



**Campos :** Polo Barbosa II - Marília - SP

**Curso :** Desenvolvedor Full Stack

**Turma :** 2023.3

**Integrante :** Rafael Leal Altero

## 1º Procedimento :

Criando um banco de dados para um sistema de comércio eletrônico

**Objetivo da Prática :**

- 1 - Criar um modelo de dados para um sistema de comércio eletrônico Utilizando o DER (Diagrama Entidade-Relacionamento);
- 2 - Criar um banco de dados para o sistema de comércio eletrônico usando o SQL Server Management Studio;
- 3 - Definir uma SEQUENCE para Pessoa
- 4 - Definir um IDENTITY para geração dos identificadores para todas as tabelas que tem o id\_ \* PRIMARY KEY.

segue o **LINK** 🖱️ **Codigos**

**Códigos solicitados no roteiro de aula:**

**Link GitHub :**

**main:**

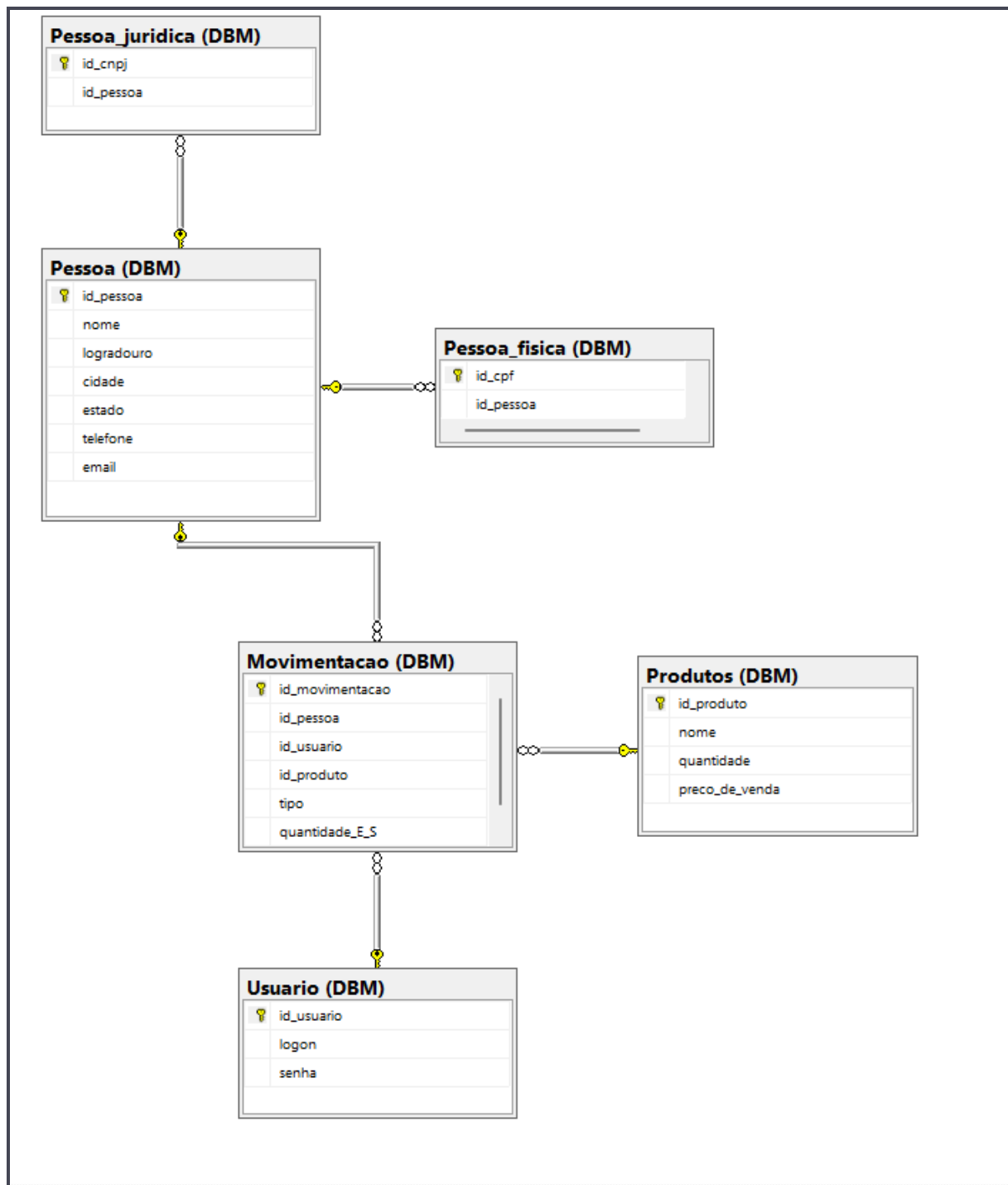
<https://github.com/Rafa1a/CadastroPOO/blob/main/src/cadastropoo/CadastroPOO.java>,

**models:**

<https://github.com/Rafa1a/CadastroPOO/tree/main/src/model>

Resultados da execução dos códigos:

DER :



## CREATE TABLES :

The screenshot displays the SQL Server Enterprise Manager interface on the left and a SQL Query window on the right. The Enterprise Manager shows a server named 'RAFAEL (SQL Server 16.0.1000.6 - loja)' with a database 'DBM' under the 'Bancos de Dados' folder. The 'Tabelas' (Tables) folder is expanded, showing a list of tables including 'DBM.Pessoa', 'DBM.Pessoa\_fisica', 'DBM.Pessoa\_juridica', and 'DBM.Produtos'.

The SQL Query window shows the following T-SQL code for creating the tables:

```
CREATE SCHEMA DBM
GO

--Pessoa
CREATE TABLE DBM.Pessoa
(
    id_pessoa int IDENTITY(1,1) PRIMARY KEY,
    nome varchar(50) not null,
    logradouro varchar(50) not null,
    cidade varchar(20) not null,
    estado varchar(3) not null,
    telefone varchar(11),
    email varchar(35)
);
GO

--Pessoa Fisica e Juridica
CREATE TABLE DBM.Pessoa_fisica
(
    id_cpf int not null,
    id_pessoa int not null,
    CONSTRAINT PK_Pessoa_fisica PRIMARY KEY (id_cpf),
    CONSTRAINT FK_Pessoa_f FOREIGN KEY (id_pessoa)
    REFERENCES DBM.Pessoa
);
GO

CREATE TABLE DBM.Pessoa_juridica
(
    id_cnpj int not null,
    id_pessoa int not null,
    CONSTRAINT PK_Pessoa_juridica PRIMARY KEY (id_cnpj),
    CONSTRAINT FK_Pessoa_j FOREIGN KEY (id_pessoa)
    REFERENCES DBM.Pessoa
);
GO

--Produtos e Usuarios
CREATE TABLE DBM.Produtos
```

The bottom of the SQL Query window shows the execution results:

```
Mensagens
Comandos concluídos com êxito.

Horário de conclusão: 2023-07-30T09:52:17.8826469-03:00
```

A status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

## Análise e Conclusão:

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

As diferentes cardinalidades são implementadas em um banco de dados relacional usando chaves estrangeiras (FOREIGN). Uma chave estrangeira é uma coluna ou um conjunto de colunas em uma tabela que faz referência a uma chave primária em outra tabela.

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

Pelo o'que foi estudado e o que entendi o mais próximo de herança seria as foreign ou seja chave estrangeiras passadas para outras tabelas “filhas” de suas tabelas “pais”

Um exemplo seria : A tabela "Pessoa" é a tabela base que contém os atributos comuns a todas as pessoas. As tabelas "Pessoa Física" e "Pessoa Jurídica" são tabelas especializadas que herdam o atributo "id\_pessoa" da tabela base. Através dessa herança, podemos estabelecer relacionamentos com outras tabelas e representar diferentes tipos de pessoas no banco de dados.

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

Bom existem vários fatores mas os principais que mais impactaram na minha opinião foram a :

- Uma interface gráfica do usuário (GUI) intuitiva que facilita a navegação pelos bancos de dados e as suas tabelas.
- DER com a possibilidade de criar e gerenciar os diagramas através das próprias tabelas podendo ainda criar tabelas utilizando os diagramas.
- Administrar bancos de dados, como a criação e gerenciamento de usuários, o gerenciamento de backups, gerenciamento de estatísticas e o gerenciamento e criação de índices.
- Editor SQL Avançado: O SSMS inclui um editor de código SQL.