



# Estácio

**ALUNO: FILIPE MACIEL DE SOUZA ANDRADE**

**TURMA: 9001**

**CURSO: DESENVOLVIMENTO FULL STACK**

**DISCIPLINA: Vamos Manter as Informações?**

**MATRÍCULA: 2023.04.65842-1**

**POLO PORTO ALEGRE-RS**

**3º PERÍODO**

## **Missão Prática | Nível 2 | Mundo 3**

### **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.

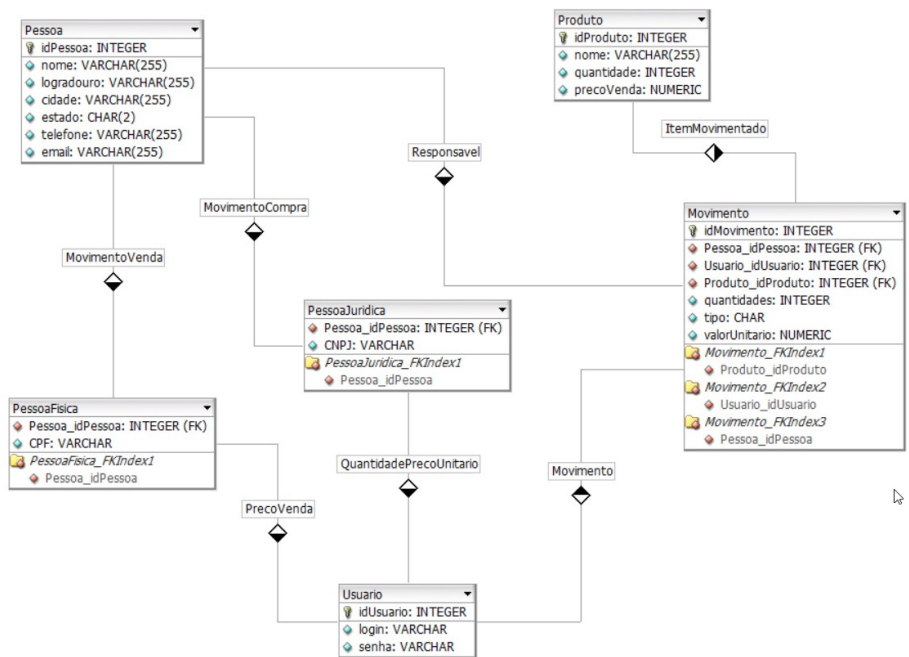
## **1º Procedimento | Criando o Banco de Dados**

```
scriptUM.sql - disconnected x
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

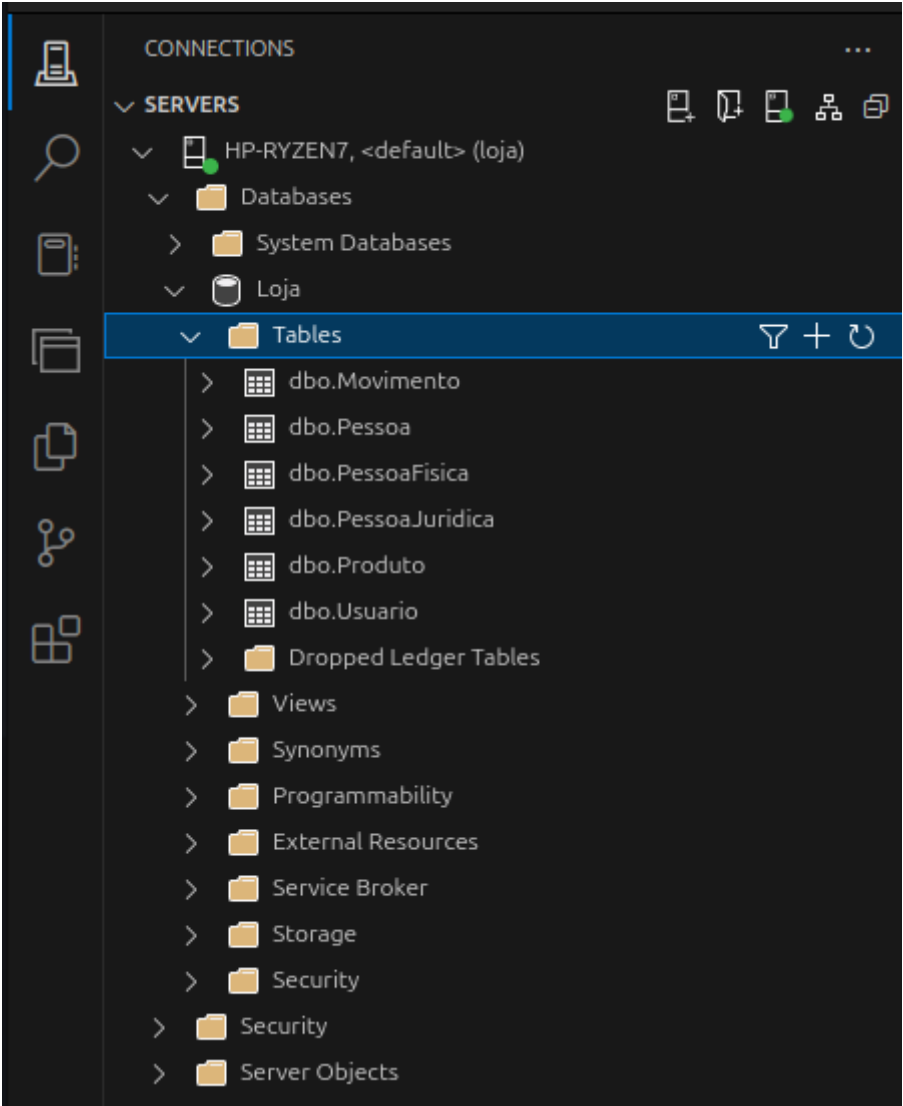
1  -- script um
2  USE Loja;
3  GO
4
5  CREATE SEQUENCE sequenciaPessoaID
6  START WITH 1
7  INCREMENT BY 1;
8
9  CREATE TABLE Pessoa
10 (
11     idPessoa INTEGER NOT NULL CONSTRAINT PK_Pessoa PRIMARY KEY,
12     nome VARCHAR(255) NOT NULL,
13     logradouro VARCHAR(255),
14     cidade VARCHAR(255),
15     estado CHAR(2) NOT NULL,
16     telefone VARCHAR(11),
17     email VARCHAR(255)
18 );
19 GO
20
21 CREATE TABLE PessoaFisica
22 (
23     idPessoa INTEGER NOT NULL CONSTRAINT PK_PessoaFisica PRIMARY KEY,
24     CPF VARCHAR(11) NOT NULL,
25     CONSTRAINT FK_PessoaFisica_Pessoa FOREIGN KEY(idPessoa) REFERENCES Pessoa(idPessoa)
26 );
27 GO
28
29 CREATE TABLE PessoaJuridica
30 (
31     idPessoa INTEGER NOT NULL CONSTRAINT PK_PessoaJuridica PRIMARY KEY,
32     CNPJ VARCHAR(14) NOT NULL,
33     CONSTRAINT FK_PessoaJuridica_Pessoa FOREIGN KEY(idPessoa) REFERENCES Pessoa(idPessoa)
34 );
```

```
scriptUM.sql - disconnected x
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

34 );
35 GO
36
37 CREATE TABLE Usuario
38 (
39     idUsuario INTEGER NOT NULL CONSTRAINT PK_Usuario PRIMARY KEY IDENTITY,
40     login VARCHAR(20) NOT NULL,
41     senha VARCHAR(20) NOT NULL
42 );
43 GO
44
45 CREATE TABLE Produto
46 (
47     idProduto INTEGER NOT NULL CONSTRAINT PK_Produto PRIMARY KEY,
48     nome VARCHAR(255) NOT NULL,
49     quantidade INTEGER,
50     valorVenda NUMERIC(5, 2)
51 );
52 GO
53
54 CREATE TABLE Movimento
55 (
56     idMovimento INTEGER NOT NULL CONSTRAINT PK_Movimento PRIMARY KEY,
57     idUsuario INTEGER NOT NULL,
58     idPessoa INTEGER NOT NULL,
59     idProduto INTEGER,
60     quantidade INTEGER,
61     tipo CHAR(1),
62     valorUnitario NUMERIC(5, 2),
63     CONSTRAINT FK_Movimento_Usuario FOREIGN KEY(idUsuario) REFERENCES Usuario(idUsuario),
64     CONSTRAINT FK_Movimento_Pessoa FOREIGN KEY(idPessoa) REFERENCES Pessoa(idPessoa),
65     CONSTRAINT FK_Movimento_Produto FOREIGN KEY(idProduto) REFERENCES Produto(idProduto)
66 );
67 GO
```



Resultados:



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

R: 1X1: Uma instância em uma entidade se relaciona com no máximo uma instância na outra, usando chaves primárias e estrangeiras.

1XN: Uma instância em uma entidade se relaciona com zero, uma ou várias instâncias na outra, porém cada registro feito na segunda tabela é associado com apenas um registro da primeira tabela, usando um chave estrangeira na segunda tabela.

NxN: Uma instância em uma entidade se relaciona com zero, uma ou várias instâncias na outra.

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

R: Tabela por Hierarquia (Single Table Inheritance - STI)

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

R: Possui uma interface gráfica de certa forma amigável, apesar de não ser multiplataforma e já ser ultrapassado, sendo uma opção mais viável e lógica o Azure Data Studio.

## 2º Procedimento | Alimentando a Base

```
scriptDOIS.sql - disconnected X
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

1 -- script dois
2 USE Loja;
3 INSERT INTO Usuario
4     (login, senha)
5 VALUES('op1', 'op1'),
6        ('op2', 'op2'),
7        ('op3', 'op3'),
8        ('op4', 'op4');
9
10 INSERT INTO Produto
11     (idProduto, nome, quantidade, valorVenda)
12 VALUES('1', 'banana', '100', '5.00'),
13        ('3', 'laranja', '500', '2.00'),
14        ('4', 'manga', '800', '4.00');
15
16 INSERT INTO Pessoa
17     (idPessoa, nome, logradouro, cidade, estado, telefone, email)
18 VALUES
19     (NEXT VALUE FOR sequenciaPessoaID, 'Ian', 'Rua A, 1', 'Resende', 'RJ', '1111-1111', 'ian@gmail.com'),
20     (NEXT VALUE FOR sequenciaPessoaID, 'Sofia', 'Rua B, 2', 'Araguari', 'MG', '2222-2222', 'sofia@gmail.com'),
21     (NEXT VALUE FOR sequenciaPessoaID, 'Alan', 'Rua C, 3', 'Porto Alegre', 'RS', '3333-3333', 'alan@gmail.com'),
22     (NEXT VALUE FOR sequenciaPessoaID, 'Empresa de Produtos', 'Avenida D, 4', 'Garanhuns', 'PE', '4444-4444', 'empresaproduto@gmail.com'),
23     (NEXT VALUE FOR sequenciaPessoaID, 'Empresa de Madeira', 'Avenida E, 5', 'Brasilia', 'DF', '5555-5555', 'empresamadeira@gmail.com');
24
25 INSERT INTO PessoaFisica
26     (idPessoa, CPF)
27 VALUES(1, '11111111111'),
28        (2, '22222222222'),
29        (3, '33333333333');
30
```

```
scriptDOIS.sql - disconnected X
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

31 INSERT INTO PessoaJuridica
32     (idPessoa, CNPJ)
33 VALUES(4, '444444444444'),
34     (5, '555555555555');
35
36 INSERT INTO Movimento
37     (idMovimento, idUsuario, idPessoa, idProduto, quantidade, tipo, valorUnitario)
38 VALUES(1, 1, 5, 1, 4, 'E', 5.00),
39     (3, 2, 3, 3, 2, 'S', 2.00),
40     (5, 1, 4, 4, 60, 'E', 4.00),
41     (6, 2, 1, 1, 15, 'S', 5.00),
42     (7, 4, 2, 4, 25, 'S', 4.00),
43     (9, 3, 5, 3, 50, 'E', 2.00);
```

```
scriptTRES.sql - disconnected X
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

1  -- script tres
2
3  -- pessoa fisica
4  SELECT *
5  FROM PessoaFisica
6      INNER JOIN Pessoa ON PessoaFisica.idPessoa = Pessoa.idPessoa
7
8  -- pessoa juridica
9  SELECT *
10 FROM PessoaJuridica
11     INNER JOIN Pessoa ON PessoaJuridica.idPessoa = Pessoa.idPessoa
12
13 SELECT
14     Produto.nome AS 'produto', Pessoa.nome AS 'fornecedor',
15     Movimento.quantidade, Movimento.valorUnitario,
16     Movimento.quantidade * Movimento.valorUnitario AS 'valorTotal'
17 FROM Movimento
18     INNER JOIN Produto ON Movimento.idProduto = Produto.idProduto
19     INNER JOIN Pessoa ON Movimento.idPessoa = Pessoa.idPessoa
20 WHERE Movimento.tipo = 'E';
21
22 SELECT
23     Produto.nome AS 'produto', Pessoa.nome AS 'comprador',
24     Movimento.quantidade, Movimento.valorUnitario,
25     Movimento.quantidade * Movimento.valorUnitario AS 'valorTotal'
26 FROM Movimento
27     INNER JOIN Produto ON Movimento.idProduto = Produto.idProduto
28     INNER JOIN Pessoa ON Movimento.idPessoa = Pessoa.idPessoa
29 WHERE Movimento.tipo = 'S';
30
31 SELECT
32     Produto.nome AS 'produto',
33     SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalEntrada'
34 FROM Movimento
```



```
scriptTRES.sql - disconnected x
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

34 FROM Movimento
35     JOIN Produto ON Movimento.idProduto = Produto.idProduto
36 WHERE Movimento.tipo = 'E'
37 GROUP BY Produto.nome;
38
39 SELECT
40     Produto.nome AS 'produto',
41     SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalSaida'
42 FROM Movimento
43     JOIN Produto ON Movimento.idProduto = Produto.idProduto
44 WHERE Movimento.tipo = 'S'
45 GROUP BY Produto.nome;
46
47 SELECT DISTINCT
48     Usuario.idUsuario, Usuario.login, Movimento.idMovimento
49 FROM Usuario
50     LEFT JOIN Movimento ON Usuario.idUsuario = Movimento.idUsuario AND Movimento.tipo = 'E'
51 WHERE idMovimento IS NULL;
52
53 SELECT
54     Usuario.login AS 'operador',
55     SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalEntrada'
56 FROM Movimento
57     JOIN Usuario ON Movimento.idUsuario = Usuario.idUsuario
58 WHERE Movimento.tipo = 'E'
59 GROUP BY Usuario.login;
60
61 SELECT
62     Usuario.login AS 'operador',
63     SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalSaida'
64 FROM Movimento
65     JOIN Usuario ON Movimento.idUsuario = Usuario.idUsuario
66 WHERE Movimento.tipo = 'S'
67 GROUP BY Usuario.login;
```

```
scriptTRES.sql - disconnected x
Run Cancel Connect Change Database: Select Database Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

68
69 SELECT
70     SUM(Movimento.valorUnitario * Movimento.quantidade) / SUM(Movimento.quantidade) AS 'mediaPonderada'
71 FROM Movimento
72 WHERE Movimento.tipo = 'S';
```



# Resultados:

ResultsMessages

	idPessoa	CPF	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	11111111111	1	Ian	Rua A, 1	Resende	RJ	1111-1111	ian@gmail.com
2	2	22222222222	2	Sofia	Rua B, 2	Araguari	MG	2222-2222	sofia@gmail.com
3	3	33333333333	3	Alan	Rua C, 3	Porto Alegre	RS	3333-3333	alan@gmail.com

	idPessoa	CNPJ	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	4	44444444444	4	Empresa de Produtos	Avenida D, 4	Garanhuns	PE	4444-4444	empresaproduto@gmail.com
2	5	55555555555	5	Empresa de Madeira	Avenida E, 5	Brasilia	DF	5555-5555	empresamadeira@gmail.com

	produto	fornecedor	quantidade	valorUnitario	valorTotal
1	banana	Empresa de Madeira	4	5.00	20.00
2	manga	Empresa de Produtos	60	4.00	240.00
3	laranja	Empresa de Madeira	50	2.00	100.00

	produto	comprador	quantidade	valorUnitario	valorTotal
1	laranja	Alan	2	2.00	4.00
2	banana	Ian	15	5.00	75.00
3	manga	Sofia	25	4.00	100.00

	produto	valorTotalEntrada
1	banana	20.00
2	laranja	100.00
3	manga	240.00

	produto	valorTotalSaida
1	banana	75.00
2	laranja	4.00
3	manga	100.00

## Results Messages

1	banana	20.00
2	laranja	100.00
3	manga	240.00

	produto	valorTotalSaida
1	banana	75.00
2	laranja	4.00
3	manga	100.00

	idUsuario	login	idMovimento
1	2	op2	NULL
2	4	op4	NULL

	operador	valorTotalEntrada
1	op1	260.00
2	op3	100.00

	operador	valorTotalSaida
1	op2	79.00
2	op4	100.00

	mediaPonderada
1	4.261904

### **1. Quais as diferenças no uso de sequence e identity?**

**R:** Sequence – Escopo independente, gerenciamento manual, visibilidade global, com numeração personalizada em diferentes contextos.

Identity – Escopo vinculado à tabela, gerenciamento automático, visibilidade limitada, com chave primária e numeração automática

### **2. Qual a importância das chaves estrangeiras para a consistência do banco?**

**R:** As chaves estrangeiras são importantes para a consistência e confiabilidade em bancos de dados relacionais. Utilizando elas você garante integridade dos seus dados, facilita a manutenção do banco de dados e constrói um sistema robusto e confiável.

### **3. Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?**

**R:** Álgebra Relacional – Operações básicas para manipular conjuntos de dados, como projeção, seleção, união e junção.

Cálculo Relacional – Expressões mais complexas utilizando fórmulas e variáveis, como restrição, agrupamento e ordenação.

### **4. Como é feito o agrupamento em consultas, e qual requisito é obrigatório?**

**R:** O agrupamento é feito usando o GROUP BY, que é usado para combinar linhas de dados em grupos com base em valores semelhantes em uma ou mais colunas. O requisito obrigatório do GROUP BY é de especificar todas as colunas não agregadas no SELECT.