

**Universidade Pitágoras Unopar**

**Anhanguera**

**ENGENHARIA DE SOFTWARE**

**Elisson Nadson Souza Marques - RA 4372521401**

**PORTFÓLIO - RELATÓRIO DE AULA PRÁTICA**

NOME DA DISCIPLINA: Programação e Desenvolvimento de Banco de Dados

**Elisson Nadson Souza Marques - RA 4372521401**

**PORTFÓLIO - RELATÓRIO DE AULA PRÁTICA**

NOME DA DISCIPLINA: Programação e Desenvolvimento de Banco de Dados

Trabalho de portfólio apresentado como requisito parcial para a obtenção de pontos para a média semestral

Orientador: Vinicius Camargo Prattes

# INTRODUÇÃO

Neste trabalho, desenvolvi a estrutura de um banco de dados relacional utilizando o **MySQL Workbench**.

O objetivo foi criar e manipular um banco de dados chamado **"Loja"**, seguindo um modelo Entidade-Relacionamento (DER) previamente definido.

O foco principal foi criar tabelas relacionadas, inserir registros de exemplo e implementar uma visão (VIEW) para exibir contas pendentes.

# OBJETIVOS

1. Criar a estrutura física do banco de dados **Loja**.
2. Implementar tabelas com relacionamentos (chaves primárias e estrangeiras).
3. Inserir registros fictícios para simular dados reais.
4. Criar uma visão (VIEW) para exibir contas pendentes de clientes.
5. Validar a consistência do modelo com um diagrama Entidade-Relacionamento.

# DESENVOLVIMENTO

**Etapa 1: Criação do Banco de Dados**

1. **Configuração**:
   * A base de dados **"Loja"** foi criada utilizando o comando:

CREATE DATABASE IF NOT EXISTS Loja;

USE Loja;

1. **Criação das Tabelas**:
   * Foram implementadas as tabelas **Estado**, **Município**, **Cliente** e **ContaReceber**, conforme descrito no modelo DER.
   * Exemplo da tabela **Estado**:

CREATE TABLE Estado (

ID INT PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(50) NOT NULL,

UF CHAR(2) NOT NULL

);

**Etapa 2: Inserção de Dados**

1. **População das Tabelas**:
   * Foram inseridos registros fictícios em cada tabela utilizando comandos **INSERT**.
   * Exemplo de inserção na tabela **Cliente**:

INSERT INTO Cliente (Nome, CPF, Celular, EndLogradouro, EndNumero, EndMunicipio, EndCep, Municipio\_ID)

VALUES

('João Silva', '12345678901', '11999999999', 'Rua A', '123', 'São Paulo', '01234567', 1),

('Maria Souza', '98765432101', '21999999999', 'Rua B', '456', 'Rio de Janeiro', '21000000', 2);

**Etapa 3: Criação de uma Visão**

1. **Objetivo da VIEW**:
   * Criar uma visão chamada **ContasPendentes** para exibir apenas contas com situação = 1 (pendentes).
   * Comando utilizado:

CREATE VIEW ContasPendentes AS

SELECT

CR.ID AS "ID da Conta a Receber",

C.Nome AS "Nome do Cliente",

C.CPF,

CR.DataVencimento,

CR.Valor

FROM ContaReceber CR

JOIN Cliente C ON C.ID = CR.Cliente\_ID

WHERE CR.Situacao = '1';

1. **Teste da VIEW**:
   * A consulta:

SELECT \* FROM ContasPendentes;

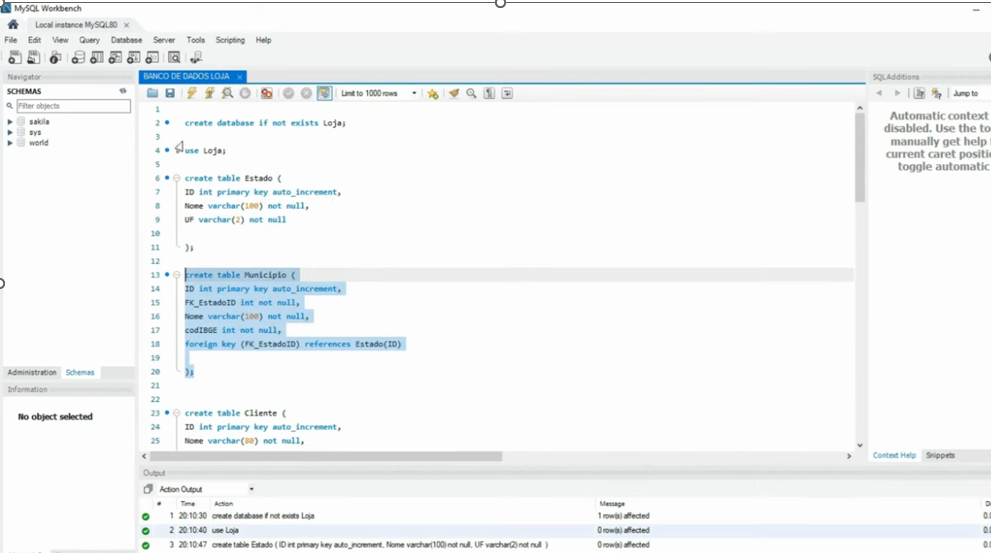
* + Retornou corretamente os registros pendentes, como demonstrado na imagem fornecida.

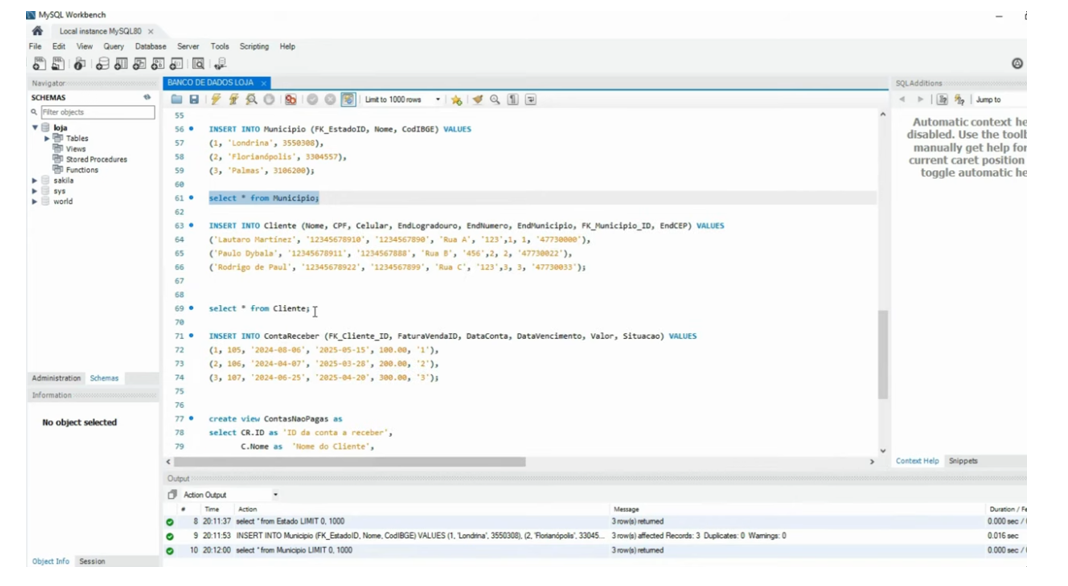
1. **RESULTADOS**

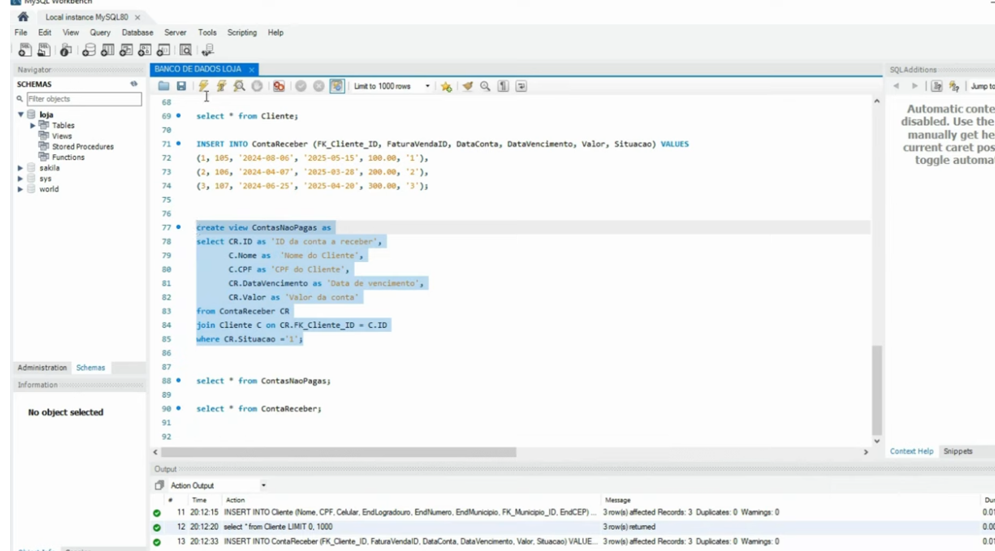
Os resultados obtidos foram os seguintes:

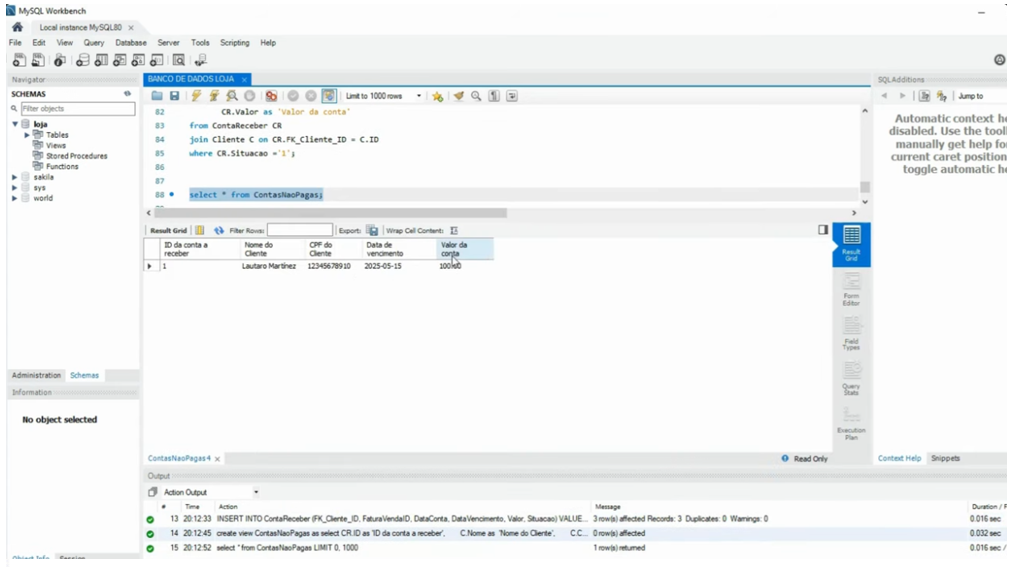
1. **Criação do Banco**:
   * O banco de dados foi criado com sucesso e todas as tabelas foram implementadas conforme o modelo DER.
2. **Inserção de Dados**:
   * Todos os registros foram adicionados corretamente às tabelas.
3. **VIEW Funcional**:
   * A visão **ContasPendentes** exibiu com precisão os dados esperados.
4. **Validação com DER**:
   * O modelo Entidade-Relacionamento foi validado, conforme mostrado no diagrama gerado pelo MySQL Workbench.

# EXPLICAÇÃO DAS IMAGENS

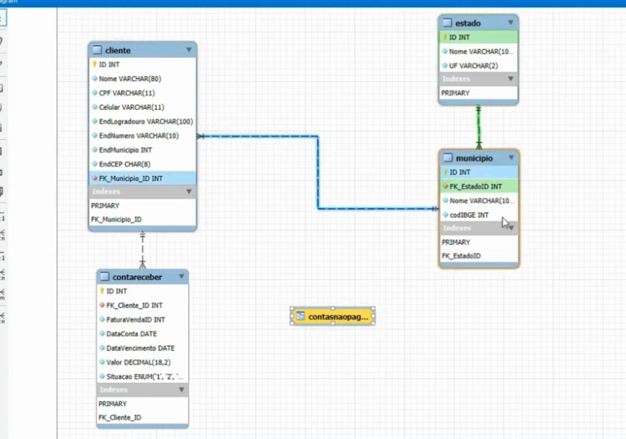
1. **Imagem 1**: Criação do banco de dados e das tabelas Estado, Município e Cliente.  
   
2. **Imagem 2**: Inserção de dados nas tabelas Município e Cliente.



1. **Imagem 3**: Inserção de dados na tabela ContaReceber e criação da visão **ContasPendentes**.  
     
   
2. **Imagem 4**: Visualização dos registros na visão **ContasPendentes**.



1. **Imagem 5**: Diagrama Entidade-Relacionamento (DER) validando os relacionamentos entre as tabelas.



# CONCLUSÃO

Essa atividade foi essencial para consolidar os conceitos de modelagem e manipulação de banco de dados relacional. Desde a criação das tabelas até a consulta de dados com a visão **ContasPendentes**, foi possível observar a importância de um bom planejamento do modelo de dados.

O uso do **MySQL Workbench** foi fundamental para organizar e validar o banco, garantindo a consistência das tabelas e relacionamentos. Essa prática reforça a aplicabilidade dos conhecimentos em projetos futuros.

# REFERÊNCIAS

* MySQL Documentation: <https://dev.mysql.com/doc/>
* MySQL Workbench User Guide: <https://dev.mysql.com/doc/workbench/en/>
* Curso de Banco de Dados - Gustavo Guanabara.