

Engenharia de Software

Portfólio – Relatório de Aula Prática:

Criar um banco de dados utilizando a linguagem SQL e realizar operações de manipulação e acesso aos dados.

Hávila Júnior Alves Da Silva

Trabalho de Portfólio apresentado como requisito parcial para a obtenção de pontos para a média semestral Orientador: Anderson Emidio de Macedo Goncalves

Sumário

1	1. INTRODUÇÃO	3
2	2. METODOLOGIA	3
3	2.1. Ferramentas e Infraestrutura	3
4	2.2. Etapas de Desenvolvimento.....	3
5	2.3. Modelo de Dados.....	3
6	3. RESULTADOS E IMPLEMENTAÇÃO	4
7	4. RESULTADOS OBTIDOS.....	6
8	5. ANÁLISE E DISCUSSÃO	7
9	6. CONCLUSÃO	7

1 1. INTRODUÇÃO

Este relatório documenta a implementação prática de um sistema de banco de dados para uma loja, desenvolvido durante a aula de Programação e Desenvolvimento de Banco de Dados. A atividade teve como objetivo principal a criação de um banco de dados utilizando a linguagem SQL e a realização de operações de manipulação e acesso aos dados, aplicando os conceitos teóricos aprendidos em sala de aula.

O projeto focou no desenvolvimento de um sistema de controle financeiro para uma loja, com ênfase no gerenciamento de contas a receber e no cadastro de clientes, seguindo as melhores práticas de modelagem de dados e normalização.

2 2. METODOLOGIA

3 2.1. Ferramentas e Infraestrutura

Sistema Gerenciador de Banco de Dados: MySQL Community Server 8.0

Ferramenta de Desenvolvimento: MySQL Workbench 8.0

Linguagem de Programação: SQL (Structured Query Language)

Sistema Operacional: Ambiente Windows/Linux Licença: Freeware

4 2.2. Etapas de Desenvolvimento

A atividade foi dividida em três etapas principais:

1. Modelagem e Criação da Estrutura: Implementação do modelo físico baseado no DER
2. População do Banco de Dados: Inserção de dados de teste
3. Criação de Consultas e Relatórios: Desenvolvimento de VIEWS para análise

5 2.3. Modelo de Dados

O diagrama entidade-relacionamento implementado seguiu a seguinte estrutura:

text

Cliente (1) ——— (N) ContaReceber

| |

6 3. RESULTADOS E IMPLEMENTAÇÃO

3.1. Etapa 1 - Criação da Estrutura (DDL)

Script de Criação do Banco de Dados:

```
sql
-- Criação do banco de dados Loja
CREATE DATABASE IF NOT EXISTS Loja;
USE Loja;

-- Tabela Cliente CREATE TABLE
Cliente (    id INT
AUTO_INCREMENT PRIMARY KEY,
nome VARCHAR(100) NOT NULL,
cpf VARCHAR(14) NOT NULL UNIQUE,
telefone VARCHAR(15),      email
VARCHAR(100) );

-- Tabela ContaReceber CREATE TABLE
ContaReceber (    id INT
AUTO_INCREMENT PRIMARY KEY,
cliente_id INT NOT NULL,
data_vencimento DATE NOT NULL,
valor DECIMAL(10,2) NOT NULL,
situacao ENUM('1', '2', '3') NOT
NULL,
FOREIGN KEY (cliente_id) REFERENCES
Cliente(id) );
```

Características da Implementação:

- Chaves primárias com auto incremento
- Campo situacao como ENUM com valores 1, 2, 3
- Constraints de integridade referencial
- Tipos de dados apropriados para cada campo
- Respeito aos relacionamentos e restrições de não nulo

3.2. Etapa 2 - População do Banco (DML)

Script: inserir.sql

sql

```
-- Inserção de dados na tabela Cliente
INSERT INTO Cliente (nome, cpf, telefone, email) VALUES
('João Silva', '123.456.789-00', '(11) 9999-8888', 'joao.silva@email.com'),
('Maria Santos', '987.654.321-00', '(11) 7777-6666', 'maria.santos@email.com'),
('Pedro Oliveira', '456.789.123-00', '(11) 5555-4444', 'pedro.oliveira@email.com'),
('Ana Costa', '321.654.987-00', '(11) 3333-2222', 'ana.costa@email.com'),
('Carlos Souza', '654.321.987-00', '(11) 1111-0000',
'carlos.souza@email.com');

-- Inserção de dados na tabela ContaReceber
INSERT INTO ContaReceber (cliente_id, data_vencimento, valor, situacao) VALUES
(1, '2024-01-15', 150.00, '1'), -- Conta
registrada (1, '2024-01-20', 200.50, '3'), --
Conta paga
(2, '2024-01-18', 300.00, '1'), -- Conta
registrada (2, '2024-01-25', 450.75, '2'), --
Conta cancelada
(3, '2024-02-01', 125.00, '1'), -- Conta
registrada (4, '2024-02-05', 600.00, '3'), --
Conta paga
(5, '2024-02-10', 275.25, '1'); -- Conta
registrada
```

3.3. Etapa 3 - Consultas e Relatórios (DQL)

Script: consulta.sql

sql

```
-- Criação da VIEW para contas não pagas
CREATE VIEW
ContasNaoPagas AS
SELECT      cr.id AS
'ID_Conta',
c.nome AS 'Nome_Cliente',
c.cpf AS 'CPF_Cliente',
cr.data_vencimento AS
```

```
'Data_Vencimento',      cr.valor AS  
'Valor_Contra'  
FROM ContaReceber cr  
INNER JOIN Cliente c ON cr.cliente_id = c.id  
WHERE cr.situacao = '1';  
-- Consulta para visualizar os resultados  
SELECT * FROM ContasNaoPagas;
```

7 4. RESULTADOS OBTIDOS

4.1. Estrutura Criada com Sucesso

Banco de dados "Loja" criado

2 tabelas implementadas (Cliente e ContaReceber)

Relacionamento 1:N estabelecido corretamente Constraints e validações aplicadas

4.2. Dados Inseridos

5 clientes cadastrados

7 contas a receber registradas

Dados distribuídos entre diferentes situações

4.3. Resultado da VIEW ContasNaoPagas

A consulta da VIEW retornou os seguintes resultados:

	Nome_Cliente	CPF_Cliente	Data_Vencimento	Valor_Contra
I				
D				
-				
C				
O				
n				

t

a

1	João Silva	123.456.789-00	2024-01-15	R\$ 150,00
3	Maria Santos	987.654.321-00	2024-01-18	R\$ 300,00
5	Pedro Oliveira	456.789.123-00	2024-02-01	R\$ 125,00
7	Carlos Souza	654.321.987-00	2024-02-10	R\$ 275,25

Total de contas não pagas: 4 registros **Valor total em aberto:** R\$ 850,25

8 5. ANÁLISE E DISCUSSÃO

5.1. Aspectos Técnicos Implementados

Uso de ENUM: O campo situacao foi implementado como ENUM, garantindo que apenas os valores 1, 2 e 3 sejam aceitos, conforme especificado:

- = Conta registrada
- = Conta cancelada
- = Conta paga

Integridade Referencial: A chave estrangeira cliente_id garante que não existam contas sem cliente associado

Auto Incremento: Todas as chaves primárias implementadas com auto incremento, simplificando a inserção de dados

5.2. Benefícios da VIEW Criada

A VIEW ContasNaoPagas proporciona:

Simplificação de consultas complexas

Reutilização de código SQL

Maior segurança no acesso aos dados

Melhor organização da lógica de negócio

9 6. CONCLUSÃO

A atividade prática foi concluída com sucesso, atendendo a todos os requisitos especificados no roteiro. Através deste projeto, foi possível:

Consolidar conhecimentos em SQL para criação e manipulação de bancos de dados

Aplicar conceitos de modelagem entidade-relacionamento na prática

3. Desenvolver habilidades em DDL, DML e DQL
4. Implementar soluções reais para problemas de gestão de dados

O banco de dados desenvolvido demonstra robustez e escalabilidade, podendo ser expandido para incluir novas funcionalidades como controle de estoque, vendas e relatórios financeiros mais complexos.