

**Desenvolvimento Full Stack**

Aluno: Victor de A. Costa

Missão Prática | Nível 2 | Mundo 3 - 2024.3
VAMOS MANTER AS INFORMAÇÕES!

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

1º Procedimento | Criando o Banco de Dados**--> CÓDIGOS UTILIZADOS**

```
CREATE TABLE Pessoa (
    idPessoa INT IDENTITY(1,1) PRIMARY KEY,
    nome NVARCHAR(255),
    logradouro NVARCHAR(255),
    cidade NVARCHAR(255),
    estado CHAR(2),
    telefone NVARCHAR(255),
    email NVARCHAR(255)
);

INSERT INTO Pessoa (nome, logradouro, cidade, estado, telefone, email) VALUES ('Victor', 'Av. N. S. das Graças', 'SJM', 'RJ', '21994921277', 'victorcosta@estacio.com');

SELECT * FROM Pessoa;

CREATE TABLE Produto (
    idProduto INT IDENTITY(1,1) PRIMARY KEY,
    nome NVARCHAR(255),
    quantidade INT,
    precoVenda NUMERIC
);

INSERT INTO Produto (nome, quantidade, precoVenda) VALUES ('Camiseta CREATEE', 50, 5000);

SELECT * FROM Produto;

CREATE TABLE Usuario (
    idUsuario INT IDENTITY(1,1) PRIMARY KEY,
    login NVARCHAR(255),
    senha NVARCHAR(255)
);

INSERT INTO Usuario (login, senha) VALUES ('loja', 'loja');

SELECT * FROM Usuario;

CREATE TABLE Pessoa_Fisica (
    cpf NVARCHAR(11) PRIMARY KEY,
    idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa)
);

CREATE TABLE Pessoa_Juridica (
    cnpj NVARCHAR(14) PRIMARY KEY,
    idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa)
);

INSERT INTO Pessoa_Fisica (cpf, idPessoa) VALUES ('454516456', 1);

SELECT * FROM Pessoa_Fisica;

INSERT INTO Pessoa_Juridica (cnpj, idPessoa) VALUES ('454516213456', 1);

SELECT * FROM Pessoa_Juridica;
```

```

CREATE TABLE Mover (
    idMovimento INT IDENTITY(1,1) PRIMARY KEY,
    idUsuario INT FOREIGN KEY REFERENCES Usuario(idUsuario),
    idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa),
    idProduto INT FOREIGN KEY REFERENCES Produto(idProduto),
    quantidade INT,
    tipo CHAR(1),
    valorUnitario FLOAT
);

```

```

INSERT INTO Mover(idUsuario, idPessoa, idProduto, quantidade, tipo, valorUnitario) VALUES
(1, 1, 1, 2, 'E', 5000);

```

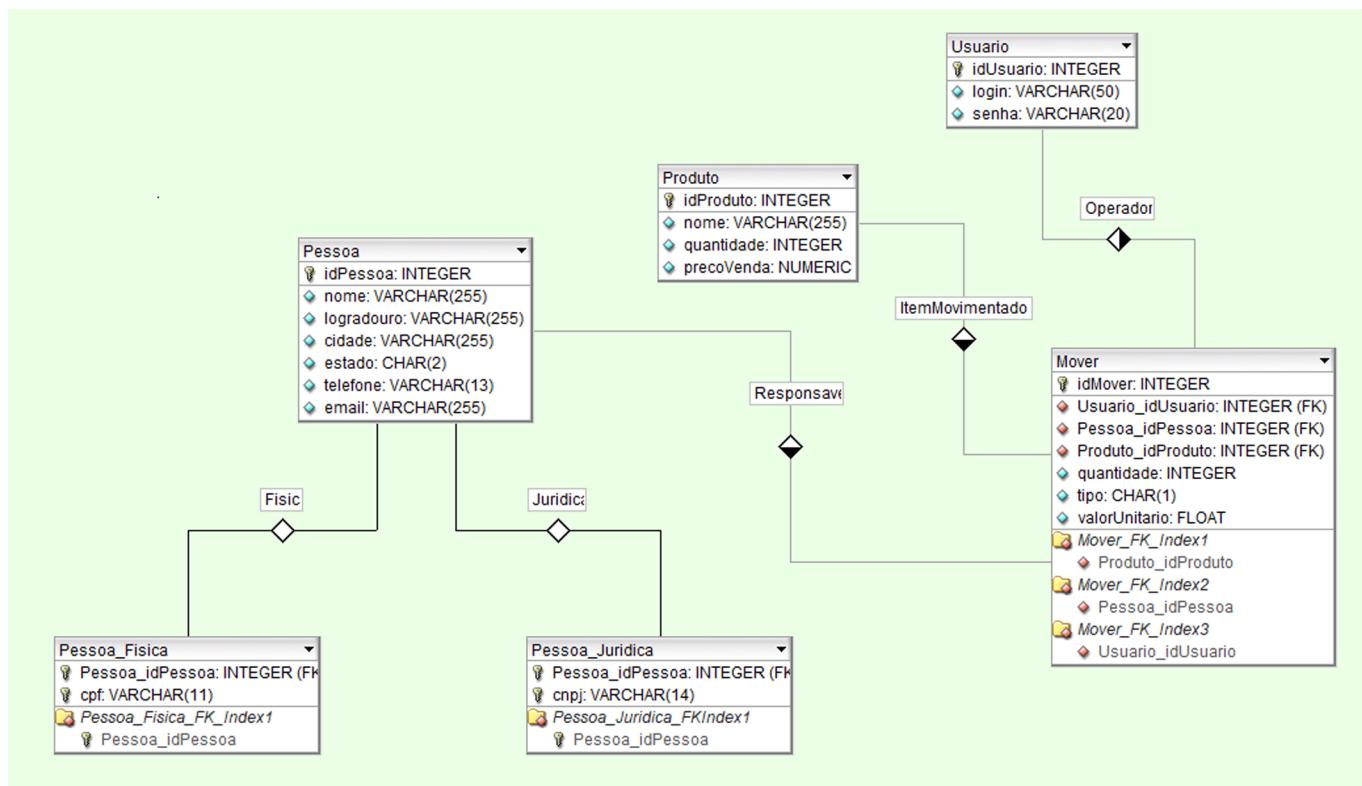
```

SELECT * FROM Mover;

```

---> RESULTADO

Resultados						
Mensagens						
1	idPessoa	nome	logradouro	cidade	estado	telefone
1	1	Victor	Av. N. S. das Graças	SJM	RJ	21994921277
1	email					victorcosta@estacio.com
1	idProduto	nome	quantidade	precoVenda		
1	1	Camiseta CREATEEE	50	5000		
1	idUsuario	login	senha			
1	1	loja	loja			
1	cpf	idPessoa				
1	454516456	1				
1	cnpj	idPessoa				
1	454516213456	1				



---> ANÁLISE E CONCLUSÃO

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

As cardinalidades em bancos de dados relacionais descrevem o número de instâncias de uma entidade que podem estar associadas a uma instância de outra entidade. As cardinalidades mais comuns são: 1x1 (Um para Um), 1xN (Um para Muitos), NxN (Muitos para Muitos).

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

Tabela por entidade: Cada tabela contém os dados da classe pai e da classe filha.

Tabela única + coluna discriminadora: Uma única tabela contém todos os dados, e uma coluna discriminadora é usada para diferenciar entre entidades.

Tabela principal + tabela filhas com FK para a tabela principal: Este modelo é adequado quando a normalização é essencial e as tabelas filhas têm muitas colunas.

Estratégias mistas: É possível modelar parte da hierarquia conforme uma estratégia e parte conforme outra.

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

SQL Server Management Studio (SSMS) otimiza a produtividade com diversos recursos que tornam o gerenciamento de bancos de dados mais eficiente. Ele fornece uma interface gráfica intuitiva, que facilita a navegação e a execução de tarefas complexas sem a necessidade de digitar longos comandos SQL.