

Campus: POLO VILA DOS REMÉDIOS – OSASCO – SP

Curso: Desenvolvimento Full Stack

Nome Disciplina: Nível 2: Vamos Manter as Informações?

Matrícula: 2023 0397 9797

Semestre Letivo: 3º semestre

Nome: Anderson Barbosa Almeida

Repositório no GIT:

<https://github.com/andydevbarbosa/andydevbarbosa/blob/main/RPG0015>

[%20-](#)

[%20Vamos%20manter%20as%20informa%C3%A7%C3%B5es!/Criando%20o%20Banco%20de%20Dados.sql](#)

Título da Prática: Criando Banco de Dados

Objetivo da Prática:

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

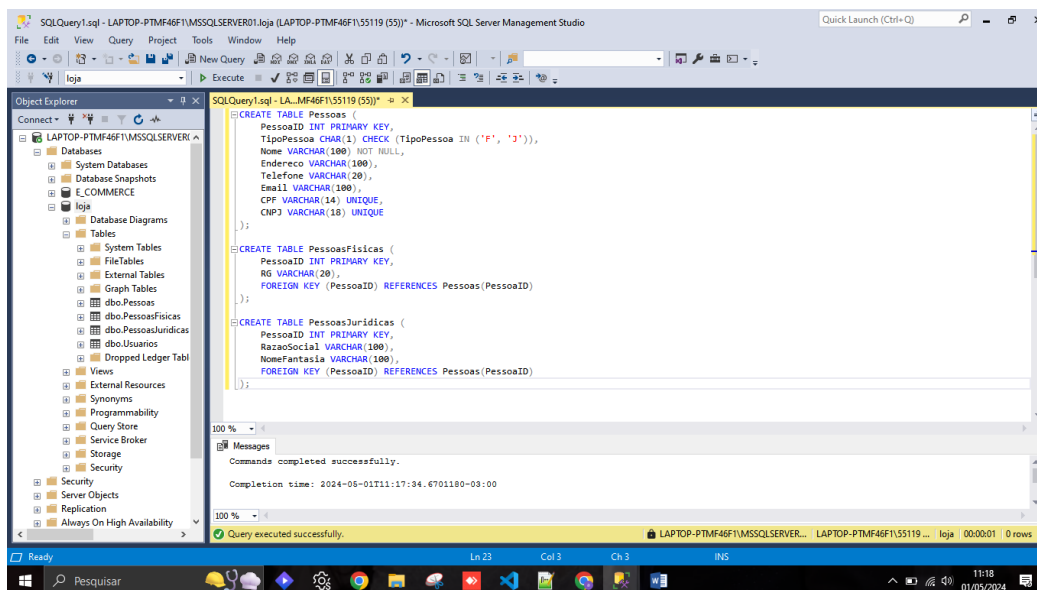
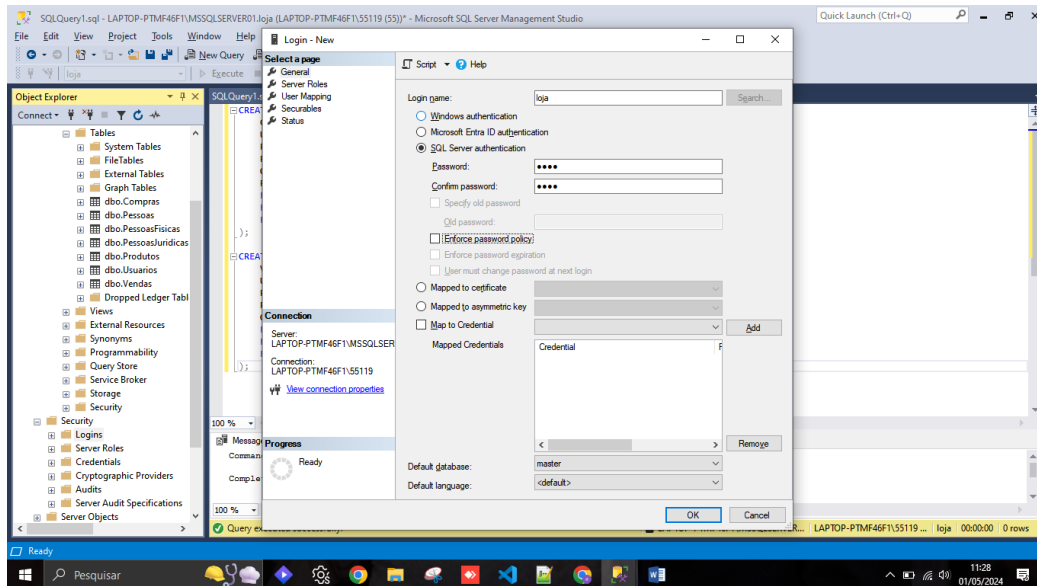
Todos os códigos solicitados nesse roteiro:

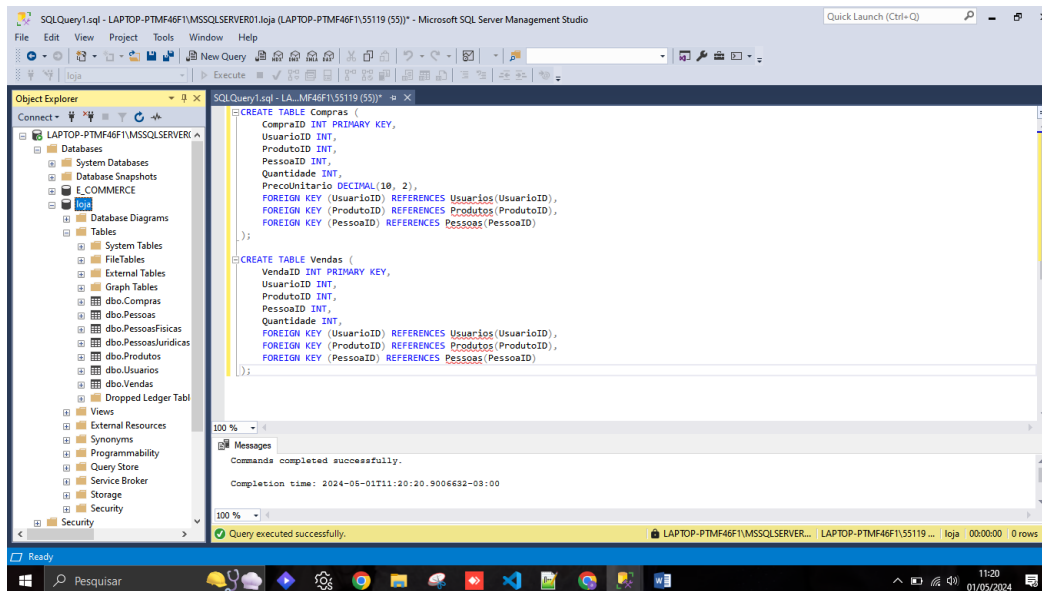
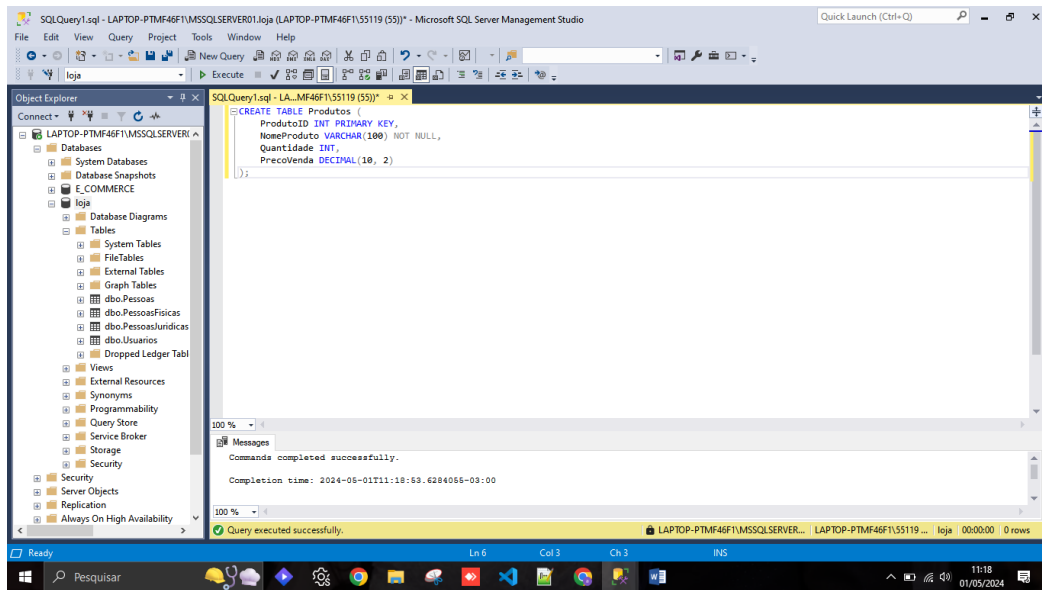
```
2. CREATE TABLE Usuarios (  
3.     UsuarioID INT PRIMARY KEY,  
4.     NomeUsuario VARCHAR(50) UNIQUE NOT NULL,  
5.     Senha VARCHAR(100) NOT NULL,  
6.     NomeCompleto VARCHAR(100),  
7.     Email VARCHAR(100)  
8. );  
9.  
10. -- Criar tabela de pessoas  
11. CREATE TABLE Pessoas (  
12.     PessoaID INT PRIMARY KEY,  
13.     TipoPessoa CHAR(1) CHECK (TipoPessoa IN ('F', 'J')),  
14.     Nome VARCHAR(100) NOT NULL,  
15.     Endereco VARCHAR(100),  
16.     Telefone VARCHAR(20),  
17.     Email VARCHAR(100),  
18.     CPF VARCHAR(14) UNIQUE,  
19.     CNPJ VARCHAR(18) UNIQUE  
20. );  
21.
```

```
22.-- Criar tabela de pessoas físicas
23.CREATE TABLE PessoasFisicas (
24.  PessoaID INT PRIMARY KEY,
25.  RG VARCHAR(20),
26.  FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID)
27.);
28.
29.-- Criar tabela de pessoas jurídicas
30.CREATE TABLE PessoasJuridicas (
31.  PessoaID INT PRIMARY KEY,
32.  RazaoSocial VARCHAR(100),
33.  NomeFantasia VARCHAR(100),
34.  FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID)
35.);
36.
37.-- Criar tabela de produtos
38.CREATE TABLE Produtos (
39.  ProdutoID INT PRIMARY KEY,
40.  NomeProduto VARCHAR(100) NOT NULL,
41.  Quantidade INT,
42.  PrecoVenda DECIMAL(10, 2)
43.);
44.
45.-- Criar tabela de compras
46.CREATE TABLE Compras (
47.  CompralID INT PRIMARY KEY,
48.  UsuarioID INT,
49.  ProdutoID INT,
50.  PessoaID INT,
51.  Quantidade INT,
52.  PrecoUnitario DECIMAL(10, 2),
53.  FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID),
54.  FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID),
55.  FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID)
56.);
57.
58.-- Criar tabela de vendas
59.CREATE TABLE Vendas (
60.  VendaID INT PRIMARY KEY,
61.  UsuarioID INT,
62.  ProdutoID INT,
63.  PessoaID INT,
64.  Quantidade INT,
65.  FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID),
66.  FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID),
67.  FOREIGN KEY (PessoaID) REFERENCES Pessoas(PessoaID)
68.);
```



Resultados de execução do Códigos:





Análise e Conclusão:

- **Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NXN, em um banco de dados relacional?**

As diferentes cardinalidades em um banco de dados relacional são implementadas através de chaves estrangeiras que estabelecem os relacionamentos entre as tabelas.

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

Para representar o uso de herança em bancos de dados relacionais, geralmente é utilizado um tipo de relacionamento chamado "tabela de união" ou "tabela de subtipo". Neste caso, uma tabela principal contém os atributos compartilhados por todas as entidades, enquanto tabelas secundárias contêm os atributos específicos de cada tipo de entidade. As tabelas secundárias têm uma relação 1 para 1 com a tabela principal.

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

Interface gráfica intuitiva para gerenciar banco de dados, tabelas, procedimentos armazenados, visões, etc.

Editor de consultas SQL avançado com destaque de sintaxe e sugestões automáticas.

Capacidade de visualizar e modificar a estrutura do banco de dados de forma visual.

Ferramentas de administração, como backup e restauração de banco de dados, otimização de consultas, monitoramento de desempenho, entre outros.

Integração com outras ferramentas da Microsoft, como Visual Studio e Azure, facilitando o desenvolvimento e a implantação de aplicativos.