

**Campus:** Estácio EAD  
**Curso:** Desenvolvimento Full Stack  
**Disciplina:** Nível 2: Vamos Manter as Informações?  
**Turma:** 3º Semestre  
**Semestre Letivo:** 2025.1  
**Integrantes:**

**202310043203 -** Felipe Cavalcante de Oliveira

**Título da Prática**

**Modelagem e Implementação de Banco de Dados para Controle de Compras e Vendas com SQL Server**

**Objetivo da Prática**

Desenvolver a modelagem e implementação de um sistema de banco de dados que permita o controle de compras e vendas de produtos, com distinção entre pessoas físicas e jurídicas, operadores do sistema, e aplicação de conceitos como chaves primárias, estrangeiras, identidade e relacionamento.  
  
CREATE TABLE usuarios (

id\_usuario INT PRIMARY KEY IDENTITY(1,1),

nome VARCHAR(100),

email VARCHAR(100),

senha VARCHAR(100)

);

CREATE TABLE pessoas (

id\_pessoa INT PRIMARY KEY IDENTITY(1,1),

nome VARCHAR(100),

endereco VARCHAR(150),

telefone VARCHAR(20),

email VARCHAR(100),

tipo\_pessoa CHAR(1) -- 'F' ou 'J'

);

CREATE TABLE pessoas\_fisicas (

id\_pessoa INT PRIMARY KEY,

cpf VARCHAR(14),

FOREIGN KEY (id\_pessoa) REFERENCES pessoas(id\_pessoa)

);

CREATE TABLE pessoas\_juridicas (

id\_pessoa INT PRIMARY KEY,

cnpj VARCHAR(18),

FOREIGN KEY (id\_pessoa) REFERENCES pessoas(id\_pessoa)

);

CREATE TABLE produtos (

id\_produto INT PRIMARY KEY IDENTITY(1,1),

nome VARCHAR(100),

quantidade INT,

preco\_venda DECIMAL(10,2)

);

CREATE TABLE compras (

id\_compra INT PRIMARY KEY IDENTITY(1,1),

id\_usuario INT,

id\_produto INT,

id\_pessoa\_juridica INT,

quantidade INT,

preco\_unitario DECIMAL(10,2),

data\_compra DATETIME DEFAULT GETDATE(),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario),

FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto),

FOREIGN KEY (id\_pessoa\_juridica) REFERENCES pessoas\_juridicas(id\_pessoa)

);

CREATE TABLE vendas (

id\_venda INT PRIMARY KEY IDENTITY(1,1),

id\_usuario INT,

id\_produto INT,

id\_pessoa\_fisica INT,

quantidade INT,

preco\_unitario DECIMAL(10,2),

data\_venda DATETIME DEFAULT GETDATE(),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario),

FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto),

FOREIGN KEY (id\_pessoa\_fisica) REFERENCES pessoas\_fisicas(id\_pessoa)

### ); Resultados da Execução dos Códigos

* Tabelas criadas com sucesso no SQL Server.
* Relacionamentos entre as entidades corretamente definidos com **chaves estrangeiras**.
* Testes de inserção realizados com dados fictícios, validando integridade relacional.
* Sistema pronto para registrar movimentações de compra (com PJ) e venda (com PF).

### ****Análise e Conclusão****

#### ▪ Quais as diferenças no uso de SEQUENCE e IDENTITY?

* **IDENTITY** é associado diretamente a uma coluna da tabela, gerando valores incrementais automaticamente ao inserir registros.
* **SEQUENCE** é um objeto separado do banco, que pode ser compartilhado entre tabelas e usado com mais flexibilidade, especialmente quando é necessário gerar números fora do processo de inserção.

#### ▪ Qual a importância das chaves estrangeiras para a consistência do banco?

* Garantem a **integridade referencial**, evitando que registros de tabelas dependentes fiquem órfãos (por exemplo, uma venda sem produto válido).
* São fundamentais para a lógica de relacionamento entre entidades e para manter o banco confiável e sem inconsistências.

#### ▪ Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?

* **Álgebra Relacional:**
  + SELECT, PROJECT (colunas), JOIN, UNION, INTERSECT, DIFFERENCE, CARTESIAN PRODUCT.
* **Cálculo Relacional:**
  + Linguagens baseadas em lógica (como o WHERE em SQL), com variáveis de tupla e condições booleanas.

#### ▪ Como é feito o agrupamento em consultas, e qual requisito é obrigatório?

* Usando o comando GROUP BY.
* É **obrigatório** que todas as colunas selecionadas (que não sejam agregadas com COUNT(), SUM(), AVG() etc.) estejam presentes no GROUP BY.

**Repositório Git:**  
  
https://github.com/FelipeCavalcante77/SISTEMALOJA\_BDO