

## **Entrega 2**

### **Projeto de banco de dados SQL.**

**Adilson Lima Dario**

**Turma 70960**

## 1. Introdução

Este documento apresenta a segunda entrega do projeto de banco de dados para o gerenciamento de uma biblioteca. A primeira entrega estabeleceu a base estrutural do banco de dados, incluindo a modelagem das entidades principais, a definição dos relacionamentos e a criação do esquema do banco de dados através de um script SQL.

Dando continuidade ao desenvolvimento, esta segunda etapa tem como objetivo aprimorar a funcionalidade e a eficiência do banco de dados, implementando componentes que facilitam a manipulação e o acesso aos dados. Para isso, foram desenvolvidas *Views*, Funções e *Stored Procedures*, que proporcionam uma camada de abstração e otimização nas operações do banco de dados.

Além disso, este documento detalha o script de inserção de dados nas tabelas, garantindo a disponibilidade de informações para testes e demonstração do sistema. Também são apresentadas as descrições detalhadas das *Views*, Funções e *Stored Procedures*, incluindo seus objetivos, tabelas envolvidas e benefícios para o projeto. A implementação desses recursos visa aprimorar a usabilidade, a segurança e o desempenho do sistema de gerenciamento da biblioteca, fornecendo uma base sólida para futuras expansões e integrações.

## 2. 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.

### 3. 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

### 4. Descrição dos *Stored Procedures*

- **sp\_registrar\_aluguel:** Insere um novo registro na tabela Aluguel com o id\_livro, id\_cliente e data\_aluguel fornecidos.,Aluguel. Objetivo: padronizar e simplificar o processo de registrar um novo aluguel, garantindo a integridade dos dados;
- **sp\_registrar\_devolucao:** Atualiza a data\_devolucao de um aluguel existente na tabela Aluguel com a data\_devolucao fornecida, com base no id\_aluguel.,Aluguel. Objetivo: facilitar o registro da devolução de um livro, mantendo o controle dos aluguéis em aberto e fechados;
- **sp\_adicionar\_livro\_autor:** Insere um novo relacionamento entre um livro e um autor na tabela Livro\_Autor, verificando previamente se a relação já existe.,Livro\_Autor. Objetivo: permitir a associação de múltiplos autores a um livro de forma organizada e evitar a duplicação de registros na tabela de relacionamento.