



Relatório de Modelagem e Implementação de Banco de Dados

Curso: Desenvolvimento FullStack
Nome: Marília Regis
Período:2024.4
Polo: Messejana

Introdução

Este relatório apresenta o processo de modelagem e implementação de um banco de dados para um sistema de compra e venda de produtos, conforme os requisitos estabelecidos. O desenvolvimento foi realizado utilizando o DBDesigner Fork para modelagem e o SQL Server Management Studio para implementação e criação do banco de dados.

Modelagem do Banco de Dados

A modelagem foi realizada no DBDesigner Fork, utilizando entidades e relacionamentos que atendem aos requisitos do sistema. As principais entidades são:

1. Usuários id_usuario nome email senha	2. Pessoas idpessoa nome endereço telefone tipo	3. Pessoa Física idpessoa cpf	4. Pessoa Jurídica : idpessoa cnpj
5. Produtos: id_produto nome quantidade preco_venda	6. Movimentos de Compra: id_movimento id_usuario id_produto id pessoa_juridica quantidade preço unitário	7. Movimentos de Venda: id_movimento id_usuario id_produto id pessoa_fisica quantidade preco_total	

Relacionamentos

- **Usuários** possuem permissão para realizar movimentos de compra e venda.
- **Movimentos de Compra** sempre envolvem uma **Pessoa Jurídica**.
- **Movimentos de Venda** sempre envolvem uma **Pessoa Física**.

Modelo Entidade-Relacionamento

O modelo foi estruturado com as tabelas relacionadas de forma clara, respeitando os relacionamentos 1x1 entre Pessoas e suas especializações (Física e Jurídica), e os relacionamentos 1xN entre Movimentos e as demais entidades.

Implementação no SQL Server Management Studio

A implementação do banco de dados foi realizada utilizando a linguagem SQL para criar as tabelas, relacionamentos, e constraints necessárias.

1. Configuração Inicial

Adicionar logon do usuário loja:

```
CREATE LOGIN loja WITH PASSWORD = 'loja';
CREATE USER loja FOR LOGIN loja;
ALTER ROLE db_owner ADD MEMBER loja;
```

Criação das tabelas:

-- Banco de Dados

```
CREATE DATABASE LojaDB;
USE LojaDB;
```

-- Sequence para ID de Pessoa

```
CREATE SEQUENCE seq_pessoa_id
START WITH 1
INCREMENT BY 1;
```

-- Tabela Usuários

```
CREATE TABLE Usuarios (
    id_usuario INT PRIMARY KEY IDENTITY,
    nome NVARCHAR(100),
    email NVARCHAR(100) UNIQUE,
    senha NVARCHAR(100)
);
```

-- Tabela Pessoas

```
CREATE TABLE Pessoas (
    id_pessoa INT PRIMARY KEY DEFAULT NEXT VALUE FOR seq_pessoa_id,
    nome NVARCHAR(100),
    endereco NVARCHAR(200),
    telefone NVARCHAR(20),
    tipo NVARCHAR(20)
);
```

-- Tabela Pessoa Física

```
CREATE TABLE PessoaFisica (
    id_pessoa INT PRIMARY KEY,
    cpf NVARCHAR(11) UNIQUE,
    FOREIGN KEY (id_pessoa) REFERENCES Pessoas(id_pessoa)
);
```

-- Tabela Pessoa Jurídica

```
CREATE TABLE PessoaJuridica (  
    id_pessoa INT PRIMARY KEY,  
    cnpj NVARCHAR(14) UNIQUE,  
    FOREIGN KEY (id_pessoa) REFERENCES Pessoas(id_pessoa)  
);
```

-- Tabela Produtos

```
CREATE TABLE Produtos (  
    id_produto INT PRIMARY KEY IDENTITY,  
    nome NVARCHAR(100),  
    quantidade INT,  
    preco_venda DECIMAL(10, 2)  
);
```

-- Tabela Movimentos de Compra

```
CREATE TABLE MovimentosCompra (  
    id_movimento INT PRIMARY KEY IDENTITY,  
    id_usuario INT,  
    id_produto INT,  
    id_pessoa_juridica INT,  
    quantidade INT,  
    preco_unitario DECIMAL(10, 2),  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario),  
    FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto),  
    FOREIGN KEY (id_pessoa_juridica) REFERENCES PessoaJuridica(id_pessoa)  
);
```

-- Tabela Movimentos de Venda

```
CREATE TABLE MovimentosVenda (  
    id_movimento INT PRIMARY KEY IDENTITY,  
    id_usuario INT,  
    id_produto INT,  
    id_pessoa_fisica INT,  
    quantidade INT,  
    preco_total AS (quantidade * (SELECT preco_venda FROM Produtos WHERE id_produto  
= MovimentosVenda.id_produto)),  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario),  
    FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto),  
    FOREIGN KEY (id_pessoa_fisica) REFERENCES PessoaFisica(id_pessoa)  
);
```

3. Exportação do Script SQL

O script completo foi salvo em um arquivo `.sql` para futura reutilização ou alterações no banco de dados.

Conclusão

A modelagem e implementação do banco de dados atenderam aos requisitos propostos, utilizando boas práticas de organização e normalização. O DBDesigner Fork foi eficaz na criação do modelo, e o SQL Server Management Studio permitiu a criação eficiente das estruturas necessárias. Este trabalho demonstra o entendimento básico da modelagem de banco de dados e o uso de SQL para implementação.