



Missão Prática - Nível 1 Mundo 3

Campus : POLO COPACABANA

Curso : DESENVOLVIMENTO FULL STACK

Disciplina : RPG0014 INICIANDO O CAMINHO PELO JAVA

Turma : 9003

Semestre: 2023.3 FLEX

Nome : MARCO SERGIO ALBINO VITTORIO BAROZZI

2º Procedimento | Alimentando a Base









Objetivos:

Utilizar o SQL Server Management Studio para alimentar as tabelas com dados básicos do sistema

Criar dados para as tabelas usuario, produtos, movimento, pessoas físicas e jurídicas na base de dados e visualizar

Códigos desenvolvidos :

Execução de scripts

 01_Pratica.m3,n2.part1 banco user.sql	15/09/2023 20:13	Arquivo SQL	4 KB
 02_Pratica.m3,n2.part2 create tbs views.sql	03/10/2023 04:55	Arquivo SQL	9 KB
 03_Pratica.m3,n2.part3 insert tbs pessoas usuario.sql	02/10/2023 04:35	Arquivo SQL	3 KB
 04_Pratica.m3,n2.part4 prod.sql	14/09/2023 07:02	Arquivo SQL	1 KB
 05_Pratica.m3,n2.part5 mov E.sql	14/09/2023 07:52	Arquivo SQL	4 KB
 06_Pratica.m3,n2.part6 mov S.sql	14/09/2023 08:05	Arquivo SQL	4 KB
 07_Compras Extras part8.sql	14/09/2023 19:20	Arquivo SQL	2 KB
 08_Vendas Extrass part7.sql	14/09/2023 20:14	Arquivo SQL	2 KB

- Criando usuarios pessao fisica e juridica

03_Pratica.m3,n2.part3 insert tbs pessoas usuario.sql

- Criando Produtos

04_Pratica.m3,n2.part4 prod.sql

- Criando movimentos de entrada

05_Pratica.m3,n2.part5 mov E.sql

- Criando movimentos de saida

06_Pratica.m3,n2.part6 mov S.sql

- Inserindo vendas extras

07_Compras Extras part8.sql

- Inserindo compras extras

08_Vendas Extrass part7.sql

01 - Populando tabelas

---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'
--- insere pessoa fisica, juridica e operadores (usuarios)
USE loja;
GO

delete from Movimento

delete from pessoa

-- Insere em Pessoa fisica

-- pf1

INSERT INTO Pessoa (id,nome, logradouro,cidade,estado ,telefone,
email, tipo)
VALUES (NEXT VALUE FOR PessoaIdSeq,'Fulano Silva', 'Rua X',
'Cidade X', 'MG', '(21)3333333', 'fulano@example.com', 'F');

-- Insere em PessoaFisica

INSERT INTO PessoaFisica (idpessoa, cpf)
VALUES ((SELECT MAX(id) FROM Pessoa where tipo='F'),
'69418874083');

-- pf2

INSERT INTO Pessoa (id,nome, logradouro,cidade,estado ,telefone,
email, tipo)
VALUES (NEXT VALUE FOR PessoaIdSeq,'Ciclana Souza', 'Rua Y',
'Cidade Y', 'SP', '(21)4444444', 'ciclana@example.com', 'F');

INSERT INTO PessoaFisica (idpessoa, cpf)

```
VALUES ((SELECT MAX(id) FROM Pessoa where tipo='F'),  
'66203143049');
```

-- pf3

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

```
VALUES ( NEXT VALUE FOR PessoaIdSeq,'Mari Silva', 'Rua X',  
'Cidade T', 'SP', '(21)4445675', 'mari@example.com', 'F');
```

```
INSERT INTO PessoaFisica (idpessoa, cpf)
```

```
VALUES ((SELECT MAX(id) FROM Pessoa where tipo='F'),  
'65241594043');
```

-- Insere em Pessoa jurídica

-- pj1

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

```
VALUES ( NEXT VALUE FOR PessoaIdSeq,'Hortifruti sucesso', 'Rua  
A', 'Cidade A', 'SP', '(11)1111111', 'empresaA@example.com', 'J');
```

```
INSERT INTO PessoaJuridica (idpessoa, cnpj)
```

```
VALUES ((SELECT MAX(id) FROM Pessoa where tipo='J'),  
'30333306000120');
```

-- pj2

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

```
VALUES ( NEXT VALUE FOR PessoaIdSeq,'Mercado principal', 'Rua  
B', 'Cidade B', 'RJ', '(11)2222222', 'empresaB@example.com', 'J');
```

```
INSERT INTO PessoaJuridica (idpessoa, cnpj)
```

```
VALUES ((SELECT MAX(id) FROM Pessoa where tipo='J'),  
'33102645000194');
```

```
-----  
-- insere usuario  
-----
```

```
delete from Usuario
```

```
insert into Usuario (login,senha)  
values('op1','op1'),  
      ('op2','op2'),  
      ('op3','op3');
```

03 – Popula tabela produto

```
---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'  
--- insere produtos
```

```
delete from produto
```

```
-- insert produto  
INSERT INTO Produto ( nome, quantidade, precoVenda)  
VALUES ('banana', 100, 10.5),  
      ('laranja', 50, 20.0),  
      ('manga', 200, 5.75);
```

```
select 'produtos inseridos' as acao
```

```
select * from produto
```

04 – Populando tabela movimento com entradas

```
---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'  
--- insere movimentações de entradas(compras)
```

```
-- 1º insert movimento
```

```
delete from Movimento where tipo='E'
```

--- **OP3 compra 30 bananas do fornecedor Hortifruti sucesso**

```
INSERT INTO Movimento
```

```
(
    idUsuario,
    idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

```
SELECT
    (SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op3'),
    (SELECT TOP 1 id FROM Pessoa WHERE tipo='J' and nome LIKE
'Hort%'),
    (SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'bana%'),
    30,
    (SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'bana%'),
    'E';
```

--- **ver 1º compra**

```
select 'op3 compra 30 bananas' as acao1
```

```
select
    u.login as operador,
    mv.tipo as operacao,
    p.nome as produto,
    mv.quantidade,
    mv.valorUnitario,
    (mv.valorUnitario*mv.quantidade) as total,
    pe.nome as fornecedor,
    pj.cnpj
from (((movimento mv inner join produto p on
p.idProduto=mv.idProduto)
```

```

inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join
Pessoa pe on pe.id=pj.idpessoa)
where mv.tipo='E'

```

-- 2º insert movimento

--- **OP3 compra 10 laranjas do fornecedor Hortifruti sucesso**

INSERT INTO Movimento

```

(
    idUsuario,
    idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)

```

```

SELECT
    (SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op3'),
    (SELECT TOP 1 id FROM Pessoa WHERE tipo='J' and nome LIKE
'Hort%'),
    (SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'lara%'),
    10,
    (SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'lara%'),
    'E';

```

--- **ver 2º compra**

select 'op3 compra 10 laranjas' as acao2

```

select
u.login as operador,

```

```

mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pj.cnpj
from (((movimento mv inner join produto p on
p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join
Pessoa pe on pe.id=pj.idpessoa)
where mv.tipo='E'

```

-- 3º insert movimento

--- OP1 compra 50 mangas do fornecedor Mercado principal

```

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

SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='J' and nome LIKE
'Merc%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'man%'),
50,

```



```
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE  
'man%'),  
'E';
```

--- ver 3º compra

```
select 'op1 compra 50 mangas' as acao2
```

```
select  
u.login as operador,  
mv.tipo as operacao,  
p.nome as produto,  
mv.quantidade,  
mv.valorUnitario,  
(mv.valorUnitario*mv.quantidade) as total,  
pe.nome as fornecedor,  
pj.cnpj  
from (((movimento mv inner join produto p on  
p.idProduto=mv.idProduto)  
inner join usuario u on u.idUsuario=mv.idUsuario)  
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join  
Pessoa pe on pe.id=pj.idpessoa)  
where mv.tipo='E'
```

05 – Populando tabela movimento com saídas

---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'
--- insere movimentações de saída (vendas)

-- 1º insert movimento

```
delete from Movimento where tipo='S'
```

--- OP1 vende 10 bananas

```
INSERT INTO Movimento  
(  
    idUsuario,  
    idPessoa,
```

```

        idProduto,
        quantidade,
        valorUnitario,
        tipo
    )

```

```

SELECT
    (SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
    (SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'fu%'),
    (SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'bana%'),
    10,
    (SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'bana%'),
    'S';

```

--- ver 1° compra

select 'op1 vende 10 bananas para fulano' as acao1

```

select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pf.cpf
from (((movimento mv inner join produto p on
p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa
pe on pe.id=pf.idpessoa)
where mv.tipo='S'

```


-- 2° insert movimento

--- OP1 vende 5 mangas

INSERT INTO Movimento

```
(
    idUsuario,
    idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

```
SELECT
    (SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
    (SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'cic%'),
    (SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'man%'),
    5,
    (SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'man%'),
    'S';
```

--- ver 2º venda

select 'op1 vende 5 mangas para ciclana' as acao2

```
select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pf.cpf
```

```

from (((movimento mv inner join produto p on
p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa
pe on pe.id=pf.idpessoa)
where mv.tipo='S'

```

-- 3° insert movimento

--- OP1 vende 15 laranjas

INSERT INTO Movimento

```

(
    idUsuario,
    idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

```

SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'ma%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE 'la%'),
15,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'la%'),
'S';

```

--- ver 3° venda

select 'op1 vende 15 laranjas para mari' as acao3

select

```

u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pf.cpf
from (((movimento mv inner join produto p on
p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa
pe on pe.id=pf.idpessoa)
where mv.tipo='S'

```

06 - Compras Extras

---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'

--- insere movimentações de entradas(compras) extras

-- 1º insert movimento

```

-----
--- OP3 compra 20 bananas do fornecedor Hortifruti sucesso
-----

```

INSERT INTO Movimento

```

(
    idUsuario,
    idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

SELECT

```

(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op3'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='J' and nome LIKE
'Hort%'),

```

```

(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'bana%'),
20,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'bana%'),
'E';

```

--- OP3 compra 38 laranjas do fornecedor Hortifruti sucesso

```

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

```

```

SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op3'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='J' and nome LIKE
'Hort%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'lar%'),
38,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'lar%'),
'E';

```

07 - Vendas Extras

---ATENÇÃO - NECESSÁRIO ESTAR LOGADO COM 'loja'
--- insere movimentações de saidas(vendas) extras

--- OP1 vende 20 mangas a fulano

```
INSERT INTO Movimento
```

```
(
    idUsuario,
idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

```
SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'ful%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'man%'),
20,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'man%'),
'S';
```

--- OP1 vende 20 laranjas a ciclana

```
INSERT INTO Movimento
```

```
(
    idUsuario,
idPessoa,
    idProduto,
    quantidade,
    valorUnitario,
    tipo
)
```

```
SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op1'),
```

```

(SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'cic%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE 'la%'),
20,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'la%'),
'S';

```

```

-----
--- OP2 vende 30 bananas a Mari
-----

```

```

INSERT INTO Movimento

```

```

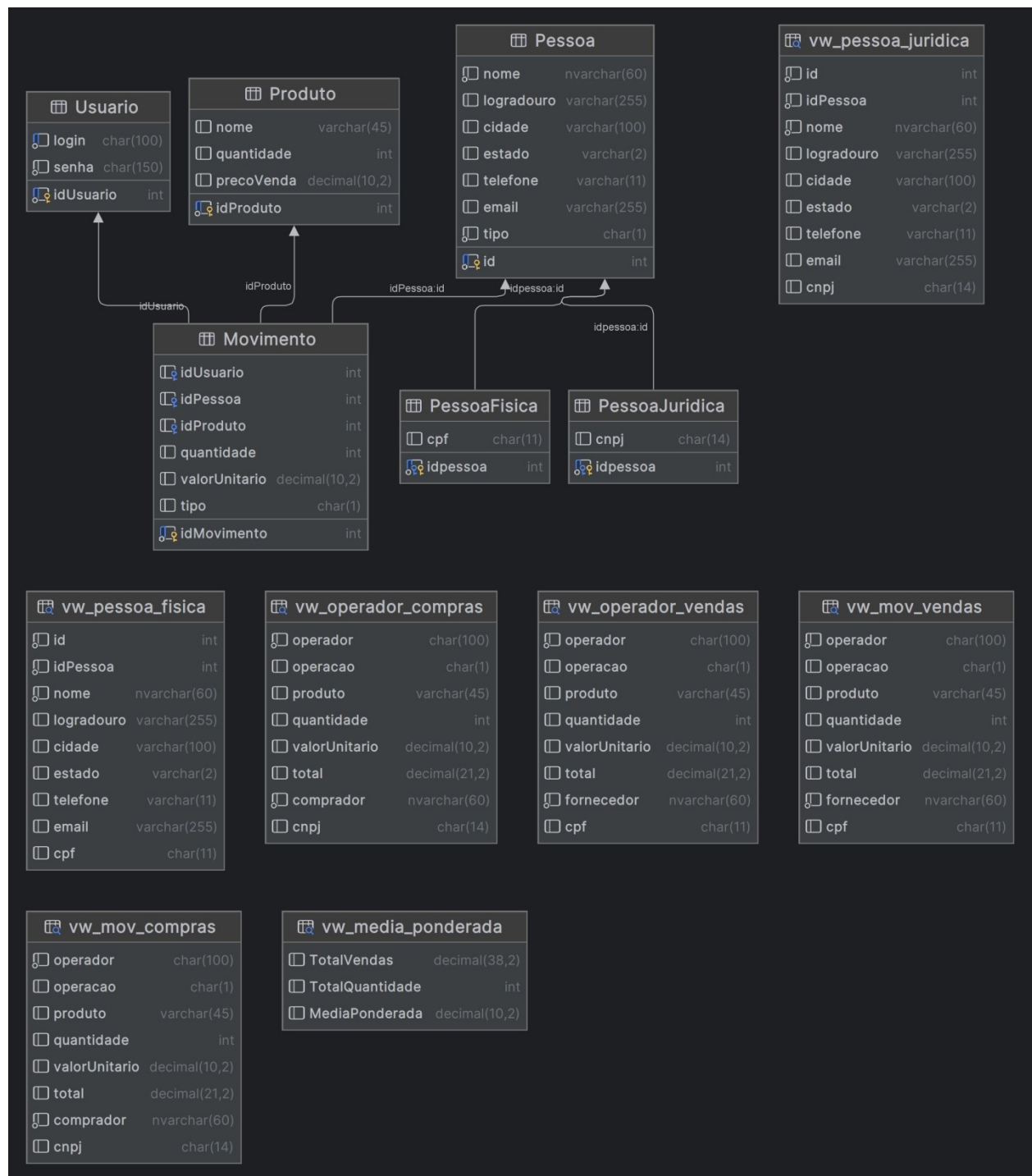
    (
        idUsuario,
    idPessoa,
        idProduto,
        quantidade,
        valorUnitario,
        tipo
    )

```

```

SELECT
(SELECT TOP 1 idUsuario FROM Usuario WHERE login = 'op2'),
(SELECT TOP 1 id FROM Pessoa WHERE tipo='F' and nome LIKE
'mar%'),
(SELECT TOP 1 idProduto FROM Produto WHERE nome LIKE
'ba%'),
30,
(SELECT TOP 1 precoVenda FROM Produto WHERE nome LIKE
'ba%'),
'S';

```

Resultados da execução dos códigos

1- Dados completos de pessoas físicas.

```
select * from vw_pessoa_fisica
```

	id	idPessoa	nome	logradouro	cidade	estado	telefone	email	cpf
1	1	1	Fulano Silva	Rua X	Cidade X	MG	(21)3333333	fulano@example.com	69418874083
2	2	2	Ciclana Souza	Rua Y	Cidade Y	SP	(21)4444444	ciclana@example.com	66203143049
3	3	3	Mari Silva	Rua X	Cidade T	SP	(21)4445675	mari@example.com	65241594043

```
-----  
-- Fisica  
-----  
  
CREATE VIEW vw_pessoa_fisica AS  
select p.id,pf.idPessoa,p.nome,p.logradouro,p.cidade,p.estado,p.telefone,p.email,pf.cpf  
from (pessoa p inner join pessoafisica pf on p.id=pf.idpessoa)
```

2 - Dados completos de pessoas Jurídicas

```
select * from vw_pessoa_juridica
```

	id	idPessoa	nome	logradouro	cidade	estado	telefone	email	cnpj
1	4	4	Hortifruti sucesso	Rua A	Cidade A	SP	(11)1111111	empresaA@example.com	30333306000120
2	5	5	Mercado principal	Rua B	Cidade B	RJ	(11)2222222	empresaB@example.com	33102645000194

```
-----  
-- Juridica  
-----  
  
CREATE VIEW vw_pessoa_juridica AS  
select p.id,pj.idPessoa,p.nome,p.logradouro,p.cidade,p.estado,p.telefone,p.email,pj.cnpj  
from (pessoa p inner join pessoaJuridica pj on p.id=pj.idpessoa)
```

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

```
select * from vw_mov_compras
```

	operador	operacao	produto	quantidade	valorUnitario	total	comprador	cnpj
1	op3	E	banana	30	10.50	315.00	Hortifruti sucesso	30333306000120
2	op3	E	laranja	10	20.00	200.00	Hortifruti sucesso	30333306000120
3	op1	E	manga	50	5.75	287.50	Mercado principal	33102645000194

3.1 compras extras

```
select * from vw_mov_compras
```

	operador	operacao	produto	quantidade	valorUnitario	total	comprador	cnpj
1	op3	E	banana	30	10.50	315.00	Hortifruti sucesso	30333306000120
2	op3	E	laranja	10	20.00	200.00	Hortifruti sucesso	30333306000120
3	op1	E	manga	50	5.75	287.50	Mercado principal	33102645000194
4	op3	E	banana	20	10.50	210.00	Hortifruti sucesso	30333306000120
5	op3	E	laranja	38	20.00	760.00	Hortifruti sucesso	30333306000120

Compras de fornecedores (E)

- 1 - OP3 compra 30 bananas do fornecedor Hortifruti sucesso
- 2 - OP3 compra 10 laranjas do fornecedor Hortifruti sucesso
- 3 - OP1 compra 50 mangas do fornecedor Mercado principal
- 4 - OP3 compra 20 bananas do fornecedor Hortifruti sucesso(extras)
- 5 - OP3 compra 38 laranjas do fornecedor Hortifruti sucesso(extras)

```

-- Movimentações de compras
-----

CREATE VIEW vw_mov_compras AS
select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as comprador,
pj.cnpj
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join Pessoa pe on pe.id=pj.idPessoa)
where mv.tipo='E'

```

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

`select * from vw_mov_vendas`

	operador	operacao	produto	quantidade	valorUnitario	total	fornecedor	cpf
1	op1	S	banana	10	10.50	105.00	Fulano Silva	69418874083
2	op1	S	laranja	15	20.00	300.00	Mari Silva	65241594043
3	op1	S	manga	5	5.75	28.75	Ciclana Souza	66203143049

4.1 – vendas extras

`select * from vw_mov_vendas`

	operador	operacao	produto	quantidade	valorUnitario	total	fornecedor	cpf
1	op1	S	banana	10	10.50	105.00	Fulano Silva	69418874083
2	op1	S	manga	5	5.75	28.75	Ciclana Souza	66203143049
3	op1	S	laranja	15	20.00	300.00	Mari Silva	65241594043
4	op1	S	manga	20	5.75	115.00	Fulano Silva	69418874083
5	op1	S	laranja	20	20.00	400.00	Ciclana Souza	66203143049
6	op2	S	banana	30	10.50	315.00	Mari Silva	65241594043

Vendas (S)

- 1 - OP1 vende 10 bananas a Fulano Silva
- 2 - OP1 vende 5 mangas a Ciclana Souza
- 3 - OP1 vende 15 laranjas a Mari Silva
- 4 - OP1 vende 20 mangas a fulano(extras)
- 5 - OP1 vende 20 laranjas a ciclana(extras)
- 6 - OP2 vende 30 bananas a Mari(extras)

```
-----  
-- Movimentações de vendas  
-----  
  
CREATE VIEW vw_mov_vendas AS  
select  
u.login as operador,  
mv.tipo as operacao,  
p.nome as produto,  
mv.quantidade,  
mv.valorUnitario,  
(mv.valorUnitario*mv.quantidade) as total,  
pe.nome as fornecedor,  
pf.cpf  
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)  
inner join usuario u on u.idUsuario=mv.idUsuario)  
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa pe on pe.id=pf.idpessoa)  
where mv.tipo='S'
```


5- Valor total das entradas agrupadas por produto.

```

1 select * from vw_mov_compras
2 select produto, sum(total) as total from vw_mov_compras
   group by produto

```

operador	operacao	produto	quantidade	valorUnitario	total	comprador	cnpj
op3	E	banana	30	10.50	315.00	Hortifruti sucesso	30333306000120
op3	E	laranja	10	20.00	200.00	Hortifruti sucesso	30333306000120
op1	E	manga	50	5.75	287.50	Mercado principal	33102645000194
op3	E	banana	20	10.50	210.00	Hortifruti sucesso	30333306000120
op3	E	laranja	38	20.00	760.00	Hortifruti sucesso	30333306000120

produto	total
banana	525.00
laranja	960.00
manga	287.50

```

-- Movimentações de compras
CREATE VIEW vw_mov_compras AS
select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as comprador,
pj.cnpj
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join Pessoa pe on pe.id=pj.idPessoa)
where mv.tipo='E'

```

6- Valor total das saídas agrupadas por produto.

```
select * from vw_mov_vendas 1
```

```
select produto,sum(total) as total from vw_mov_vendas
group by produto 2
```

operador	operacao	produto	quantidade	valorUnitario	total	fornecedor	cpf
op1	S	banana	10	10.50	105.00	Fulano Silva	69418874083
op1	S	manga	5	5.75	28.75	Ciclana Souza	66203143049
op1	S	laranja	15	20.00	300.00	Mari Silva	65241594043
op1	S	manga	20	5.75	115.00	Fulano Silva	69418874083
op1	S	laranja	20	20.00	400.00	Ciclana Souza	66203143049
op2	S	banana	30	10.50	315.00	Mari Silva	65241594043

produto	total
banana	420.00
laranja	700.00
manga	143.75

```
-- Movimentações de vendas
```

```
CREATE VIEW vw_mov_vendas AS
select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pf.cpf
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa pe on pe.id=pf.idpessoa)
where mv.tipo='S'
```

]

7- Operadores que não efetuaram movimentações de entrada (compra).

```
select * from vw_operador_compras 1
```

```
select * from Usuario where login not in(  
select operador from vw_operador_compras  
) 2
```

	operador	operacao	produto	quantidade	valorUnitario	total	comprador	cnpj	
1	op3	E	banana	30	10.50	315.00	Hortifruti sucesso	30333306000120	
2	op3	E	laranja	10	20.00	200.00	Hortifruti sucesso	30333306000120	1
3	op1	E	manga	50	5.75	287.50	Mercado principal	33102645000194	
4	op3	E	banana	20	10.50	210.00	Hortifruti sucesso	30333306000120	
5	op3	E	laranja	38	20.00	760.00	Hortifruti sucesso	30333306000120	

	idUsuario	login	senha	
1	2	op2	op2	2 operadores que não efetuaram compras

```
-----  
--- Operadores compras  
-----  
  
CREATE VIEW vw_operador_compras AS  
select  
u.login as operador,  
mv.tipo as operacao,  
p.nome as produto,  
mv.quantidade,  
mv.valorUnitario,  
(mv.valorUnitario*mv.quantidade) as total,  
pe.nome as comprador,  
pj.cnpj  
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)  
inner join usuario u on u.idUsuario=mv.idUsuario)  
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join Pessoa pe on pe.id=pj.idPessoa)  
where mv.tipo='E';  
GO
```


8- Valor total de entrada, agrupado por operador.

```
select * from vw_operador_compras
```

1

```
select operador ,sum(total) as total  
from vw_operador_compras group by operador
```

2

	operador	operacao	produto	quantidade	valorUnitario	total	comprador	cnpj
1	op3	E	banana	30	10.50	315.00	Hortifruti sucesso	30333306000120
2	op3	E	laranja	10	20.00	200.00	Hortifruti sucesso	30333306000120
3	op1	E	manga	50	5.75	287.50	Mercado principal	33102645000194
4	op3	E	banana	20	10.50	210.00	Hortifruti sucesso	30333306000120
5	op3	E	laranja	38	20.00	760.00	Hortifruti sucesso	30333306000120

1

	operador	total
1	op1	287.50
2	op3	1485.00

2

```
-----  
--- Operadores compras  
-----  
  
CREATE VIEW vw_operador_compras AS  
select  
u.login as operador,  
mv.tipo as operacao,  
p.nome as produto,  
mv.quantidade,  
mv.valorUnitario,  
(mv.valorUnitario*mv.quantidade) as total,  
pe.nome as comprador,  
pj.cnpj  
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)  
inner join usuario u on u.idUsuario=mv.idUsuario)  
left join PessoaJuridica pj on mv.idPessoa=pj.idPessoa) inner join Pessoa pe on pe.id=pj.idPessoa)  
where mv.tipo='E';  
GO
```

9- Valor total de saída, agrupado por operador.

SQL queries and results for operator sales aggregation.

Query 1:

```
select * from vw_operador_vendas
```

Query 2:

```
select operador ,sum(total) as total
from vw_operador_vendas group by operador
```

Results of Query 1:

operador	operacao	produto	quantidade	valorUnitario	total	fornecedor	cpf
op1	S	banana	10	10.50	105.00	Fulano Silva	69418874083
op1	S	manga	5	5.75	28.75	Ciclana Souza	66203143049
op1	S	laranja	15	20.00	300.00	Mari Silva	65241594043
op1	S	manga	20	5.75	115.00	Fulano Silva	69418874083
op1	S	laranja	20	20.00	400.00	Ciclana Souza	66203143049
op2	S	banana	30	10.50	315.00	Mari Silva	65241594043

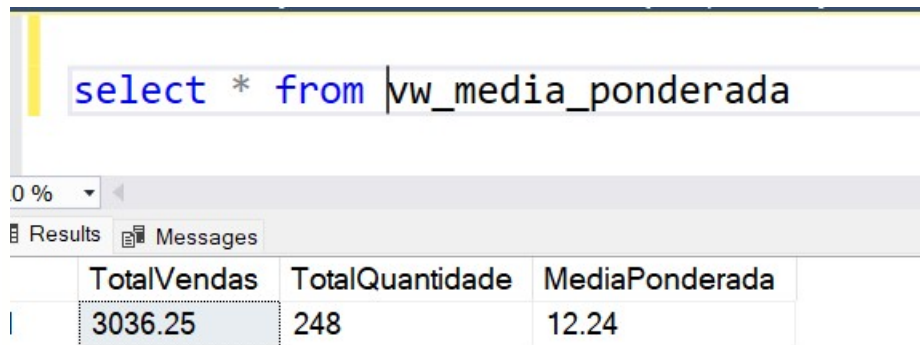
Results of Query 2:

operador	total
op1	948.75
op2	315.00

View Definition:

```
CREATE VIEW vw_operador_vendas AS
select
u.login as operador,
mv.tipo as operacao,
p.nome as produto,
mv.quantidade,
mv.valorUnitario,
(mv.valorUnitario*mv.quantidade) as total,
pe.nome as fornecedor,
pf.cpf
from (((movimento mv inner join produto p on p.idProduto=mv.idProduto)
inner join usuario u on u.idUsuario=mv.idUsuario)
left join PessoaFisica pf on mv.idPessoa=pf.idPessoa) inner join Pessoa pe on pe.id=pf.idpessoa)
where mv.tipo='S'
```

10- Valor médio de venda por produto, utilizando média ponderada



The screenshot shows a SQL query execution interface. The query entered is `select * from vw_media_ponderada`. Below the query, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with three columns: 'TotalVendas', 'TotalQuantidade', and 'MediaPonderada'. The table contains one row of data.

TotalVendas	TotalQuantidade	MediaPonderada
3036.25	248	12.24

```
-----  
-- Operadores media ponderada  
-----  
  
CREATE VIEW vw_media_ponderada AS  
SELECT  
    SUM(valorUnitario * quantidade) AS TotalVendas,  
    SUM(quantidade) AS TotalQuantidade,  
    CAST(FLOOR(SUM(valorUnitario * quantidade) * 100 / SUM(quantidade)) / 100 AS DECIMAL(10, 2)) AS MediaPonderada  
FROM  
    Movimento;
```

Análise e Conclusão:

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

A sequence se trata de um objeto separado ao qual pode ser usado em qualquer coluna, desde que numérica enquanto que identity é um atributo que por ser restrita e única por coluna, é usada como chave primaria.

A sequence porem, não é impeditida de ser usada como chave primaria desde que com o controle adequado.

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

A integridade referencial ou consistencia dos dados relacionados entre duas tabelas

Evitar dados orfãos. Como dito acima, consistencia dos dados

Perfomance pois otimizam as consultas

Garantem transações seguras |

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

Seleção (select * from)

Projeção (select col1,col2 from)

Produto cartesiano(from Tab1,Tab2)

União (union)

Diferença (EXCEPT SELECT, where col not in(),where not(col=))

Junção (Join)

Divisão - (feito através de uma combinação de operações)

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

Feito através da clausula group by onde somente colunas declaradas poderão ser declaradas na query ou outras colunas que seja aplicadas funções agregadas com SUM,AVG,COUNT.etc..|