



Estácio

Universidade Estácio de Sá

Paulo amancio de souza

Desenvolvimento Full Stack

Criando o Banco de dados
Mundo 3

Palhoça
2024

Acesso Github:<https://github.com/robertoliveira1993/Projeto-Loja>

Título da Prática

Implementação de um Sistema de Cadastro de Usuários e Movimentação de Produtos

Objetivo da Prática

- Desenvolver um modelo de dados relacional para um sistema de cadastro de usuários, produtos, e movimentação de compras e vendas, utilizando o SQL Server Management Studio para criação e gerenciamento do banco de dados.

Desenvolvimento da Prática

```
CREATE DATABASE Loja;
```

```
CREATE SEQUENCE PessoaSeq
```

```
AS INT
```

```
START WITH 1
```

```
INCREMENT BY 1;
```

```
CREATE TABLE Usuarios (
```

```
    UsuarioID INT IDENTITY(1,1) PRIMARY KEY,
```

```
    NomeUsuario NVARCHAR(50) NOT NULL,
```

```
    Senha NVARCHAR(50) NOT NULL,
```

```
    Email NVARCHAR(100) NOT NULL
```

```
);
```

```
CREATE TABLE Pessoas (
```

```
    PessoaID INT PRIMARY KEY DEFAULT NEXT VALUE FOR PessoaSeq,
```

```
    TipoPessoa VARCHAR(10) CHECK (TipoPessoa IN ('Física', 'Jurídica')) NOT NULL,
```

```
    Nome VARCHAR(100) NOT NULL,
```

```
Endereco VARCHAR(200),  
Telefone VARCHAR(20),  
Email VARCHAR(100),  
CPF CHAR(11) NULL,  
CNPJ CHAR(14) NULL  
);
```

```
CREATE TABLE Produtos (  
    ProdutoID INT IDENTITY(1,1) PRIMARY KEY,  
    NomeProduto VARCHAR(100) NOT NULL,  
    Quantidade INT NOT NULL,  
    PrecoVenda DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE Compras (  
    CompraID INT IDENTITY(1,1) PRIMARY KEY,  
    UsuarioID INT NOT NULL,  
    PessoaID INT NOT NULL,  
    ProdutoID INT NOT NULL,  
    Quantidade INT NOT NULL,  
    PrecoUnitario DECIMAL(10, 2) NOT NULL,  
    DataCompra DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID),  
    FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID),  
    FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID)  
);
```

```

CREATE TABLE Vendas (

    VendaID INT IDENTITY(1,1) PRIMARY KEY,

    UsuarioID INT NOT NULL,

    PessoaID INT NOT NULL,

    ProdutoID INT NOT NULL,

    Quantidade INT NOT NULL,

    PrecoUnitario DECIMAL(10, 2) NOT NULL,

    DataVenda DATETIME DEFAULT GETDATE(),

    FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID),

    FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID),

    FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID)

);

CREATE LOGIN loja WITH PASSWORD = 'loja';

CREATE USER loja FOR LOGIN loja;

ALTER ROLE db_owner ADD MEMBER loja;

```

Resultados da Execução dos Códigos

Criação do Banco de Dados:

- O banco de dados **Loja** foi criado com sucesso.

Criação da Sequence e Tabelas:

- A sequence **PessoaSeq** e as tabelas **Usuarios**, **Pessoas**, **Produtos**, **Compras**, e **Vendas** foram criadas conforme o script.

Criação de Login e Usuário:

- O login **loja** foi criado e o usuário **loja** foi associado ao banco de dados **Loja** com permissões de **db_owner**.

Análise e Conclusão

Implementação das Diferentes Cardinalidades:

- **1x1 (um-para-um):** Utilizado para o relacionamento entre **Pessoas** e identificadores únicos, como no uso da sequence **PessoaSeq** para garantir que cada pessoa tenha um identificador único.
- **1xN (um-para-muitos):** Implementado em relacionamentos como **Usuarios** e **Compras**, onde um usuário pode fazer muitas compras.
- **NxN (muitos-para-muitos):** Para relacionamentos muitos-para-muitos, normalmente utilizamos uma tabela de junção que referencia as chaves primárias de ambas as tabelas relacionadas. Embora este exemplo não inclua diretamente um relacionamento NxN, isso pode ser estendido para outros cenários de aplicação.

Herança em Bancos de Dados Relacionais:

- **Relacionamento para Herança:** A herança em bancos de dados relacionais pode ser representada utilizando uma abordagem de tabela única, tabela por classe ou tabela de junção. No nosso exemplo, a diferenciação entre **Pessoa Física** e **Pessoa Jurídica** é feita através da coluna **TipoPessoa** e campos específicos (**CPF** e **CNPJ**), que é uma forma de representação de herança utilizando a abordagem de tabela única.