



POLO BARRA WORLD
CURSO DESENVOLVIMENTO FULL STACK

Disciplina: Nível 2 – Vamos manter as informações?

Turma: 2024.3 Flex / 3º Semestre

Aluno: Paola Savedra Barreiros

Matrícula: 2023.0701.4731

Repositório Github: [pasavedra/RPG0015-Vamos-manter-as-informacoes](https://github.com/pasavedra/RPG0015-Vamos-manter-as-informacoes)
(github.com)

Missão Prática | Nível 1 | Mundo 3

Título da prática: RPG0015 – Vamos manter as informações?

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 SOLICITADOS NO ROTEIRO

PROCEDIMENTO 01: CRIANDO O BANCO DE DADOS

```
create database Loja;

drop database Loja;

use Loja;

create table pessoa(
    idpessoa int NOT NULL ,
    nome varchar(255) NOT NULL ,
    logradouro varchar(255) NOT NULL ,
    cidade varchar(255) NOT NULL ,
    estado char(2) NOT NULL ,
    telefone varchar(11) NOT NULL ,
    email varchar(255) NOT NULL ,
    primary key(idpessoa));

create table pessoa_fisica (
    idpessoa int NOT NULL,
    cpf varchar(255) NOT NULL,
    primary key (idpessoa),
    foreign key (idpessoa) references pessoa(idpessoa));

create table pessoa_juridica (
    idpessoa int NOT NULL,
    cnpj varchar(255) NOT NULL,
    primary key (idpessoa),
    foreign key (idpessoa) references pessoa(idpessoa));

create table produto (
    idproduto int NOT NULL ,
```

```
nome varchar(255) NOT NULL ,  
quantidade varchar(255) NOT NULL ,  
preco_venda numeric(5,2) NOT NULL ,  
primary key(idproduto));
```

```
create table usuario (  
idusuario int NOT NULL ,  
login varchar(255) NOT NULL ,  
senha varchar(255) NOT NULL ,  
primary key(idusuario));
```

```
create table movimento (  
idmovimento int NOT NULL,  
Usuario_idUsuario int NOT NULL ,  
pessoa_idpessoa int NOT NULL ,  
produto_idproduto int NOT NULL ,  
quantidade int NOT NULL ,  
tipo char NOT NULL ,  
valorUnitario numeric(5,2) NOT NULL ,  
primary key(idmovimento),  
foreign key (Usuario_idUsuario) references usuario(idusuario),  
foreign key (produto_idproduto) references produto(idproduto),  
foreign key (pessoa_idpessoa) references pessoa(idpessoa));
```

```
create sequence seq_Pessoa  
as numeric  
start with 1  
increment by 1  
no cycle;
```

PROCEDIMENTO 02: ALIMENTANDO A BASE

use Loja;

insert into usuario

values (1, 'op1', 'op1'),(2, 'op2', 'op2');

insert into produto

values (1, 'Banana', 100, 5.00),(3, 'Laranja', 500, 2.00),(4, 'Manga', 800, 4.00);

insert into pessoa

values (NEXT VALUE FOR seq_Pessoa, 'Joao', 'Rua 12, cas 3, Quitanda',
'Riacho do Sul', 'PA', '1111-1111','joao@riacho.com');

insert into pessoa

values (NEXT VALUE FOR seq_Pessoa, 'JJC', 'Rua 11, Centro',
'Riacho do Norte', 'PA', '1212-1212','jjc@riacho.com');

insert into pessoa_fisica

values (1,'111111111111');

insert into pessoa_juridica

values (2,'22222222222222');

insert into movimento

values (1,1,1,1,20,'S',4.00),

(4,1,1,3,15,'S',2.00),

(5,2,1,3,10,'S',3.00),

(7,1,2,3,15,'E',5),

(8,1,2,4,20,'E',4.00);

-- Dados completos de pessoas físicas.

```
select *  
from pessoa, pessoa_fisica  
where pessoa.idpessoa = pessoa_fisica.idpessoa_fisica;
```

--Dados completos de pessoas jurídicas.

```
select *  
from pessoa, pessoa_juridica  
where pessoa.idpessoa = pessoa_juridica.idpessoa_juridica;
```

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

```
select idmovimento, produto_idproduto, produto.nome as 'Produto',pessoa_idpessoa,  
pessoa.nome as 'Fornecedor', movimento.quantidade, valorUnitario,  
(movimento.quantidade * valorUnitario) as valor_total  
from movimento  
join pessoa  
on movimento.pessoa_idpessoa = pessoa.idpessoa  
join produto  
on movimento.produto_idproduto = produto.idproduto  
where movimento.tipo = 'E';
```

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

```
select idmovimento, produto_idproduto, produto.nome as 'Produto',pessoa_idpessoa,  
pessoa.nome as 'Comprador', movimento.quantidade, valorUnitario,  
(movimento.quantidade * valorUnitario) as valor_total  
from movimento  
join pessoa  
on movimento.pessoa_idpessoa = pessoa.idpessoa  
join produto  
on movimento.produto_idproduto = produto.idproduto
```

```
where movimento.tipo = 'S';
```

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

```
select produto.nome, SUM (movimento.quantidade * movimento.valorUnitario) AS 'VALOR  
TOTAL ENTRADAS'
```

```
from movimento
```

```
JOIN produto
```

```
on produto.idproduto = movimento.produto_idproduto
```

```
where movimento.tipo = 'E'
```

```
group by produto.nome;
```

```
--Valor total das saídas agrupadas por produto.
```

```
select produto.nome, SUM (movimento.quantidade * movimento.valorUnitario) AS 'VALOR  
TOTAL SAIDAS'
```

```
from movimento
```

```
JOIN produto
```

```
on produto.idproduto = movimento.produto_idproduto
```

```
where movimento.tipo = 'S'
```

```
group by produto.nome;
```

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

```
select movimento.Usuario_idUsuario AS 'ID DO OPERADOR'
```

```
from movimento
```

```
except
```

```
select movimento.Usuario_idUsuario
```

```
from movimento
```

```
where movimento.tipo = 'E';
```

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

```
select usuario.login AS OPERADOR, SUM (movimento.quantidade *  
movimento.valorUnitario) AS 'VALOR TOTAL ENTRADAS'
```

```
from movimento
```

JOIN usuario

on usuario.idusuario = movimento.Usuario_idUsuario

where movimento.tipo = 'E'

group by usuario.login;

--Valor total de saída, agrupado por operador.

select usuario.login AS OPERADOR, SUM (movimento.quantidade *
movimento.valorUnitario) AS 'VALOR TOTAL SAIDAS'

from movimento

JOIN usuario

on usuario.idusuario = movimento.Usuario_idUsuario

where movimento.tipo = 'S'

group by usuario.login;

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

select produto.nome, SUM (movimento.quantidade * movimento.valorUnitario) /
SUM(movimento.quantidade) as 'Valor médio de venda'

from movimento

JOIN produto

on produto.idproduto = movimento.produto_idproduto

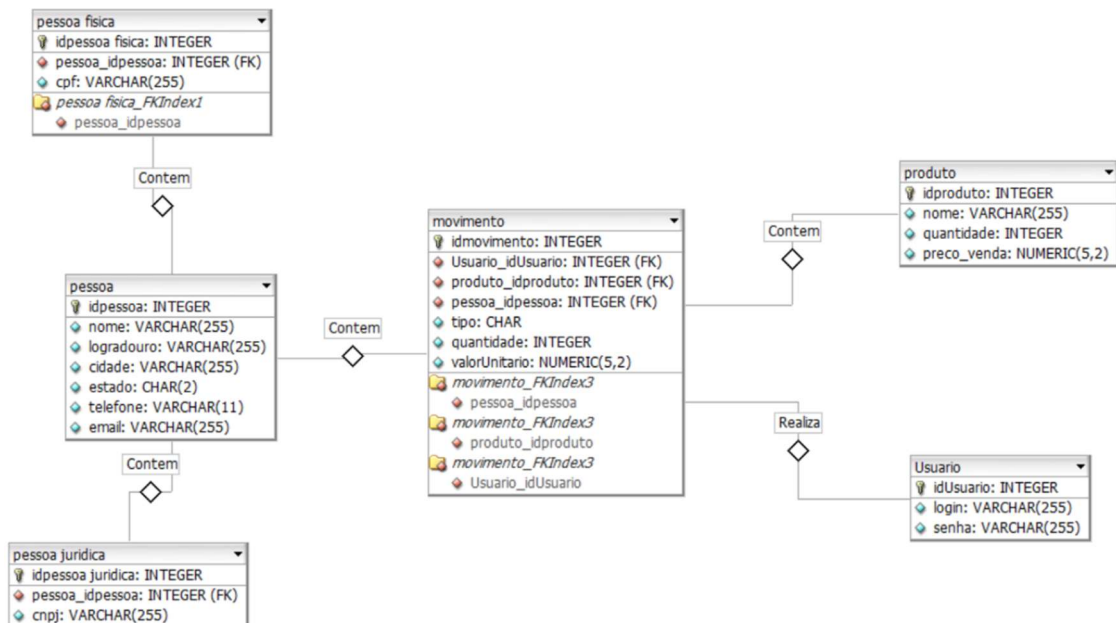
where movimento.tipo = 'S'

group by produto.nome;

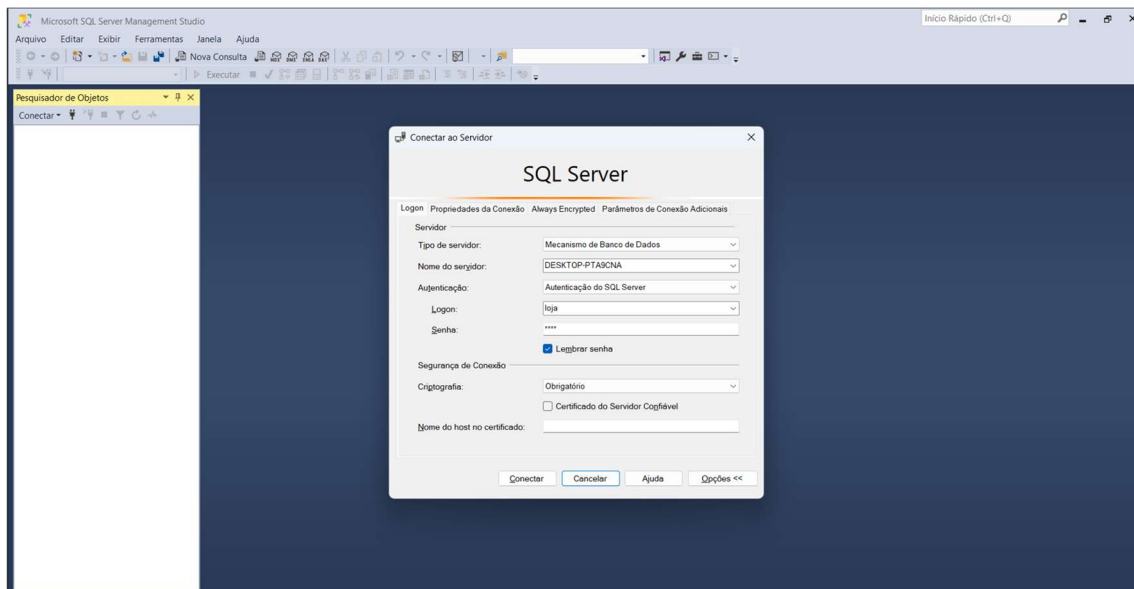
RESULTADO DAS EXECUÇÕES

PROCEDIMENTO 01: CRIANDO O BANCO DE DADOS

Modelo de Dados:



Banco de dados e tabelas:



Excluir Objeto

Selecionar uma página

Geral

Conexão

Servidor:
DESKTOP-PTA9CNA

Conexão:
DESKTOP-PTA9CNA

Exibir propriedades

Progresso

Pronto

Propriedades da Conexão

Propriedades da conexão atual:

A

Z

Ambiente de Servidor	
Nome do Computador	DESKTOP-PTA9CNA
Plataforma Host	Windows
Distribuição Host	Windows 10 Pro
Versão Host	10.0
Nível do Service Pack de Host	
Processadores	12
Memória do Sistema Operacional	7971
Autenticação	
Método de Autenticação	Autenticação do Windows
Nome de Usuário	loja
Conexão	
Banco de Dados	master
SPID	68
Protocolo de Rede	<padrão>
Tamanho do Pacote de Rede	4096
Tempo Limite da Conexão	30
Tempo Limite de Execução	0
Criptografado	Sim
Produto	
Nome do Produto	Microsoft SQL Server Enterprise Evaluation Edition (64-
Versão do Produto	16.0.1125 RTM
Nome do Servidor	DESKTOP-PTA9CNA
Nome da Instância	
Idioma	Português (Brasil)
Ordenação	Latin1_General_CI_AS

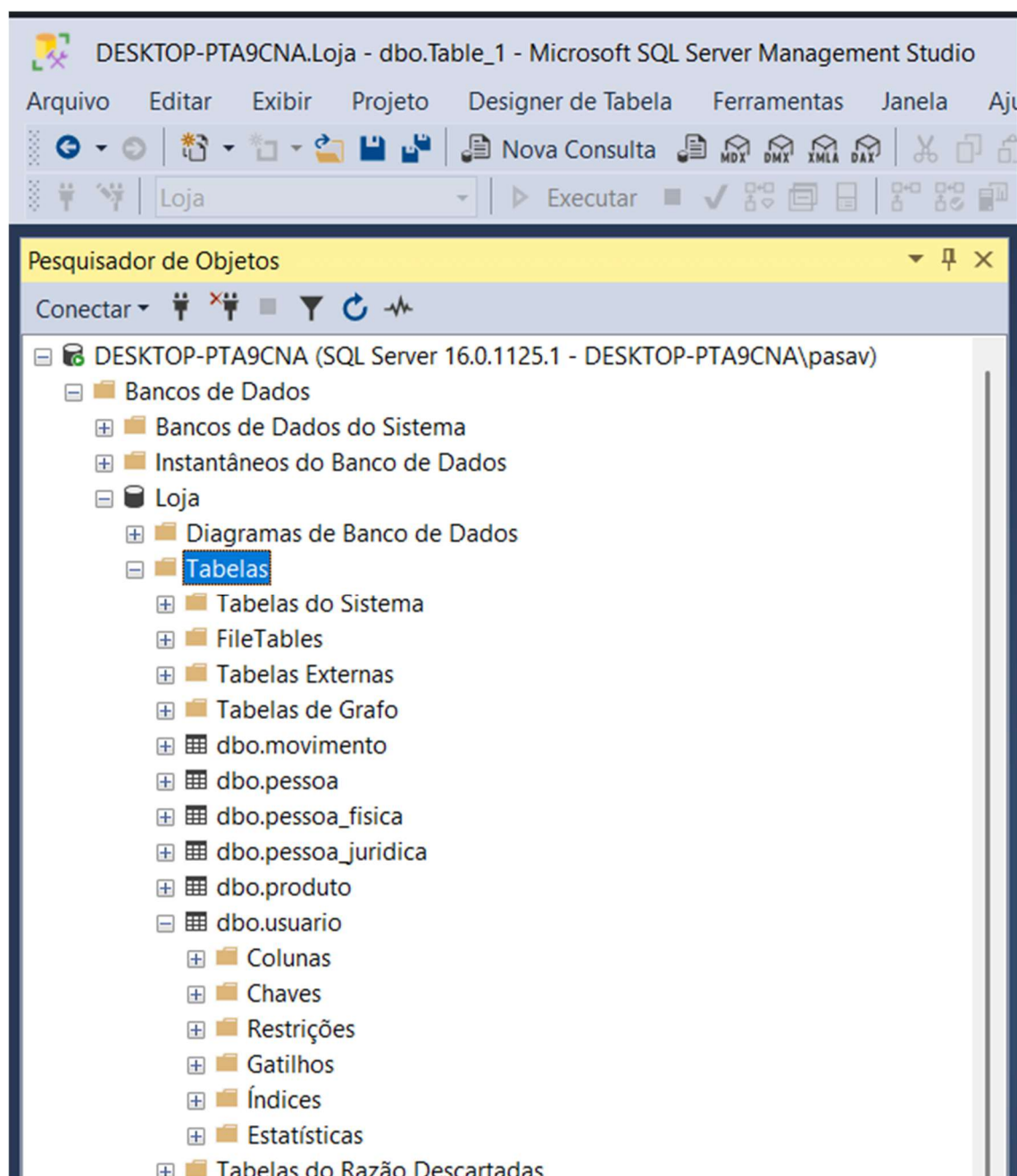
Nome de Usuário

O nome de usuário ou logon usado para conexão.

Fechar

Ajuda

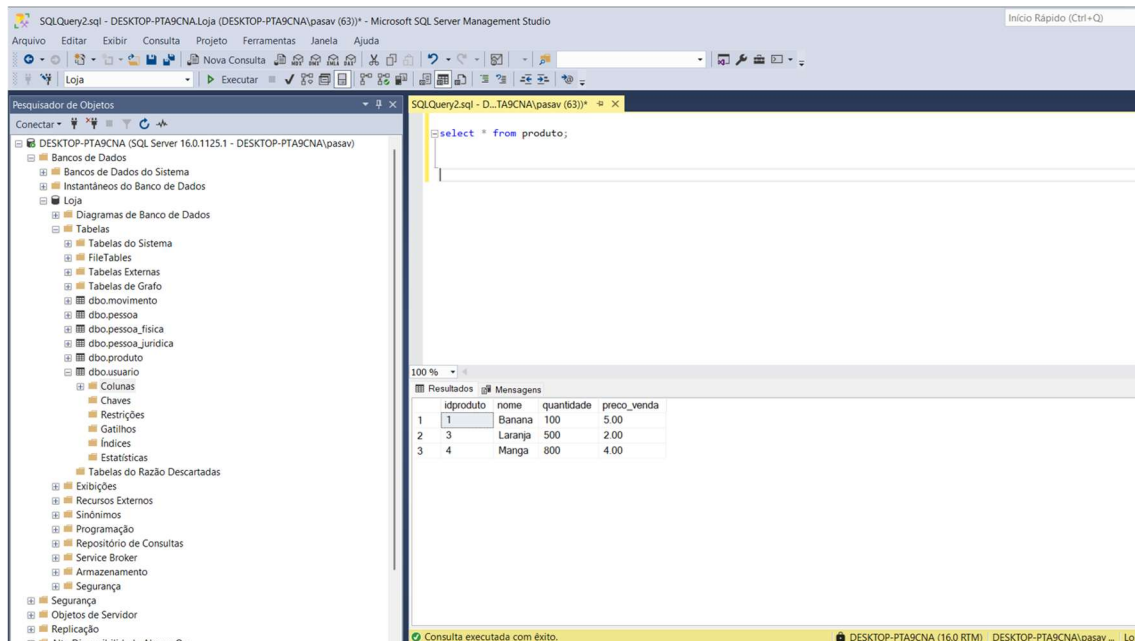
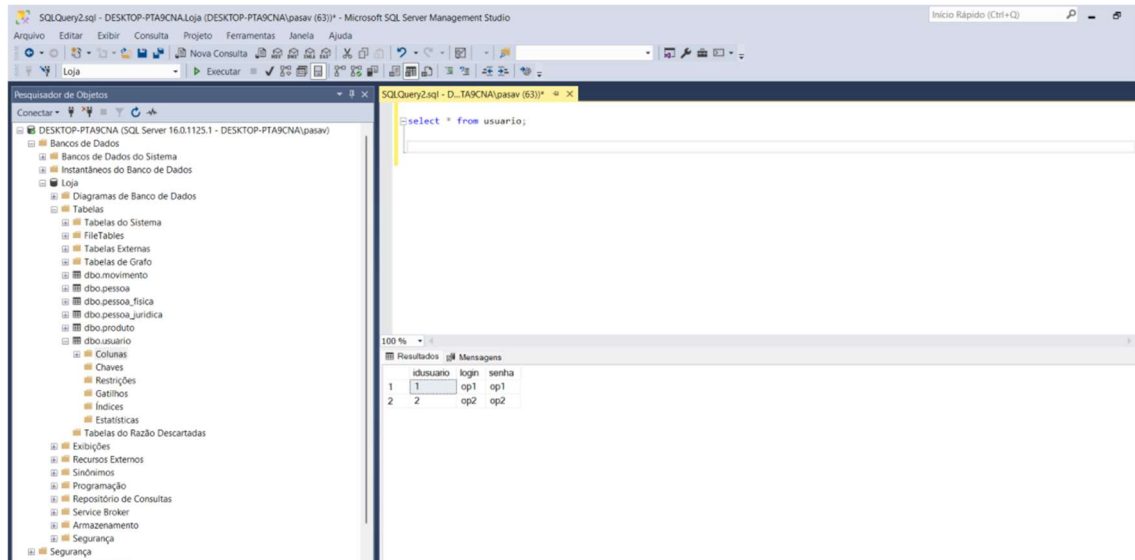
OK



RESULTADO DAS EXECUÇÕES

PROCEDIMENTO 02: ALIMENTANDO A BASE

BANCOS E TABELAS:



SQLQuery2.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA)pasav (63)* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar - [+] [v] [f] [r] [t] [u] [d] [e] [s] [c] [o] [n] [e] [c] [t] [a] [r]

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA)pasav

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razo Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

Alto Desempenho

SQLQuery2.sql - D...TA9CNA)pasav (63)*

```
-- Dados completos de pessoas físicas.
select *
from pessoa, pessoa_fisica
where pessoa.idpessoa = pessoa_fisica.idpessoa_fisica;

--Dados completos de pessoas jurídicas.
select *
from pessoa, pessoa_juridica
where pessoa.idpessoa = pessoa_juridica.idpessoa_juridica;
```

100 %

Resultados Mensagens

idpessoa	nome	logradouro	cidade	estado	telefone	email	idpessoa_fisica	cpf
1	Joao	Rua 12, cas 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	1	11111111111

idpessoa	nome	logradouro	cidade	estado	telefone	email	idpessoa_juridica	cnpj
2	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjo@riacho.com	2	2222222222222

Consulta executada com êxito. DESKTOP-PTA9CNA (16.0 RTM) DESKTOP-PTA9CNA)pasav -- Loja 00:00:00 2 linhas

SQLQuery2.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA)pasav (63)* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar - [+] [v] [f] [r] [t] [u] [d] [e] [s] [c] [o] [n] [e] [c] [t] [a] [r]

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA)pasav

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razo Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

SQLQuery2.sql - D...TA9CNA)pasav (63)*

```
--Valor médio de venda por produto, utilizando média ponderada.
select produto.nome, SUM (movimento.quantidade * movimento.valorUnitario) / SUM(movimento.quantidade) as 'V'
from movimento
JOIN produto
on produto.idproduto = movimento.produto_idproduto
where movimento.tipo = 'S'
group by produto.nome;

select * from movimento;
```

100 %

Resultados Mensagens

idmovimento	Usuario_idUsuario	pessoa_idpessoa	produto_idproduto	quantidade	tipo	valorUnitario
1	1	1	1	20	S	4.00
2	4	1	3	15	S	2.00
3	5	2	3	10	S	3.00
4	7	1	2	15	E	5.00
5	8	1	4	20	E	4.00

SQLQuery2.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA) (pasav (63)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja Nova Consulta Executar

Resquisitor de Objetos

Conectar

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA) (pasav)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Gráfico

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razão Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

Alts.Dimensions.Altasur.On

SQLQuery2.sql - D..TAS9CNA (pasav (63))

```
-- Dados completos de pessoas físicas.
select *
from pessoa, pessoa_fisica
where pessoa.idpessoa = pessoa_fisica.idpessoa_fisica;

--Dados completos de pessoas jurídicas.
select *
from pessoa, pessoa_juridica
where pessoa.idpessoa = pessoa_juridica.idpessoa_juridica;
```

100 %

Resultados Mensagens

idpessoa	nome	logradouro	cidade	estado	telefone	email	idpessoa_fisica	cpf
1	Joao	Rua 12, cas 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	1	11111111111

idpessoa	nome	logradouro	cidade	estado	telefone	email	idpessoa_juridica	cnpj
2	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjc@riacho.com	2	22222222222222

Consulta executada com êxito. DESKTOP-PTA9CNA (16.0 RTM) DESKTOP-PTA9CNA (pasav ... Loja 00:00:00 2 linhas

SQLQuery6.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA) (pasav (72)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja Nova Consulta Executar

Resquisitor de Objetos

Conectar

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA) (pasav)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Gráfico

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razão Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

Alts.Dimensions.Altasur.On

SQLQuery6.sql - D..TAS9CNA (pasav (72))

```
--Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e valor total.
select idmovimento, produto_idproduto, produto.nome as 'Produto', pessoa_idpessoa, pessoa.nome as 'Fornecedor', movimento.quantidade
(movimento.quantidade * valorUnitario) as valor_total
from movimento
join pessoa
on movimento.pessoa_idpessoa = pessoa.idpessoa
join produto
on movimento.produto_idproduto = produto.idproduto
where movimento.tipo = 'E';

--Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total
select idmovimento, produto_idproduto, produto.nome as 'Produto', pessoa_idpessoa, pessoa.nome as 'Comprador', movimento.quantidade
(movimento.quantidade * valorUnitario) as valor_total
from movimento
join pessoa
on movimento.pessoa_idpessoa = pessoa.idpessoa
join produto
```

100 %

Mensagens

idmovimento	produto_idproduto	Produto	pessoa_idpessoa	Fornecedor	quantidade	valorUnitario	valor_total
7	3	Laranja	2	JJC	15	5.00	75.00
8	4	Manga	2	JJC	20	4.00	80.00

idmovimento	produto_idproduto	Produto	pessoa_idpessoa	Comprador	quantidade	valorUnitario	valor_total
1	1	Banana	1	Joao	20	4.00	80.00
2	4	Laranja	1	Joao	15	2.00	30.00
3	5	Laranja	1	Joao	10	3.00	30.00

SQLQuery6.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA) (pasav (72)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razoão Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

SQLQuery6.sql - D.:TA9CNA) (pasav (72))

```
--Valor total das entradas agrupadas por produto.
select produto.nome, sum (movimento.quantidade * movimento.valorUnitario) AS 'VALOR TOTAL ENTRADAS'
from movimento
JOIN produto
on produto.idproduto = movimento.produto_idproduto
where movimento.tipo = 'E'
group by produto.nome;

--Valor total das saídas agrupadas por produto.
select produto.nome, sum (movimento.quantidade * movimento.valorUnitario) AS 'VALOR TOTAL SAIDAS'
from movimento
JOIN produto
on produto.idproduto = movimento.produto_idproduto
where movimento.tipo = 'S'
group by produto.nome;
```

100 %

Resultados Mensagens

	nome	VALOR TOTAL ENTRADAS
1	Laranja	75.00
2	Manga	60.00

	nome	VALOR TOTAL SAIDAS
1	Banana	60.00
2	Laranja	60.00

Consulta executada com êxito.

DESKTOP-PTA9CNA (16.0 RTM) | DESKTOP-PTA9CNA) (pasav ... | Loja | 00:00:00 | 4 linhas

SQLQuery6.sql - DESKTOP-PTA9CNA Loja (DESKTOP-PTA9CNA) (pasav (72)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Resquisitor de Objetos

Conectar

DESKTOP-PTA9CNA (SQL Server 16.0.1125.1 - DESKTOP-PTA9CNA)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

Loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.movimento

dbo.pessoa

dbo.pessoa_fisica

dbo.pessoa_juridica

dbo.produto

dbo.usuario

Colunas

Chaves

Restrições

Gatilhos

Índices

Estatísticas

Tabelas do Razoão Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Objetos de Servidor

Replicação

SQLQuery6.sql - D.:TA9CNA) (pasav (72))

```
--Operadores que não efetuaram movimentações de entrada (compra).
select movimento.Usuario_idUsuario AS 'ID DO OPERADOR'
from movimento
except
select movimento.Usuario_idUsuario
from movimento
where movimento.tipo = 'E';

--Valor total de entrada, agrupado por operador.
select usuario.login AS OPERADOR, sum (movimento.quantidade * movimento.valorUnitario) AS 'VALOR TOTAL ENTRADAS'
from movimento
on usuario.idusuario = movimento.Usuario_idUsuario
where movimento.tipo = 'E'
group by usuario.login;

--Valor total de saída, agrupado por operador.
select usuario.login AS OPERADOR, sum (movimento.quantidade * movimento.valorUnitario) AS 'VALOR TOTAL SAIDAS'
from movimento
on usuario.idusuario = movimento.Usuario_idUsuario
```

100 %

Resultados Mensagens

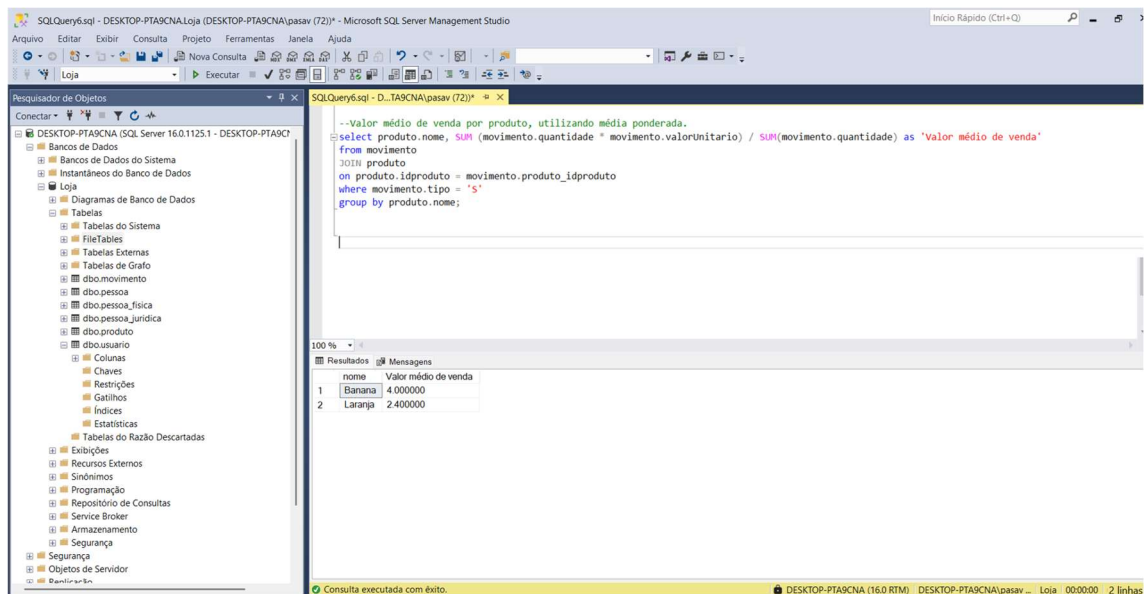
	ID DO OPERADOR
1	2

	OPERADOR	VALOR TOTAL ENTRADAS
1	op1	155.00

	OPERADOR	VALOR TOTAL SAIDAS
1	op1	110.00
2	op2	30.00

Consulta executada com êxito.

DESKTOP-PTA9CNA (16.0 RTM) | DESKTOP-PTA9CNA) (pasav ... | Loja | 00:00:00 | 4 linhas



ANÁLISE E CONCLUSÃO

PROCEDIMENTO 1:

1. Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

São implementadas por meio de relacionamentos de tabelas, usando chaves estrangeiras e chaves primárias.

2. Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

1x1

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

Interface gráfica amigável facilitando navegação e execução de tarefas, criação de script para automatizar tarefas, ferramentas avançadas de desempenho e análise de dados, gerenciamento de múltiplos servidores, gerenciamento de permissões.

PROCEDIMENTO 2:

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

Sequence: É um objeto separado que pode ser usado por várias tabelas, permite reiniciar, alterar incremento e outros ajustes sem afetar a tabela, pode ser reutilizada em diferentes contextos.

Identify: é específica de uma tabela, definida diretamente na coluna, fácil de implementar, geralmente é usado para chaves primárias e não podem ser compartilhadas entre tabelas.

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

Possibilita a implementação da integridade de dados diretamente no banco de dados, conhecida como integridade referencial. Uma chave estrangeira é a representação de um relacionamento entre tabelas.

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

Álgebra Relacional: Restrição, Projeção, Seleção, União, Diferença de Conjuntos, interseção, Junção, Produto Cartesiano, Divisão, Atribuição e Renomeação.

Cálculo Relacional: Maior, Menor, Maior ou Igual, Igual, Diferente, menor ou igual.

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

Utilizando o Grupo "By"

