



1. Modelagem e Implementação de Banco de Dados Relacional com Herança e Cardinalidades

2. OBJETIVO DA PRÁTICA

O objetivo desta prática é projetar e implementar um banco de dados relacional utilizando o SQL Server, aplicando os conceitos de modelagem com **herança** e diferentes **cardinalidades** (1x1, 1xN, NxN), de forma a consolidar os conhecimentos teóricos com atividades práticas.

3. CÓDIGOS DESENVOLVIDOS

3.1 Criação da sequência:

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

3.2 Criação das tabelas:

Usuário

```
CREATE TABLE usuario (  
  idUsuario INT PRIMARY KEY IDENTITY,  
  nome VARCHAR(100) NOT NULL,  
  login VARCHAR(50) NOT NULL UNIQUE  
  senha VARCHAR(50) NOT NULL);
```

Pessoa

```
CREATE TABLE pessoa (  
  idPessoa INT PRIMARY KEY IDENTITY,  
  nome VARCHAR(100) NOT NULL,  
  endereco VARCHAR(100),  
  telefone VARCHAR(20),  
  email VARCHAR(1000);
```

Pessoa_Fisica

```
CREATE TABLE pessoa_fisica (  
  idPessoa INT PRIMARY KEY,  
  cpf CHAR(11) NOT NULL UNIQUE,  
  FOREIGN KEY (idPessoa) REFERENCES pessoa(idPessoa);
```

Pessoa_Juridica

```
CREATE TABLE pessoa_juridica (  
  idPessoa INT PRIMARY KEY,  
  cnpj CHAR(14) NOT NULL UNIQUE,  
  FOREIGN KEY (idPessoa) REFERENCES pessoa(idPessoa);
```

Produto

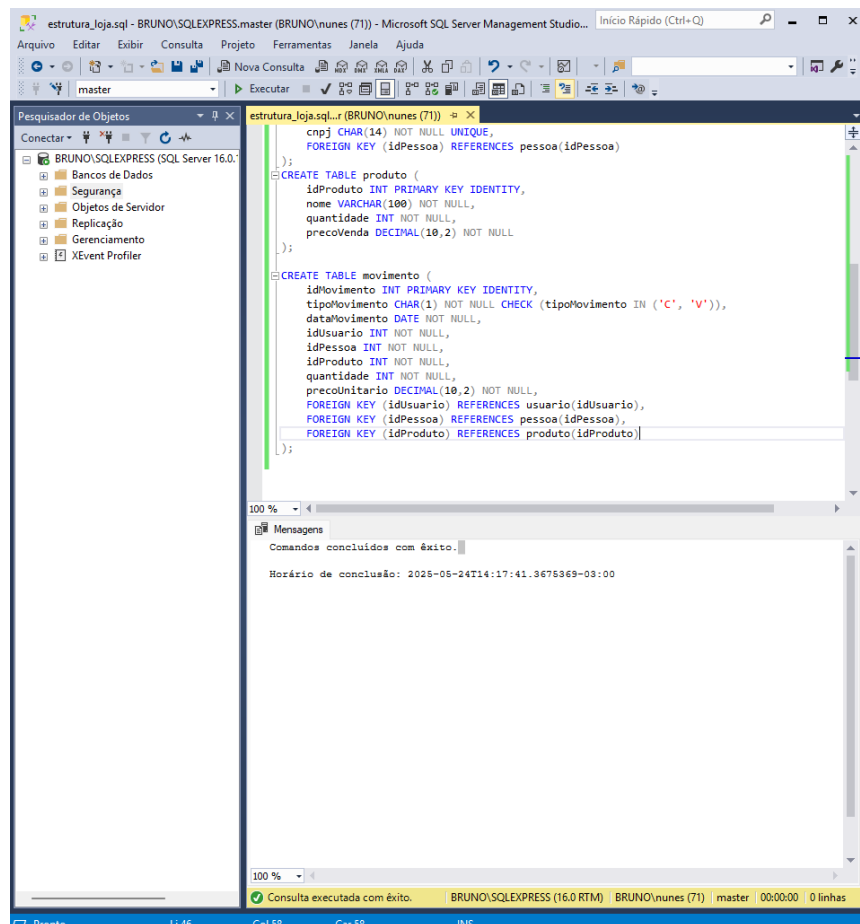
```
CREATE TABLE produto (  
  idProduto INT PRIMARY KEY IDENTITY,  
  nome VARCHAR(100) NOT NULL,  
  quantidade INT NOT NULL,  
  precoVenda DECIMAL(10,2) NOT NULL);
```

Movimento

```
CREATE TABLE movimento (  
  idMovimento INT PRIMARY KEY IDENTITY,  
  tipoMovimento CHAR(1) NOT NULL CHECK (tipoMovimento IN ('C', 'V')),  
  dataMovimento DATE NOT NULL,  
  idUsuario INT NOT NULL,  
  idPessoa INT NOT NULL,  
  idProduto INT NOT NULL,
```

quantidade INT NOT NULL,
precoUnitario DECIMAL(10,2) NOT NULL,
FOREIGN KEY (idUserio) REFERENCES usuario(idUsuario),
FOREIGN KEY (idPessoa) REFERENCES pessoa(idPessoa),
FOREIGN KEY (idProduto) REFERENCES produto(idProduto);

4. RESULTADOS DA EXECUÇÃO



5. ANÁLISE E CONCLUSÃO

5.1 Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

- **1x1:** Relacionamentos 1 para 1 são implementados usando chaves primárias e estrangeiras que compartilham o mesmo valor. Exemplo: `pessoa` e `pessoa_fisica`, onde a `idPessoa` é chave primária e estrangeira.
- **1xN (um para muitos):** Um registro de uma tabela se relaciona com vários da outra. Exemplo: `usuario` possui vários `movimentos` →

movimento.Usuario_idUsuario é chave estrangeira para
usuario.idUsuario.

- **NxN (muitos para muitos):** Implementado por meio de uma **tabela intermediária**. Exemplo: item_movimentado relaciona produto e movimento.

5.2 Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

A herança é implementada por meio da **técnica de generalização/especialização**, criando:

- Uma tabela principal (pessoa) com os atributos comuns.
- Tabelas filhas (pessoa_fisica, pessoa_juridica) com os atributos específicos e referência à chave primária da tabela pai.

5.3 Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?

O SQL Server Management Studio (SSMS) contribui para a produtividade com:

- Interface gráfica amigável para modelagem de tabelas, relacionamentos e visualização de dados.
- Recursos de **auto-complete**, execução de scripts e verificação de erros em tempo real.
- Ferramentas de **geração automática de scripts**, backups, restauração e análise de desempenho.
- Suporte a **intellisense** e **documentação visual**.