

# **Prática: RPG0015 - Vamos manter as informações!**

## **Título da Prática:**

Modelagem e Implementação de um Banco de Dados Simples

## **Objetivo da Prática:**

Desenvolver habilidades em modelagem de dados, criação de banco de dados relacional e execução de consultas SQL utilizando SQL Server.

## **Modelagem do Banco de Dados**

Modelo Entidade-Relacionamento

O modelo de banco de dados foi elaborado com as seguintes entidades e relacionamentos:

1. Pessoa: Base comum para pessoas físicas e jurídicas.
2. Usuário: Armazena dados de login para operadores.
3. Produto: Informações sobre itens comercializados.
4. Movimentação: Histórico de entradas e saídas de produtos.

Relacionamentos:

- Pessoa Física e Pessoa Jurídica derivam de Pessoa (herança 1:1).
- Movimentação referencia Produto e Usuário.

## **Scripts SQL**

Criação do Banco de Dados e Tabelas:

```
CREATE DATABASE LojaDB;
```

GO

USE LojaDB;

GO

-- Tabela Pessoa

```
CREATE TABLE Pessoa (  
    id_pessoa INT PRIMARY KEY,  
    nome NVARCHAR(100),  
    endereco NVARCHAR(200),  
    contato NVARCHAR(100),  
    tipo CHAR(1) -- F para Física, J para Jurídica  
);
```

-- Tabela Pessoa Física

```
CREATE TABLE PessoaFisica (  
    id_pessoa INT PRIMARY KEY,  
    cpf CHAR(11),  
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)  
);
```

-- Tabela Pessoa Jurídica

```
CREATE TABLE PessoaJuridica (  
    id_pessoa INT PRIMARY KEY,  
    cnpj CHAR(14),  
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)  
);
```

-- Tabela Usuário

```
CREATE TABLE Usuario (  
    id_usuario INT IDENTITY PRIMARY KEY,  
    username NVARCHAR(50),  
    senha NVARCHAR(50)  
);
```

-- Tabela Produto

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

-- Tabela Movimentação

```
CREATE TABLE Movimentacao (  
    id_movimentacao INT IDENTITY PRIMARY KEY,  
    tipo CHAR(1), -- E para entrada, S para saída  
    id_usuario INT,  
    id_produto INT,  
    quantidade INT,  
    preco_unitario DECIMAL(10, 2),  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),  
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)  
);  
  
GO
```

-- Inserção de Dados (Exemplo resumido)

```
INSERT INTO Usuario (username, senha) VALUES ('op1', 'op1'), ('op2', 'op2');
```

```
INSERT INTO Produto (nome, quantidade, preco_venda) VALUES ('Produto A', 100, 10.00);
```

## Consultas SQL

1. Dados completos de pessoas físicas:

```
SELECT p.*, pf.cpf
```

```
FROM Pessoa p
```

```
INNER JOIN PessoaFisica pf ON p.id_pessoa = pf.id_pessoa;
```

2. Valor total das entradas por produto:

```
SELECT p.nome, SUM(m.quantidade * m.preco_unitario) AS total
```

```
FROM Movimentacao m
```

```
INNER JOIN Produto p ON m.id_produto = p.id_produto
```

```
WHERE m.tipo = 'E'
```

```
GROUP BY p.nome;
```

## Análise e Conclusão

- A herança foi implementada por meio de relacionamentos 1:1 entre a tabela Pessoa e as tabelas PessoaFisica e PessoaJuridica.

- Relacionamentos 1:N foram utilizados entre Usuários e Movimentações.

## Git

Repositório do projeto disponível em: **[\*\*Link do Repositório\*\*]**