

#### 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](#_Toc187098937)

[1.2. Introdução 5](#_Toc187098938)

[1.4. Entregáveis 5](#_Toc187098939)

[1.5. Metodologia 5](#_Toc187098940)

[1.6. Requisitos 5](#_Toc187098941)

[Requisitos Funcionais: 5](#_Toc187098942)

[Requisitos Não Funcionais: 5](#_Toc187098943)

[1.7. Sprints 6](#_Toc187098944)

[1.8. Critérios de Aceitação 6](#_Toc187098945)

[2.1. Arquitetura do Banco de Dados 8](#_Toc187098946)

[Modelo Conceitual: 8](#_Toc187098947)

[Entidades principais: 8](#_Toc187098948)

[Diagrama UML 8](#_Toc187098949)

[Modelo Lógico: 9](#_Toc187098950)

[Tabelas e relacionamentos: 9](#_Toc187098951)

[2.2. Tecnologias e Ferramentas 12](#_Toc187098952)

[2.3.Procedimentos Armazenados 13](#_Toc187098953)

[2.4.Consultas com Joins 15](#_Toc187098954)

[2.5.Criação de Views 17](#_Toc187098955)

[2.6.Consultas com Order by 19](#_Toc187098956)

[2.7.Consultas com Like 19](#_Toc187098957)

[2.4.Between(Not in, In) 20](#_Toc187098958)

[2.5. Plano de Testes 21](#_Toc187098959)

[2.6. Plano de Manutenção 21](#_Toc187098960)

[APENDICE: A 22](#_Toc187098961)

**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**

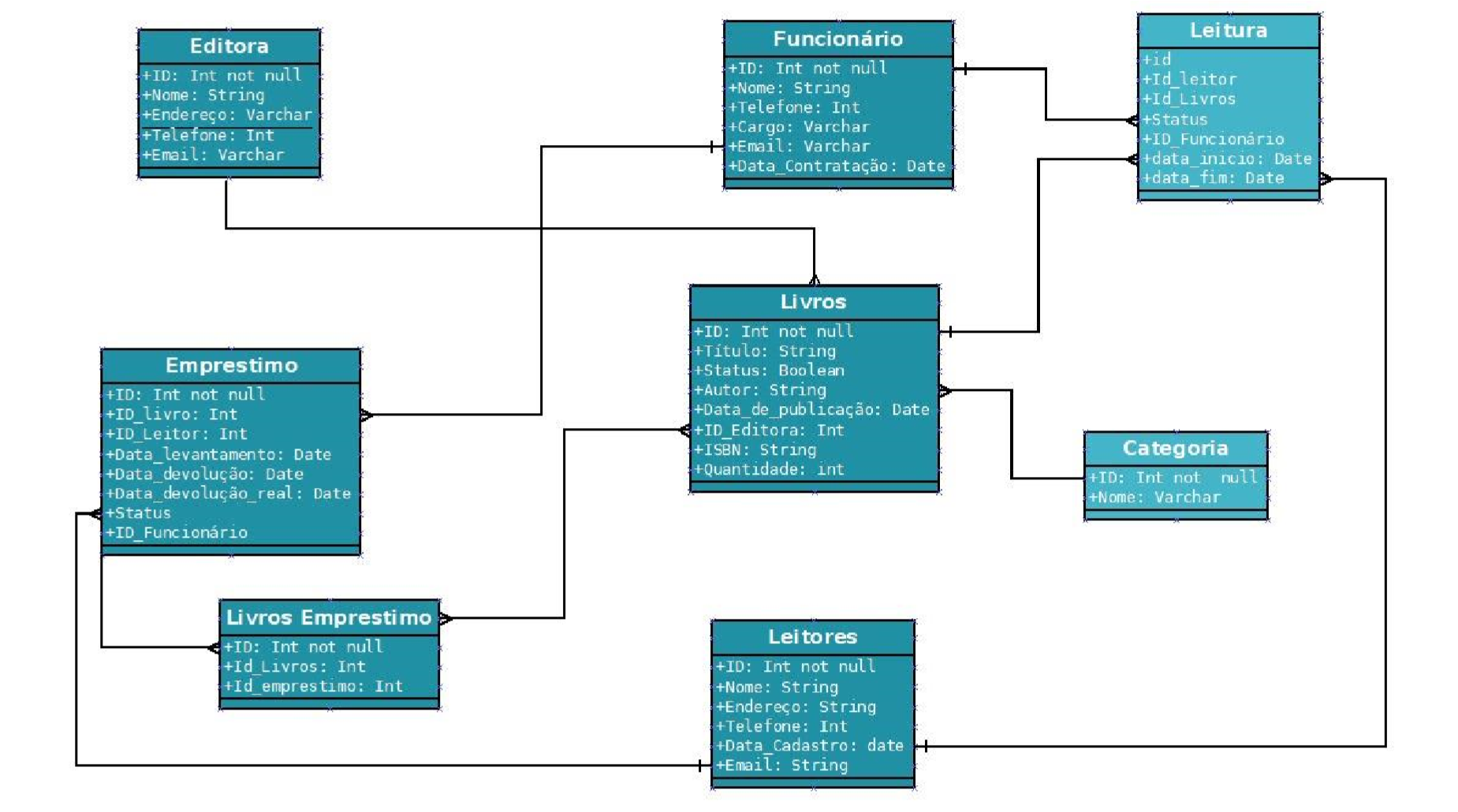
# 2.1. Arquitetura do Banco de Dados

# 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, endereco, 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 emprestimo. |
| 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

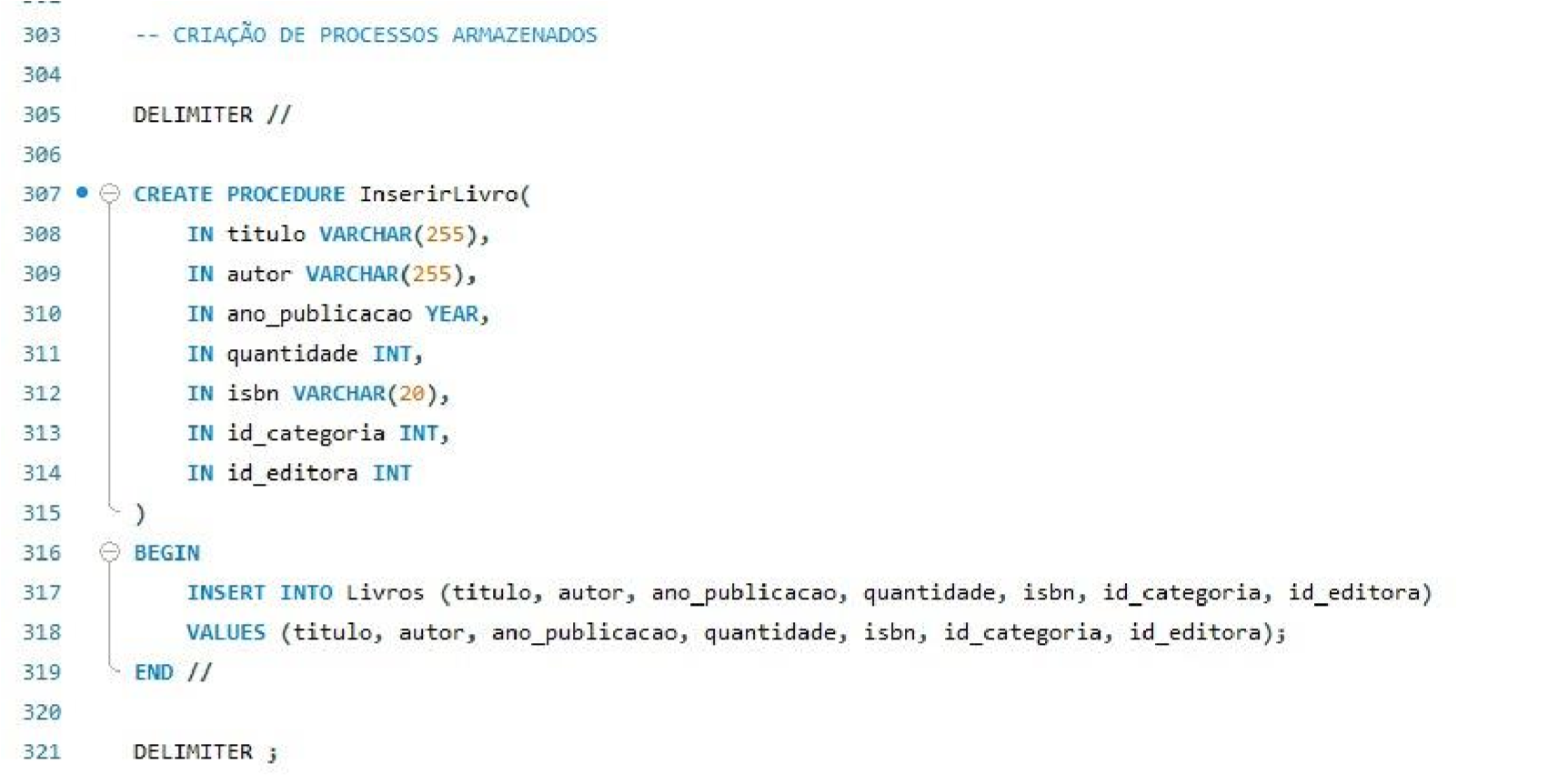
# 2.2. Tecnologias e Ferramentas

Banco de Dados: **MySQL.**

Controle de Versão: **Git.**

**Ferramentas Adicionais:** MySQL Workbench, GitHub ou GitLab.

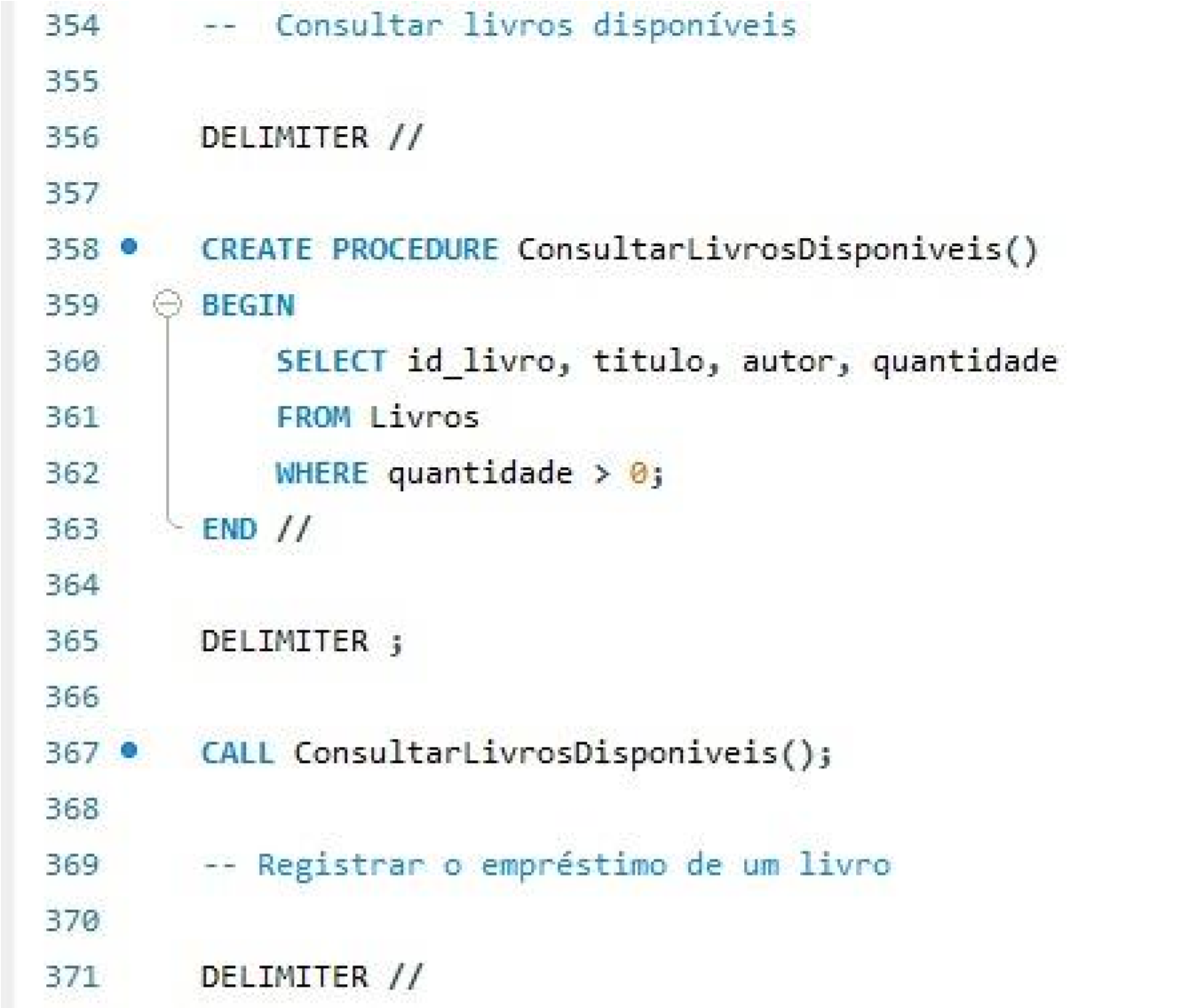
# 2.3.Procedimentos Armazenados



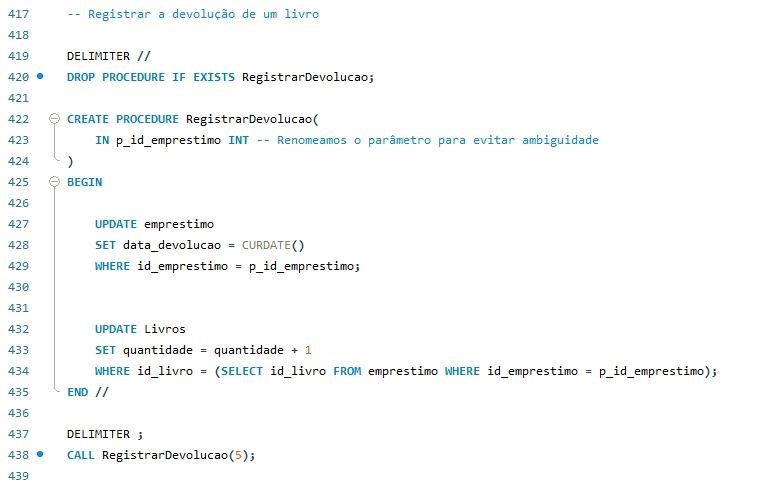
#### Procedimento para inserção de livros



Procedimento para cadastrar um empréstimo

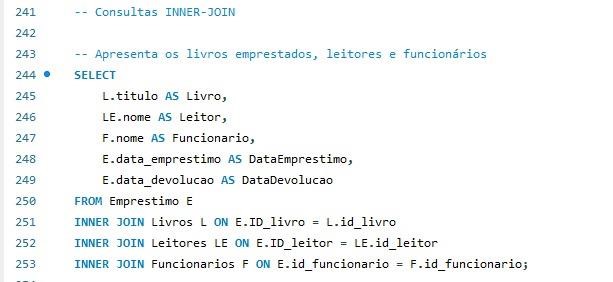


#### Procedimento para consultar os livros disponíveis

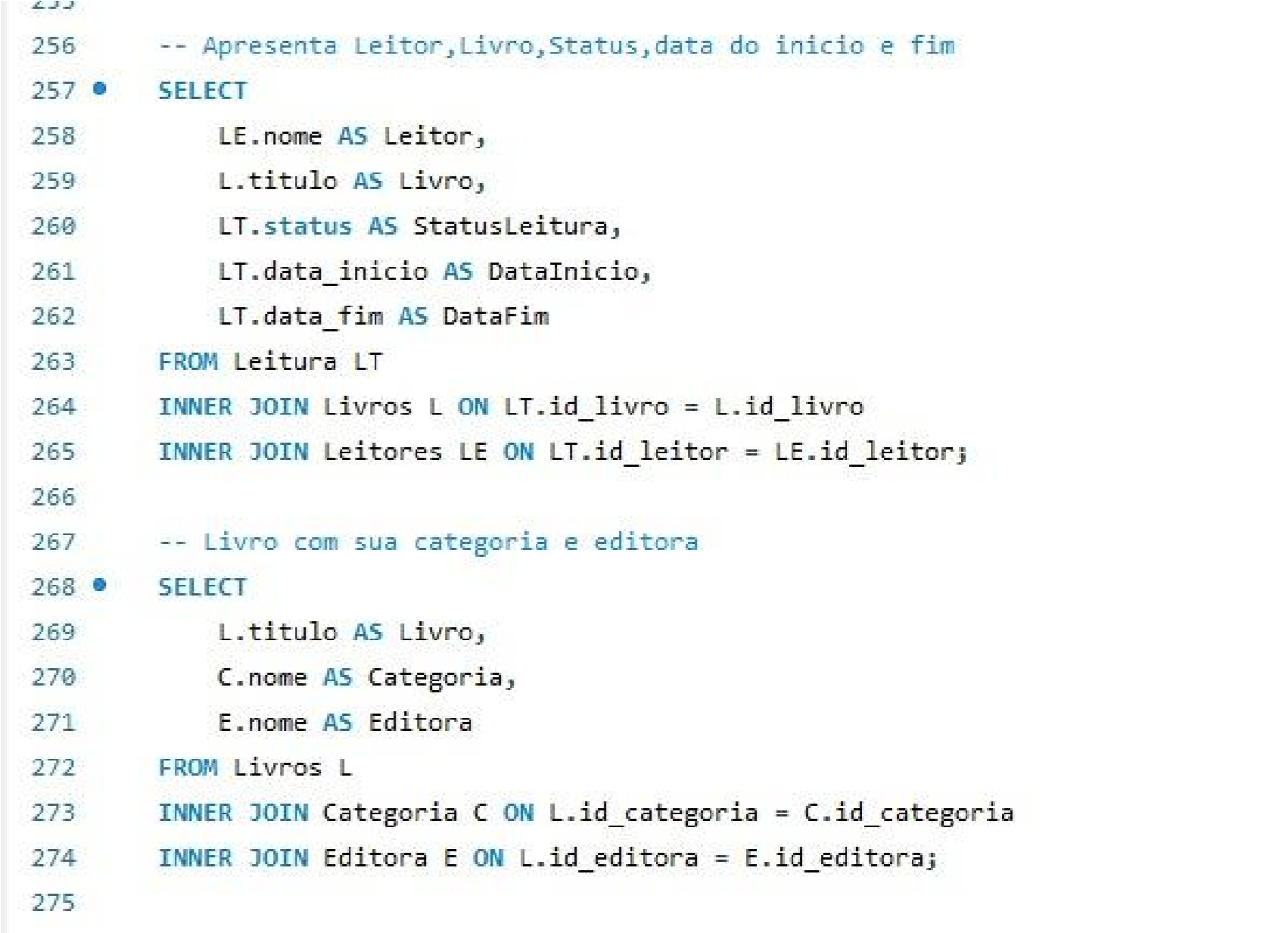


#### Procedimento para registar a devolução de livros

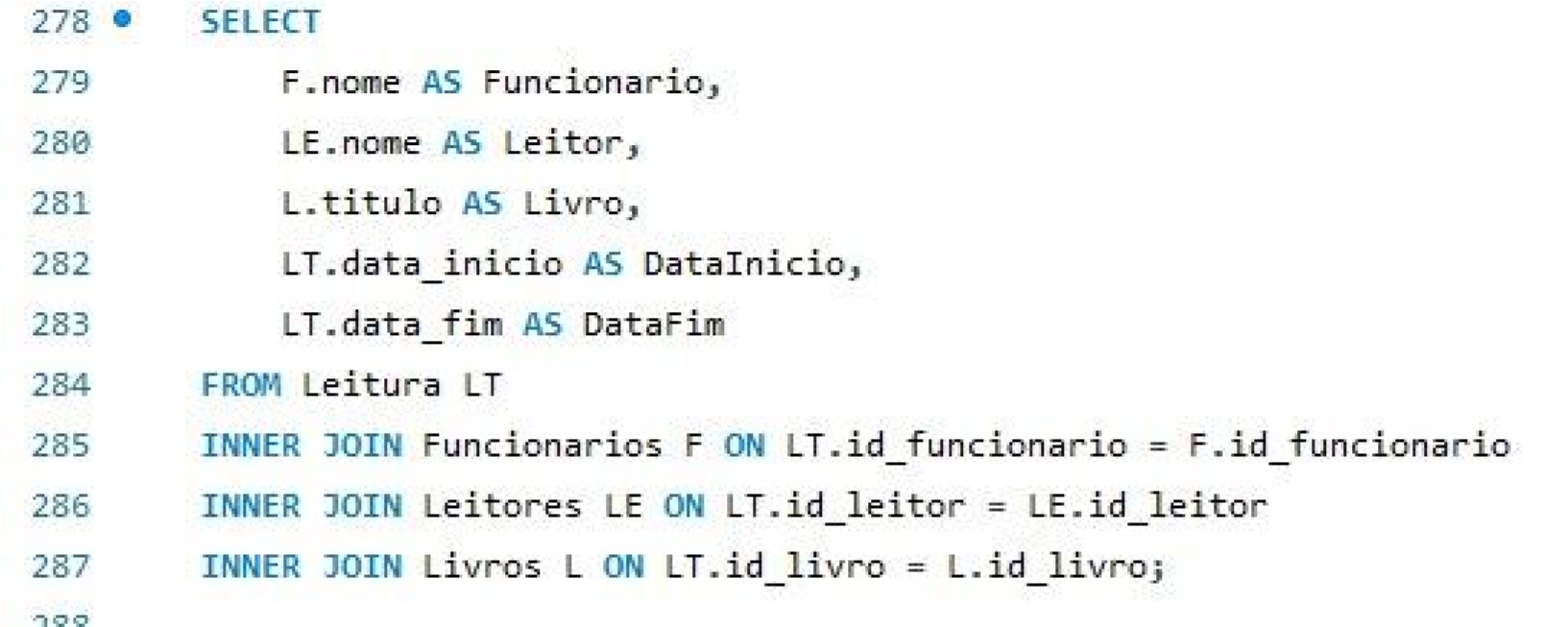
# 2.4.Consultas com Joins



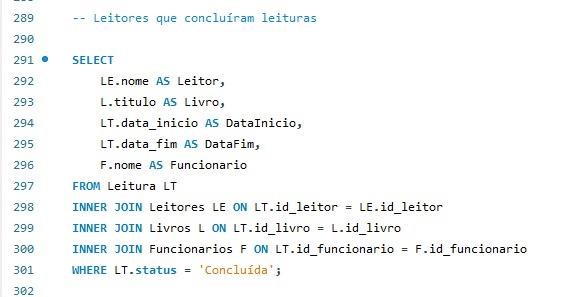
#### Apresenta os livros emprestados, leitores e funcionários



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

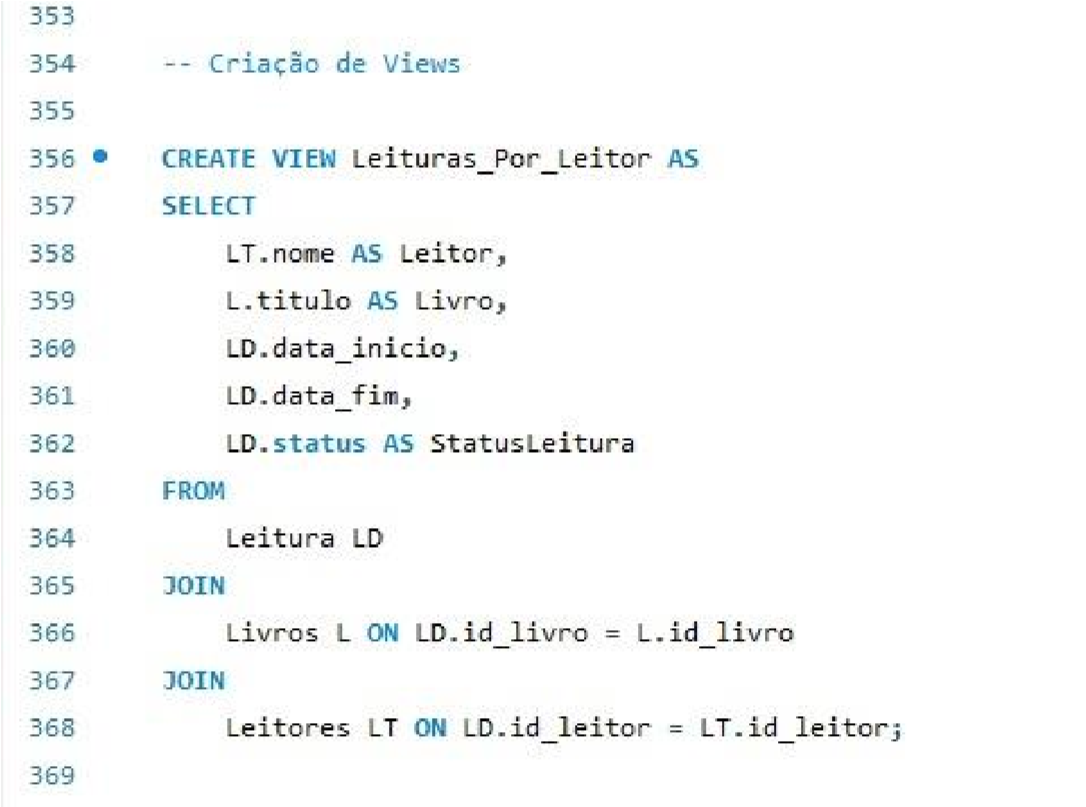


#### Funcionários que registraram

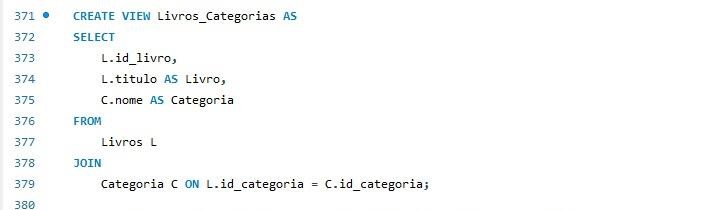


#### Leitores que concluíram leituras

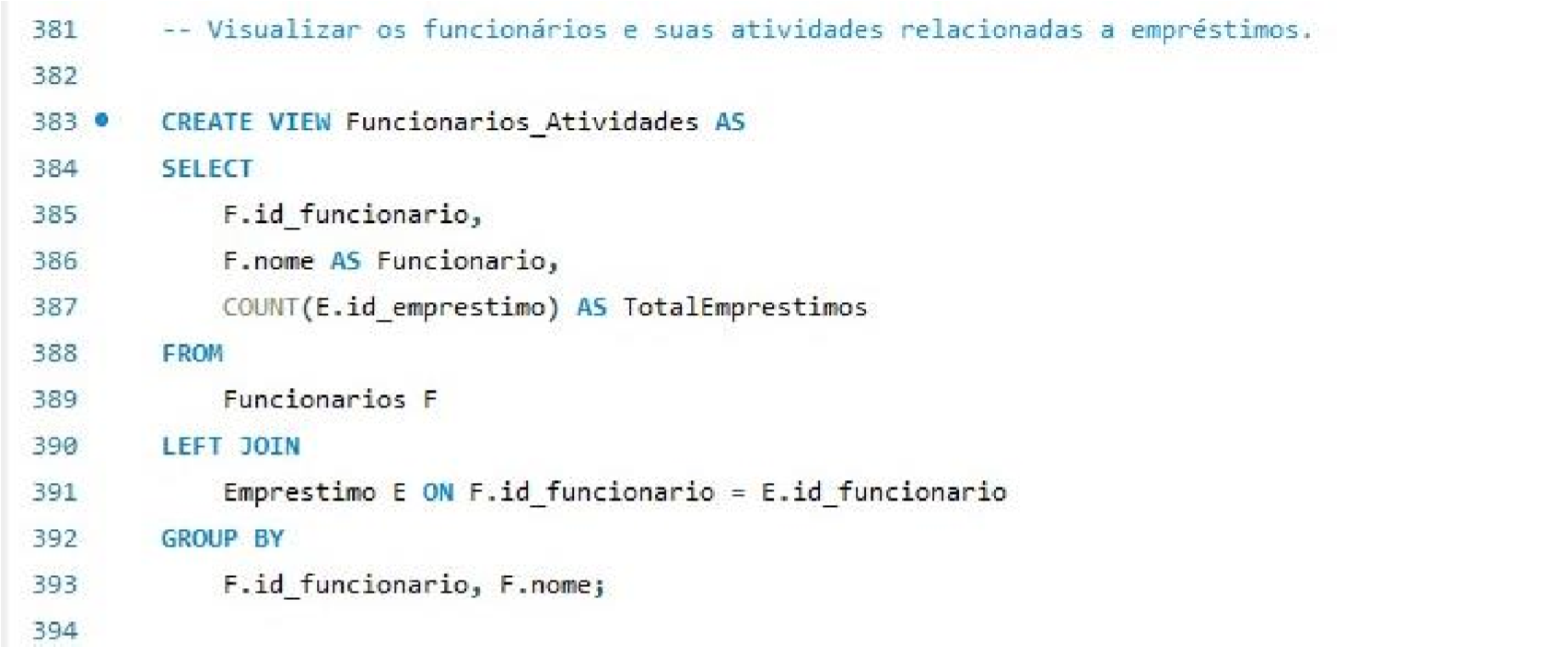
# 2.5.Criação de Views



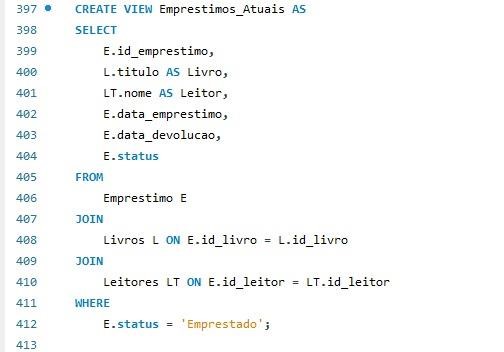
#### Apresenta os leitores e o status da leitura



#### Apresenta os livros e a categoria.



Apresenta os funcionários e suas atividades relacionadas a empréstimos.



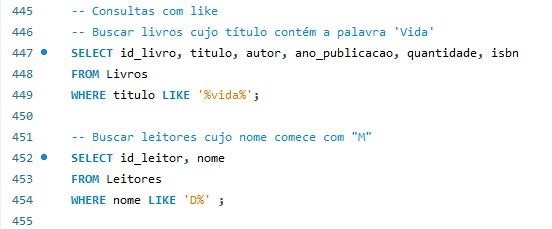
Apresenta os empréstimos feitos

# 2.6.Consultas com Order by

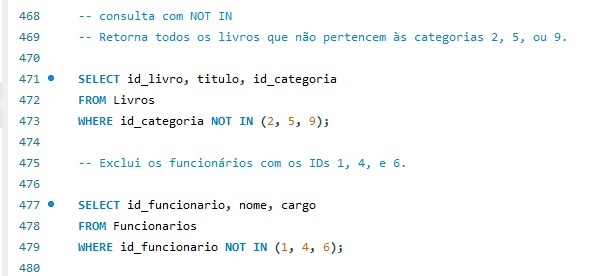


#### Consultas com order by por ordem alfabética

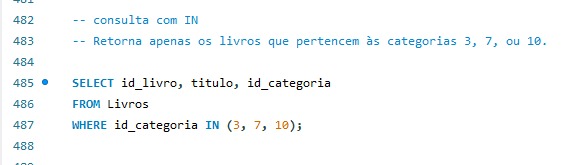
# 2.7.Consultas com Like



# 2.4.Between(Not in, In)







## 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.Professsor Reinaldo dos Santos Nº42 Lisboa', 'livros@esferadoslivros.pt','+351-21-340-4060'),

('SHEDD Publicações','Rua Carlos Gomes,993 Santo Amaro-São Paulo','shedd@sheddpublicaaoes.com.br','11-3223-6594'),

('Editora Esperança','','pedidos@editoraesperanca.com.br','41-3022-3390'),

('Editora Vida','Rua Conde de Sarzedas,246 São Pauol','atendimento@editoravida.com.br','11-2618-7000'),

('Plural Editores','Centro de Logistica Talatona,LojaC3,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@thomasnelson.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!!! Teenho 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','Colecção 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 Esbolços Bíblicos de Gênesis a Apocalipse','Georg Brinke','2017','15','978-85-86249-53-2','9','3'),

('Esbolços 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'),

('Hazaela Lemos','Bibliotecária Chefe','2018-02-12','923-504-399','hazaelaluisa@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','leopoldopiedado80@gmail.com'),

('Fernanda Silva','Contadora','2016-05-22','923-768-549','fernandasilva@gmail.com');

-- Inserção tabela Leitors.

INSERT INTO Leitores(nome, email, telefone, endereco, data\_cadastro) VALUES

('Dinétria Lemos Camilo','dinetrialemos16@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);