### **Analise e Desenvolvimento de Sistemas Tecnológico**

Eduardo Estigarribia Oliveira

título do trabalho:

Aula prática de Programação e Desenvolvimento de Banco de Dados

Goiânia - Goiás

2024

eduardo estigarribia Oliveira

título do trabalho:

Programação e Desenvolvimento de Banco de Dados

Aula prática de Programação e Desenvolvimento de Banco de Dados apresentado como requisito parcial para a obtenção de média semestral no curso Análise e Desenvolvimento de Sistemas.

Tutor: Tiago Barroso dos Santos

Goiânia - Goiás

2024

SUMÁRIO

[**Analise e Desenvolvimento de Sistemas Tecnológico** 0](#_Toc181188394)

[1 INTRODUÇÃO 3](#_Toc181188395)

[2 DESENVOLVIMENTO 4](#_Toc181188396)

[2.1 CRIAÇÃO DO BANCO DE DADOS “LOJA” 4](#_Toc181188397)

[2.2 INSERSÃO DE DADOS 6](#_Toc181188398)

[2.3 CRIAÇÃO DA vIEW 7](#_Toc181188399)

[3 RESULTADO 9](#_Toc181188400)

[3.1 CRIAÇÃO DO BANCO DE DADOS E TABELAS 9](#_Toc181188401)

[3.2 PREENCHIMENTO DE TABELAS 10](#_Toc181188402)

[3.3 CRIAÇÃO DA VIEW PARA CONTAS A RECEBER 12](#_Toc181188403)

[4 CONCLUSÃO 13](#_Toc181188404)

# INTRODUÇÃO

A compreensão e a aplicação de técnicas de modelagem e manipulação de banco de dados são fundamentais para qualquer profissional da área de tecnologia da informação. A disciplina de **Programação e Desenvolvimento de Banco de Dados** proporciona uma base sólida para o entendimento da estruturação de dados, relacionamentos entre entidades e gerenciamento eficiente das informações em sistemas empresariais. Esses conhecimentos são essenciais para o desenvolvimento de soluções que demandam consistência, integridade e segurança no armazenamento e na recuperação de dados. Em um mercado cada vez mais voltado para a análise e a gestão eficiente de grandes volumes de dados, o domínio dessas técnicas torna-se um diferencial competitivo para qualquer desenvolvedor.

Neste trabalho, serão desenvolvidas atividades práticas que envolvem a criação de um banco de dados utilizando o **MySQL Workbench** e comandos SQL para definição, manipulação e consulta de dados. Partindo de um diagrama entidade-relacionamento pré-definido, serão realizadas etapas que envolvem a implementação das tabelas, inserção de registros e a criação de uma visão para exibir informações específicas do banco. Ao final, será elaborado um relatório contendo os scripts desenvolvidos e os resultados obtidos, visando reforçar o aprendizado e a prática dos conceitos estudados em sala de aula.

# DESENVOLVIMENTO

Conforme orientações sobre esta tarefa vamos criar o banco de dados, suas tabelas e relações conforme o diagrama entidade-relacionamento (DER- fig. 1) e posteriormente popular com dados.

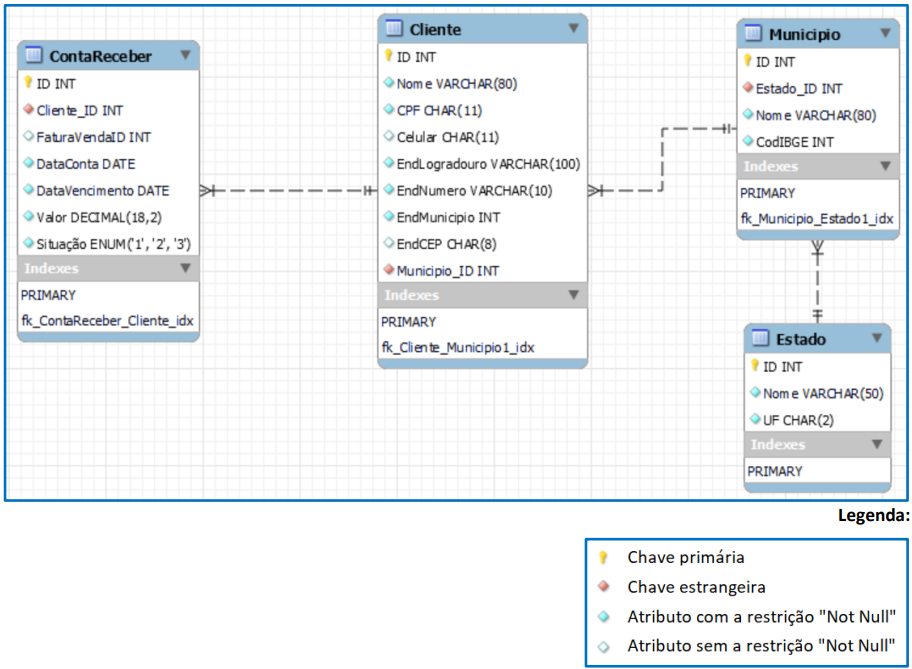


Figura 1

## CRIAÇÃO DO BANCO DE DADOS “LOJA”

No desenvolvimento de um banco de dados, utilizamos diferentes tipos de comandos SQL para cada etapa do processo. Na criação do banco de dados "Loja", aplicamos comandos de **DDL (Data Definition Language)**, responsáveis pela definição da estrutura das tabelas e das relações entre elas, incluindo a criação das chaves primárias e estrangeiras, além de definir restrições como NOT NULL e tipos de dados apropriados para cada campo.

A seguir temos os comandos para criação do banco e suas tabelas:

*-- Criando o banco de dados "Loja"*

*CREATE DATABASE Loja;*

*USE Loja;*

*-- Criando a tabela "Estado"*

*CREATE TABLE Estado (*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*nome VARCHAR(50) NOT NULL,*

*uf CHAR(2) NOT NULL*

*);*

*-- Criando a tabela "Municipio"*

*CREATE TABLE Municipio (*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*nome VARCHAR(80) NOT NULL,*

*codIBGE INT,*

*estado\_id INT NOT NULL,*

*FOREIGN KEY (estado\_id) REFERENCES Estado(id)*

*);*

*-- Criando a tabela "Cliente"*

*CREATE TABLE Cliente (*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*nome VARCHAR(80) NOT NULL,*

*cpf CHAR(11) NOT NULL UNIQUE,*

*celular CHAR(11),*

*endereco\_logradouro VARCHAR(100),*

*endereco\_numero VARCHAR(10),*

*endereco\_municipio INT,*

*endereco\_cep CHAR(8),*

*municipio\_id INT,*

*FOREIGN KEY (municipio\_id) REFERENCES Municipio(id)*

*);*

*-- Criando a tabela "ContaReceber"*

*CREATE TABLE ContaReceber (*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*cliente\_id INT,*

*fatura\_venda\_id INT,*

*data\_conta DATE,*

*data\_vencimento DATE,*

*valor DECIMAL(18, 2) NOT NULL,*

*situacao ENUM('1', '2', '3') NOT NULL,*

*FOREIGN KEY (cliente\_id) REFERENCES Cliente(id)*

*);*

## INSERSÃO DE DADOS

Para a realização de operações de manipulação de dados, utilizamos comandos de DML (Data Manipulation Language), que incluem as instruções de inserção, atualização e exclusão de registros. Esses comandos são essenciais para garantir a interação adequada com os dados armazenados em nossas tabelas. Seguindo as diretrizes propostas, desenvolvemos um script no arquivo ***inserir.sql*** para popular o banco de dados 'Loja', garantindo que as tabelas sejam preenchidas com informações coerentes e realistas (fig. 2).

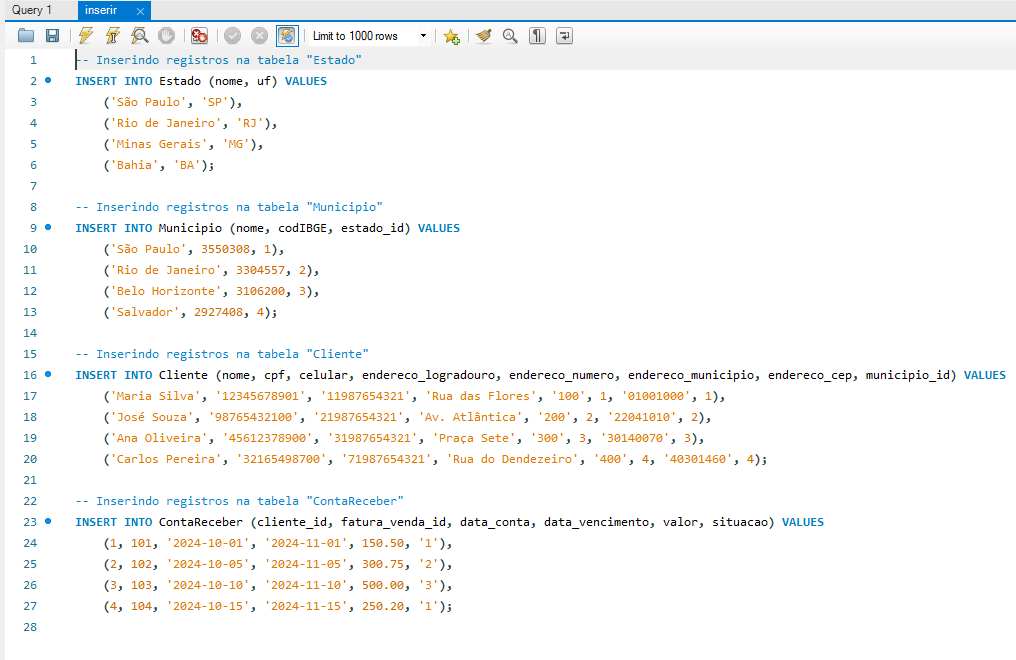


Figura 2

## CRIAÇÃO DA vIEW

Com as tabelas devidamente populadas, agora é possível criar a view conforme descrito na proposta. Para isso, desenvolvemos um script no arquivo consulta.sql que cria uma visão (VIEW) destinada a retornar todas as contas que ainda não foram pagas (Situação = 1). Essa view vai incluir as seguintes informações:

* ID da conta a receber
* Nome e CPF do cliente associado à conta
* Data de vencimento da conta
* Valor da conta

Abaixo segue imagem (fig. 3) da view criada:

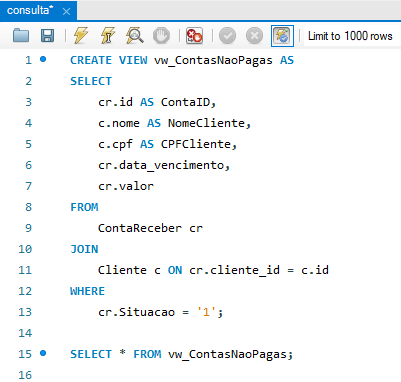


Figura 3 - View

# RESULTADO

Após cada ação do desenvolvimento tivemos os seguintes resultados conforme descrito a seguir:

## CRIAÇÃO DO BANCO DE DADOS E TABELAS

Na Figura *4*, apresentamos a estrutura inicial do banco de dados ‘Loja’, composta pelas tabelas **Estado**, **Municipio**, **Cliente** e **ContaReceber**. Essas tabelas foram criadas com base em um modelo DER previamente definido, assegurando a integridade referencial entre elas por meio de chaves primárias e estrangeiras. Essa estruturação permite uma organização lógica e eficiente dos dados.

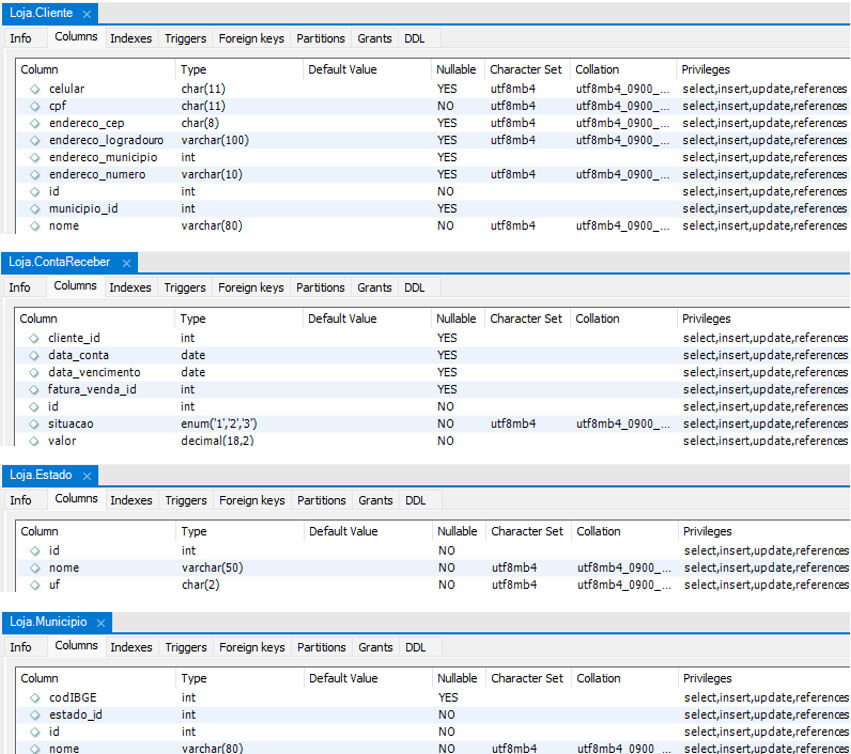


Figura 4 – Tabelas criadas

## PREENCHIMENTO DE TABELAS

Após a criação das tabelas, foi realizado o processo de inserção de dados utilizando comandos de DML. Na sequência, apresentamos as tabelas preenchidas (Figuras 4, 5, 6 e 7), contendo registros reais de **Estados**, **Municípios**, **Clientes** e **Contas a Receber**. O objetivo é demonstrar a consistência dos dados inseridos e a interligação entre as informações em diferentes tabelas.

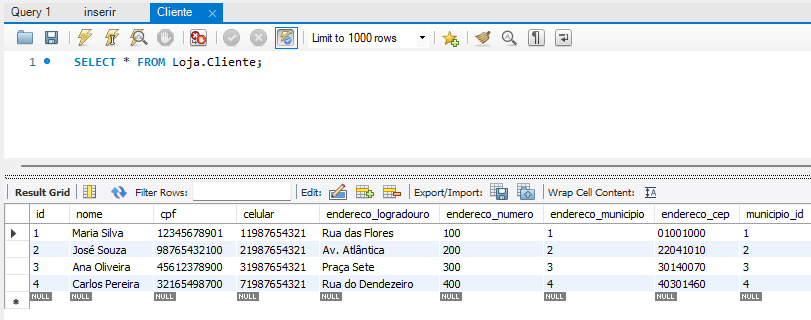


Figura 4 – Tabela Cliente

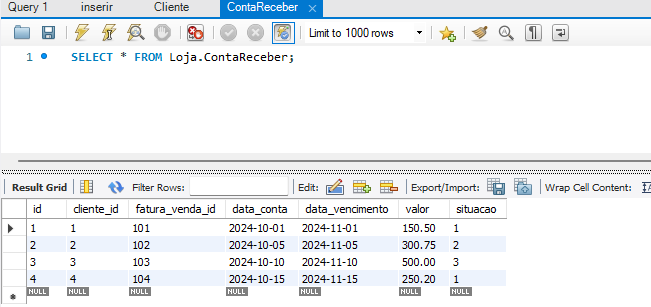


Figura 5 – Tabela ContasReceber

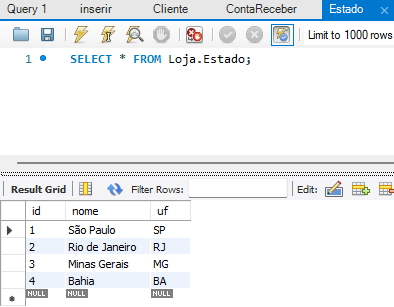


Figura 6 – Tabela Estado

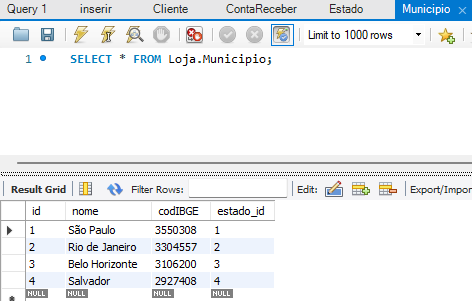


Figura 7 – Tabela Municipio

## CRIAÇÃO DA VIEW PARA CONTAS A RECEBER

A Figura 8 exibe o resultado da execução da view vw\_ContasNaoPagas, criada para listar as contas a receber ainda não pagas (Situação = 1). Essa visualização seria fundamental para o acompanhamento financeiro, fornecendo um relatório detalhado de contas pendentes de pagamento.

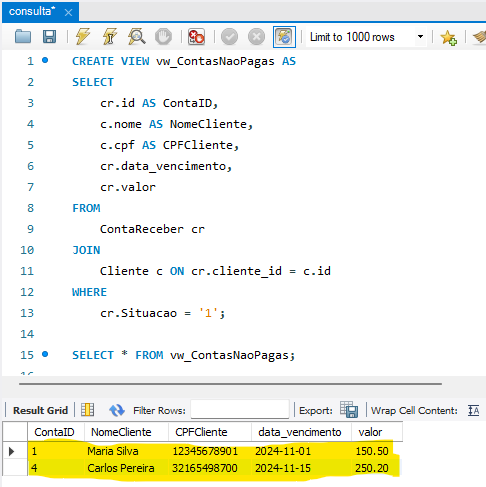


Figura 8 – vw\_ContasNaoPagas

# CONCLUSÃO

A realização deste trabalho prático foi fundamental para reforçar o nosso conhecimento sobre programação e desenvolvimento de banco de dados. Através da criação de um banco relacional, da estruturação de tabelas, da inserção de dados e da construção de uma view para consultas específicas, tivemos a oportunidade de aplicar conceitos teóricos de forma prática e consolidar o entendimento sobre a linguagem SQL e suas diferentes operações.

Além disso, essa disciplina é essencial para quem estuda Análise e Desenvolvimento de Sistemas, pois o domínio sobre bancos de dados e a capacidade de manipular informações de forma eficiente e segura são habilidades indispensáveis para qualquer profissional da área. A compreensão desses conceitos não apenas facilita o desenvolvimento de aplicações mais robustas e escaláveis, mas também nos prepara para desafios reais do mercado de trabalho, onde o gerenciamento correto de dados é uma peça-chave para o sucesso de projetos e soluções de software.

.