

Relatório da Aula Prática - Implementação de Banco de Dados MySQL

Aluno : Nivando Soares da Silva

Data : 11/08/2024

Introdução

Nesta aula prática, foi desenvolvida a implementação de um banco de dados utilizando o MySQL Server e o MySQL Workbench. O objetivo principal foi construir a estrutura física do banco de dados com base no Diagrama Entidade-Relacionamento (DER) fornecido, além de criar scripts para inserção de dados e consultas SQL. Este relatório descreve os métodos utilizados, os resultados obtidos e as conclusões finais.

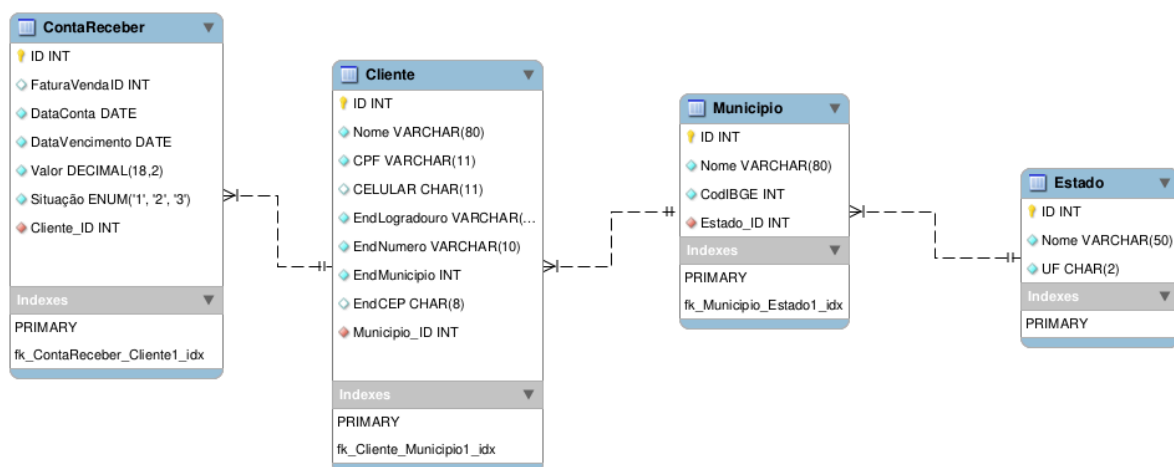
Métodos

1. Instalação do MySQL Community Server e MySQL Workbench

Foi realizado o download e a instalação do MySQL Community Server e do MySQL Workbench num ambiente Linux Ubuntu 23.10 para a criação e gerenciamento do banco de dados. O MySQL Workbench foi utilizado como ferramenta de modelagem e execução de scripts SQL.

2. Elaboração do Modelo Físico

Com base no DER fornecido, foi elaborado o modelo físico do banco de dados no MySQL Workbench. A estrutura das tabelas, seus relacionamentos, chaves primárias e estrangeiras foram definidos conforme o modelo lógico.



3. Criação das tabelas do banco em linguagem SQL

Com base na imagem do modelo físico foi desenvolvido um script sql para criar a estrutura do modelo no banco mysql

4. Criação do Script `inserir.sql`

Foi desenvolvido o script `inserir.sql`, responsável por inserir dados em todas as tabelas criadas. Este script inclui comandos `INSERT INTO` para cada tabela, garantindo que os dados iniciais fossem inseridos de acordo com as especificações.

5. Criação do Script `consulta.sql`

O script `consulta.sql` foi elaborado para criar uma visão que retorna todas as contas que ainda não foram pagas. Este script utiliza a cláusula `CREATE VIEW` para definir a consulta desejada.

Resultados

Estrutura do Banco de Dados

Abaixo estão os scripts SQL utilizados para a criação das tabelas e definição dos relacionamentos:

```
CREATE TABLE Estado (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Nome VARCHAR(50) NOT NULL,  
    UF CHAR(2) NOT NULL  
);  
  
CREATE TABLE Municipio (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Nome VARCHAR(80) NOT NULL,  
    CodIBGE INT,  
    Estado_ID INT NOT NULL,  
    FOREIGN KEY (Estado_ID) REFERENCES Estado(ID)  
);  
  
CREATE TABLE Cliente (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Nome VARCHAR(80) NOT NULL,  
    CPF VARCHAR(11) NOT NULL UNIQUE, -- Garantindo a unicidade do CPF  
    Celular CHAR(11),  
    End_Logradouro VARCHAR(100),  
    End_Numero VARCHAR(10),  
    End_Municipio INT NOT NULL,  
    End_CEP CHAR(8),  
    Municipio_ID INT NOT NULL,  
    FOREIGN KEY (Municipio_ID) REFERENCES Municipio(ID)  
);  
  
CREATE TABLE ContaReceber (  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    FaturaVendaID INT,  
    DataConta DATE,  
    DataVencimento DATE,  
    Valor DECIMAL(18,2) NOT NULL,  
    Situacao ENUM('1', '2', '3') NOT NULL DEFAULT '1', -- Situação inicial como  
    'Conta registrada'  
    Cliente_ID INT NOT NULL,  
    FOREIGN KEY (Cliente_ID) REFERENCES Cliente(ID)
```

```
);
```

Script para inserção dos dados:

```
-- insert.sql

-- Inserir dados na tabela Estado
INSERT INTO Estado (Nome, UF) VALUES
    ('São Paulo', 'SP'),
    ('Rio de Janeiro', 'RJ'),
    ('Minas Gerais', 'MG');

-- Inserir dados na tabela Municipio
INSERT INTO Municipio (Nome, CodIBGE, Estado_ID) VALUES
    ('São Paulo', 12001, 1),
    ('Rio de Janeiro', 33045, 2),
    ('Belo Horizonte', 31062, 3);

-- Inserir dados na tabela Cliente
INSERT INTO Cliente (Nome, CPF, Celular, End_Logradouro, End_Numero,
End_Municipio, End_CEP, Municipio_ID) VALUES
    ('João da Silva', '12345678901', '11987654321', 'Rua Principal', '123', 1,
'01234567', 1),
    ('Maria Souza', '98765432109', '21987654321', 'Avenida Central', '456', 2,
'23456789', 2),
    ('Pedro Santos', '54321678901', '31987654321', 'Rua das Flores', '789', 3,
'34567890', 3);

-- Inserir dados na tabela ContaReceber
INSERT INTO ContaReceber (FaturaVendaID, DataConta, DataVencimento, Valor,
Situacao, Cliente_ID) VALUES
    (1001, '2023-11-01', '2023-11-30', 1000.00, 1, 1),
    (1002, '2023-12-01', '2023-12-31', 1500.00, 1, 2),
    (1003, '2024-01-01', '2024-01-31', 2000.00, 1, 3);
```

Criação da View para consulta de contas a receber

```
CREATE VIEW ContasAReceberPendentes AS
SELECT
    cr.ID AS ID_Conta,
    c.Nome AS Nome_Cliente,
    c.CPF,
    cr.DataVencimento,
    cr.Valor
FROM
    ContaReceber cr
INNER JOIN Cliente c ON cr.Cliente_ID = c.ID
WHERE
    cr.Situacao = 1;

SELECT * FROM ContasAReceberPendentes;
```

Resultado da consulta após execução do código a acima:

14 •
15
16

SELECT * FROM ContasARceberPendentes;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

| # | ID_Conta | Nome_Cliente | CPF | DataVencimento | Valor |
|---|----------|---------------|-------------|----------------|---------|
| 1 | 1 | João da Silva | 12345678901 | 2023-11-30 | 1000.00 |
| 2 | 2 | Maria Souza | 98765432109 | 2023-12-31 | 1500.00 |
| 3 | 3 | Pedro Santos | 54321678901 | 2024-01-31 | 2000.00 |

Conclusão

A atividade prática permitiu a compreensão das etapas necessárias para a implementação de um banco de dados físico a partir de um DER. Através da utilização do MySQL Workbench, foi possível modelar e gerenciar o banco de dados de maneira eficaz, além de elaborar scripts SQL que automatizam a inserção e consulta de dados. A prática reforçou a importância de um planejamento adequado na construção de bases de dados, garantindo a integridade e a consistência das informações armazenadas.