

Universidade Estácio

Campus: POLO FSP-RO

Curso: Desenvolvimento Full Stack

Disciplina: Missão prática do 2º nível de conhecimento

Turma: 2025

Semestre Letivo: 1

Integrantes: Eliton Rodriguês de Oliveira

2º Procedimento Alimentando a Base de Dados

Título da Prática:

Alimentando a Base de Dados no SQL Server Management Studio

Objetivo da Prática:

Popular as tabelas criadas no banco de dados com dados básicos, incluindo usuários, produtos, pessoas físicas e jurídicas, e movimentações. Realizar consultas SQL para verificar a consistência dos dados.

1. Inserção de Usuários

Os seguintes comandos SQL foram utilizados para inserir os usuários operadores no banco de dados:

INSERT INTO Usuario (nome, email, senha) VALUES ('Operador1', 'op1@empresa.com', 'op1');

INSERT INTO Usuario (nome, email, senha) VALUES ('Operador2', 'op2@empresa.com', 'op2');

Observação: Para sistemas reais, as senhas devem ser armazenadas em formato hash, codificado em Base64, para garantir segurança.

2. Inserção de Produtos

Os seguintes produtos foram adicionados à base de dados:

INSERT INTO Produto (nome, quantidade, precoVenda) VALUES ('Produto A', 100, 50.00);

INSERT INTO Produto (nome, quantidade, precoVenda) VALUES ('Produto B', 200, 30.00);

3. Inserção de Pessoas

Obter o próximo ID da sequence:

SELECT NEXT VALUE FOR PessoaSeq;

Inserir Pessoa Comum:

INSERT INTO Pessoa (idPessoa, nome, endereco, telefone) VALUES (1, 'Empresa XYZ', 'Rua 1, 100', '(11) 99999-9999');

INSERT INTO Pessoa (idPessoa, nome, endereco, telefone) VALUES (2, 'João Silva', 'Avenida 2, 200', '(11) 98888-8888');

Inserir Pessoa Jurídica:

INSERT INTO PessoaJuridica (idPessoa, CNPJ) VALUES (1, '12.345.678/0001-99');

Inserir Pessoa Física:

INSERT INTO PessoaFisica (idPessoa, CPF) VALUES (2, '123.456.789-00');

4. Criação de Movimentações

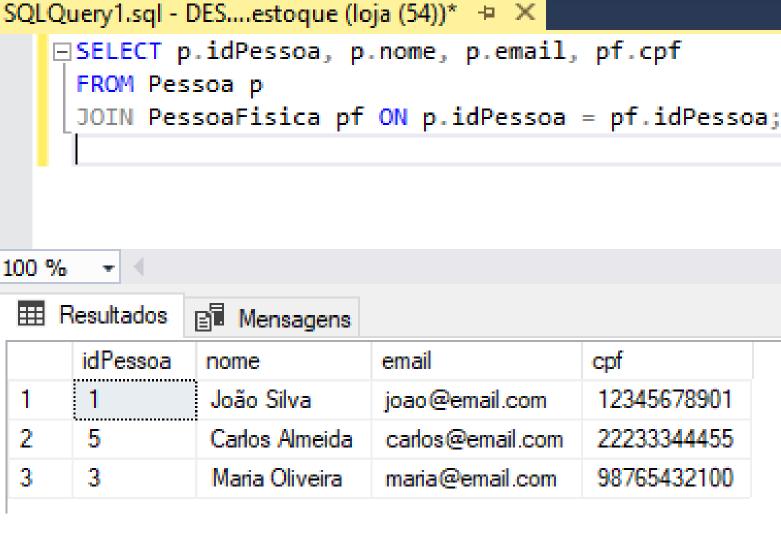
Inserir Movimentações de Entrada e Saída:

INSERT INTO Movimento (tipoMovimento, idProduto, idPessoa, quantidade, precoUnitario) VALUES ('E', 1, 1, 50, 45.00);

INSERT INTO Movimento (tipoMovimento, idProduto, idPessoa, quantidade, precoUnitario) VALUES ('S', 1, 2, 10, 50.00);

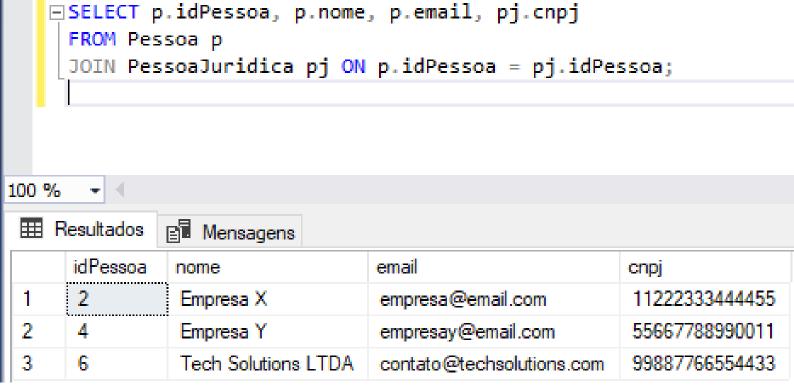
5. Consultas Realizadas

Dados completos de pessoas físicas:



Dados completos de pessoas jurídicas:

SQLQuery1.sql - DES....estoque (loja (54))* □ ×



Movimentações de Entrada:

Im. quantituaue, im. precoonittario, m. vaioriotai

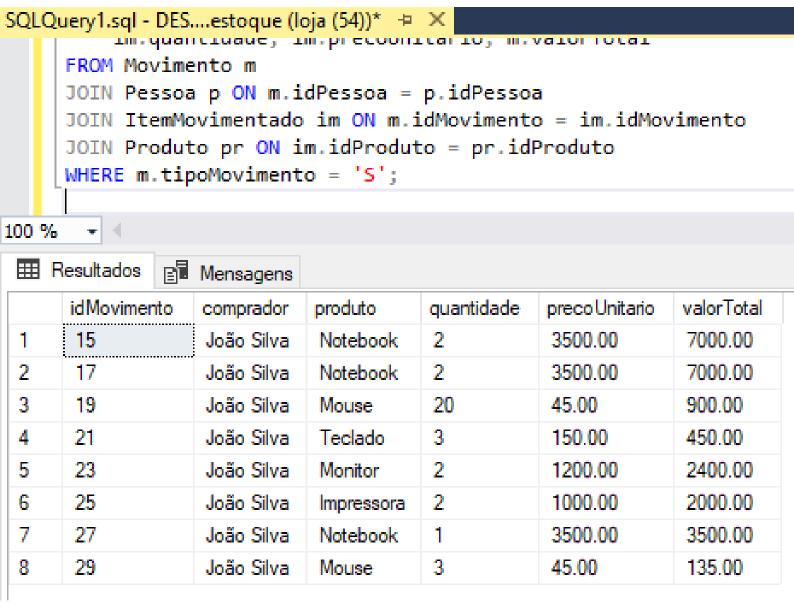
SQLQuery1.sql - DES....estoque (loja (54))* □ ×

```
FROM Movimento m
      JOIN Pessoa p ON m.idPessoa = p.idPessoa
      JOIN ItemMovimentado im ON m.idMovimento = im.idMovimento
      JOIN Produto pr ON im.idProduto = pr.idProduto
      WHERE m.tipoMovimento = 'E';
100 %
        + [4]
 Resultados

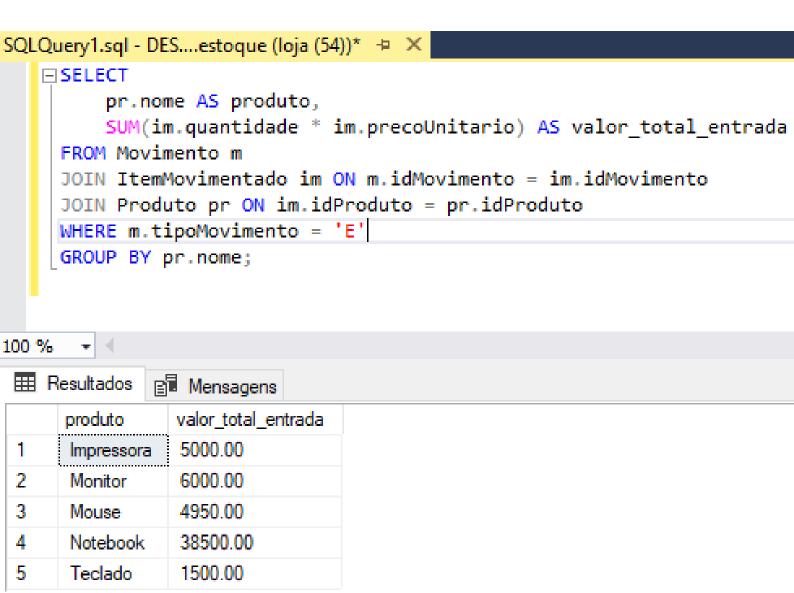
    Mensagens

                                                        preco Unitario
      idMovimento.
                    fomecedor.
                                produto
                                            guantidade
                                                                      valorTotal
                    Empresa X
                                            4
 1
                                Notebook
                                                        3500.00
                                                                      14000.00
      14
                    Empresa X
 2
       16
                                Mouse
                                            50
                                                        45.00
                                                                      2250.00
 3
       18
                    Empresa X
                                Notebook
                                            5
                                                        3500.00
                                                                      17500.00
 4
       20
                    Empresa X
                                Teclado
                                            10
                                                        150.00
                                                                      1500.00
                    Empresa X
                                            5
 5
       22
                                Monitor
                                                        1200 00
                                                                      6000 00
       24
                    Empresa X
                                                        1000.00
 6
                                Impressora
                                            5
                                                                      5000 00
 7
                    Empresa X
                                Notebook
                                            2
                                                        3500.00
       26
                                                                      7000.00
                    Empresa X
 8
       28
                                Mouse
                                            10
                                                        45 00
                                                                      450.00
       30
                    Empresa X
                                                        45 00
 9
                                Mouse
                                            50
                                                                      250 00
```

Movimentações de Saída:



Valor total das entradas agrupadas por produto:



Valor total das saídas agrupadas por produto:

```
SQLQuery1.sql - DES....estoque (loja (54))* → ×

SELECT

pr.nome AS produto,

SUM(im.quantidade * im.precoUnitario) AS valor_total_saida
FROM Movimento m

JOIN ItemMovimentado im ON m.idMovimento = im.idMovimento

JOIN Produto pr ON im.idProduto = pr.idProduto

WHERE m.tipoMovimento = 'S'

GROUP BY pr.nome;
```

Resultados Mensagens		
	produto	valor_total_saida
1	Impressora	2000.00
2	Monitor	2400.00
3	Mouse	1035.00
4	Notebook	17500.00
5	Teclado	450.00

100 %

Análise e Conclusão

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

- Sequence permite reutilização de valores e pode ser usada em múltiplas tabelas.
- Identity é específica de uma tabela e gera valores automáticos para uma coluna.

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

- As chaves estrangeiras garantem a integridade referencial, impedindo registros "soltos" e assegurando que relações entre tabelas sejam mantidas corretamente.
- 3. Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?
 - Operadores da álgebra relacional: SELECT, PROJECT, JOIN, UNION, INTERSECT, DIFFERENCE.

 Operadores do cálculo relacional: EXISTS, FOR ALL, NOT EXISTS, QUANTIFIED PREDICATES.

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

 O agrupamento é realizado com GROUP BY, e qualquer coluna selecionada fora de funções agregadas deve estar no GROUP BY.

Conclusão:

A prática permitiu a alimentação adequada do banco de dados e a realização de consultas SQL, garantindo a consistência dos dados inseridos. O uso de sequence, chaves estrangeiras e operadores SQL foi essencial para estruturar corretamente as informações, garantindo integridade e facilidade de consulta.

Repositório Git

O projeto está armazenado no seguinte repositório Git: ## GitHub - Elitonr65/banco_estoque