	<p>Universidade Estácio</p> <p>Polo São Lourenço da Mata</p> <p>Desenvolvimento Full Stack</p> <p>Semestre 2024.1</p>	<p>Disciplina: Vamos Manter as Informações?</p> <p>Aluno: Manoel José Matrícula: 202301361117 Turma: 2023.1</p>
---	---	---

Modelagem e implementação de um banco de dados simples, utilizando como base o SQL Server.

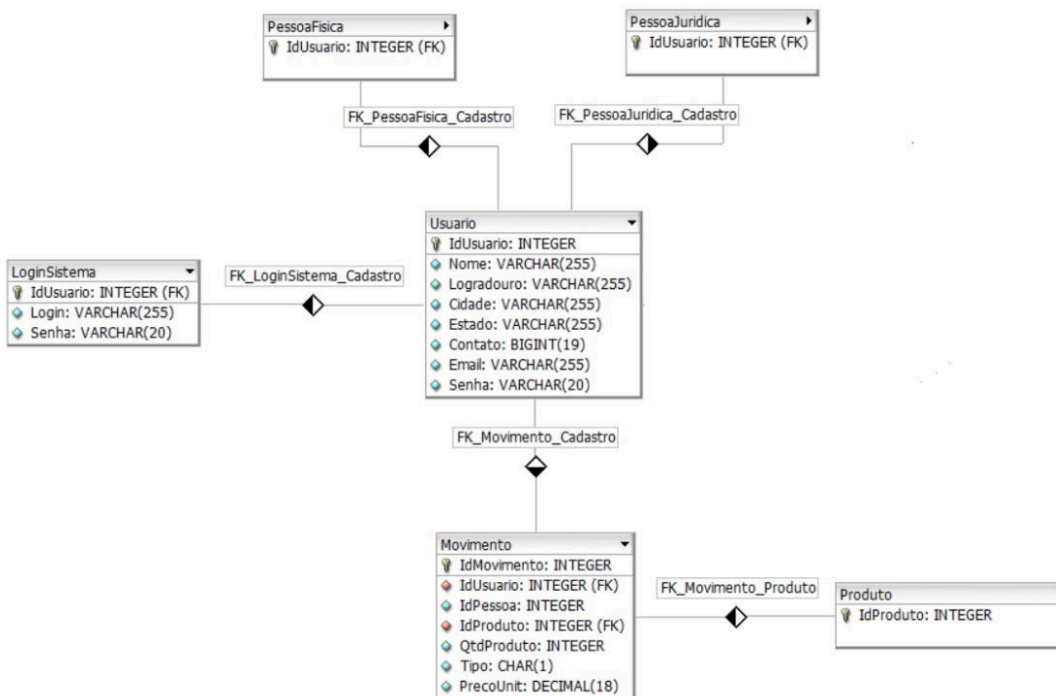
Objetivos da Prática:

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 dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server.

1º Procedimento | Criando o Banco de Dados

Todos os códigos solicitados neste roteiro de aula:

1 Modelo Relacional (DBDesigner):



2- Criando o Banco de Dados e as Tabelas com suas Colunas e Tipos de Dados:

-- Cria um novo banco de dados chamado "loja"

```
CREATE DATABASE loja4;
```

```
GO
```

-- Usa o banco de dados recém-criado

```
USE loja;
```

```
GO
```

```
CREATE TABLE Cadastro (
    IdUsuario INT PRIMARY KEY,
    Nome NVARCHAR(255),
    Localizacao VARCHAR(255),
    Contato BIGINT,
    Email VARCHAR(255),
    Senha VARCHAR (20),
    TipoPessoa CHAR(1),
    CPF BIGINT,
    CNPJ BIGINT );
```

```
CREATE TABLE LoginSistema(  
    IdUsuario INT,  
    Senha VARCHAR(20),  
    TipoPessoa CHAR(1) );
```

```
CREATE TABLE CPF (  
    CPF BIGINT,  
    MovCompra INT PRIMARY KEY);
```

```
CREATE TABLE CNPJ (  
    CNPJ BIGINT,  
    MovVenda INT PRIMARY KEY);
```

```
CREATE TABLE Mov_Compra (  
    IdCompra INT PRIMARY KEY,  
    IdProduto INT);
```

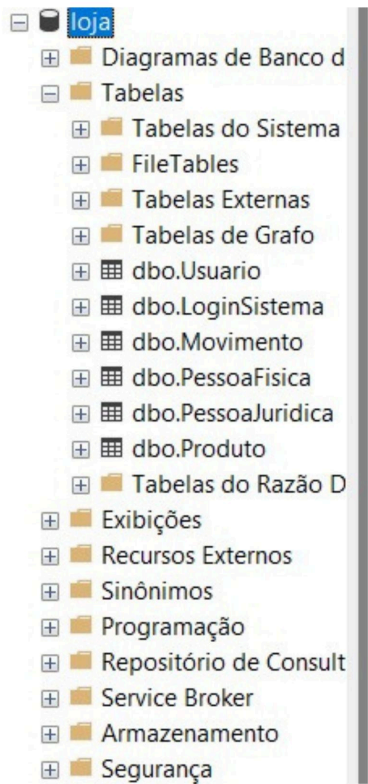
```
CREATE TABLE MovVenda (  
    IdVenda INT PRIMARY KEY,  
    IdProduto INT );
```

```
CREATE TABLE Produto (  
    IdProduto INT PRIMARY KEY,  
    NomeProduto VARCHAR(255),  
    QtdProduto INT,  
    PrecoVenda DECIMAL );
```

GO

3- Utilizando o SQL Server Management Studio para criar a base de dados modelada:

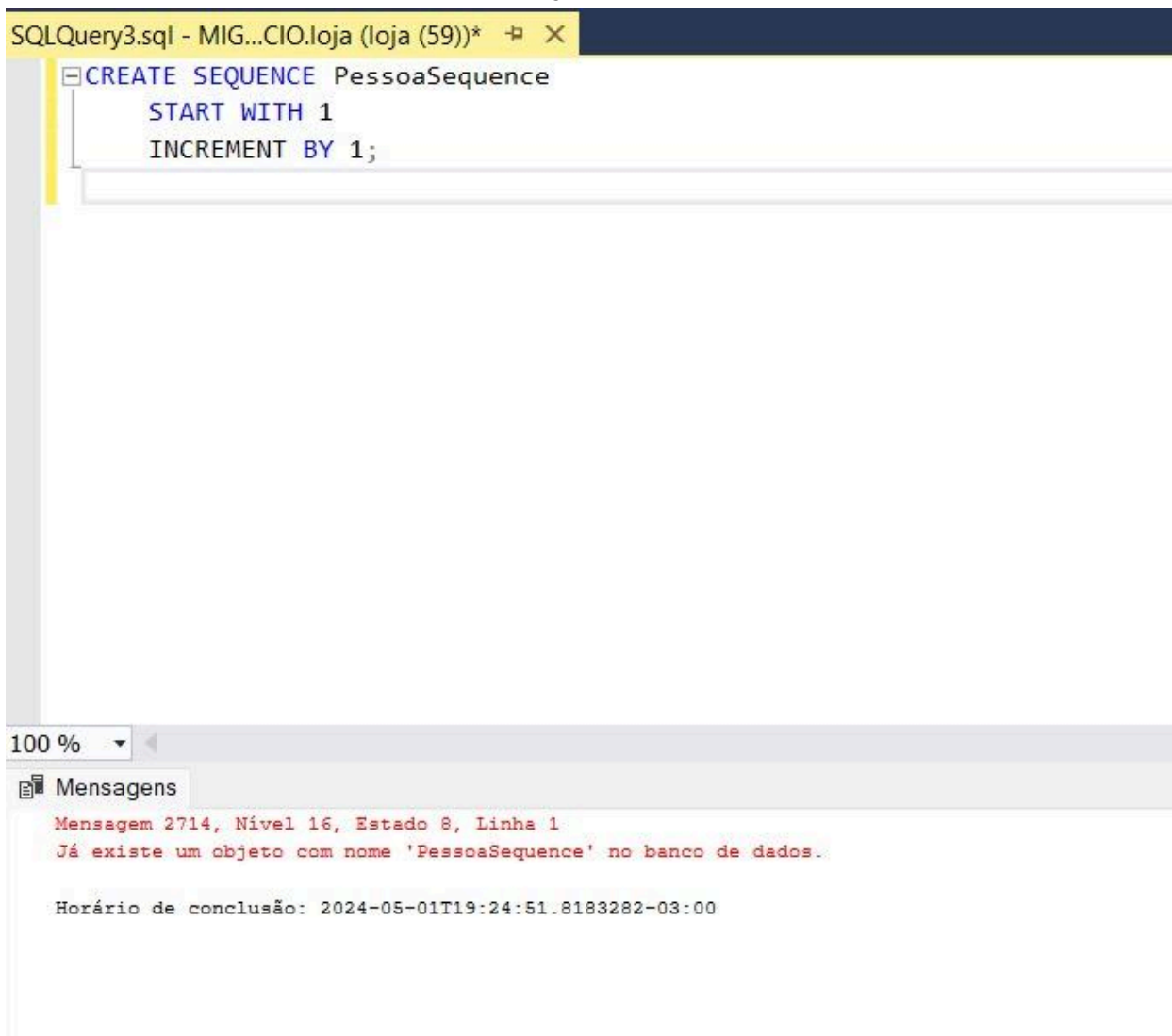
- **Banco e Tabelas:**



- **Tabelas e Colunas:**

- [-] **dbo.Usuario**
 - [-] Colunas
 - 🔑 IdUsuario (PK, int, não nulo)
 - 📄 Nome (varchar(255), não nulo)
 - 📄 Logradouro (varchar(255), não nulo)
 - 📄 Cidade (varchar(255), não nulo)
 - 📄 Estado (varchar(255), não nulo)
 - 📄 Contato (bigint, não nulo)
 - 📄 Email (varchar(255), não nulo)
 - 📄 Senha (varchar(20), não nulo)
 - [+] Chaves
 - [+] Restrições
 - [+] Gatilhos
 - [+] Índices
 - [+] Estatísticas
- [-] **dbo.LoginSistema**
 - [-] Colunas
 - 🔑 IdUsuario (PK, FK, int, não nulo)
 - 📄 Login (varchar(255), não nulo)
 - 📄 Senha (varchar(20), não nulo)
 - [+] Chaves
 - [+] Restrições
 - [+] Gatilhos
 - [+] Índices
 - [+] Estatísticas
- [-] **dbo.Movimento**
 - [-] Colunas
 - 🔑 IdMovimento (PK, int, não nulo)
 - 🔗 IdUsuario (FK, int, não nulo)
 - 📄 IdPessoa (int, não nulo)
 - 🔗 IdProduto (FK, int, não nulo)
 - 📄 QtdProduto (int, não nulo)
 - 📄 Tipo (char(1), não nulo)
 - 📄 PrecoUnit (decimal(18,0), não nulo)
 - [+] Chaves
 - [+] Restrições
 - [+] Gatilhos
 - [+] Índices
 - [+] Estatísticas
- [-] **dbo.PessoaFisica**
 - [-] Colunas
 - 🔑 IdUsuario (PK, FK, int, não nulo)
 - 📄 CPF (bigint, não nulo)
 - [+] Chaves
 - [+] Restrições
 - [+] Gatilhos
 - [+] Índices
 - [+] Estatísticas
- [-] **dbo.PessoaJuridica**
 - [-] Colunas
 - 🔑 IdUsuario (PK, FK, int, não nulo)
 - 📄 CNPJ (bigint, não nulo)
 - [+] Chaves
 - [+] Restrições

4- Definindo uma sequence para geração dos identificadores de pessoa, dado o relacionamento 1x1 com pessoa física ou jurídica:



The screenshot shows a SQL query editor window titled "SQLQuery3.sql - MIG...CIO.loja (loja (59))*". The query text is:

```
CREATE SEQUENCE PessoaSequence  
START WITH 1  
INCREMENT BY 1;
```

Below the query editor, there is a "Mensagens" (Messages) pane. It displays an error message in red text:

Mensagem 2714, Nível 16, Estado 8, Linha 1
Já existe um objeto com nome 'PessoaSequence' no banco de dados.

At the bottom of the messages pane, it shows the completion time: "Horário de conclusão: 2024-05-01T19:24:51.8183282-03:00".

Obs.: Eu executei o script e fechei logo em seguida, por isso que o script acima não exibe o resultado de execução mostrando sucesso.

Análise e Conclusão:

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

R- As diferentes cardinalidades em um banco de dados relacional são implementadas através do uso de chaves estrangeiras e chaves primárias, bem como da estruturação das tabelas de banco de dados.

1 para 1 (1x1):

Nesse tipo de relação, uma entrada em uma tabela está associada a exatamente uma entrada em outra tabela e vice-versa.

1 para muitos (1xN):

Aqui, uma entrada em uma tabela pode estar associada a várias entradas em outra tabela, mas cada entrada nesta outra tabela está associada a apenas uma entrada na primeira tabela.

Muitos para muitos (NxN):

Este é um pouco mais complexo e requer uma tabela intermediária (também conhecida como tabela de junção ou de associação) para mapear as relações entre as duas tabelas.

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

R- Para representar o conceito de herança em bancos de dados relacionais, geralmente é usado o modelo de herança de tipo, que é uma técnica de modelagem de dados que permite representar hierarquias de classes ou tipos.

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

R- O SQL Server Management Studio (SSMS) oferece várias ferramentas e recursos que podem melhorar significativamente a produtividade nas tarefas relacionadas ao gerenciamento de banco de dados. Aqui estão alguns dos principais benefícios:

Interface Integrada e Intuitiva: O SSMS fornece uma interface gráfica de usuário (GUI) bem projetada e fácil de usar, que permite aos usuários realizar uma ampla gama de tarefas de gerenciamento de banco de dados sem a necessidade de escrever consultas SQL manualmente.

Edição de Esquema Visual: O SSMS permite visualizar e editar facilmente o esquema do banco de dados usando designers visuais intuitivos para tabelas, procedimentos armazenados, funções, visões e outros objetos do banco de dados.

Ferramentas de Desenvolvimento: O SSMS inclui um conjunto abrangente de ferramentas de desenvolvimento, como um editor de consultas com realce de sintaxe, autocompletar e sugestões de código, que ajudam os desenvolvedores a escrever e depurar consultas SQL de forma eficiente.

Monitoramento e Otimização de Desempenho: O SSMS oferece recursos de monitoramento de desempenho integrados, como o Monitor de Atividade e o SQL Profiler, que permitem aos administradores de banco de dados identificar e resolver rapidamente problemas de desempenho do servidor SQL.

Tarefas de Administração Automatizadas: O SSMS permite automatizar várias tarefas de administração por meio de scripts SQL, Agentes do SQL Server, Jobs e integração com o PowerShell, economizando tempo e reduzindo erros.

Integração com Outras Ferramentas: O SSMS se integra perfeitamente com outras ferramentas e tecnologias da Microsoft, como o Azure SQL Database, o SQL Server Analysis Services (SSAS), o SQL Server Reporting Services (SSRS) e o SQL Server Integration Services (SSIS), facilitando o desenvolvimento e a administração de soluções de dados complexas.