**Parte 1: Modelo Conceitual (Diagrama Entidade-Relacionamento - DER)**

O primeiro passo é identificar as entidades (os "objetos" principais do sistema), seus atributos (as características de cada entidade) e como elas se relacionam.

**Entidades Identificadas:**

* **LIVRO:** Armazena as informações gerais de cada obra.
* **AUTOR:** Armazena os dados dos autores.
* **EXEMPLAR:** Representa cada cópia física de um livro que a biblioteca possui.
* **MEMBRO:** Armazena os dados dos usuários da biblioteca.
* **EMPRÉSTIMO:** Registra a ação de um membro pegar um exemplar emprestado.

**Relacionamentos:**

* Um **LIVRO** pode ser escrito por **vários AUTORES**, e um **AUTOR** pode escrever **vários LIVROS**. Isso é um relacionamento **N:N (muitos-para-muitos)**. Para resolver isso, criamos uma entidade associativa chamada LIVRO\_AUTOR.
* Um **LIVRO** pode ter **vários EXEMPLARES**.
* Um **MEMBRO** pode realizar **vários EMPRÉSTIMOS**.
* Um **EXEMPLAR** pode ser pego em **vários EMPRÉSTIMOS** (ao longo do tempo).

**Diagrama (representação textual):**

Plain Text

[LIVRO] --1..N-- (possui) --1..1-- [EXEMPLAR]

| |

| |

(é escrito por) N..N (é pego em) 1..N

| |

| |

[AUTOR] [EMPRÉSTIMO] --N..1-- (realiza) -- [MEMBRO]

**Parte 2: Modelo Lógico (Esquema Relacional)**

Agora, convertemos o DER em tabelas, definindo as colunas, Chaves Primárias (PK) e Chaves Estrangeiras (FK).

* **AUTORES**
  + id\_autor (PK, INT, AUTO\_INCREMENT)
  + nome (VARCHAR(255))
  + nacionalidade (VARCHAR(100))
* **LIVROS**
  + isbn (PK, VARCHAR(13))
  + titulo (VARCHAR(255))
  + ano\_publicacao (YEAR)
  + editora (VARCHAR(150))
* **LIVRO\_AUTOR** (Tabela associativa para resolver o N:N)
  + fk\_livro\_isbn (FK para LIVROS.isbn)
  + fk\_autor\_id (FK para AUTORES.id\_autor)
  + *Chave Primária Composta: (fk\_livro\_isbn, fk\_autor\_id)*
* **EXEMPLARES**
  + codigo\_patrimonio (PK, VARCHAR(50))
  + status (ENUM('Disponível', 'Emprestado', 'Em Manutenção'))
  + fk\_livro\_isbn (FK para LIVROS.isbn)
* **MEMBROS**
  + matricula (PK, INT, AUTO\_INCREMENT)
  + nome\_completo (VARCHAR(255))
  + endereco (VARCHAR(255))
  + telefone (VARCHAR(20))
* **EMPRESTIMOS**
  + id\_emprestimo (PK, INT, AUTO\_INCREMENT)
  + data\_emprestimo (DATE)
  + data\_prevista\_devolucao (DATE)
  + data\_devolucao\_real (DATE, pode ser NULL)
  + fk\_membro\_matricula (FK para MEMBROS.matricula)
  + fk\_exemplar\_codigo (FK para EXEMPLARES.codigo\_patrimonio)

**Parte 3: Modelo Físico (Script SQL - DDL)**

Este é o código SQL para criar todas as tabelas e seus relacionamentos no banco de dados.

SQL

*-- Criação da tabela AUTORES*

CREATE TABLE AUTORES (

id\_autor INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

nacionalidade VARCHAR(100)

);

*-- Criação da tabela LIVROS*

CREATE TABLE LIVROS (

isbn VARCHAR(13) PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

ano\_publicacao YEAR,

editora VARCHAR(150)

);

*-- Criação da tabela associativa LIVRO\_AUTOR*

CREATE TABLE LIVRO\_AUTOR (

fk\_livro\_isbn VARCHAR(13) NOT NULL,

fk\_autor\_id INT NOT NULL,

PRIMARY KEY (fk\_livro\_isbn, fk\_autor\_id),

FOREIGN KEY (fk\_livro\_isbn) REFERENCES LIVROS(isbn),

FOREIGN KEY (fk\_autor\_id) REFERENCES AUTORES(id\_autor)

);

*-- Criação da tabela EXEMPLARES*

CREATE TABLE EXEMPLARES (

codigo\_patrimonio VARCHAR(50) PRIMARY KEY,

status ENUM('Disponível', 'Emprestado', 'Em Manutenção') NOT NULL,

fk\_livro\_isbn VARCHAR(13) NOT NULL,

FOREIGN KEY (fk\_livro\_isbn) REFERENCES LIVROS(isbn)

);

*-- Criação da tabela MEMBROS*

CREATE TABLE MEMBROS (

matricula INT PRIMARY KEY AUTO\_INCREMENT,

nome\_completo VARCHAR(255) NOT NULL,

endereco VARCHAR(255),

telefone VARCHAR(20)

);

*-- Criação da tabela EMPRESTIMOS*

CREATE TABLE EMPRESTIMOS (

id\_emprestimo INT PRIMARY KEY AUTO\_INCREMENT,

data\_emprestimo DATE NOT NULL,

data\_prevista\_devolucao DATE NOT NULL,

data\_devolucao\_real DATE,

fk\_membro\_matricula INT NOT NULL,

fk\_exemplar\_codigo VARCHAR(50) NOT NULL,

FOREIGN KEY (fk\_membro\_matricula) REFERENCES MEMBROS(matricula),

FOREIGN KEY (fk\_exemplar\_codigo) REFERENCES EXEMPLARES(codigo\_patrimonio)

);

**Parte 4: Administração e Manutenção do Banco de Dados (DDL e DCL)**

Aqui estão os comandos SQL para as tarefas de administração solicitadas.

1. **Criar usuário estagiario e conceder permissão ALTER:**

SQL

*-- Cria um novo usuário chamado 'estagiario' com a senha 'mudar123'*

CREATE USER 'estagiario'@'localhost' IDENTIFIED BY 'mudar123';

*-- Concede APENAS a permissão para alterar (ALTER) a estrutura da tabela LIVROS*

*-- Substitua 'nome\_do\_banco' pelo nome real do seu banco de dados*

GRANT ALTER ON nome\_do\_banco.LIVROS TO 'estagiario'@'localhost';

1. **Adicionar a coluna genero na tabela LIVROS:**

SQL

*-- Adiciona uma nova coluna chamada 'genero' na tabela LIVROS*

ALTER TABLE LIVROS

ADD COLUMN genero VARCHAR(50);

1. **Exportar o banco de dados:** Para esta etapa, você usaria uma ferramenta como o **MySQL Workbench**. O processo geralmente é:
   * No menu superior, vá em **Server > Data Export**.
   * Na tela que abrir, selecione o banco de dados da biblioteca (nome\_do\_banco) na lista à esquerda.
   * Certifique-se de que a opção **"Export to Self-Contained File"** esteja marcada.
   * Clique no botão **"Start Export"**.
   * Isso irá gerar um único arquivo .sql contendo toda a estrutura e os dados do seu banco, que é o arquivo a ser entregue.