

## Entrega 3

Projeto de banco de dados SQL.

Adilson Lima Dario

Turma 70960

## 1. Introdução

Este documento descreve o desenvolvimento e a conclusão da entrega do projeto final, que consiste na modelagem e criação de um banco de dados para o gerenciamento de uma biblioteca. O objetivo principal deste projeto é criar uma estrutura robusta e eficiente para armazenar, organizar e acessar informações cruciais para o funcionamento da biblioteca, como detalhes de livros, autores, temas, clientes e registros de aluguel.

## 2. Objetivo

O objetivo desta entrega é demonstrar toda a base, estrutura, modelo de entidade e relacionamentos, tabelas, views, funções e stored procedures da biblioteca no formato SQL.

Especificamente, busca-se:

- Modelar as principais entidades envolvidas no sistema de uma biblioteca;
- Definir os relacionamentos entre essas entidades de forma clara e precisa;
- Criar um diagrama de entidade-relacionamento (DER) que represente visualmente a estrutura do banco de dados;
- Listar e descrever detalhadamente cada tabela (entidade), seus campos, tipos de dados e tipos de chaves;
- Gerar um script SQL funcional para a criação do banco de dados e suas tabelas, incluindo a definição de chaves primárias e estrangeiras para garantir a integridade referencial;
- Aprimorar a funcionalidade e a eficiência do banco de dados, implementando componentes que facilitam a manipulação e o acesso aos dados;
- Desenvolver *Views*, *Funções* e *Stored Procedures*, que proporcionam uma camada de abstração e otimização nas operações do banco de dados;
- Detalhar as descrições das *Views*, *Funções* e *Stored Procedures*, incluindo seus objetivos, tabelas envolvidas e benefícios para o projeto;
- Detalhar o script de inserção de dados nas tabelas, garantindo a disponibilidade de informações para testes e demonstração do sistema;
- Preparar a documentação em formato PDF, conforme solicitado, contendo todas as informações relevantes desta etapa.

### 3. Situação Problemática

A implementação de um banco de dados para uma biblioteca surge da necessidade de superar as limitações de métodos tradicionais de gerenciamento, como fichários manuais ou planilhas eletrônicas. Esses métodos podem ser ineficientes para lidar com grandes volumes de dados, dificultam a busca rápida por informações específicas, podem levar a inconsistências nos dados e complicam o controle de empréstimos e devoluções.

Um banco de dados relacional bem projetado pode solucionar essas brechas ao:

- **Centralizar as informações:** Reunindo todos os dados relevantes em um único sistema, facilitando o acesso e a gestão.
- **Melhorar a organização:** Estruturando os dados em tabelas inter-relacionadas, permitindo consultas eficientes e a manutenção da integridade dos dados.
- **Otimizar processos:** Agilizando tarefas como a catalogação de novos livros, o registro de clientes, o controle de empréstimos e a geração de relatórios.
- **Reduzir erros:** Impondo restrições e relacionamentos entre as tabelas para garantir a consistência e a validade dos dados.
- **Facilitar a escalabilidade:** Permitindo que o sistema acompanhe o crescimento da biblioteca e o aumento do número de usuários e acervo.

A implementação deste banco de dados fornecerá uma base sólida para o desenvolvimento de um sistema de gerenciamento de biblioteca completo e eficiente.

### 4. Modelo de Negócio

A organização que utiliza esta solução é uma biblioteca tradicional, que oferece serviços de empréstimo de livros. A biblioteca possui um acervo diversificado de livros categorizados por temas, com múltiplos autores contribuindo para diferentes obras. Os clientes (usuários da biblioteca) se registram para poderem alugar os livros disponíveis. O processo de aluguel envolve o registro de qual livro foi emprestado a qual cliente e as datas de empréstimo e devolução esperada.

### 5. Diagrama de Entidade-Relação (DER)

O Diagrama de Entidade-Relação (DER) para o banco de dados da biblioteca é composto pelas seguintes entidades (tabelas) e seus relacionamentos:

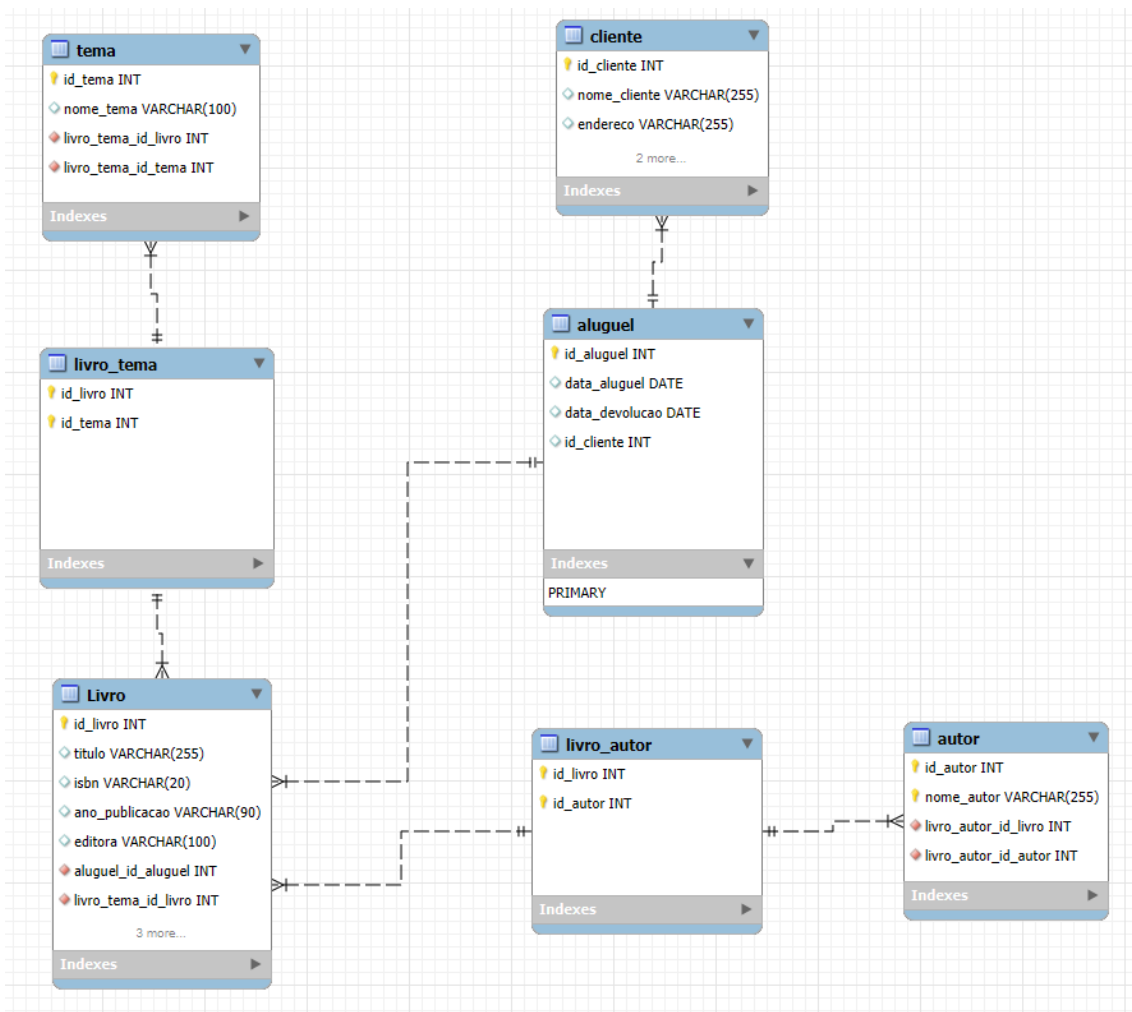
- **Livro:** Contém informações sobre cada livro.
- **Tema:** Contém informações sobre os temas dos livros.
- **Autor:** Contém informações sobre os autores dos livros.
- **Cliente:** Contém informações sobre os clientes da biblioteca.
- **Aluguel:** Registra os empréstimos de livros aos clientes.
- **Livro\_Tema:** Tabela associativa para o relacionamento muitos-para-muitos entre Livro e Tema.
- **Livro\_Autor:** Tabela associativa para o relacionamento muitos-para-muitos entre Livro e Autor.

#### **Relacionamentos Representados no DER:**

- **Livro** possui um relacionamento muitos-para-muitos com **Tema**, mediado pela tabela **Livro\_Tema**.
- **Livro** possui um relacionamento muitos-para-muitos com **Autor**, mediado pela tabela **Livro\_Autor**.
- **Cliente** possui um relacionamento um-para-muitos com **Aluguel** (um cliente pode ter muitos aluguéis).
- **Livro** possui um relacionamento um-para-muitos com **Aluguel** (um livro pode ser alugado muitas vezes por diferentes clientes).

As chaves primárias de cada entidade são utilizadas como chaves estrangeiras nas tabelas de relacionamento (Livro\_Tema, Livro\_Autor e Aluguel) para estabelecer as ligações e garantir a integridade referencial.

Modelo de entidade-relacionamento:



## 6. Listagem de Tabelas

A seguir, a listagem das tabelas que compõem o banco de dados da biblioteca, com a descrição de cada tabela, seus campos, abreviações, nomes completos, tipos de dados e tipos de chaves:

Tabela	Descrição	Campo Abreviado	Nome Completo	Tipo de Dado	Chave
Livro	Informações sobre cada livro disponível.	id_livro	Identificador Livro	INT	PK, AI
		titulo	Título do Livro	VARCHAR(255)	
		isbn	ISBN do Livro	VARCHAR(20)	UNIQUE
		ano_publicacao	Ano de Publicação	YEAR	
		editora	Editora do Livro	VARCHAR(100)	

Tema	Categorias ou assuntos dos livros.	id_tema	Identificador Tema	INT	PK, AI
		nome_tema	Nome do Tema	VARCHAR(100)	UNIQUE
Autor	Informações sobre os autores dos livros.	id_autor	Identificador Autor	INT	PK, AI
		nome_autor	Nome do Autor	VARCHAR(255)	
Cliente	Informações sobre os clientes da biblioteca.	id_cliente	Identificador Cliente	INT	PK, AI
		nome_cliente	Nome do Cliente	VARCHAR(255)	
		endereco	Endereço do Cliente	VARCHAR(255)	
		telefone	Telefone do Cliente	VARCHAR(20)	
Aluguel	Registro dos aluguéis de livros.	id_aluguel	Identificador Aluguel	INT	PK, AI
		id_livro	Identificador Livro	INT	FK
		id_cliente	Identificador Cliente	INT	FK
		data_aluguel	Data do Aluguel	DATE	
		data_devolucao	Data de Devolução	DATE	
Livro_Tema	Tabela de ligação entre Livros e Temas.	id_livro	Identificador Livro	INT	PK, FK
		id_tema	Identificador Tema	INT	PK, FK
Livro_Autor	Tabela de ligação entre Livros e Autores.	id_livro	Identificador Livro	INT	PK, FK
		id_autor	Identificador Autor	INT	PK, FK

## 7. Script SQL de Criação da Base de Dados e Tabelas

Link para github:

[https://github.com/AdyDario/coderhouse\\_sql](https://github.com/AdyDario/coderhouse_sql)

## 8. Descrição das Views

- **[vw\\_livros\\_detalhados](#)**: Exibe informações detalhadas sobre os livros, incluindo título, ISBN, ano de publicação, editora e os nomes dos autores e temas associados.,Livro, Livro\_Autor, Autor, Livro\_Tema, Tema. Objetivo: facilitar a consulta de informações completas sobre os livros, evitando a necessidade de múltiplas junções de tabelas;
- **[vw\\_alugueis\\_ativos](#)**: Lista todos os alugueis que ainda não foram devolvidos (onde data\_devolucao é NULL). Inclui o ID do aluguel, título do livro, nome do cliente e data do aluguel.,Aluguel, Livro, Cliente. Objetivo: fornece uma visão rápida dos livros que estão atualmente emprestados;
- **[vw\\_clientes\\_com\\_alugueis](#)**: Exibe os nomes dos clientes e o número de livros que cada um alugou.,Cliente, Aluguel. Objetivo: Permitir identificar clientes com maior histórico de aluguel.

## 9. Descrição das Funções

- **[fn\\_calcular\\_dias\\_aluguel](#)**: Calcula o número de dias entre duas datas fornecidas. Retorna NULL se a data\_final for NULL.,Nenhuma (opera com os parâmetros de entrada), pode ser utilizada para calcular o tempo de aluguel de um livro.,INT;
- **[fn\\_verificar\\_disponibilidade\\_livro](#)**: Verifica se um determinado livro (pelo id\_livro) está atualmente alugado. Retorna TRUE se não estiver alugado e FALSE caso contrário., Aluguel. Auxilia na verificação rápida se um livro pode ser alugado.,BOOLEAN