Estácio

Campus: Polo Centro – Nilópolis – RJ
Curso: Desenvolvimento Full stack
Disciplina: Vamos manter as informações!
N° Turma: 9001
Semestre Letivo: 2025.1
Integrante: Davison Rodrigues Barbosa

Título da Prática

RPG0015 - Vamos manter as informações!

Objetivo da Prática

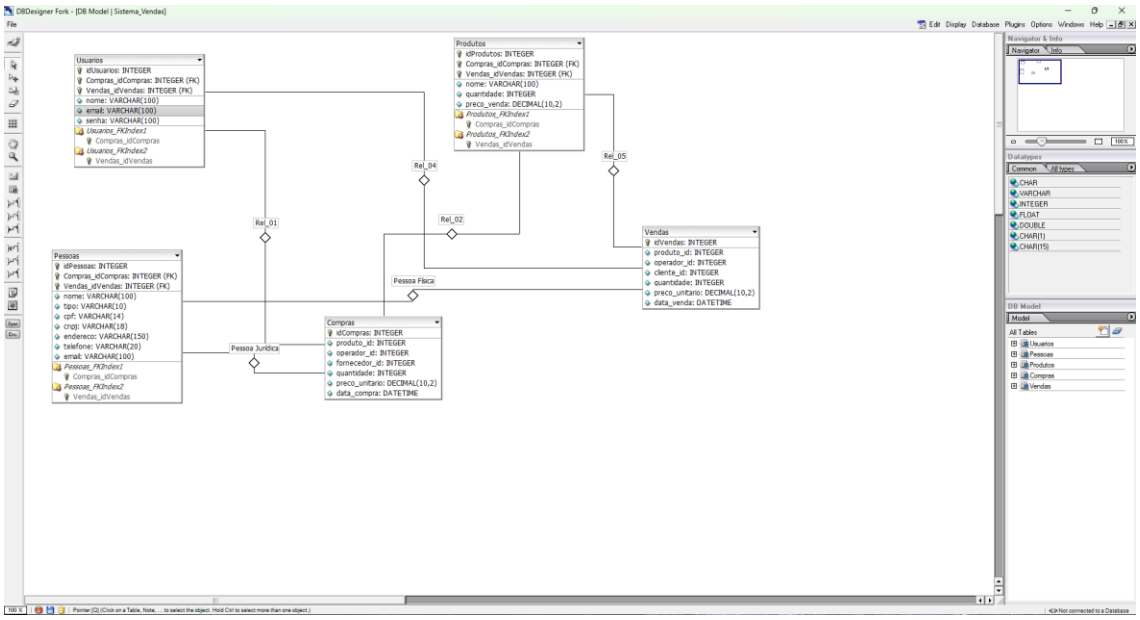
1. Identificar os requisitos de um sistema e transformá-los no modelo adequado.
2. Utilizar ferramentas de modelagem para bases de dados relacionais.
3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
4. Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
5. No final do exercício, o aluno terá vivenciado a experiência de modelar a base de
6. dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na
7. plataforma do SQL Server.

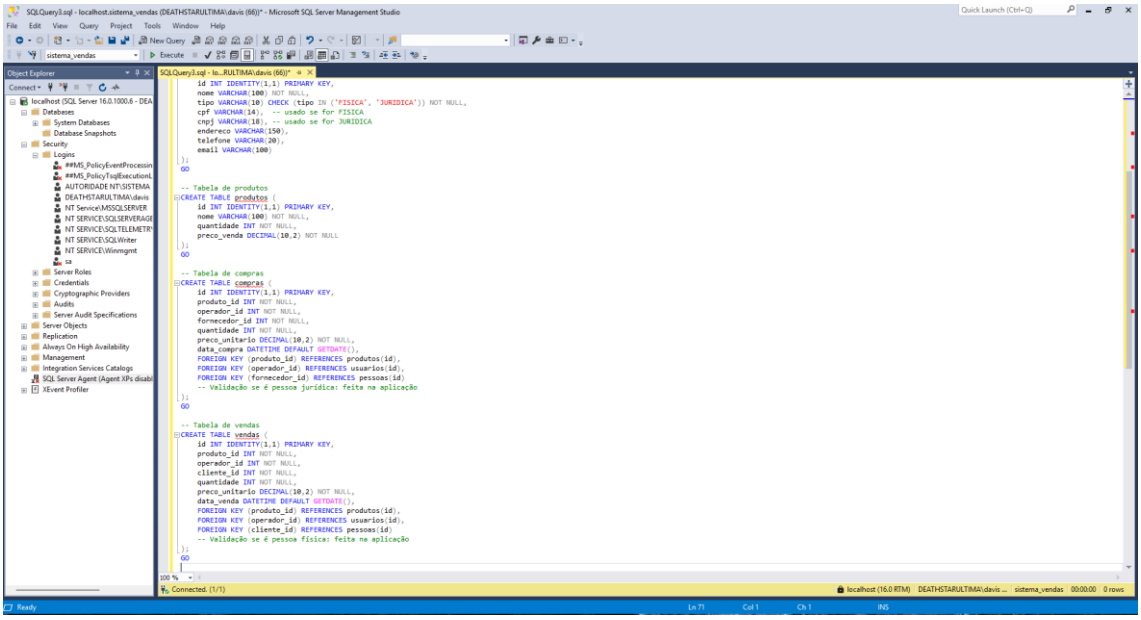
Código-Fonte

```
CREATE DATABASE sistema_vendas;
GO

-- Usar o banco criado
USE sistema_vendas;
GO

-- Tabela de usuários (operadores)
CREATE TABLE usuarios (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(100) NOT NULL
);
GO

-- Tabela de pessoas (física ou jurídica)
CREATE TABLE pessoas (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
```

```
    tipo VARCHAR(10) CHECK (tipo IN ('FISICA', 'JURIDICA')) NOT NULL,
    cpf VARCHAR(14),  -- usado se for FISICA
    cnpj VARCHAR(18), -- usado se for JURIDICA
    endereco VARCHAR(150),
    telefone VARCHAR(20),
    email VARCHAR(100)
);
GO

-- Tabela de produtos
CREATE TABLE produtos (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    quantidade INT NOT NULL,
    preco_venda DECIMAL(10,2) NOT NULL
);
GO

-- Tabela de compras
CREATE TABLE compras (
    id INT IDENTITY(1,1) PRIMARY KEY,
    produto_id INT NOT NULL,
    operador_id INT NOT NULL,
    fornecedor_id INT NOT NULL,
    quantidade INT NOT NULL,
    preco_unitario DECIMAL(10,2) NOT NULL,
    data_compra DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (produto_id) REFERENCES produtos(id),
    FOREIGN KEY (operador_id) REFERENCES usuarios(id),
    FOREIGN KEY (fornecedor_id) REFERENCES pessoas(id)
    -- Validação se é pessoa jurídica: feita na aplicação
);
GO

-- Tabela de vendas
CREATE TABLE vendas (
    id INT IDENTITY(1,1) PRIMARY KEY,
    produto_id INT NOT NULL,
    operador_id INT NOT NULL,
    cliente_id INT NOT NULL,
    quantidade INT NOT NULL,
    preco_unitario DECIMAL(10,2) NOT NULL,
    data_venda DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (produto_id) REFERENCES produtos(id),
    FOREIGN KEY (operador_id) REFERENCES usuarios(id),
    FOREIGN KEY (cliente_id) REFERENCES pessoas(id)
    -- Validação se é pessoa física: feita na aplicação
);
GO
```

Resultados da Execução





Análise e Conclusão

- **Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?**
  Cardinalidade descreve quantas ocorrências de uma entidade se relacionam com quantas da outra.

  Os principais tipos:

Tipo	Nome técnico	Exemplo simples
1:1	Um para Um	Pessoa e RG
1:N	Um para Muitos	Cliente e Pedidos
N:N	Muitos para Muitos	Alunos e Disciplinas

- **Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?**
  A herança em bancos de dados relacionais não é nativa como nas linguagens orientadas a objetos, mas pode ser simulada com algumas abordagens de modelagem. Cada abordagem representa um tipo de relacionamento entre tabelas.

- **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) é uma ferramenta completa e poderosa que aumenta muito a produtividade no gerenciamento de bancos de dados SQL Server. Ele é como um "canivete suíço" para desenvolvedores, DBAs e estudantes.

Repositório no Git

https://github.com/DavisRodriguesB/Missao_Pratica_Nivel_2-_Mundo_3