

**UNIVERSIDADE ESTÁCIO DE SÁ**

**FULLSTACK**

**Mundo 03 - Nível 02   
Implementação de um cadastro de clientes em modo texto,**

**com persistência em arquivos, baseado na tecnologia Java.**

Herval Rosano Dantas  
Matrícula 202205119203

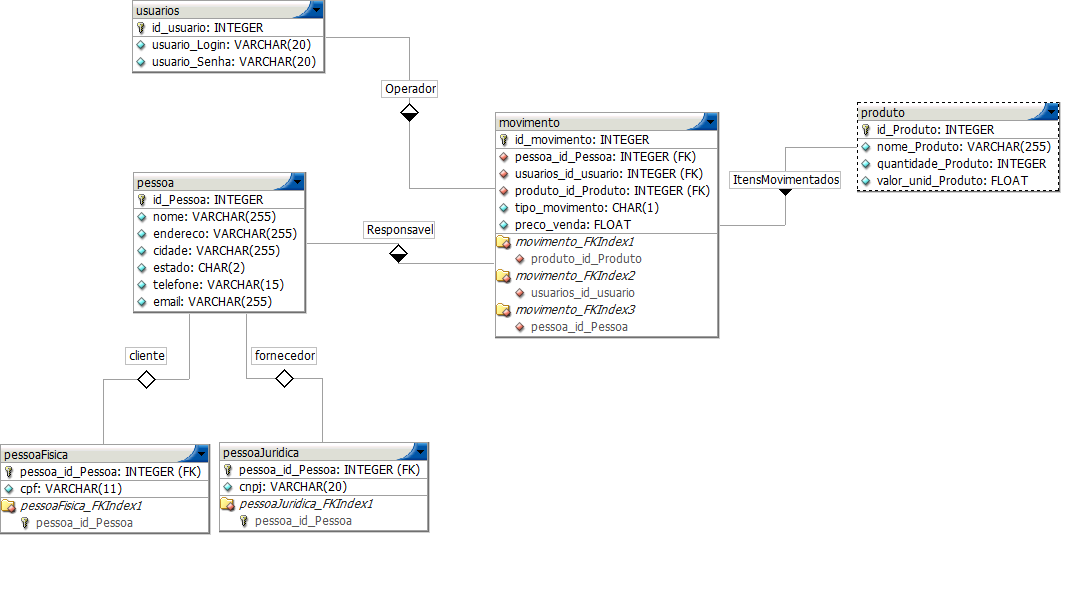
RIO DE JANEIRO – RJ  
2023

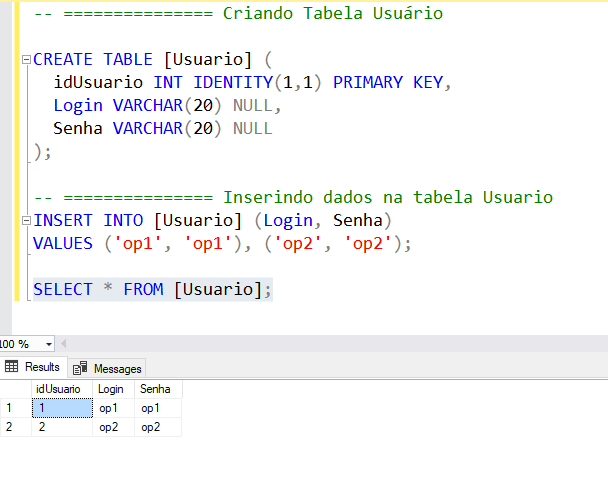
**Objetivo da Prática**

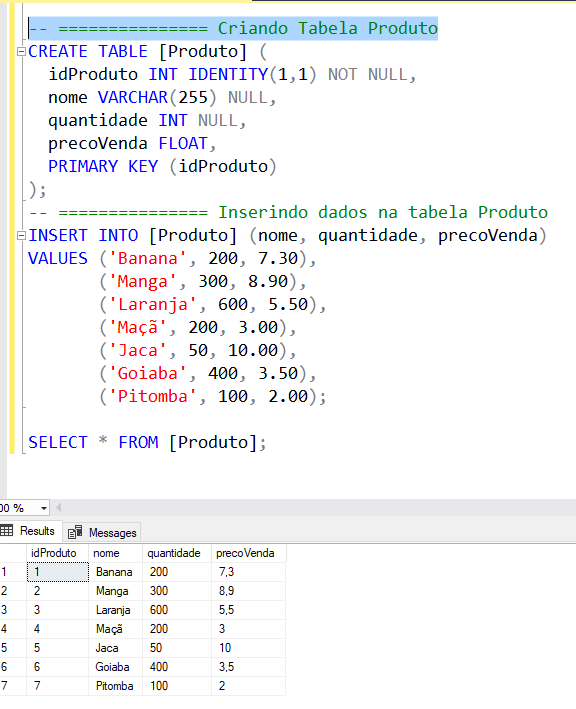
* Identificar os requisitos de um sistema e transformá-los no modelo adequado.
* Utilizar ferramentas de modelagem para bases de dados relacionais.
* Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
* Explorar a sintaxe SQL na consulta e manipulação de dados (DML).

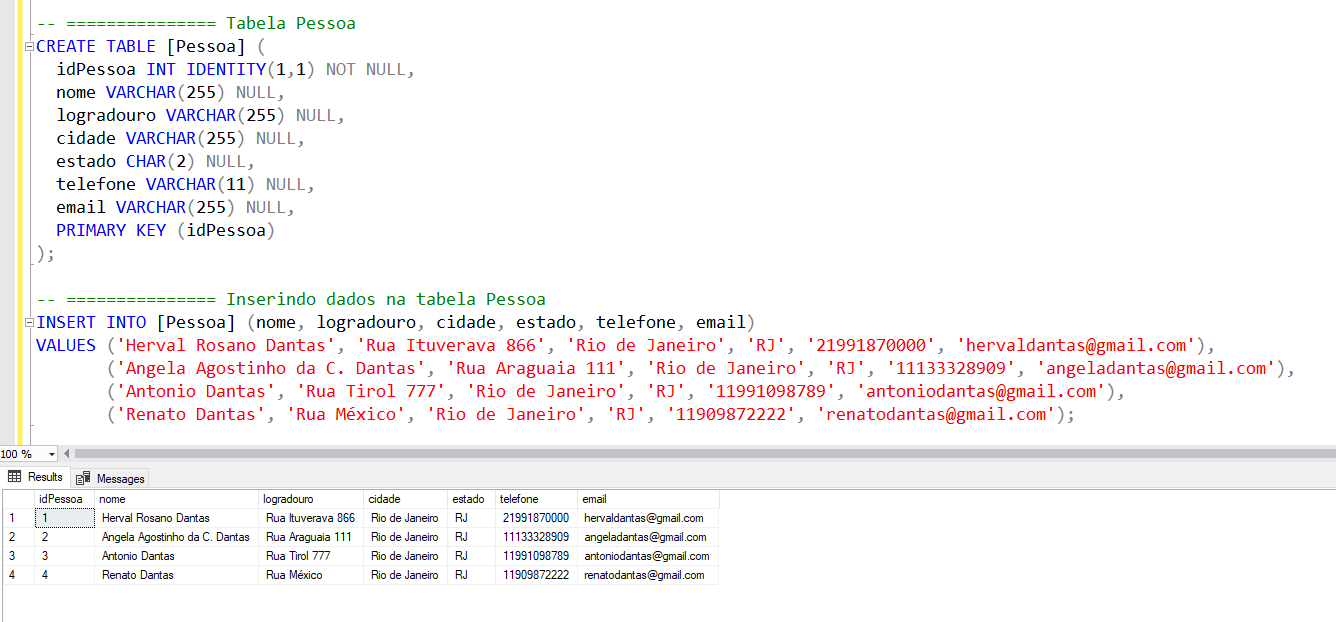
**1º Procedimento – Criando o Banco de Dados**

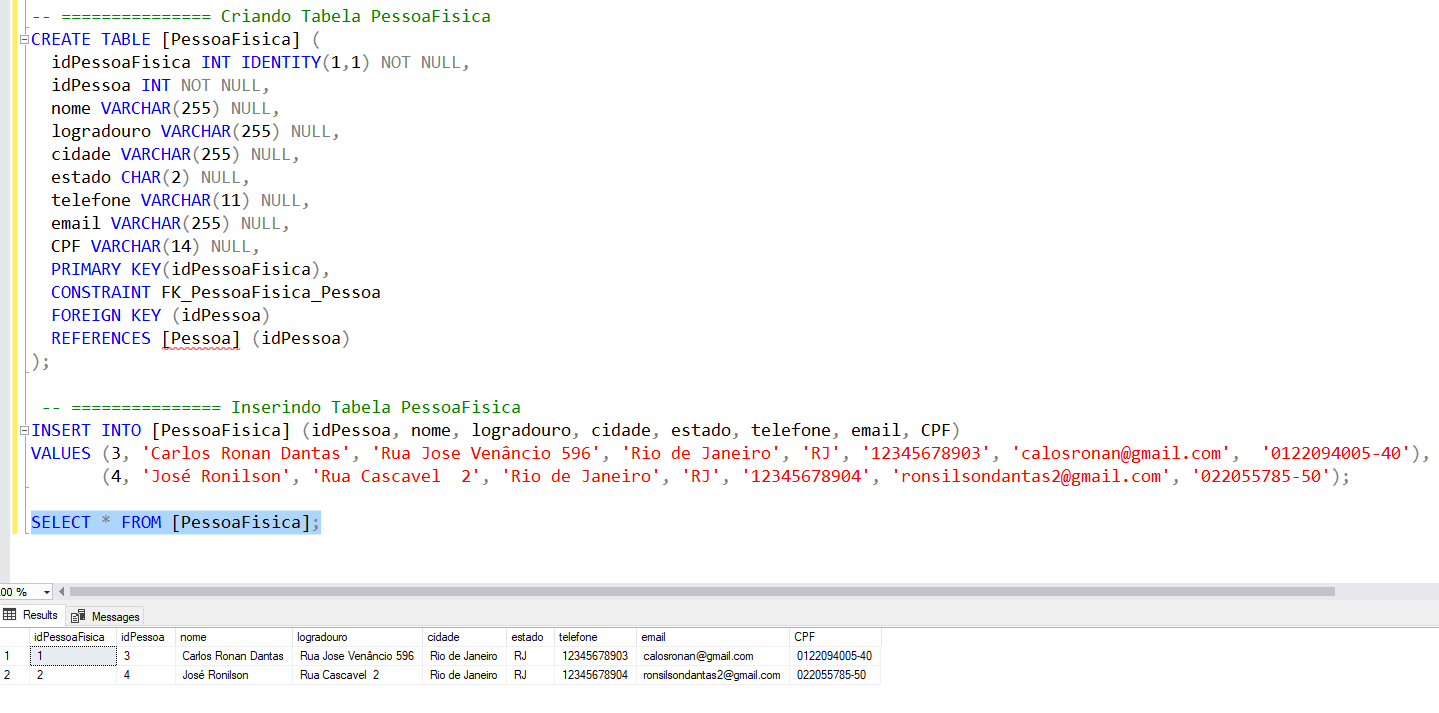
Modelagem do Banco de Dados usando o DBDesigner Fork

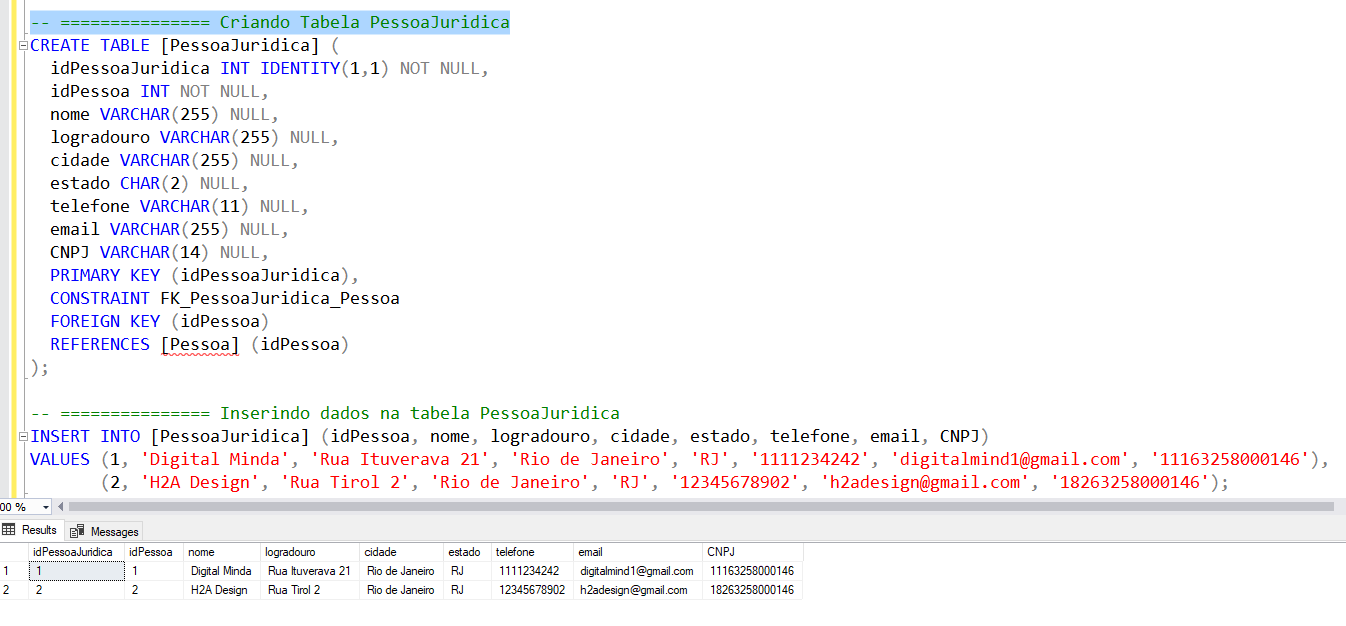


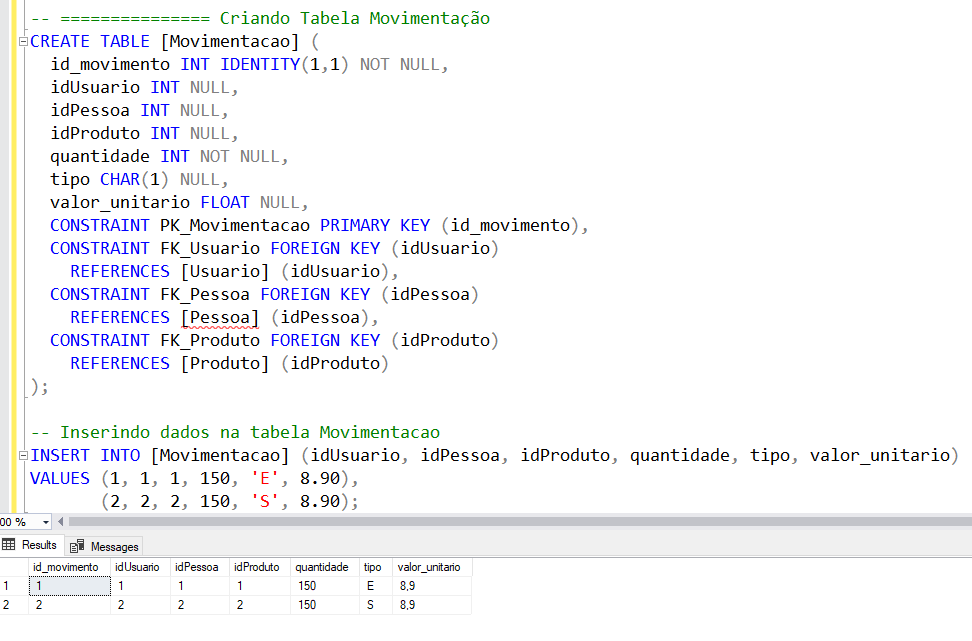
Com a ajuda do DBDesigner, passamos a criar as tabelas da loja. Veja arquivo loja\_create\_tables.sql





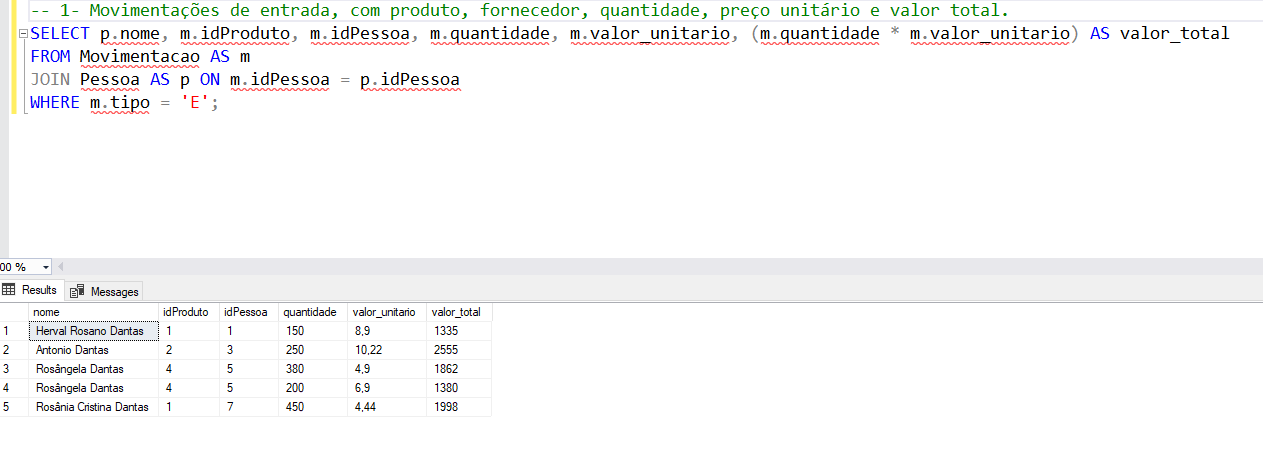


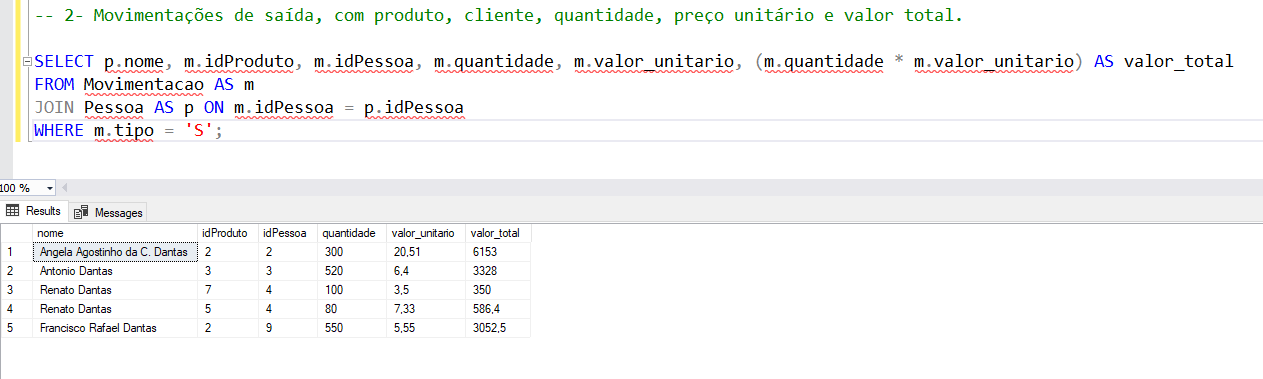


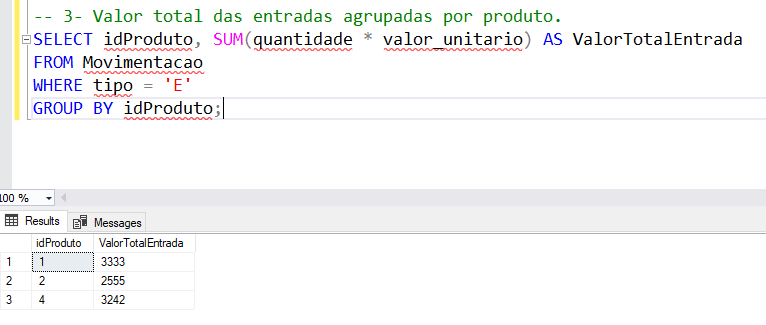
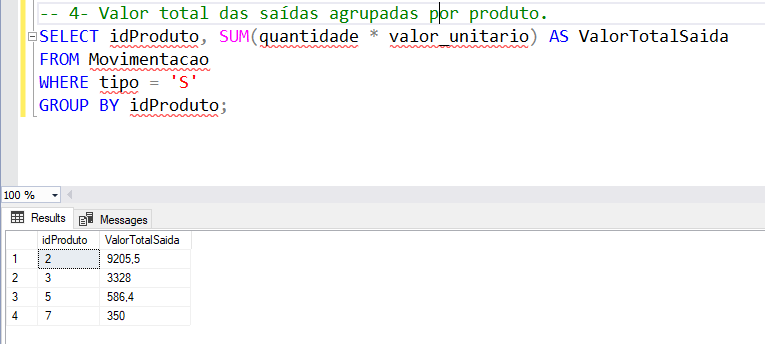


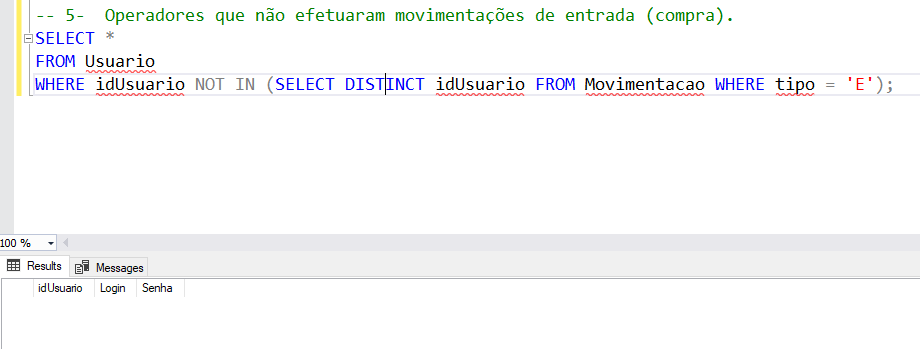
Conforme solicitado pela missão. Vamos efetuar diversas pesquisas. Veja arquivo loja\_searches.sql.

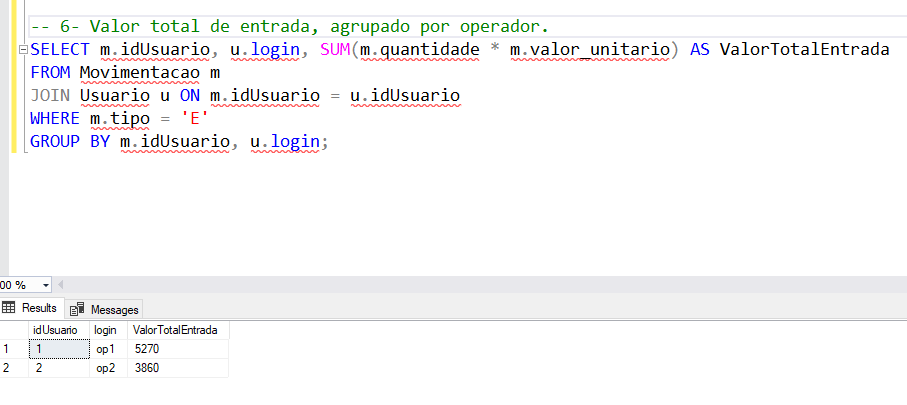
Os dados completos de pessoas físicas e pessoas jurídicas já foram apresentados acima quando da criação dos mesmos.

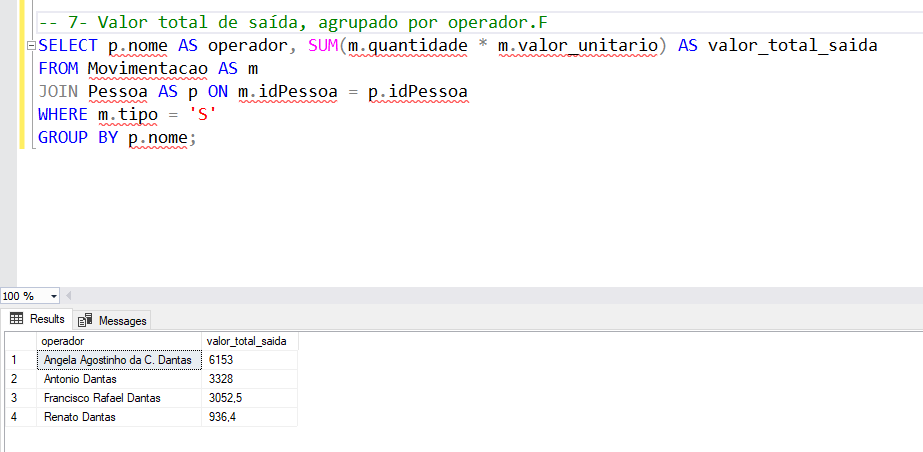


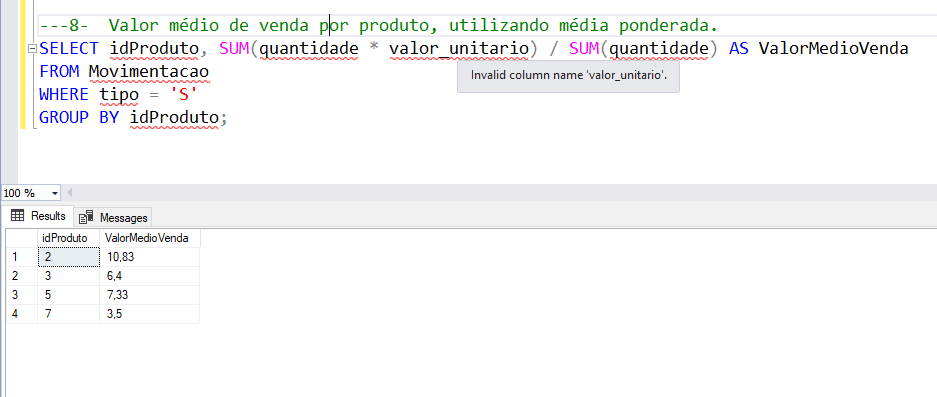












**Análise e Conclusão:**

**Quais as diferenças no uso de sequence e identity?**

Ambos são geradores de sequência numérica num Banco de dado. A diferença é que identity é criado na coluna da própria tabela onde está sendo declarada. Já o sequence é um objeto que não está vinculado a uma tabela específica e que por isso pode ser usada em mais de uma.

**Qual a importância das chaves estrangerias para a consistência do banco?**

As chaves estrangeiras (foreign Keys – FK) é justamente o faz com que haja relação entre um dado em uma row (linha) de uma tabela com a row na outra tabela. É essa linkagem que mantém a integridade das informações, por isso que chamamos de Bando de Dados Relacional. Nesse relacionamento é que se fortalece a integridade das informações, nos permite fazer as junções (Join), integridade referencial, prevenção de dados inválidos, e muito mais.

**Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no**

**cálculo relacional?**

Os operadores SQL que pertencem à álgebra relacional: seleção, projeção, união, interseção e diferença.  
Seleção, é quando se recupera uma linha de uma tabela que atenda a uma condição específica do tipo: coluna idade > 30.  
Projeção, é usado para selecionar colunas específicas de uma tabela e com elas criar uma nova tabela.

União, é usado para combinar duas tabelas e retorna todas as linhas distintas.

Interseção, é usado para encontrar as linhas que estão em ambas as tabelas.

Diferença, é quando retorna as linhas que estão numa tabela mas não na outra.

Já o cálculo relacional fica mais próximo da lógica matemática e pode ser mais abstrato do que a álgebra relacional, que é mais operacional. Na linguagem SQL, os operadores como SELECT, FROM, WHERE, JOIN, GROUP BY, HAVING, ORDER BY, entre outros, são usados para realizar consultas e manipulações de dados. Eles representam operações mais práticas e funcionais que são usadas em bancos de dados do mundo real.

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

O agrupamento é feito agrupando-se linhas (row) de dados com valores comuns em mais de uma coluna (column) onde se aplica uma função de ajuntamento como SUM, COUNT, AVG, MAX OU MIN a esses grupos resultantes. E claro que o requisito obrigatório é GROUP BY.

Um exemplo básico e padrão para um agrupamento:

**SELECT coluna1, coluna2, funcao\_agregacao(coluna3)** // aqui na coluna3 pode ser aplicada uma função de agregação do tipo: um somatório, uma média, valores a partir de um certo número e por aí vai.

**FROM** tabela

**GROUP BY coluna1, coluna2**;

SELECT: Especifica as colunas que você deseja selecionar na consulta.

FROM: Especifica a tabela da qual você está selecionando os dados.

GROUP BY: Indica as colunas que servirão como critério de agrupamento. As linhas de dados serão agrupadas com base nos valores dessas colunas.

Atenção: apenas as colunas listadas na cláusula GROUP BY podem ser selecionadas na lista de seleção (SELECT). Uma coisa que observei no SQL é que a ordem das variáveis pode vir antes de serem criadas. Exemplo. Está se selecionando colunas antes de dizer em qual tabela elas estão e além disso qual a restrição “GROUP BY”, digo: só se pode usar as colunas que estão especificadas no agrupamento.