Alexandria

**IA utilizada**: Deepseek

**Pedido feito**:

Estou montando um banco de dados pelo MySQL para um sistema de gerenciamento de biblioteca. No momento, o código está da seguinte forma:

create database ALEXANDRIA;

use ALEXANDRIA;

CREATE TABLE TIPO\_USUARIO (

pk\_tipo\_user INT PRIMARY KEY auto\_increment,

tipo\_user\_nome VARCHAR(50) not null,

tipo\_user\_descricao VARCHAR(300)

);

CREATE TABLE STATUS\_MEMBRO (

pk\_status\_mem INT PRIMARY KEY auto\_increment,

status\_mem\_nome VARCHAR(50) not null,

status\_mem\_descricao VARCHAR(200)

);

CREATE TABLE CATEGORIA (

pk\_cat INT PRIMARY KEY auto\_increment,

cat\_nome VARCHAR(100) not null

);

CREATE TABLE AUTOR (

pk\_aut INT PRIMARY KEY auto\_increment,

aut\_nome VARCHAR(100) not null,

aut\_sobrenome VARCHAR(150) not null,

aut\_data\_nascimento DATE

);

CREATE TABLE FORNECEDOR (

pk\_forn INT PRIMARY KEY auto\_increment,

forn\_nome VARCHAR(250) not null,

forn\_cnpj VARCHAR(18) unique,

forn\_telefone VARCHAR(16),

forn\_email VARCHAR(100),

forn\_endereco VARCHAR(250)

);

CREATE TABLE LIVRO (

pk\_liv INT PRIMARY KEY auto\_increment,

liv\_titulo VARCHAR(200) not null,

liv\_isnb VARCHAR(13) not null unique,

liv\_edicao INT,

liv\_anoPublicacao YEAR,

liv\_sinopse VARCHAR(3000),

liv\_estoque INT,

liv\_dataAlteracaoEstoque DATE,

liv\_idioma VARCHAR(30),

liv\_num\_paginas INT,

liv\_capa VARCHAR(255)

);

CREATE TABLE PLANO (

pk\_plan INT PRIMARY KEY auto\_increment,

plan\_nome VARCHAR(100) not null,

plan\_valor DECIMAL(10,2),

plan\_duracao VARCHAR(50),

plan\_descricao VARCHAR(3000),

plan\_limite\_emp INT default 2

);

CREATE TABLE USUARIO (

pk\_user INT PRIMARY KEY auto\_increment,

user\_nome VARCHAR(250) not null,

user\_cpf VARCHAR(14) unique not null,

user\_email VARCHAR(250),

user\_telefone VARCHAR(16),

user\_senha VARCHAR(255) not null,

user\_login VARCHAR(20) unique,

user\_dataAdmissao DATE,

user\_dataDemissao DATE,

user\_foto VARCHAR(255),

fk\_tipoUser INT not null,

FOREIGN KEY(fk\_tipoUser) references TIPO\_USUARIO(pk\_tipo\_user) on delete restrict on update cascade

);

CREATE TABLE MEMBRO (

pk\_mem INT PRIMARY KEY auto\_increment,

mem\_nome VARCHAR(250) not null,

mem\_cpf VARCHAR(14) unique not null,

mem\_senha VARCHAR(250) not null,

mem\_email VARCHAR(250),

mem\_telefone VARCHAR(16),

fk\_status INT not null,

fk\_plan INT,

FOREIGN KEY(fk\_status) references STATUS\_MEMBRO(pk\_status\_mem) on delete restrict on update cascade,

FOREIGN KEY(fk\_plan) references PLANO(pk\_plan) on delete restrict on update cascade

);

CREATE TABLE EMPRESTIMO (

pk\_emp INT PRIMARY KEY auto\_increment,

emp\_prazo INT,

emp\_dataEmp DATE,

emp\_dataDev DATE,

emp\_dataDevReal DATE,

emp\_valorMultaDiaria DECIMAL(10,2) default 1.50,

emp\_status ENUM('Empréstimo Ativo', 'Empréstimo Atrasado', 'Renovação Ativa', 'Renovação Atrasada', 'Finalizado') not null default 'Empréstimo Ativo',

fk\_mem INT not null,

fk\_user INT not null,

FOREIGN KEY(fk\_mem) references MEMBRO(pk\_mem) on delete restrict on update cascade,

FOREIGN KEY(fk\_user) references USUARIO(pk\_user) on delete restrict on update cascade

);

CREATE TABLE RESERVA (

pk\_res INT PRIMARY KEY auto\_increment,

res\_prazo INT,

res\_dataMarcada DATE,

res\_dataVencimento DATE,

res\_dataFinalizada DATE,

res\_observacoes VARCHAR(1000),

res\_status ENUM('Aberta', 'Cancelada', 'Finalizada', 'Atrasada') not null default 'Aberta',

fk\_mem INT not null,

fk\_liv INT not null,

fk\_user INT not null,

FOREIGN KEY(fk\_mem) references MEMBRO(pk\_mem) on delete restrict on update cascade,

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade,

FOREIGN KEY(fk\_user) references USUARIO(pk\_user) on delete restrict on update cascade

);

CREATE TABLE Multa (

pk\_mul INT PRIMARY KEY auto\_increment,

mul\_valor DECIMAL(10,2),

mul\_qtdDias INT,

mul\_status ENUM('Aberta', 'Finalizada'),

fk\_mem INT not null,

fk\_emp INT not null,

FOREIGN KEY(fk\_mem) references MEMBRO(pk\_mem) on delete restrict on update cascade,

FOREIGN KEY(fk\_emp) references EMPRESTIMO(pk\_emp) on delete restrict on update cascade

);

CREATE TABLE PAG\_PLANO (

pk\_pag\_plan INT PRIMARY KEY auto\_increment,

pag\_plan\_preco DECIMAL(10,2),

pag\_plan\_valorPag DECIMAL(10,2),

pag\_plan\_dataPag DATE,

pag\_plan\_dataVen DATE,

pag\_plan\_comprovante VARCHAR(255),

pag\_plan\_status ENUM('Em dia', 'Atrasado') not null,

fk\_mem INT not null,

fk\_plan INT not null,

FOREIGN KEY(fk\_mem) references MEMBRO(pk\_mem) on delete restrict on update cascade,

FOREIGN KEY(fk\_plan) references PLANO(pk\_plan) on delete restrict on update cascade

);

CREATE TABLE REMESSA (

pk\_rem INT PRIMARY KEY auto\_increment,

rem\_data DATE not null,

rem\_qtd INT not null,

fk\_forn INT not null,

fk\_liv INT not null,

fk\_user INT not null,

FOREIGN KEY(fk\_forn) references FORNECEDOR(pk\_forn) on delete restrict on update cascade,

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade,

FOREIGN KEY(fk\_user) references USUARIO(pk\_user) on delete restrict on update cascade

);

/\*--------------------- TABELAS JUNÇÃO ---------------------\*/

CREATE TABLE FORN\_LIV (

fk\_liv INT,

fk\_forn INT,

PRIMARY KEY(fk\_liv, fk\_forn),

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade,

FOREIGN KEY(fk\_forn) references FORNECEDOR(pk\_forn) on delete restrict on update cascade

);

CREATE TABLE CAT\_LIV (

fk\_liv INT,

fk\_cat INT,

PRIMARY KEY(fk\_liv, fk\_cat),

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade,

FOREIGN KEY(fk\_cat) references CATEGORIA(pk\_cat) on delete restrict on update cascade

);

CREATE TABLE AUT\_LIV (

fk\_liv INT,

fk\_aut INT,

PRIMARY KEY(fk\_liv, fk\_aut),

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade,

FOREIGN KEY(fk\_aut) references AUTOR(pk\_aut) on delete restrict on update cascade

);

CREATE TABLE EMP\_LIV (

fk\_emp INT,

fk\_liv INT,

PRIMARY KEY(fk\_emp, fk\_liv),

FOREIGN KEY(fk\_emp) references EMPRESTIMO(pk\_emp) on delete restrict on update cascade,

FOREIGN KEY(fk\_liv) references LIVRO(pk\_liv) on delete restrict on update cascade

);

Eu gostaria que você inserisse valores para cada uma das tabelas, de forma a conduzir testes no banco de dados criado. É necessário que sejam inseridos ao menos:

2 dados na tabela TIPO\_USUARIO (Administrador e Funcionário), 2 dados na tabela STATUS\_MEMBRO (Ativo e Suspenso), 5 dados na tabela CATEGORIA, 10 dados na tabela AUTOR, 10 dados na tabela FORNECEDOR, 20 dados na tabela LIVRO, 3 dados na tabela USUARIO, 10 dados na tabela MEMBRO, 3 dados na tabela PLANO, e um mínimo de 5 dados nas demais tabelas. Ao responder, coloque já em código SQL (Exemplo: insert into CATEGORIA values (dados);).

Leve em consideração que a tabela USUARIO se refere aos funcionários da biblioteca, a tabela MEMBRO aos membros que solicitam empréstimos de livros, e a tabela PLANO aos planos disponíveis para os membros (Plano Individual ou Plano Família, por exemplo).