UNIVERSIDADE ESTÁCIO

CURSO DE DESENVOLVIMENTO FULL STACK

RELATÓRIO DA MISSÃO PRÁTICA 2 - MUNDO 3

TURMA: 2023.1

DISCIPLINA: RPG0015 - VAMOS MANTER AS INFORMAÇÕES

CAMPUS: VIA CORPVS - FORTALEZA (CE)

MOISÉS EDUARDO GOMES DA COSTA

Relatório de Prática de Modelagem e Criação Banco de Dados com SQL

1° Procedimento - Criação de Banco de Dados em SQL

Objetivo 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.

Códigos

Link do Github com os códigos da Prática:
https://github.com/BeaterOfWar/Mundo-3_MP2

Criacao_tabelas.sql

```
CREATE TABLE Pessoa (
    idPessoa INT PRIMARY KEY IDENTITY(1,1),
    nome VARCHAR(255),
    logradouro VARCHAR(255),
    cidade VARCHAR(255),
    estado CHAR(2),
    telefone VARCHAR(11),
    email VARCHAR(255)
);

CREATE TABLE Movimento (
    idMovimento INT PRIMARY KEY IDENTITY(1,1),
    Pessoa_idPessoa INT,
    Usuario_idUsuario INT,
    Produto_idProduto INT,
    quantidade INT,
    precoUnitario NUMERIC,
    FOREIGN KEY (Pessoa_idPessoa) REFERENCES Pessoa(idPessoa)
);

CREATE TABLE PessoaFisica (
    idPF INT PRIMARY KEY IDENTITY(1,1),
    cpf INT,
    Pessoa_idPessoa INT,
    FOREIGN KEY (Pessoa_idPessoa) REFERENCES Pessoa(idPessoa)
```

```
);

CREATE TABLE PessoaJuridica (
    idPJ INT PRIMARY KEY IDENTITY(1,1),
    cnpj INT,
    Pessoa_idPessoa INT,
    FOREIGN KEY (Pessoa_idPessoa) REFERENCES Pessoa(idPessoa)
);

CREATE TABLE Usuario (
    idUsuario INT PRIMARY KEY IDENTITY(1,1),
    Login VARCHAR(25),
    Senha VARCHAR(30)
);

CREATE TABLE Produto (
    idProduto INT PRIMARY KEY IDENTITY(1,1),
    nome_produto VARCHAR(255),
    quantidade_produto INT,
    precoVenda_produto NUMERIC
);

CREATE SEQUENCE PessoaSeq
    AS INT
    START WITH 1
    INCREMENT BY 1;
```

Resultados da Execução dos Códigos

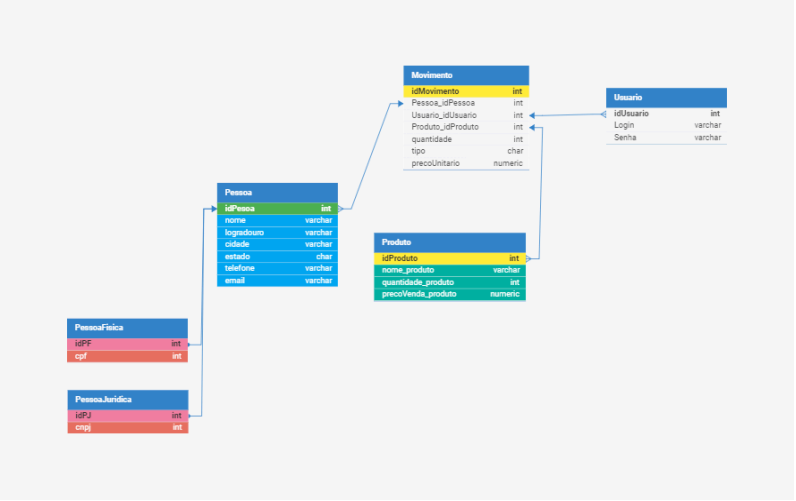


Figura 1: Modelagem do banco de dados feito no https://www.dbdesigner.net/

Análise e Conclusão

(A) Como são implementadas as diferentes cardinalidades, basicamente 1 x 1, 1 x N ou N x N, em um banco de dados relacional?

1x1: Cada item em uma tabela se conecta a um item em outra.

Exemplo: Uma pessoa tem um único CPF.

1xN: Um item em uma tabela se conecta a muitos itens em outra. Exemplo: Uma loja tem muitos produtos.

NxN: Muitos itens em uma tabela se conectam a muitos itens em outra. Isso é feito usando uma tabela extra.

Exemplo: um produto pode pertencer a mais de uma categoria e uma categoria pode ter mais de um produto.

Esses relacionamentos permitem que os dados sejam organizados de maneira lógica e eficiente, facilitando consultas complexas e garantindo a integridade dos dados.

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

Neste tipo de relação, uma entidade ampla também conhecida como superclasse é subdividida em entidades mais detalhadas ou subclasses, que adquirem os atributos da entidade ampla. Por exemplo, uma entidade “Pessoa” pode ser dividida em “PessoaFisica” e “PessoaJuridica”, ambas herdando as características da entidade “Pessoa”.

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

SQL Server Management Studio é uma ferramenta que facilita o gerenciamento de bancos de dados. Ele possui um editor de consultas avançado, um designer de tabelas visual, recursos de monitoramento de desempenho e segurança robusta. Facilmente também ele pode ser integrado a ferramentas da microsoft o que transforma ele em uma ferramenta bastante versátil e útil

2° Procedimento - Alimentando a Base

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.

Códigos

Link do Github com os códigos da Prática:
https://github.com/BeaterOfWar/Mundo-3_MP2

adicicao_dados.sql

```
ALTER TABLE PessoaFisica

ALTER COLUMN cpf BIGINT;



ALTER TABLE PessoaJuridica

ALTER COLUMN cnpj BIGINT;



INSERT INTO Usuario (Login, Senha)

VALUES ('op1', 'op1'),

       ('op2', 'op2');



INSERT INTO Pessoa (nome, logradouro, cidade, estado, telefone, email)

VALUES ('Fernanda', 'Rua Fulano de tal, 34', 'Maceio', 'AL', '3135-1311',
'fernandap@gmail.com'),

                ('Gabriela', 'Avenida Beltrano, 62', 'Recife', 'PE', '3212-2482',
'gabrielat@gmail.com'),

           ('Eduarda', 'Avenida Marechal Deodoro, 55', 'Fortaleza', 'CE', '1123-5745',
'duda554@gmail.com'),
```

```
            ('Alvenaria Agil', 'Rua das Maquinas, 88', 'S◆o Paulo', 'SP', '4074-5321',
'alvenariaagil@gmail.com'),

            ('Cozinha da Luciana', 'Rua Augusto Peixoto, 75', 'Rio de Janeiro', 'RJ',
'4126-7575', 'lucianareceitas@gmail.com');



INSERT INTO PessoaFisica (cpf, Pessoa_idPessoa)

VALUES ('52898121112', 1),

       ('31927426784', 2),

       ('51234124113', 3);



INSERT INTO PessoaJuridica (cnpj, Pessoa_idPessoa)

VALUES ('76279400000140', 4),

       ('52267334000120', 5);



INSERT INTO Produto (nome_produto, quantidade_produto, precoVenda_produto)

VALUES ('Banana', 100, 5.00),

       ('Laranja', 500, 2.00),

       ('Manga', 800, 4.00);



INSERT INTO Movimento (Pessoa_idPessoa, Usuario_idUsuario, Produto_idProduto, quantidade,
tipo, precoUnitario)

VALUES (1, 1, 1, 10, 'S', 5.00),

       (2, 2, 2, 20, 'E', 2.00),

       (3, 1, 3, 30, 'E', 4.00);
```

consulta_dados.sql

```sql
--a. Dados completos de pessoas físicas.

SELECT * FROM PessoaFisica PF

INNER JOIN Pessoa P ON PF.Pessoa_idPessoa = P.idPessoa;

--b. Dados completos de pessoas jurídicas.

SELECT * FROM PessoaJuridica PJ

INNER JOIN Pessoa P ON PJ.Pessoa_idPessoa = P.idPessoa;

--c. Movimentações de entrada, com produto,fornecedor, quantidade, preço unitário e valor total.

SELECT M.Produto_idProduto, M.Pessoa_idPessoa AS Fornecedor, M.quantidade, M.precoUnitario, M.quantidade * M.precoUnitario AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'E';

--d. Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total.

SELECT M.Produto_idProduto, M.Pessoa_idPessoa AS Comprador, M.quantidade, M.precoUnitario, M.quantidade * M.precoUnitario AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'S';

--e. Valor total das entradas agrupadas por produto.

SELECT M.Produto_idProduto, SUM(M.quantidade * M.precoUnitario) AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'E'

GROUP BY M.Produto_idProduto;

--Valor total das saídas agrupadas por produto.

SELECT M.Produto_idProduto, SUM(M.quantidade * M.precoUnitario) AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'S'

GROUP BY M.Produto_idProduto;
```

```sql
--f. Operadores que não efetuaram movimentações de entrada (compra).

SELECT U.idUsuario

FROM Usuario U

WHERE U.idUsuario NOT IN (SELECT M.Usuario_idUsuario FROM Movimento M WHERE M.tipo = 'E');

--g. Valor total de entrada, agrupado por operador.

SELECT M.Usuario_idUsuario, SUM(M.quantidade * M.precoUnitario) AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'E'

GROUP BY M.Usuario_idUsuario;

--h. Valor total de saída, agrupado por operador.

SELECT M.Usuario_idUsuario, SUM(M.quantidade * M.precoUnitario) AS ValorTotal

FROM Movimento M

WHERE M.tipo = 'S'

GROUP BY M.Usuario_idUsuario;

--j. Valor médio de venda por produto, utilizando média ponderada.

SELECT M.Produto_idProduto, SUM(M.quantidade * M.precoUnitario) / SUM(M.quantidade) AS
PrecoMedioPonderado

FROM Movimento M

WHERE M.tipo = 'S'

GROUP BY M.Produto_idProduto;
```

Resultados da Execução dos Códigos

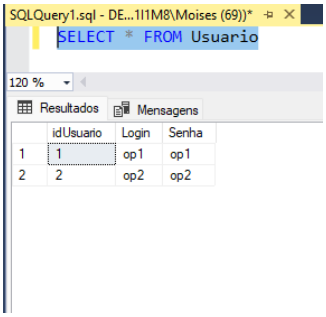
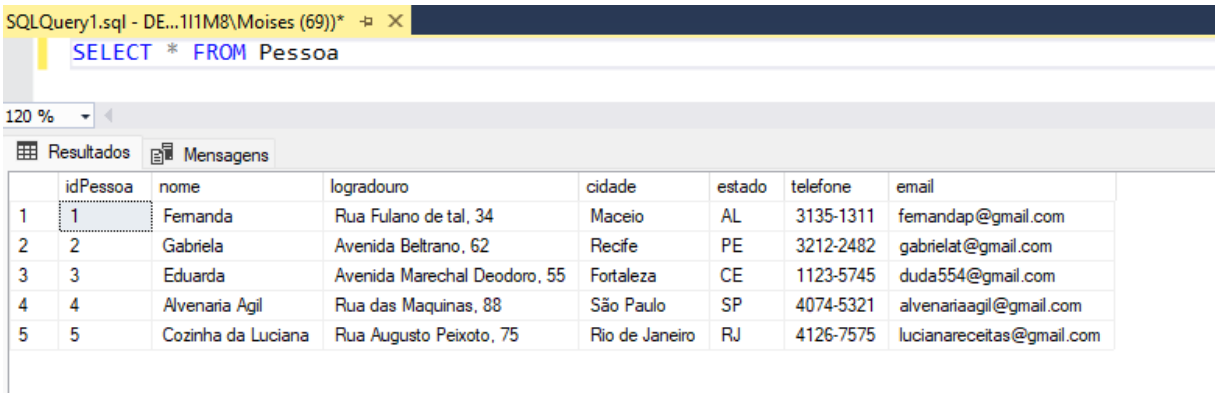
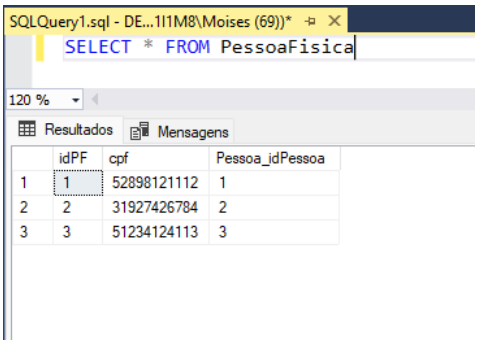


Figura 2: SELECT na tabela usuário



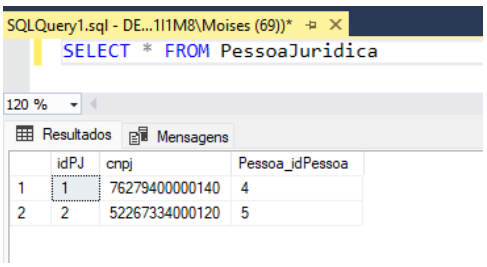
	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	Fernanda	Rua Fulano de tal, 34	Maceio	AL	3135-1311	fernandap@gmail.com
2	2	Gabriela	Avenida Beltrano, 62	Recife	PE	3212-2482	gabrielat@gmail.com
3	3	Eduarda	Avenida Marechal Deodoro, 55	Fortaleza	CE	1123-5745	duda554@gmail.com
4	4	Alvenaria Agil	Rua das Maquinas, 88	São Paulo	SP	4074-5321	alvenariaagil@gmail.com
5	5	Cozinha da Luciana	Rua Augusto Peixoto, 75	Rio de Janeiro	RJ	4126-7575	lucianareceitas@gmail.com

Figura 3: SELECT na tabela Pessoa



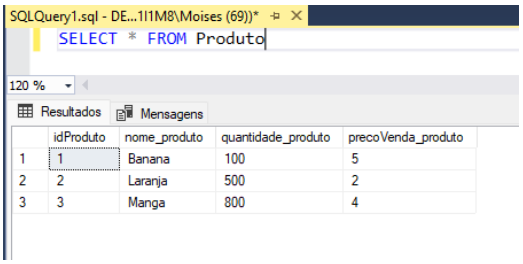
	idPF	cpf	Pessoa_idPessoa
1	1	52898121112	1
2	2	31927426784	2
3	3	51234124113	3

Figura 4: SELECT na tabela PessoaFisica



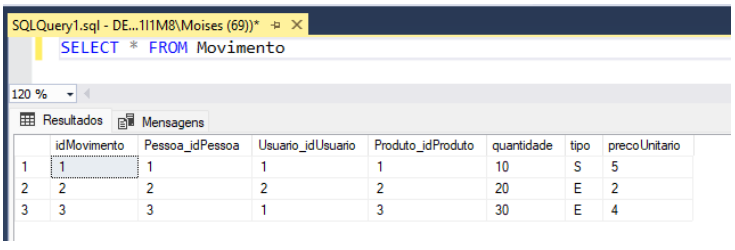
	idPJ	cnpj	Pessoa_idPessoa
1	1	76279400000140	4
2	2	52267334000120	5

Figura 5: SELECT na tabela PessoaJuridica



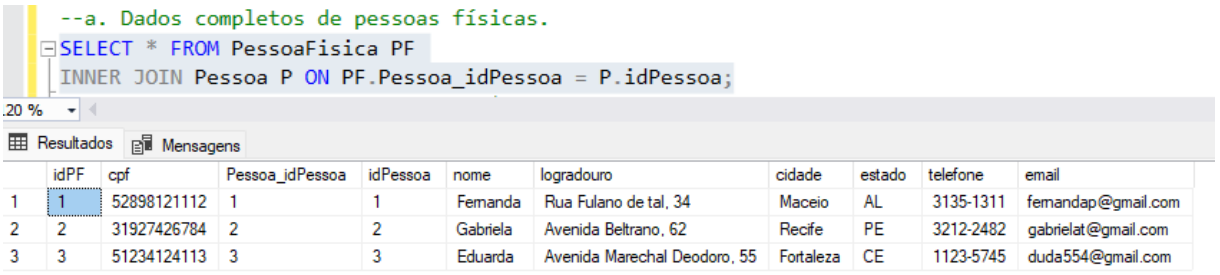
	idProduto	nome_produto	quantidade_produto	precoVenda_produto
1	1	Banana	100	5
2	2	Laranja	500	2
3	3	Manga	800	4

Figura 6: SELECT na tabela Produto



	idMovimento	Pessoa_idPessoa	Usuario_idUsuario	Produto_idProduto	quantidade	tipo	precoUnitario
1	1	1	1	1	10	S	5
2	2	2	2	2	20	E	2
3	3	3	1	3	30	E	4

Figura 7: SELECT na tabela Movimento

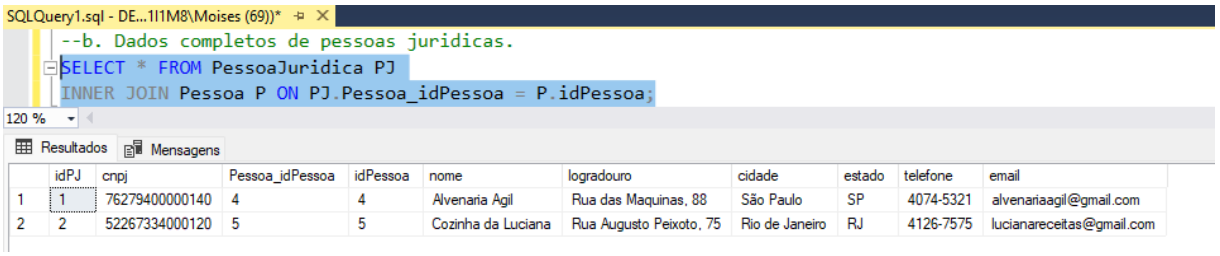


```
--a. Dados completos de pessoas físicas.
SELECT * FROM PessoaFisica PF
INNER JOIN Pessoa P ON PF.Pessoa_idPessoa = P.idPessoa;
```

Resultados | Mensagens

	idPF	cpf	Pessoa_idPessoa	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	52898121112	1	1	Fernanda	Rua Fulano de tal, 34	Maceio	AL	3135-1311	fernandap@gmail.com
2	2	31927426784	2	2	Gabriela	Avenida Beltrano, 62	Recife	PE	3212-2482	gabrielat@gmail.com
3	3	51234124113	3	3	Eduarda	Avenida Marechal Deodoro, 55	Fortaleza	CE	1123-5745	duda554@gmail.com

Figura 8: a. Dados completos de pessoas físicas.

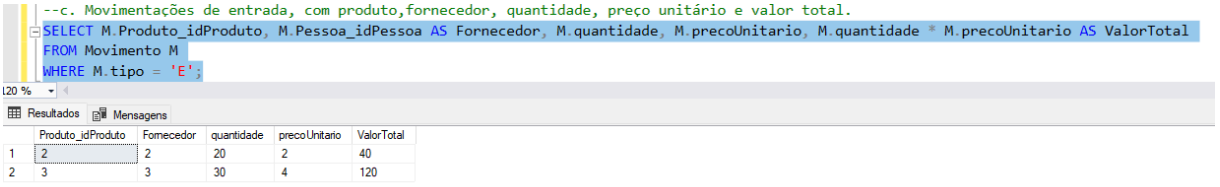


```
--b. Dados completos de pessoas juridicas.
SELECT * FROM PessoaJuridica PJ
INNER JOIN Pessoa P ON PJ.Pessoa_idPessoa = P.idPessoa;
```

Resultados | Mensagens

	idPJ	cnpj	Pessoa_idPessoa	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	76279400000140	4	4	Alvenaria Agil	Rua das Maquinas, 88	São Paulo	SP	4074-5321	alvenariaagil@gmail.com
2	2	52267334000120	5	5	Cozinha da Luciana	Rua Augusto Peixoto, 75	Rio de Janeiro	RJ	4126-7575	lucianareceitas@gmail.com

Figura 9: b. Dados completos de pessoas jurídicas.

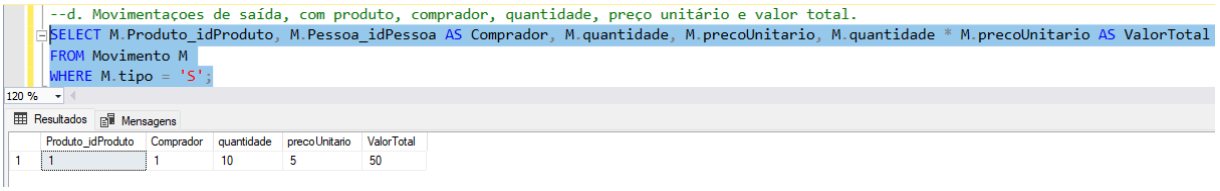


```
--c. Movimentações de entrada, com produto,fornecedor, quantidade, preço unitário e valor total.
SELECT M.Produto_idProduto, M.Pessoa_idPessoa AS Fornecedor, M.quantidade, M.precoUnitario, M.quantidade * M.precoUnitario AS ValorTotal
FROM Movimento M
WHERE M.tipo = 'E';
```

Resultados | Mensagens

	Produto_idProduto	Fornecedor	quantidade	precoUnitario	ValorTotal
1	2	2	20	2	40
2	3	3	30	4	120

Figura 10: c. Movimentações de entrada, com produto,fornecedor, quantidade, preço unitário e valor total.

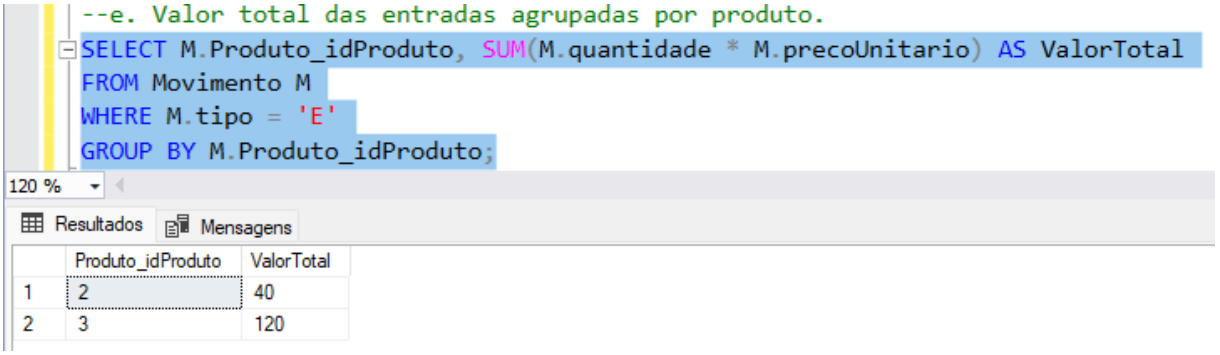


```
--d. Movimentaçoes de saída, com produto, comprador, quantidade, preço unitário e valor total.
SELECT M.Produto_idProduto, M.Pessoa_idPessoa AS Comprador, M.quantidade, M.precoUnitario, M.quantidade * M.precoUnitario AS ValorTotal
FROM Movimento M
WHERE M.tipo = 'S';
```

Resultados | Mensagens

	Produto_idProduto	Comprador	quantidade	precoUnitario	ValorTotal
1	1	1	10	5	50

Figura 11: d. Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total.



```
--e. Valor total das entradas agrupadas por produto.
SELECT M.Produto_idProduto, SUM(M.quantidade * M.precoUnitario) AS ValorTotal
FROM Movimento M
WHERE M.tipo = 'E'
GROUP BY M.Produto_idProduto;
```

Resultados | Mensagens

	Produto_idProduto	ValorTotal
1	2	40
2	3	120

Figura 12: e. Valor total das entradas agrupadas por produto.

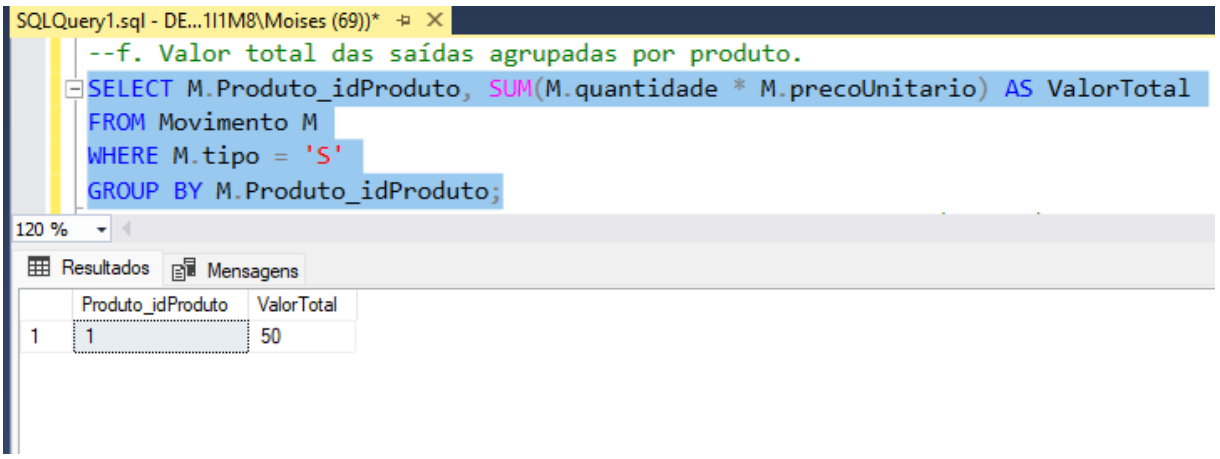


Figura 13: f. Valor total das saídas agrupadas por produto.

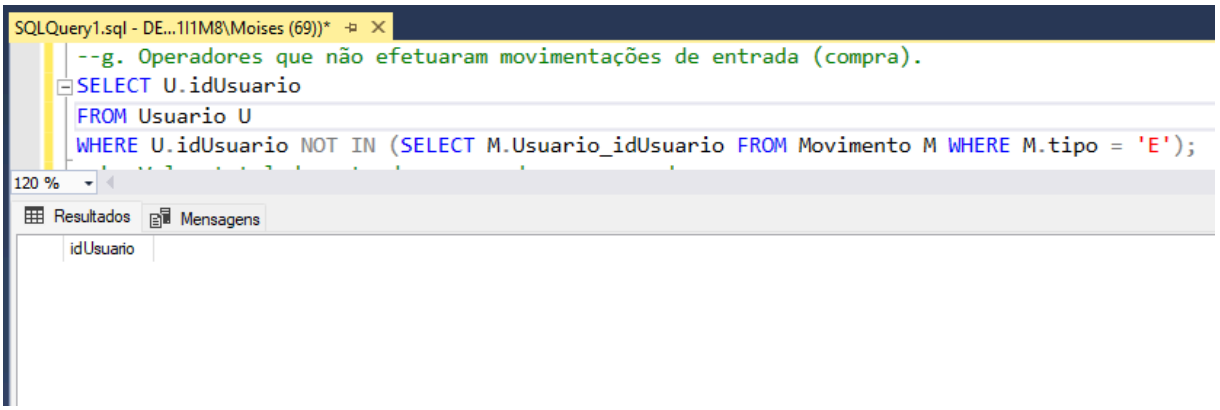


Figura 14: g. Operadores que não efetuaram movimentações de entrada (compra).

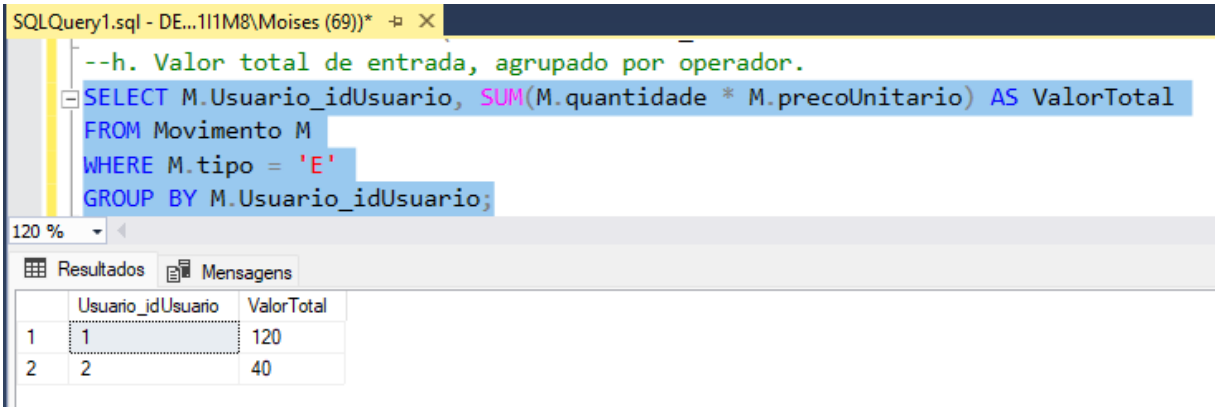


Figura 15: h. Valor total de entrada, agrupado por operador.

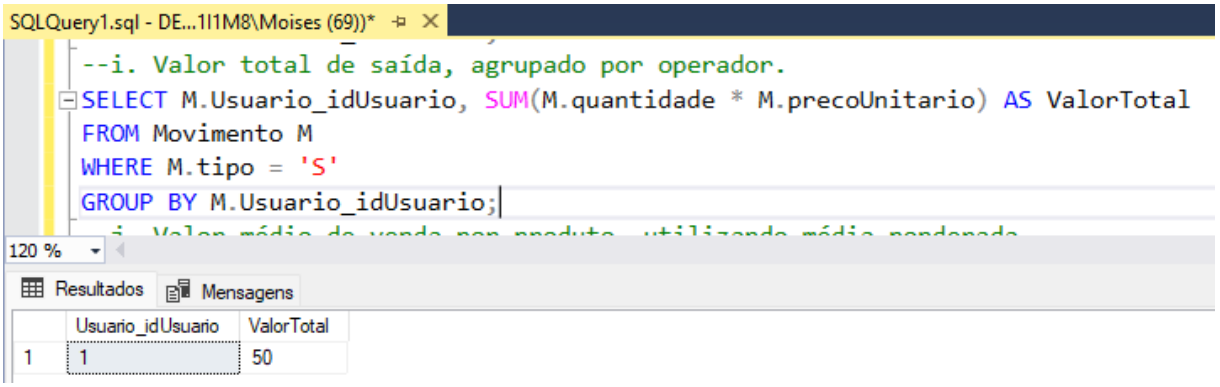


Figura 16: i. Valor total de saída, agrupado por operador.

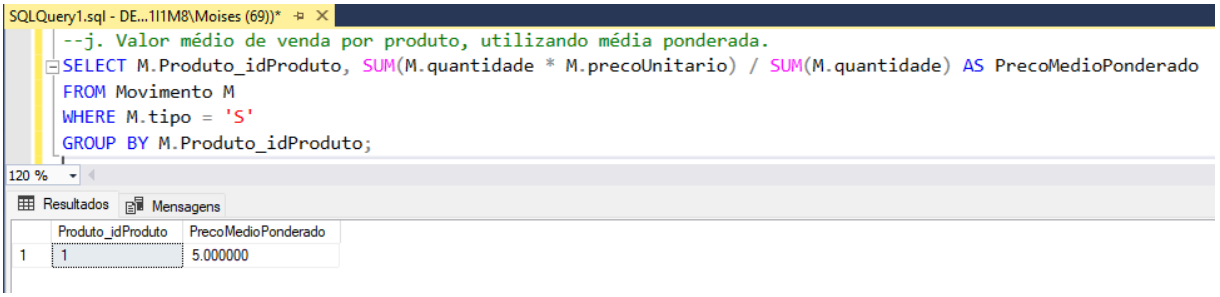


Figura 17: j. Valor médio de venda por produto, utilizando média ponderada.

Análise e Conclusão:

(A) Quais as diferenças no uso de sequence e identity?

Em SQL, identity e sequence são recursos para criar números automáticos. O identity é usado em uma coluna de uma tabela específica e só funciona para essa tabela. Já sequence é criado separadamente e pode ser utilizado em várias tabelas diferentes, porque não está ligado a uma tabela em específica.

(B) Qual a importância das chaves estrangeiras para a consistência do banco?

As chaves estrangeiras garantem a consistência e a integridade dos dados em um banco de dados relacional, permitem a criação de relacionamentos entre tabelas, facilitando a realização de consultas complexas que envolvem múltiplas tabelas, sendo assim os dados ficam mais consistentes.

(C) Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?

SELECT: Corresponde à operação de seleção na álgebra relacional.

FROM: Corresponde à operação de produto cartesiano.

WHERE: Corresponde à operação de seleção.

além desses três principais, existem também o group by e o having que são definidos como cálculo relacional

(D) Como é feito o agrupamento em consultas, e qual requisito é obrigatório?

Ele é feito usando a cláusula GROUP BY, que permite agrupar linhas que têm os mesmos valores em colunas específicas e é obrigatório que todas as colunas no SELECT que não estão na cláusula GROUP BY devem ser usadas com uma função de agregação, como SUM, AVG, COUNT, MIN, MAX, entre outras.

Referências Bibliográficas

[1] “Diferenças entre SEQUENCES x IDENTITY no Microsoft SQL Server 2012

Disponível em:
http://www.linhadecodigo.com.br/artigo/3403/diferencas-entre-sequences-x-identity-no-microsoft-sql-server-2012.aspx

Acesso em 10 de Maio de 2024.

[2] “SQL - Álgebra Relacional - Operações Fundamentais - Conceitos básicos”

Disponível em

https://www.macoratti.net/13/06/sql_arcb.htm

em 11 de Maio de 2024.

[3] “Introdução ao SQL - Agrupamento de Resultados”

Disponível em

https://www.devmedia.com.br/introducao-ao-sql-agrupamento-de-resultados/17008

Acesso em 11 de Maio de 2024.