

Campus:	Universidade Estácio de Sá
Curso:	Full Stack
Semestre:	3º Semestre
Disciplina:	RPG0015
Nome:	Temis Da Silva Teodosio
Turma:	2023.3

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

1. Título da Prática: **1º Procedimento | Criando o Banco de Dados**

2. 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: anexado ao final do relatório

-- TABLES

```
CREATE TABLE Pessoa(  
    idPessoa int,  
    nome varchar(255) not null,  
    logradouro varchar(255),  
    cidade varchar(255),  
    estado char(2),  
    telefone varchar(11),  
    email varchar(255),  
    PRIMARY KEY (idPessoa)  
);
```

```
CREATE SEQUENCE seq_Pessoa AS INT  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
MAXVALUE 1000000
```

NO CYCLE

CACHE 5;

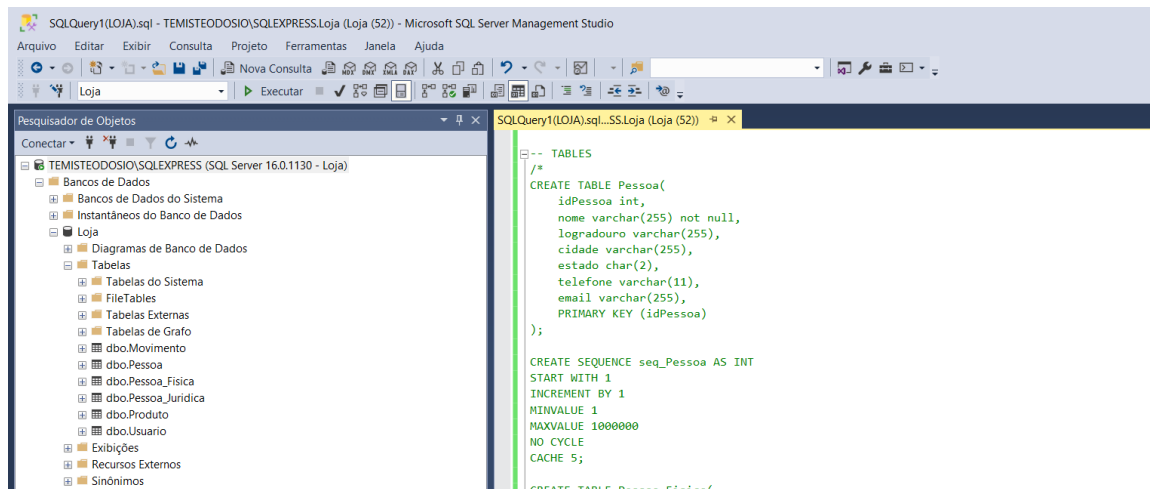
```
CREATE TABLE Pessoa_Fisica(
    idPessoa_Fisica int PRIMARY KEY,
    CPF varchar(11) NOT NULL,
    idPessoa int,
    FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)
);
```

```
CREATE TABLE Pessoa_Juridica(
    idPessoa_Juridica int PRIMARY KEY,
    CNPJ varchar(14) NOT NULL,
    idPessoa int,
    FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)
);
```

```
CREATE TABLE Produto (
    idProduto INT PRIMARY KEY,
    nome VARCHAR(255),
    quantidade INT,
    precoVenda NUMERIC(10, 2)
);
```

```
CREATE TABLE Usuario(
    idUsuario INT PRIMARY KEY,
    Logar VARCHAR(255),
    Senha VARCHAR(25)
);
```

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



```
SQLQuery1(LOJA).sql...SS.Loja (Loja (52)) * - X
);
FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)
);
CREATE TABLE Produto (
    idProduto INT PRIMARY KEY,
    nome VARCHAR(255),
    quantidade INT,
    precoVenda NUMERIC(10, 2)
);
CREATE TABLE Usuario(
    idUsuario INT PRIMARY KEY,
    Logar VARCHAR(255),
    Senha VARCHAR(25)
);
CREATE TABLE Movimento(
    idMovimento INT IDENTITY(1,1) PRIMARY KEY,
    idPessoa INT,
    idUsuario INT,
    idProduto INT,
    quantidade INT,
    tipo CHAR(1),
    valorUnitario FLOAT,
    FOREIGN KEY (idUsuario) REFERENCES Usuario(idUsuario),
    FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa),
    FOREIGN KEY (idProduto) REFERENCES Produto(idProduto)
);
--INSERT INTO
```

Análise e Conclusão:

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

1x1 => O registro de uma entidade se associada com apenas um registro da outra entidade. Elas são associadas por meio de chaves estrangeiras.

1xN => Um Registro de uma entidade pode se associar com vários registros da outra entidade. Implementado por meio de chaves estrangeiras.

NxN => Um Registro de uma entidade pode se associar com varios registros do outra entidade e vice-versa

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

Especialização, Generalização

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

Coleta, consulta, desenvolve e integram os Scripts. Gerenciam Objetos. E desempenham um papel crucial na indústria.

1. Título da Prática: **2º Procedimento | Alimentando a Base**
2. 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 dedados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server.

Todos os códigos solicitados : Anexado ao final do relatório

Os resultados

INSERT INTO Usuario VALUES

(1, 'op1', 'op1'),
(2, 'op2', 'op2');

INSERT INTO Usuario VALUES

(3, 'op3', 'op3'),
(4, 'op4', 'op4');

INSERT INTO Produto VALUES

(1, 'Banana', 100, 5.00),
(2, 'Maçã', 300, 3.50),
(3, 'Laranja', 500, 2.00),
(4, 'Manga', 800, 4.00);

select * from Usuario;

Select * from Produto;

INSERT INTO Movimento(idPessoa, idUsuario, idProduto, quantidade, tipo, valorUnitario)

VALUES (8,1,1,12,'S',4.00),
(9,2,2,2,'S',3.00),
(10,3,3,35,'E',4.00),
(11,4,4,25,'E',6.00);

INSERT INTO Pessoa

(idPessoa, nome, logradouro,cidade, estado, telefone, email)
values

```
(next value for seq_Pessoa,'Ferdinando','Rua 21, casa 411, Japiim', 'Manaus', 'AM', '921313-1313','Toro@Ferdinando.com'),  
(next value for seq_Pessoa, 'JJC', 'Rua 11, Centro', 'Riacho do Norte', 'PA', '1212-1212',  
'jjc@riacho.com');
```

INSERT INTO Pessoa

(idPessoa, nome, logradouro, cidade, estado, telefone, email)
values

(next value for seq_Pessoa, 'Marta', 'Rua 31 de março, 504', 'Manaus', 'AM', '1212-2525',
'Marta@toledo.com'),

(next value for seq_Pessoa, 'Smart', 'Laranjeiras, 0000', 'Manaus', 'AM', '4585-9674',
'Smart@CIA.com')

;

DECLARE @IdPessoa int;

SET @IdPessoa = Next value for seq_Pessoa;

INSERT INTO Pessoa_Fisica

(idPessoa_Fisica, CPF)

VALUES

(@IdPessoa, '78945612378');

INSERT INTO Pessoa_Juridica

(idPessoa_Juridica, CNPJ)

VALUES

(@IdPessoa, '15915915945682');

--DADOS

SELECT * FROM Usuario;

SELECT * FROM Produto;

SELECT * FROM Pessoa;

SELECT * FROM Pessoa_Fisica;

SELECT * FROM Pessoa_Juridica;

SELECT * FROM Movimento;

**select * from Movimento inner join Usuario on Movimento.idMovimento <>
Usuario.idUsuario;**

SELECT p.*, pf.cpf

FROM Pessoa p

INNER JOIN Pessoa_Fisica pf ON p.idPessoa = pf.idPessoa;

SELECT p.*, pj.cnpj

```
FROM Pessoa p
INNER JOIN Pessoa_Juridica pj ON p.idPessoa = pj.idPessoa;
```

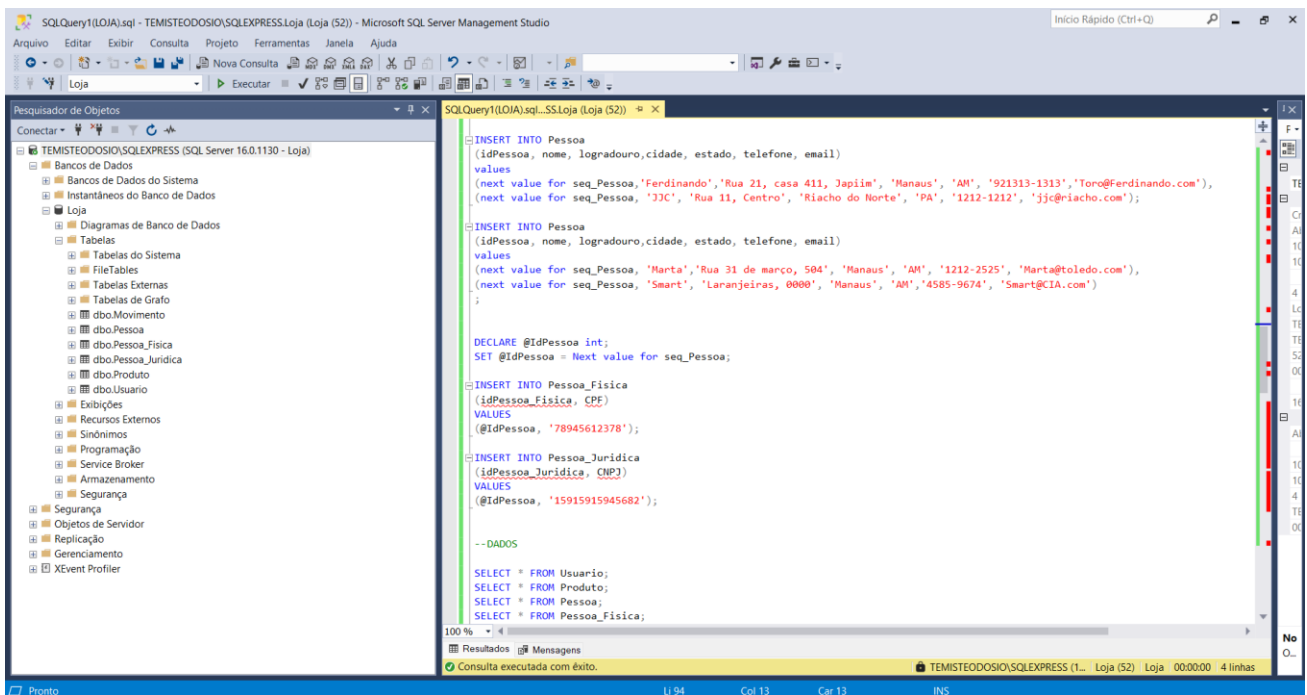
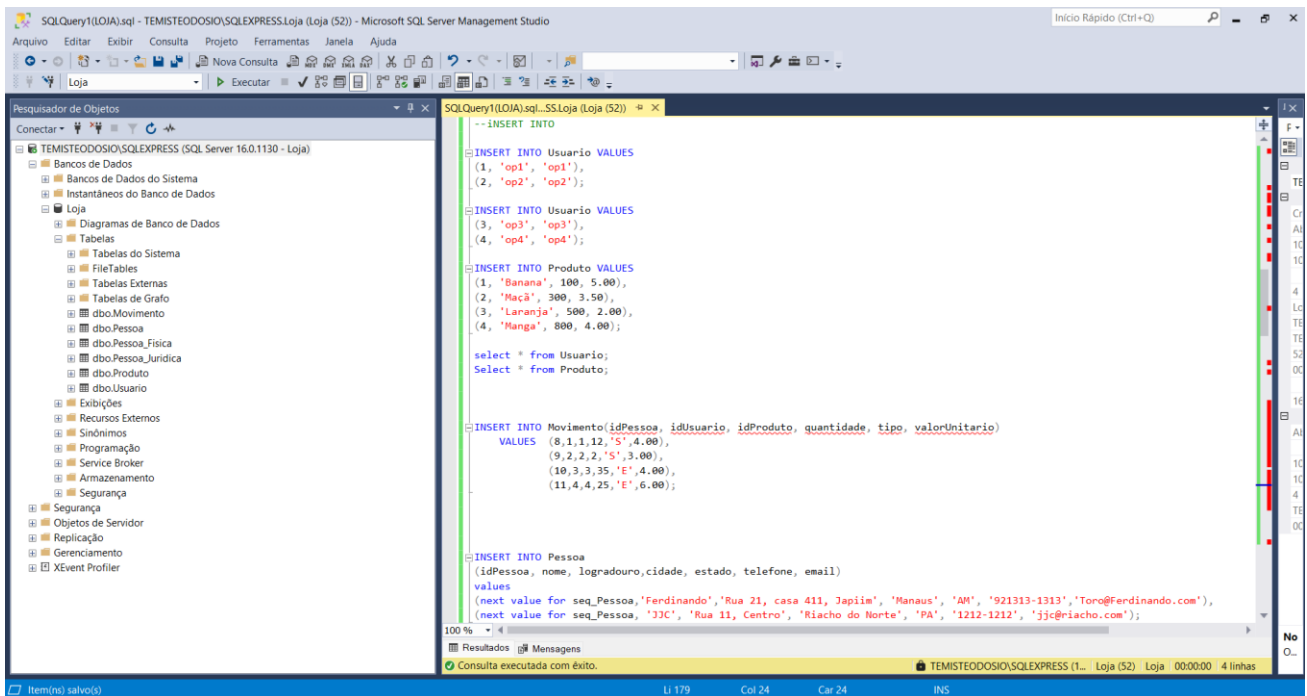
```
SELECT m.*, p.nome as fornecedor, pdt.nome as Produto, m.quantidade, m.valorUnitario,
(m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.idPessoa
INNER JOIN Produto pdt ON pdt.idProduto = m.idProduto
WHERE m.tipo = 'E';
```

```
SELECT m.*, p.nome as comprador, pdt.nome as Produto, m.quantidade, m.valorUnitario,
(m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.idPessoa = p.idPessoa
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'S';
```

```
SELECT pdt.nome, SUM(m.quantidade * m.valorUnitario) as totValor
FROM Movimento m
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'E'
GROUP BY pdt.nome;
```

```
SELECT pdt.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'S'
GROUP BY pdt.nome;
```

```
SELECT u.*
FROM Usuario u
LEFT JOIN Movimento m ON u.idUsuario = m.idUsuario AND m.tipo = 'E'
WHERE m.idMovimento is null;
```



SQLQuery1(LOJA).sql - TEMISTEODOSIO/SQLEXPRESS.Loja (Loja (S2)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar

TEMISTEODOSIO/SQLEXPRESS (SQL Server 16.0.1130 - Loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Gráfico
 - dbo.Movimento
 - dbo.Pessoa
 - dbo.Pessoa_Fisica
 - dbo.Pessoa_Juridica
 - dbo.Produto
 - dbo.Usuario
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Service Broker
 - Armazenamento
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento
 - XEvent Profiler

SQLQuery1(LOJA).sql - SS.Loja (Loja (S2))

100 %

Resultados Mensagens

idUsuario	Logar	Senha
1	op1	op1
2	op2	op2
3	op3	op3
4	op4	op4

idProduto	nome	quantidade	precoVenda
1	Banana	100	5.00
2	Maçã	300	3.50
3	Laranja	500	2.00
4	Manga	800	4.00

idPessoa	nome	logradouro	cidade	estado	telefone	email
8	Fernando	Rua 21, casa 411, Japiim	Manaus	AM	921313-1313	Toro@Ferdinando.com
9	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	je@riacho.com
10	Marta	Rua 31 de março, 504	Manaus	AM	1212-2525	Marta@toledo.com
11	Smart	Laranjeiras, 0000	Manaus	AM	4585-9674	Smart@CIA.com
14	Fernando	Rua 21, casa 411, Japiim	Manaus	AM	921313-1313	Toro@Ferdinando.com
15	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	je@riacho.com
16	Marta	Rua 31 de março, 504	Manaus	AM	1212-2525	Marta@toledo.com
17	Smart	Laranjeiras, 0000	Manaus	AM	4585-9674	Smart@CIA.com
18	Fernando	Rua 21, casa 411, Japiim	Manaus	AM	921313-1313	Toro@Ferdinando.com
20	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	je@riacho.com
21	Marta	Rua 31 de março, 504	Manaus	AM	1212-2525	Marta@toledo.com

Consulta executada com êxito.

TEMISTEODOSIO/SQLEXPRESS (1... Loja (S2) Loja 00:00:00 24 linhas

SQLQuery1(LOJA).sql - TEMISTEODOSIO/SQLEXPRESS.Loja (Loja (S2)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar

TEMISTEODOSIO/SQLEXPRESS (SQL Server 16.0.1130 - Loja)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Gráfico
 - dbo.Movimento
 - dbo.Pessoa
 - dbo.Pessoa_Fisica
 - dbo.Pessoa_Juridica
 - dbo.Produto
 - dbo.Usuario
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Service Broker
 - Armazenamento
 - Segurança
 - Objetos de Servidor
 - Replicação
 - Gerenciamento
 - XEvent Profiler

SQLQuery1(LOJA).sql - SS.Loja (Loja (S2))

100 %

Resultados Mensagens

idPessoa_Fisica	CPF	idPessoa
13	78945612378	NULL
18	78945612378	NULL
23	78945612378	NULL
28	78945612378	NULL

idPessoa_Juridica	CNPJ	idPessoa
12	15915915945682	NULL
13	15915915945682	NULL
18	15915915945682	NULL
23	15915915945682	NULL
28	15915915945682	NULL

idMovimento	idPessoa	idUsuario	idProduto	quantidade	tipo	valorUnitario
10	8	1	1	12	S	4
11	9	2	2	2	S	3
12	10	3	3	35	E	4
13	11	4	4	25	E	6
14	8	1	1	12	S	4
15	9	2	2	2	S	3
16	10	3	3	35	E	4
17	11	4	4	25	E	6
18	8	1	1	12	S	4
19	9	2	2	2	S	3
20	10	3	3	35	E	4
21	11	4	4	25	E	6
22	8	1	1	12	S	4
23	9	2	2	2	S	3
24	10	3	3	35	E	4
25	11	4	4	25	E	6

Consulta executada com êxito.

TEMISTEODOSIO/SQLEXPRESS (1... Loja (S2) Loja 00:00:00 25 linhas

Pronto Li 132 Col 1 Car 1 INS

SQLQuery1(LOJA).sql - TEMISTEODOSIO\SQLEXPRESS.Loja (Loja (52)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

- TEMISTEODOSIO\SQLEXPRESS (SQL Server 16.0.1130 - Loja)
- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grão
 - dbo.Movimento
 - dbo.Pessoa
 - dbo.Pessoa_Fisica
 - dbo.Pessoa_Juridica
 - dbo.Produto
 - dbo.Usuario
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Service Broker
 - Armazenamento
 - Segurança
- Segurança
- Objetos de Servidor
- Replicação
- Gerenciamento
- XEvent Profiler

SQLQuery1(LOJA).sql - SS.Loja (Loja (52))

```
SELECT p.**, pf.cpf
FROM Pessoa p
INNER JOIN Pessoa_Fisica pf ON p.idPessoa = pf.idPessoa;

SELECT p.**, pj.cnpj
FROM Pessoa p
INNER JOIN Pessoa_Juridica pj ON p.idPessoa = pj.idPessoa;

SELECT m.**, p.nome as fornecedor, pdt.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.idPessoa
INNER JOIN Produto pdt ON pdt.idProduto = m.idProduto
WHERE m.tipo = 'E';

SELECT m.**, p.nome as comprador, pdt.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.idPessoa = p.idPessoa
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'S';

SELECT pdt.nome, SUM(m.quantidade * m.valorUnitario) as totValor
FROM Movimento m
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'E'
GROUP BY pdt.nome;
```

100 %

	idMovimento	idPessoa	idUsuario	idProduto	quantidade	tipo	valorUnitario	fornecedor	Produto	quantidade	valorUnitario	total
1	12	10	3	3	35	E	4	Marta	Laranja	35	4	140
2	13	11	4	4	25	E	6	Smart	Manga	25	6	150
3	16	10	3	3	35	E	4	Marta	Laranja	35	4	140
4	17	11	4	4	25	E	6	Smart	Manga	25	6	150
5	20	10	3	3	35	E	4	Marta	Laranja	35	4	140
6	21	11	4	4	25	E	6	Smart	Manga	25	6	150
7	24	10	3	3	35	E	4	Marta	Laranja	35	4	140
8	25	11	4	4	25	E	6	Smart	Manga	25	6	150

Consulta executada com êxito. TEMISTEODOSIO\SQLEXPRESS (1... Loja (52) Loja 00:00:00 8 linhas

Item(ns) salvo(s) Li 149 Col 1 Car 1 INS

SQLQuery1(LOJA).sql - TEMISTEODOSIO\SQLEXPRESS.Loja (Loja (52)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

- TEMISTEODOSIO\SQLEXPRESS (SQL Server 16.0.1130 - Loja)
- Bancos de Dados
 - Bancos de Dados do Sistema
 - Instantâneos do Banco de Dados
 - Loja
 - Diagramas de Banco de Dados
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grão
 - dbo.Movimento
 - dbo.Pessoa
 - dbo.Pessoa_Fisica
 - dbo.Pessoa_Juridica
 - dbo.Produto
 - dbo.Usuario
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Service Broker
 - Armazenamento
 - Segurança
- Segurança
- Objetos de Servidor
- Replicação
- Gerenciamento
- XEvent Profiler

SQLQuery1(LOJA).sql - SS.Loja (Loja (52))

```
SELECT p.**, pj.cnpj
FROM Pessoa p
INNER JOIN Pessoa_Juridica pj ON p.idPessoa = pj.idPessoa;

SELECT m.**, p.nome as fornecedor, pdt.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.idPessoa
INNER JOIN Produto pdt ON pdt.idProduto = m.idProduto
WHERE m.tipo = 'E';

SELECT m.**, p.nome as comprador, pdt.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.idPessoa = p.idPessoa
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'S';

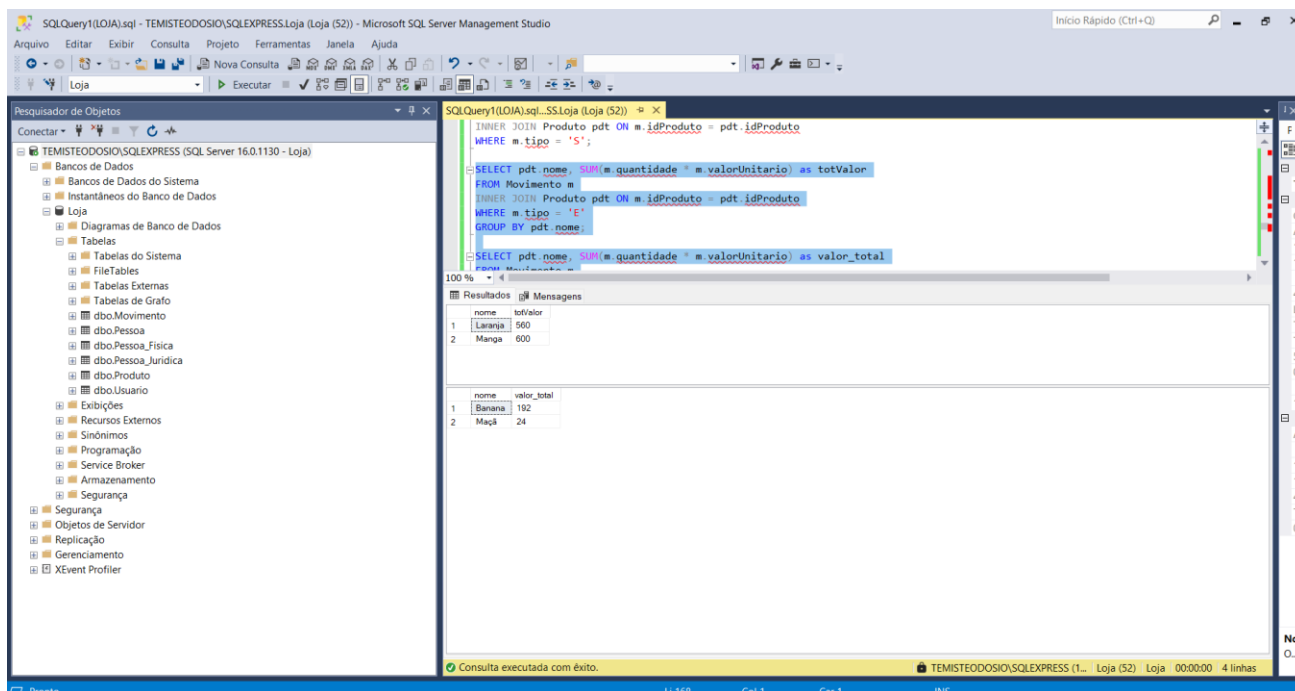
SELECT pdt.nome, SUM(m.quantidade * m.valorUnitario) as totValor
FROM Movimento m
INNER JOIN Produto pdt ON m.idProduto = pdt.idProduto
WHERE m.tipo = 'E'
GROUP BY pdt.nome;
```

100 %

	idMovimento	idPessoa	idUsuario	idProduto	quantidade	tipo	valorUnitario	comprador	Produto	quantidade	valorUnitario	total
1	10	8	1	1	12	S	4	Ferdinando	Banana	12	4	48
2	11	9	2	2	2	S	3	JJC	Maçã	2	3	6
3	14	8	1	1	12	S	4	Ferdinando	Banana	12	4	48
4	15	9	2	2	2	S	3	JJC	Maçã	2	3	6
5	18	8	1	1	12	S	4	Ferdinando	Banana	12	4	48
6	19	9	2	2	2	S	3	JJC	Maçã	2	3	6
7	22	8	1	1	12	S	4	Ferdinando	Banana	12	4	48
8	23	9	2	2	2	S	3	JJC	Maçã	2	3	6

Consulta executada com êxito. TEMISTEODOSIO\SQLEXPRESS (1... Loja (52) Loja 00:00:00 8 linhas

Pronto Li 155 Col 20 Car 20 INS



Análise :

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

O identity está associada a uma coluna de tabela especifica enquanto uma sequence pode ser compartilhada com varias tabelas.

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

Ela é uma “Cópia” da chave primaria de outra tabela. Representando um relacionamento entre entidades.

Mantem a consistência em um banco de dados. Evita perda de informações.

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

Seleção, União, Interseção, Produto cartesiano, Diferença

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

Através do GROUP BY sendo obrigatório uma função de agregação.

Conclusão:

O SQL tem muitas funcionalidades uteis quando bem utilizadas.

Link para o repositório Git

<https://github.com/TemisdaSilvaTeodosio/SQL.git>

