

# Vamos manter as informações!

Nome: Michel Angelo da Silva Portugal .

Matrícula: 202208826174 Campus: Duque de Caxias

Disciplina: RPG0015 - Vamos manter as informações! / Sem 2/2023.

#### Objetivos 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)

## Link do Repositório:

#### https://github.com/miaspe/SQL

Análise, Questionamentos e Conclusão:

P:Como são implementadas as diferentes cardinalidades, basicamente 1X1,

1XN ou NxN, em um banco de dados relacional?

- R: Através das ditas chaves primárias e das chaves estrangeiras.
- P: Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?
- R: As chaves estrangeiras fazem uma correlação entre as tabelas juntamente com uma cardinalidade de 1:X, repare que é 1:X e não 1:N, onde x pode variar de 1 pra N.
- P:Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?
- R: Melhora a produtividade por permitir a manipulação (CRUD) de objetos direto no aplicativo sem o uso de outras IDEs e editores de texto tão comuns em outras linguagens.
  - Inserir alguns produtos na base de dados, obtendo um conjunto como o que é apresentado a seguir:



Nosso código e tabela abaixo:

INSERT INTO Produto (ProdutoID,Nome\_Produto,Quantid\_Prod,Preço\_de\_Venda)
VALUES

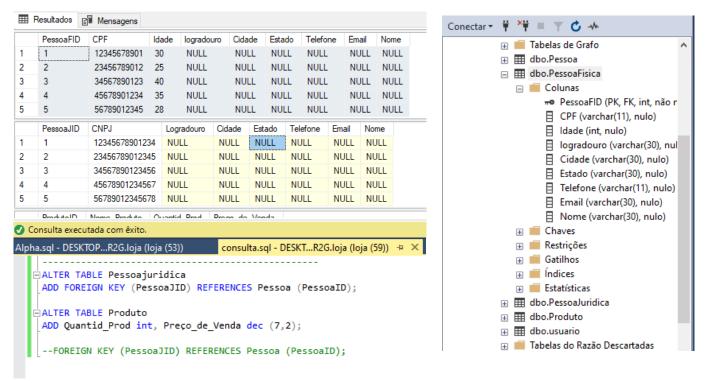
```
(1,'uva RED','15Kg', 8.31),
(2,'Abacaxi','30Unid', 6.22),
(3,'Cebolas','30Kg', 4.55);
```

	ProdutoID	Nome_Produto	Quantid_Prod	Preço_de_Venda
1	1	uva RED	15Kg	8.31
2	2	Abacaxi	30Unid	6.22
3	3	Cebolas	30Kg	4.55

Consulta executada com êxito.

Criar pessoas físicas e jurídicas na base de dados:

#### R: Nossas Tabelas e BD:



Obter o próximo id de pessoa a partir da sequence:

```
SELECT * from dbo. Pessoa
SELECT * from dbo. Pessoa
SELECT * from dbo. Pessoafisica
SELECT * from dbo. Pessoa id sea
SELECT * from dbo. Pessoa

(Nenhum nome de coluna)

1 7

(Nenhum nome de coluna)

1 9
```

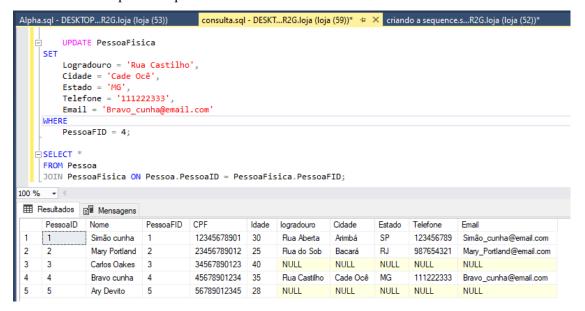
Criar algumas movimentações na base de dados, obtendo um conjunto como o que é apresentado a seguir, onde E representa Entrada e S representa Saída.

LAPTO	APTOP-HJ59HFU4 dbo.movimento 💠 🔀 SQLQuery6.sql - LAPU4.Loja (loja (51)) SQLQuery5.sql - LAPU4.Loja (loja (57))*										
	idMovimento	idUsuario	idPessoa	idProduto	quantidade	tipo	valorUnitario				
	1	1	7	1	20	S	4,00				
	4	1	7	3	15	S	2,00				
	5	2	7	3	10	S	3,00				
	7	1	15	3	15	E	5,00				
	8	1	15	4	20	E	4,00				
w	NULL	NULL	NULL	NULL	NULL	NULL	NULL				

.

- Efetuar as seguintes consultas sobre os dados inseridos:

Dados completos de pessoas físicas:



Dados completos de pessoas jurídicas:

Claro que poderíamos colocar uma tag/flag na forma de coluna a mais em pessoa alternando ou apontando o cadastro para jurídica ou física; E selecionar/apontar pra tag em nosso select PessoaJID, mas pra fins de demonstração, isso não foi implementado ...



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

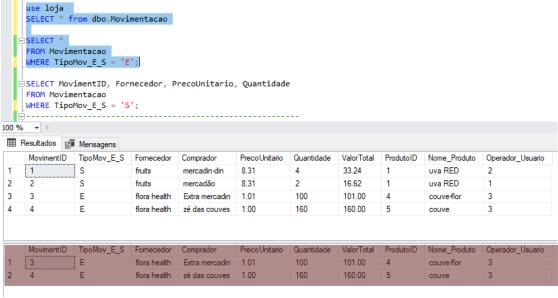


fig1:Entradas

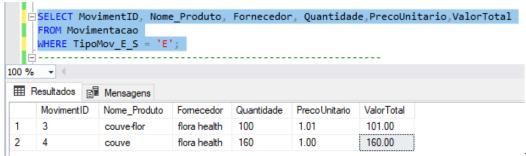


fig2:Entradas filtradas

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

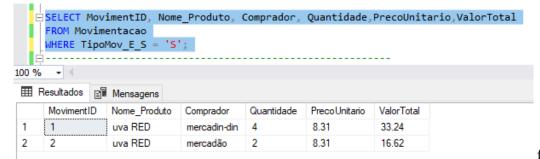
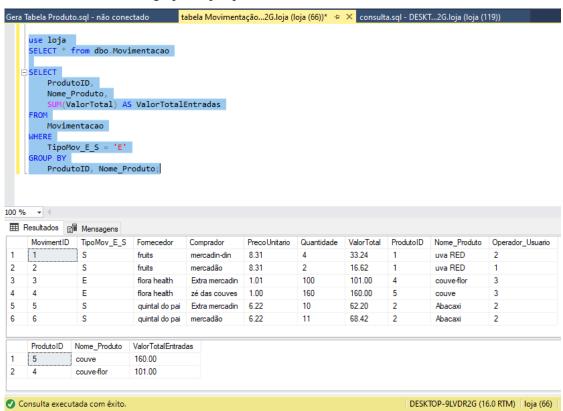


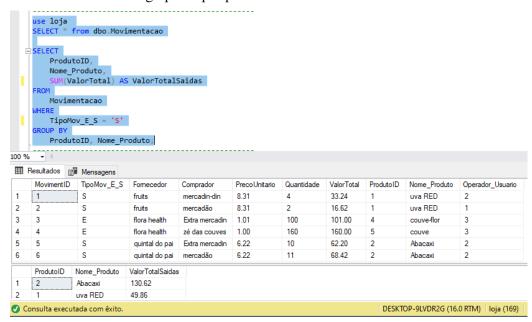
fig3:Saidas filtradas

Valor total das entradas agrupadas por produto:

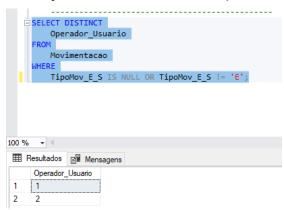


Usar - Group by SUM, faz a soma.

Valor total das saídas agrupadas por produto.

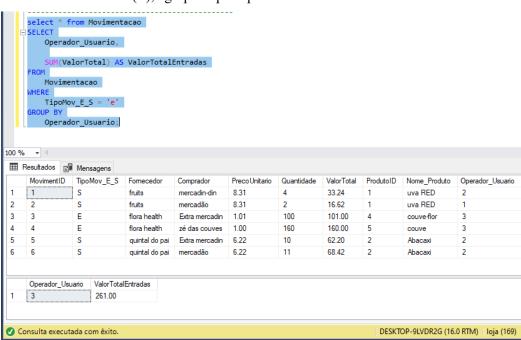


Operadores que não efetuaram movimentações de entrada (compra):



Aqui vemos/filtramos as matrículas de operadores/usuários que **Não (!=) efetuaram** entradas ou movimentações de compra.

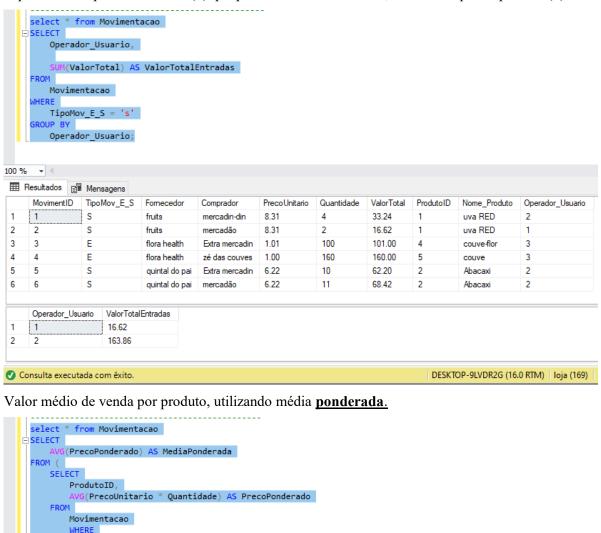
Valor total de entrada (E), agrupado por operador :

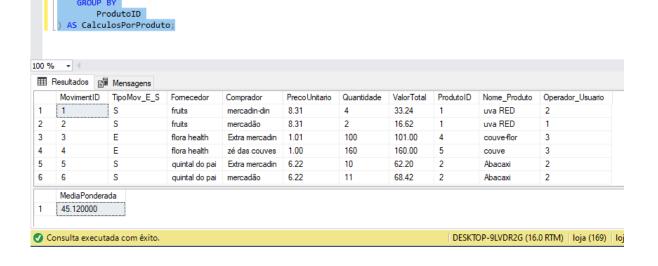


Valor total de saída (S), agrupado por operador :

[TipoMov\_E\_S] = 'S'

Aqui vemos o quanto o usuário (2) que poderia ser VENDEDOR, fez a mais que o operador (1).





Resultados da execução dos códigos também foram apresentados nas fotos;

#### Análise e Conclusão:

- P: Quais as diferenças no uso de sequence e identity?
- R: O recurso (sequence) é mais "novo" (2002) que o identity , que é mais antigo e menos versátil pra manipular/alterar.
  - P:Qual a importância das chaves estrangerias para a consistência do banco?
- R: Estabelecem **relação** entre as tabelas, com isso também temos mais segurança no tocante a informações sensíveis (confidenciais ou sigilosas)
  - P: Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?
- R: Quando falamos de relação lembramos da teoria dos conjuntos (tuplas) que no caso formam as tabelas e por sua vez os objetos que serão manipulados. No tocante as relações podemos destacar a 1:1 onde na teoria dos conjuntos essa relação é não é só uma relação ela é uma função e pode ser expressa como tal, onde cada elemento em X corresponde a um único elemento em Y (no caso de tabelas x é uma tabela e y é outra) as outras relações são somente relações não chegando ao status de função porém temos também JOIN e INNER e todas as outras operações referentes a conjuntos como pra união, interseção(INTERSECT) e outras, podendo também fazermos operações com OR, AND e NOT e ainda > < e = claro que dependendo do conteúdo dos campos(colunas) em questão, que podem armazenar dados Alpha e dados Numéricos ou a junção dos dois.
  - P: Como é feito o agrupamento em consultas, e qual requisito é obrigatório?
- R: Para isso exite a função (GROUP BY) e suas variantes agregatórias (SUM(), COUNT(), AVG(), MAX(), MIN(), etc.)

### Link do Repositório:

https://github.com/miaspe/SQL