALL InserirEmprestimo(3, 2, 1, '2024-11-30', '2024-12-15', 'Emprestado');
353
```

Procedimento para cadastrar um empréstimo

```

354 -- Consultar livros disponíveis
355
356 DELIMITER //
357
358 • CREATE PROCEDURE ConsultarLivrosDisponiveis()
359   BEGIN
360     SELECT id_livro, titulo, autor, quantidade
361     FROM Livros
362     WHERE quantidade > 0;
363   END //
364
365 DELIMITER ;
366
367 • CALL ConsultarLivrosDisponiveis();
368
369 -- Registrar o empréstimo de um livro
370
371 DELIMITER //

```

Procedimento para consultar os livros disponíveis

```

417 -- Registrar a devolução de um livro
418
419 DELIMITER //
420 • DROP PROCEDURE IF EXISTS RegistrarDevolucao;
421
422 • CREATE PROCEDURE RegistrarDevolucao(
423   IN p_id_emprestimo INT -- Renomeamos o parâmetro para evitar ambiguidade
424 )
425   BEGIN
426
427     UPDATE emprestimo
428     SET data_devolucao = CURDATE()
429     WHERE id_emprestimo = p_id_emprestimo;
430
431
432     UPDATE Livros
433     SET quantidade = quantidade + 1
434     WHERE id_livro = (SELECT id_livro FROM emprestimo WHERE id_emprestimo = p_id_emprestimo);
435   END //
436
437 DELIMITER ;
438 • CALL RegistrarDevolucao(5);
439

```

Procedimento para registrar a devolução de livros

2.4.Consultas com Joins

```
241  -- Consultas INNER-JOIN
242
243  -- Apresenta os livros emprestados, leitores e funcionários
244  • SELECT
245      L.titulo AS Livro,
246      LE.nome AS Leitor,
247      F.nome AS Funcionario,
248      E.data_emprestimo AS DataEmprestimo,
249      E.data_devolucao AS DataDevolucao
250  FROM Emprestimo E
251  INNER JOIN Livros L ON E.ID_livro = L.id_livro
252  INNER JOIN Leitores LE ON E.ID_leitor = LE.id_leitor
253  INNER JOIN Funcionarios F ON E.id_funcionario = F.id_funcionario;
```

Apresenta os livros emprestados, leitores e funcionários

```
255
256  -- Apresenta Leitor,Livro,Status,data do inicio e fim
257  • SELECT
258      LE.nome AS Leitor,
259      L.titulo AS Livro,
260      LT.status AS StatusLeitura,
261      LT.data_inicio AS DataInicio,
262      LT.data_fim AS DataFim
263  FROM Leitura LT
264  INNER JOIN Livros L ON LT.id_livro = L.id_livro
265  INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor;
266
267  -- Livro com sua categoria e editora
268  • SELECT
269      L.titulo AS Livro,
270      C.nome AS Categoria,
271      E.nome AS Editora
272  FROM Livros L
273  INNER JOIN Categoria C ON L.id_categoria = C.id_categoria
274  INNER JOIN Editora E ON L.id_editora = E.id_editora;
275
```

Apresenta Leitor, Livro, Status, Data do inicio e fim/ Livro com sua categoria e editora

```

278 • SELECT
279     F.nome AS Funcionario,
280     LE.nome AS Leitor,
281     L.titulo AS Livro,
282     LT.data_inicio AS DataInicio,
283     LT.data_fim AS DataFim
284 FROM Leitura LT
285 INNER JOIN Funcionarios F ON LT.id_funcionario = F.id_funcionario
286 INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor
287 INNER JOIN Livros L ON LT.id_livro = L.id_livro;
288

```

Funcionários que registraram

```

289 -- Leitores que concluíram leituras
290
291 • SELECT
292     LE.nome AS Leitor,
293     L.titulo AS Livro,
294     LT.data_inicio AS DataInicio,
295     LT.data_fim AS DataFim,
296     F.nome AS Funcionario
297 FROM Leitura LT
298 INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor
299 INNER JOIN Livros L ON LT.id_livro = L.id_livro
300 INNER JOIN Funcionarios F ON LT.id_funcionario = F.id_funcionario
301 WHERE LT.status = 'Concluída';
302

```

Leitores que concluíram leituras

2.5.Criação de Views

```
353
354 -- Criação de Views
355
356 • CREATE VIEW Leituras_Por_Leitor AS
357 SELECT
358     LT.nome AS Leitor,
359     L.titulo AS Livro,
360     LD.data_inicio,
361     LD.data_fim,
362     LD.status AS StatusLeitura
363 FROM
364     Leitura LD
365 JOIN
366     Livros L ON LD.id_livro = L.id_livro
367 JOIN
368     Leitores LT ON LD.id_leitor = LT.id_leitor;
369
```

Apresenta os leitores e o status da leitura

```
371 • CREATE VIEW Livros_Categorias AS
372 SELECT
373     L.id_livro,
374     L.titulo AS Livro,
375     C.nome AS Categoria
376 FROM
377     Livros L
378 JOIN
379     Categoria C ON L.id_categoria = C.id_categoria;
380
```

Apresenta os livros e a categoria.

```
381 -- Visualizar os funcionários e suas atividades relacionadas a empréstimos.
382
383 • CREATE VIEW Funcionarios_Atividades AS
384 SELECT
385     F.id_funcionario,
386     F.nome AS Funcionario,
387     COUNT(E.id_emprestimo) AS TotalEmprestimos
388 FROM
389     Funcionarios F
390 LEFT JOIN
391     Emprestimo E ON F.id_funcionario = E.id_funcionario
392 GROUP BY
393     F.id_funcionario, F.nome;
394
```

Apresenta os funcionários e suas atividades relacionadas a empréstimos.

```

397 • CREATE VIEW Emprestimos_Atuais AS
398 SELECT
399     E.id_emprestimo,
400     L.titulo AS Livro,
401     LT.nome AS Leitor,
402     E.data_emprestimo,
403     E.data_devolucao,
404     E.status
405 FROM
406     Emprestimo E
407 JOIN
408     Livros L ON E.id_livro = L.id_livro
409 JOIN
410     Leitores LT ON E.id_leitor = LT.id_leitor
411 WHERE
412     E.status = 'Emprestado';
413

```

Apresenta os empréstimos feitos

2.6.Consultas com Order by

```
427      -- Consultas com Order By
428
429      -- Consulta por ordem alfabetica livros
430 • SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
431      FROM Livros
432      ORDER BY titulo ASC;
433
434      -- consulta por ordem alfabetica leitores
435 • SELECT id_leitor, nome, email, telefone, endereco, data_cadastro
436      FROM Leitores
437      ORDER BY nome ASC;
438
439      -- consulta por ordem alfabetica Descendentes leitores
440 • SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
441      FROM Livros
442      ORDER BY titulo DESC;
443
```

Consultas com order by por ordem alfabética

2.7.Consultas com Like

```
445      -- Consultas com like
446      -- Buscar livros cujo título contém a palavra 'Vida'
447 • SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
448      FROM Livros
449      WHERE titulo LIKE '%vida%';
450
451      -- Buscar leitores cujo nome comece com "M"
452 • SELECT id_leitor, nome
453      FROM Leitores
454      WHERE nome LIKE 'D%' ;
455
```

2.4.Between(Not in, In)

```
468      -- consulta com NOT IN
469      -- Retorna todos os livros que não pertencem às categorias 2, 5, ou 9.
470
471 •   SELECT id_livro, titulo, id_categoria
472      FROM Livros
473     WHERE id_categoria NOT IN (2, 5, 9);
474
475      -- Exclui os funcionários com os IDs 1, 4, e 6.
476
477 •   SELECT id_funcionario, nome, cargo
478      FROM Funcionarios
479     WHERE id_funcionario NOT IN (1, 4, 6);
480
481
482      -- Consultas com BETWEEN
483      -- Lista de leitores cujo número id_leitor está entre 5 e 15, inclusivo.
484
485 •   SELECT id_leitor, nome, telefone
486      FROM Leitores
487     WHERE id_leitor BETWEEN 5 AND 15;
488
489
490      -- consulta com IN
491      -- Retorna apenas os livros que pertencem às categorias 3, 7, ou 10.
492
493 •   SELECT id_livro, titulo, id_categoria
494      FROM Livros
495     WHERE id_categoria IN (3, 7, 10);
496
```

2.5. Plano de Testes

- **Testes de funcionalidade:** Validar todas as operações CRUD (Create, Read, Update, Delete).
- **Testes de desempenho:** Avaliar o tempo de resposta de consultas complexas.
- **Testes de segurança:** Garantir que a base de dados está protegida contra injeção SQL.

2.6. Plano de Manutenção

- ✓ Atualização de registros e manutenção de integridade referencial.
- ✓ Backup semanal automático utilizando ferramentas nativas do MySQL. Monitoramento de logs para identificar falhas ou operações indevidas.

APENDICE: A

```
-- Criando a base de dados
create database biblioteca;
use biblioteca;

-- Criando a tabela de Categoria
CREATE TABLE Categoria (
    id_categoria INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL
);

-- Criando a tabela de Livros
CREATE TABLE Livros (
    id_livro INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    autor VARCHAR(255) NOT NULL,
    ano_publicacao YEAR,
    quantidade int,
    isbn VARCHAR(13) UNIQUE,
    id_categoria INT,
    id_editora INT,
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria),
    FOREIGN KEY (id_editora) REFERENCES Editora(id_editora)
);

-- Criando a tabela de Editora
CREATE TABLE Editora (
    id_editora INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    endereco VARCHAR(255),
    email varchar(255),
    telefone VARCHAR(20)
);

-- Criando a tabela de Funcionários
CREATE TABLE Funcionario (
    id_funcionario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    cargo VARCHAR(100),
    data_contratacao DATE,
    telefone VARCHAR(20),
    email varchar(100)
);

-- Criando a tabela de Leitores
CREATE TABLE Leitores (
    id_leitor INT AUTO_INCREMENT PRIMARY KEY,
```



```

    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE,
    telefone VARCHAR(20),
    endereco VARCHAR(255),
    data_cadastro Date);

-- Criando a tabela de Leitura
CREATE TABLE Leitura (
    id_leitura INT AUTO_INCREMENT PRIMARY KEY,
    id_leitor INT NOT NULL,
    id_livro INT NOT NULL,
    id_funcionario INT NOT NULL,
    data_inicio DATE,
    data_fim DATE,
    status ENUM('Disponível', 'Emprestado', 'Reservado') DEFAULT 'Disponível',
    FOREIGN KEY (id_leitor) REFERENCES Leitores(id_leitor),
    FOREIGN KEY (id_livro) REFERENCES Livros(id_livro),
    FOREIGN KEY (id_funcionario) REFERENCES Funcionarios(id_funcionario));

-- Criando a tabela de Empréstimos
CREATE TABLE Emprestimo (
    id_emprestimo INT AUTO_INCREMENT PRIMARY KEY,
    ID_livro INT NOT NULL,
    ID_Leitor INT NOT NULL,
    data_emprestimo DATE NOT NULL,
    data_devolucao DATE,
    Data_devolução_real Date,
    id_funcionario INT,
    FOREIGN KEY (id_funcionario) REFERENCES Funcionarios(id_funcionario),
    FOREIGN KEY (id_livro) REFERENCES Livros(id_livro),
    FOREIGN KEY (id_leitor) REFERENCES Leitores(id_leitor));

-- Criação tabela Livro_Emprestimo
CREATE TABLE Livro_Emprestimo (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_livro INT NOT NULL,
    id_emprestimo INT NOT NULL,
    FOREIGN KEY (id_livro) REFERENCES Livros(id_livro),
    FOREIGN KEY (id_emprestimo) REFERENCES Emprestimo(id_emprestimo));

-- Visualização das tabelas
DESCRIBE Categoria;
SELECT * FROM Categoria;
SELECT * FROM Editora;
SELECT * FROM Leitores;
SELECT * FROM Funcionarios;
SELECT `status` FROM Leitura;

```

```

SELECT * FROM Leitura;
SELECT * FROM Livros;
show tables;
SELECT * FROM Livro_Emprestimo;
SELECT * FROM Emprestimo;
SELECT id_livro FROM Livros;
DESCRIBE Leitores;
-- Diversão
SELECT * FROM Emprestimo WHERE status = 'Emprestado';
-- Alterações
RENAME TABLE Funcionario TO Funcionarios;
ALTER TABLE Emprestimo
ADD COLUMN status ENUM('Disponível', 'Emprestado', 'Reservado') DEFAULT
'Emprestado';
ALTER TABLE Categoria
ADD COLUMN descricao TEXT;
ALTER TABLE Livros
MODIFY COLUMN isbn VARCHAR(20);
ALTER TABLE Leitura
MODIFY COLUMN status ENUM('Em andamento', 'Concluída', 'Abandonada')
DEFAULT 'Em andamento';
ALTER TABLE Leitura
MODIFY COLUMN data_fim DATE NULL;
ALTER TABLE Emprestimo
MODIFY COLUMN Data_devolução_real DATE NULL;
-- Inserção de Registros
INSERT INTO Categoria (nome, descricao) VALUES
('Ficção', 'Livros de ficção e literatura'),
('Não-ficção', 'Livros baseados em fatos reais e temas diversos'),
('Ciência', 'Livros sobre temas científicos e pesquisa'),
('História', 'Livros de história e fatos históricos'),
('Tecnologia', 'Livros sobre tecnologia e inovação'),
('Arte', 'Livros de artes visuais e performáticas'),
('Filosofia', 'Livros de filosofia e pensamento crítico'),
('Psicologia', 'Livros sobre psicologia e comportamento humano'),
('Religião', 'Livros sobre temas religiosos e espirituais'),
('Autoajuda', 'Livros de autoaperfeiçoamento e desenvolvimento pessoal'),
('Educação', 'Livros sobre métodos educacionais e ensino'),
('Literatura Infantil', 'Livros voltados para o público infantil'),
('Biografia', 'Livros que narram a vida de uma pessoa'),
('Poesia', 'Livros de poesia e poemas'),
('Saúde', 'Livros sobre medicina, saúde e bem-estar'),
('Direito', 'Livros sobre leis e temas jurídicos'),
('Administração', 'Livros sobre negócios e administração'),
('Economia', 'Livros de economia e finanças'),
('Matemática', 'Livros sobre matemática e estatística'),
('Ciência Política', 'Livros sobre política e governança'),

```

('Geografia', 'Livros sobre geografia e estudos territoriais');

-- Inserção de dados tabela Editora.

```
INSERT INTO Editora (nome, endereco, email, telefone) VALUES
('A esfera dos livros', 'R. Professor Reinaldo dos Santos Nº42 Lisboa',
'livros@esferadoslivros.pt', '+351-21-340-4060'),
('SHEDD Publicações', 'Rua Carlos Gomes, 993 Santo Amaro-São
Paulo', 'shedd@sheddpublishing.com.br', '11-3223-6594'),
('Editora Esperança', '', 'pedidos@editoraesperanca.com.br', '41-3022-3390'),
('Editora Vida', 'Rua Conde de Sarzedas, 246 São
Paulo', 'atendimento@editoravida.com.br', '11-2618-7000'),
('Plural Editores', 'Centro de Logística Talatona, Loja C3, Belas', '', '924-351-990'),
('União dos Escritores Angolanos', '', '928-295-805'),
('Santos Editora', 'Rua Cândido Lopes 342, Centro
Curitiba', 'vendas@adsantos.com.br', '41-99790-9180'),
('Thomas Nelson Brasil', 'Pedro Pinhares
Gomes, 3990', 'vendas@thomasonelson.com.br', '');
```

-- Inserção tabela Livros

```
INSERT INTO Livros (titulo, autor, ano_publicacao, quantidade, isbn, id_categoria,
id_editora) VALUES
('Nascido para Triunfar', 'Adelino Cunha', '2012', '4', '9-789896-261528', '10', '1'),
('Casamento Blindado', 'Renato e Cristiane Cardoso', '2012', '6', '9-788578-
609443', '10', '8'),
('Socorro!!! Tenho que preparar um sermão', 'Cesar Cezar', '2010', '200', '85-7459-063-
0', '9', '7'),
('No País Chamado Árvores E Outros Contos', 'Giuseppe Mistretta', '2010', '1', '1', '6'),
('Ensina o Teu Filho a Estudar', 'Renato Paiva', '2015', '18', '9-78-9896-263683', '10', '1'),
('Lei Geral do Trabalho', 'Coleção Legislação Angola', '2019', '156', '978-989-611-486-
2', '16', '5'),
('Poder para mudar Sua Vida', 'Rick Warren', '2007', '123', '978-85-7367-987-8', '9', '4'),
('Mais de Mil Esboços Bíblicos de Gênesis a Apocalipse', 'Georg
Brinke', '2017', '15', '978-85-86249-53-2', '9', '3'),
('Esboços Bíblicos de Salmos', 'C.H. Spurgeon', '2018', '57', '978-85-88315-49-5', '9', '2'),
('Grandes Dúvidas Da Língua Portuguesa', 'Elsa Rodrigues dos Santos', '2012', '23', '9-
789896-263171', '2', '1');
```

-- Inserção tabela

```
INSERT INTO Funcionarios (nome, cargo, data_contratacao, telefone, email)
VALUES
('Carla Silva', 'Contadora', '2018-05-27', '923-728-899', 'carlasil12va@gmail.com'),
('Raquel Dimane', 'Bibliotecária', '2016-04-16', '923-758-
876', 'raqueldinmea@gmail.com'),
('Luis Guilherme', 'Contador', '2016-06-22', '924-218-
570', 'guilhermeluis123@gmail.com'),
('Hazea Lemos', 'Bibliotecária Chefe', '2018-02-12', '923-504-
399', 'hazealaluisa@gmail.com');
```

```

('Carla Camilo','Bibliotecária','2019-06-25','941-234-675','carla@camilo@gmail.com'),
('Igor Teixeira','Auxiliar de Biblioteca','2020-01-15','901-231-223','teixeiraigor@gmail.com'),
('Mateus Cazunga','Assistente de Processamento','2021-03-10','mateus29cazunga2@gmail.com'),
('Leopoldo Piedade','Responsável pelo Acervo Digital','2021-11-30','987-546-122','leopoldopiedade80@gmail.com'),
('Fernanda Silva','Contadora','2016-05-22','923-768-549','fernandasilva@gmail.com');

```

-- Inserção tabela Leitores.

```

INSERT INTO Leitores(nome, email, telefone, endereco, data_cadastro) VALUES
('Dinéttria Lemos Camilo','dinettrialemos16@gmail.com','997-501-124','Condomínio Planalto do Kinu','2020-06-12'),
('Yani de Jesus','dejesus13@gmail.com','928-234-198','Kilamba','2020-03-19'),
('Edmilton Dário','darioedmilton@gmail.com','998-332-342','Sequele','2021-02-12'),
('Richel Alves','richelalves16@gmail.com','912-454-112','Condomínio Kuditemo','2022-07-16'),
('Kassio Neves','neves2312gmail.com','923-284-983','Cassenda','2023-08-18'),
('Sandro Lopes','lopeslpoes@gmail.com','990-226-554','Jardim de Rosas','2024-11-06');

```

-- INSERÇÃO TABELA LEITURA

```

INSERT INTO Leitura(id_leitor, id_livro, id_funcionario, data_inicio, data_fim, status) VALUES
('4','11','2','2024-11-02',NULL,'Em andamento'),
('5','14','2','2024-10-04','2024-10-27','Concluída'),
('1','17','2','2024-11-07',NULL,'Em andamento'),
('5','13','2','2024-10-30',NULL,'Em andamento'),
('3','20','2','2024-09-12',NULL,'Abandonada');

```

-- inserção tabela Emprestimo

```

INSERT INTO Emprestimo(ID_livro, ID_leitor, data_emprestimo, data_devolucao, Data_devolução_real, id_funcionario, status) VALUES
('12','6','2024-11-04','2024-12-24',NULL,'3','Emprestado'),
('17','2','2024-10-16','2024-11-20',NULL,'2','Emprestado');
INSERT INTO Livro_Emprestimo(id_livro, id_emprestimo) VALUES
('12','3'),
('17','4');

```

-- Consultas INNER-JOIN

-- Apresenta os livros emprestados, leitores e funcionários

```

SELECT
    L.titulo AS Livro,
    LE.nome AS Leitor,
    F.nome AS Funcionario,
    E.data_emprestimo AS DataEmprestimo,

```

```

    E.data_devolucao AS DataDevolucao
FROM Emprestimo E
INNER JOIN Livros L ON E.ID_livro = L.id_livro
INNER JOIN Leitores LE ON E.ID_leitor = LE.id_leitor
INNER JOIN Funcionarios F ON E.id_funcionario = F.id_funcionario;

-- Apresenta Leitor,Livro,Status,data do inicio e fim
SELECT
    LE.nome AS Leitor,
    L.titulo AS Livro,
    LT.status AS StatusLeitura,
    LT.data_inicio AS DataInicio,
    LT.data_fim AS DataFim
FROM Leitura LT
INNER JOIN Livros L ON LT.id_livro = L.id_livro
INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor;

-- Livro com sua categoria e editora
SELECT
    L.titulo AS Livro,
    C.nome AS Categoria,
    E.nome AS Editora
FROM Livros L
INNER JOIN Categoria C ON L.id_categoria = C.id_categoria
INNER JOIN Editora E ON L.id_editora = E.id_editora;

-- Funcionários que registraram
SELECT
    F.nome AS Funcionario,
    LE.nome AS Leitor,
    L.titulo AS Livro,
    LT.data_inicio AS DataInicio,
    LT.data_fim AS DataFim
FROM Leitura LT
INNER JOIN Funcionarios F ON LT.id_funcionario = F.id_funcionario
INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor
INNER JOIN Livros L ON LT.id_livro = L.id_livro;

-- Leitores que concluíram leituras
SELECT
    LE.nome AS Leitor,
    L.titulo AS Livro,
    LT.data_inicio AS DataInicio,
    LT.data_fim AS DataFim,
    F.nome AS Funcionario
FROM Leitura LT

```

```

INNER JOIN Leitores LE ON LT.id_leitor = LE.id_leitor
INNER JOIN Livros L ON LT.id_livro = L.id_livro
INNER JOIN Funcionarios F ON LT.id_funcionario = F.id_funcionario
WHERE LT.status = 'Concluída';

-- CRIAÇÃO DE PROCESSOS ARMAZENADOS
DELIMITER //

CREATE PROCEDURE InserirLivro(
    IN titulo VARCHAR(255),
    IN autor VARCHAR(255),
    IN ano_publicacao YEAR,
    IN quantidade INT,
    IN isbn VARCHAR(20),
    IN id_categoria INT,
    IN id_editora INT
)
BEGIN
    INSERT INTO Livros (titulo, autor, ano_publicacao, quantidade, isbn,
id_categoria, id_editora)
    VALUES (titulo, autor, ano_publicacao, quantidade, isbn, id_categoria,
id_editora);
END //

DELIMITER ;

CALL InserirLivro(
    'Doze semanas para mudar uma vida',
    'Augusto Cury',
    2012,
    7,
    '978-989-678-052-2',
    10,
    1
);

DELIMITER //

DELIMITER //

CREATE PROCEDURE InserirEmprestimo(
    IN id_livro INT,
    IN id_leitor INT,
    IN id_funcionario INT,
    IN data_emprestimo DATE,
    IN data_devolucao DATE,
    IN status ENUM('Emprestado', 'Devolvido', 'Atrasado')

```

```
)
BEGIN
    INSERT INTO Emprestimo (id_livro, id_leitor, id_funcionario, data_emprestimo,
data_devolucao, status)
    VALUES (id_livro, id_leitor, id_funcionario, data_emprestimo, data_devolucao,
status);
END //
```

```
DELIMITER ;
```

```
CALL InserirEmprestimo(3, 2, 1, '2024-11-30', '2024-12-15', 'Emprestado');
```

```
-- Criação de Views
```

```
CREATE VIEW Leituras_Por_Leitor AS
SELECT
    LT.nome AS Leitor,
    L.titulo AS Livro,
    LD.data_inicio,
    LD.data_fim,
    LD.status AS StatusLeitura
FROM
    Leitura LD
JOIN
    Livros L ON LD.id_livro = L.id_livro
JOIN
    Leitores LT ON LD.id_leitor = LT.id_leitor;
```

```
CREATE VIEW Livros_Categorias AS
SELECT
    L.id_livro,
    L.titulo AS Livro,
    C.nome AS Categoria
FROM
    Livros L
JOIN
    Categoria C ON L.id_categoria = C.id_categoria;
```

```
-- Visualizar os funcionários e suas atividades relacionadas a empréstimos.
```

```
CREATE VIEW Funcionarios_Atividades AS
SELECT
    F.id_funcionario,
    F.nome AS Funcionario,
    COUNT(E.id_emprestimo) AS TotalEmprestimos
FROM
```

```

    Funcionarios F
LEFT JOIN
    Emprestimo E ON F.id_funcionario = E.id_funcionario
GROUP BY
    F.id_funcionario, F.nome;

-- Mostra todos os empréstimos em andamento

CREATE VIEW Emprestimos_Atuais AS
SELECT
    E.id_emprestimo,
    L.titulo AS Livro,
    LT.nome AS Leitor,
    E.data_emprestimo,
    E.data_devolucao,
    E.status
FROM
    Emprestimo E
JOIN
    Livros L ON E.id_livro = L.id_livro
JOIN
    Leitores LT ON E.id_leitor = LT.id_leitor
WHERE
    E.status = 'Emprestado';

-- Consultas Views

SELECT * FROM Leituras_Por_Leitor;

SELECT * FROM Livros_Categorias;

SELECT * FROM Funcionarios_Atividades;

SELECT * FROM Emprestimos_Atuais;

-- Consultas com Order By

-- Consulta por ordem alfabetica livros
SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
FROM Livros
ORDER BY titulo ASC;

-- consulta por ordem alfabetica leitores
SELECT id_leitor, nome, email, telefone, endereco, data_cadastro

```



```
FROM Leitores
ORDER BY nome ASC;
```

```
-- consulta por ordem alfabetica Descendentes leitores
SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
FROM Livros
ORDER BY titulo DESC;
```

```
-- Consultas com like
```

```
-- Buscar livros cujo título contém a palavra 'Vida'
```

```
SELECT id_livro, titulo, autor, ano_publicacao, quantidade, isbn
FROM Livros
WHERE titulo LIKE '%vida%';
```

```
-- Buscar leitores cujo nome comece com "M"
```

```
SELECT id_leitor, nome
FROM Leitores
WHERE nome LIKE 'D%';
```

```
-- Consultas com BETWEEN
```

```
-- Lista de leitores cujo número id_leitor está entre 5 e 15, inclusivo.
```

```
SELECT id_leitor, nome, telefone
FROM Leitores
WHERE id_leitor BETWEEN 5 AND 15;
```

```
-- consulta com NOT IN
```

```
-- Retorna todos os livros que não pertencem às categorias 2, 5, ou 9.
```

```
SELECT id_livro, titulo, id_categoria
FROM Livros
WHERE id_categoria NOT IN (2, 5, 9);
```

```
-- Exclui os funcionários com os IDs 1, 4, e 6.
```

```
SELECT id_funcionario, nome, cargo
FROM Funcionarios
WHERE id_funcionario NOT IN (1, 4, 6);
```

```
-- consulta com IN
```

```
-- Retorna apenas os livros que pertencem às categorias 3, 7, ou 10.
```

```
SELECT id_livro, titulo, id_categoria
FROM Livros
WHERE id_categoria IN (3, 7, 10);
```