DESENVOLVIMENTO FULL STACK
POLO SETOR O – CEILÂNDIA - DF
PERÍODO 2024.2 FLEX
DISCIPLINA: VAMOS MANTER AS INFORMAÇÕES
MAYCON MOURA

Título da Prática

Modelagem e Criação de Banco de Dados Relacional com SQL Server

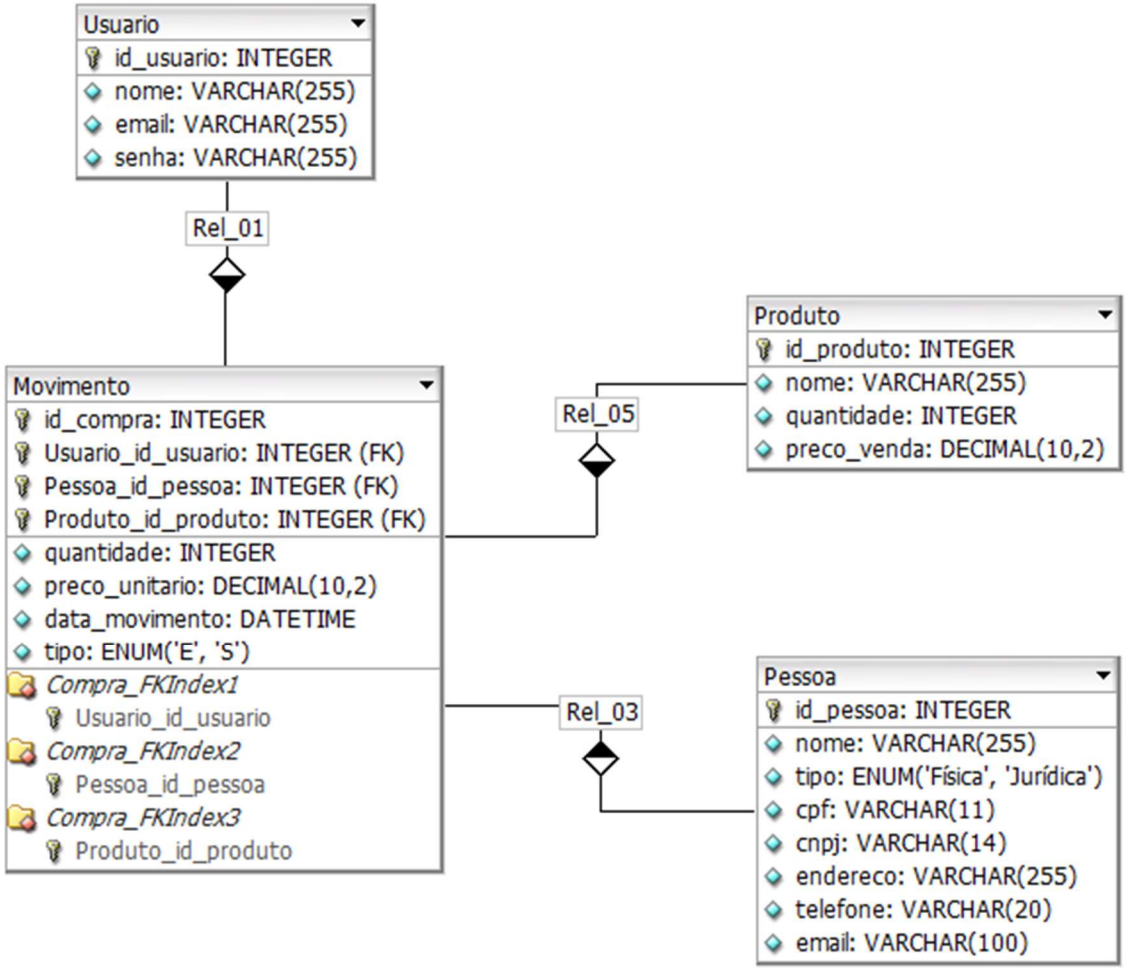
Objetivo da Prática

Demonstrar as habilidades básicas para a modelagem de um banco de dados relacional, utilizando o SQL Server Management Studio e DB Designer Fork para criar as estruturas necessárias, implementar funcionalidades e entender a sintaxe SQL.

Códigos Solicitados no Roteiro de Aula

Criação do Banco de Dados e Estruturas Necessárias

Modelagem



Criação do Banco de Dados

```
CREATE DATABASE loja;
```

1. **Usar o Banco de Dados Criado**

   ```
   USE loja;
   ```

2. **Criação da Sequence para Geração de IDs de Pessoa**

   ```
   CREATE SEQUENCE dbo.seq_pessoa_id

   START WITH 1
   INCREMENT BY 1;
   ```

3. **Criação da Tabela Usuario**

   ```
   CREATE TABLE Usuario (

      id_usuario INT IDENTITY(1,1) PRIMARY KEY,
      nome VARCHAR(100) NOT NULL,
      email VARCHAR(100) NOT NULL,
      senha VARCHAR(255) NOT NULL
   );
   ```

4. **Criação da Tabela Pessoa com Coluna Discriminadora**

```
CREATE TABLE Pessoa (

    id_pessoa INT PRIMARY KEY DEFAULT NEXT VALUE FOR dbo.seq_pessoa_id,
    nome VARCHAR(100) NOT NULL,
    tipo CHAR(1) CHECK (tipo IN ('F', 'J')) NOT NULL,
    endereco VARCHAR(255) NOT NULL,
    telefone VARCHAR(20) NOT NULL,
    email VARCHAR(100) NOT NULL
);
```

5. **Criação da Tabela PessoaFisica**

```
CREATE TABLE PessoaFisica (

    id_pessoa INT PRIMARY KEY,
    cpf VARCHAR(11) UNIQUE NOT NULL,
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)
);
```

6. **Criação da Tabela PessoaJuridica**

```
CREATE TABLE PessoaJuridica (

    id_pessoa INT PRIMARY KEY,
    cnpj VARCHAR(14) UNIQUE NOT NULL,
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)
);
```

7. **Criação da Tabela Produto**

```
CREATE TABLE Produto (

    id_produto INT IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    quantidade INT NOT NULL,
    preco_venda DECIMAL(10, 2) NOT NULL
);
```

8. **Criação da Tabela Compra**

```
CREATE TABLE Movimento (

    id_compra INT IDENTITY(1,1) PRIMARY KEY,
    id_usuario INT NOT NULL,
    id_pessoa INT NOT NULL,
    id_produto INT NOT NULL,
    quantidade INT NOT NULL,
    preco_unitario DECIMAL(10, 2) NOT NULL,
```

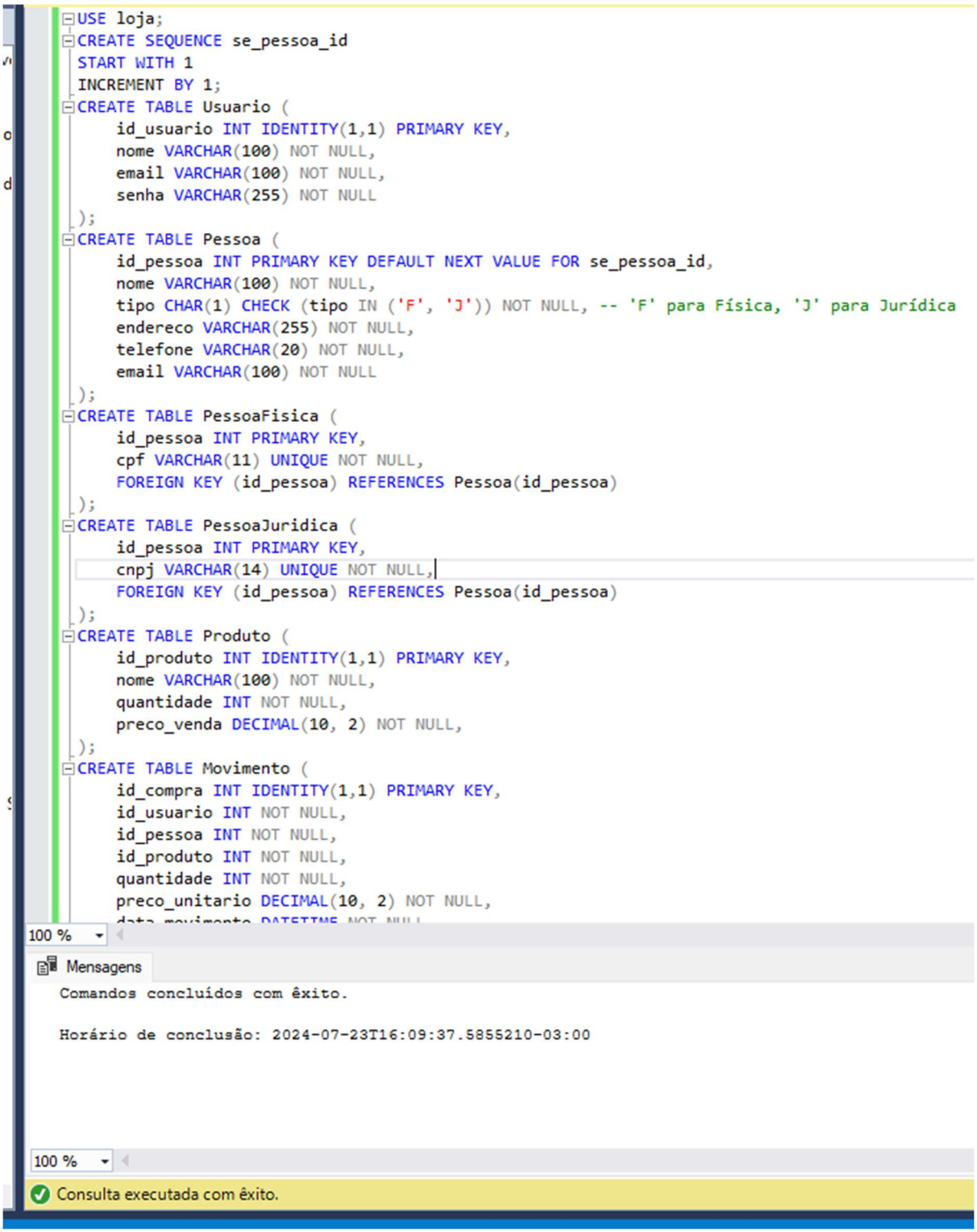
```
    data_movimento DATETIME NOT NULL,
      tipo CHAR(1) CHECK (tipo IN ('E', 'S')) NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa),
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)
);
```

9. **Criação da Tabela Venda**

```
CREATE TABLE Venda (

    id_venda INT IDENTITY(1,1) PRIMARY KEY,
    id_usuario INT NOT NULL,
    id_pessoa INT NOT NULL,
    id_produto INT NOT NULL,
    quantidade INT NOT NULL,
    preco_venda DECIMAL(10, 2) NOT NULL,
    data_venda DATETIME NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa),
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)
);
```

10. **Resultados Obtidos**



11. **Análise e Conclusão**

Cardinalidades no Banco de Dados Relacional

1. **1x1 (Um para Um)**
   - Implementada utilizando chaves primárias e estrangeiras nas tabelas Pessoa, PessoaFisica, e PessoaJuridica. Cada registro em Pessoa deve ter exatamente um registro correspondente em PessoaFisica ou PessoaJuridica.

2. **1xN (Um para Muitos)**
    - Exemplo: Usuario e Compra/Venda. Um Usuario pode fazer muitas Compras e Vendas, mas cada Compra e Venda está associada a um único Usuario.
    - Implementada através de chaves estrangeiras nas tabelas Compra e Venda referenciando a tabela Usuario.
3. **NxN (Muitos para Muitos)**
    - Exemplo hipotético: se houvesse uma relação entre Produto e Categoria onde um Produto pode pertencer a muitas Categorias e uma Categoria pode ter muitos Produtos.
    - Implementação por uso de uma tabela intermediária (associativa), como ProdutoCategoria, com chaves estrangeiras referenciando Produto e Categoria.

Herança em Bancos de Dados Relacionais

- Utiliza-se a herança na modelagem das tabelas Pessoa, PessoaFisica e PessoaJuridica.
- A coluna discriminadora tipo na tabela Pessoa diferencia entre pessoas físicas (F) e jurídicas (J).
- Relacionamentos 1x1 entre Pessoa e PessoaFisica/PessoaJuridica são utilizados para representar a herança.

SQL Server Management Studio (SSMS)

- **Produtividade**:
    - Facilita a criação, modificação e gestão de bancos de dados através de uma interface gráfica intuitiva.
    - Ferramentas de design visual, como o Database Diagrams, permitem modelar visualmente as tabelas e seus relacionamentos.
    - Funcionalidades avançadas para escrever, depurar e executar scripts SQL.
    - Geração automática de scripts de criação e alteração de tabelas e outros objetos.
    - Análise de performance e otimização de consultas através do Query Analyzer.

Conclusão

A prática demonstra como modelar e criar um banco de dados relacional completo, utilizando SQL Server para gerenciar e implementar estruturas e relacionamentos complexos, garantindo a integridade e eficiência do sistema de compras e vendas de produtos. Através do uso de herança e cardinalidades apropriadas, é possível construir um modelo de dados robusto e flexível.