



Universidade Estácio de Sá

Polo Bangu - Rio de Janeiro

-
- **Curso:** Desenvolvimento Full-Stack
 - **Disciplina:** Vamos Manter as Informações?
 - **Semestre:** 3
 - **Turma:** 2024.2
 - **Aluna:** Clara Martins Azevedo
-

1. Missão Prática - Nível 2

2. Objetivos da Missão 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.

3. Códigos do Projeto:

```
CREATE TABLE Produto (  
    idProduto INTEGER PRIMARY KEY,  
    nome VARCHAR(255),  
    quantidade INTEGER,  
    precoVenda NUMERIC  
);
```

```
CREATE TABLE Pessoa (  
    idPessoa INTEGER PRIMARY KEY,  
    nome VARCHAR(255),  
    locadouro VARCHAR(255),  
    cidade VARCHAR(255),  
    estado CHAR(2),  
    telefone VARCHAR(11),  
    email VARCHAR(255)  
);
```

```
CREATE TABLE Usuario (  
    idUsuario INTEGER PRIMARY KEY,  
    loginUsuario VARCHAR(20),  
    senha VARCHAR(20)  
);
```

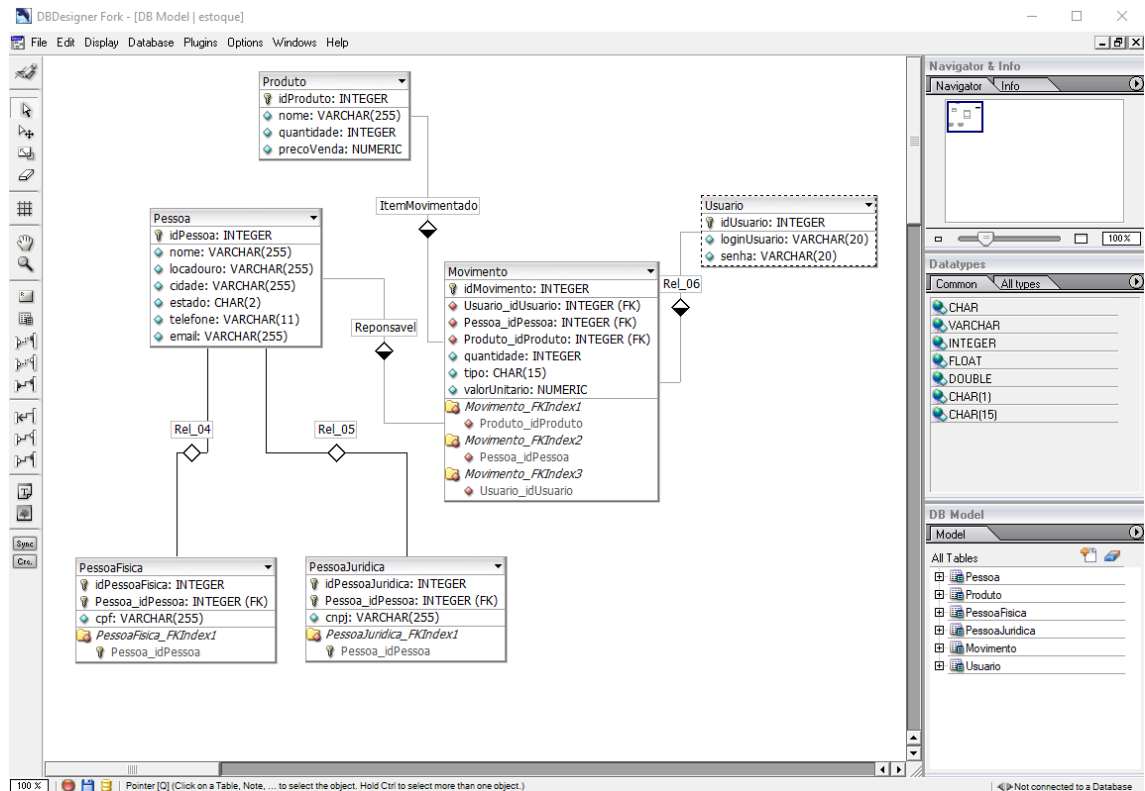
```
CREATE TABLE PessoaJuridica (  
    idPessoaJuridica INTEGER PRIMARY KEY,  
    cnpj VARCHAR(255),  
    FOREIGN KEY (idPessoaJuridica) REFERENCES Pessoa(idPessoa)  
);
```

```
CREATE TABLE PessoaFisica (  
    idPessoaFisica INTEGER PRIMARY KEY,  
    cpf VARCHAR(255),  
    FOREIGN KEY (idPessoaFisica) REFERENCES Pessoa(idPessoa)  
);
```

```
CREATE TABLE Movimento (  
    idMovimento INTEGER PRIMARY KEY NOT NULL,  
    Usuario_idUsuario INTEGER NOT NULL,  
    Pessoa_idPessoa INTEGER NOT NULL,  
    Produto_idProduto INTEGER NOT NULL,  
    quantidade INTEGER,  
    tipo CHAR(15),  
    valorUnitario NUMERIC,  
    FOREIGN KEY(Usuario_idUsuario) REFERENCES Usuario(idUsuario),  
    FOREIGN KEY(Pessoa_idPessoa) REFERENCES Pessoa(idPessoa),  
    FOREIGN KEY(Produto_idProduto) REFERENCES Produto(idProduto));
```

```
CREATE SEQUENCE PessoaIdSequence
START WITH 1
INCREMENT BY 1;
```

4. Resultado da execução do código:



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays the following SQL query:

```
idMovimento INTEGER PRIMARY KEY NOT NULL,
Usuario_idUsuario INTEGER NOT NULL,
Pessoa_idPessoa INTEGER NOT NULL,
Produto_idProduto INTEGER NOT NULL,
quantidade INTEGER,
tipo CHAR(15),
valorUnitario NUMERIC,
FOREIGN KEY(Usuario_idUsuario) REFERENCES Usuario(idUsuario),
FOREIGN KEY(Pessoa_idPessoa) REFERENCES Pessoa(idPessoa),
FOREIGN KEY(Produto_idProduto) REFERENCES Produto(idProduto));

CREATE SEQUENCE PessoaIdSequence
START WITH 1
INCREMENT BY 1;
```

The 'Mensagens' (Messages) pane at the bottom shows the execution results:

```
Comandos concluídos com êxito.

Horário de conclusão: 2024-07-30T15:15:10.1860999-03:00
```

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

5. Análise e Conclusão:

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

1X1: Quando um item de uma tabela está associado a um item de outra tabela. Para que isso aconteça, a chave estrangeira de uma das tabelas deve referenciar a chave primária da outra tabela.

1XN: Quando um item da tabela pode se associar a vários itens de outra tabela. Para que isso aconteça, semelhante a como ocorre na relação 1X1, a chave estrangeira de uma das tabelas também deve referenciar a chave primária da outra tabela, com a diferença de que a tabela com a chave estrangeira também possui uma chave primária.

NxN: Quando vários itens de uma tabela podem se associar a vários itens de outra tabela. Para que isso aconteça uma terceira tabela deve ser criada para armazenar os dados as duas tabelas que estavam relacionadas anteriormente.

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

Existem algumas estratégias que podem ser adotadas para o uso de herança em banco de dados, tudo depende da situação. Uma dessas estratégias, por exemplo, é a “tabela filha” com a chave estrangeira para a tabela principal.

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

O SQL Server Management Studio é uma ferramenta que possui inúmeras funcionalidades pensadas para facilitar o gerenciamento de banco de dados, consequentemente melhorando de forma significativa a produtividade. Dentre as principais funcionalidades estão editor de consultas, monitoramento de desempenho, designer de tabelas, interface intuitiva, entre outros.

1. Missão Prática - Nível 2

2. Objetivos da Missão 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.

3. Códigos do Projeto:

```
INSERT INTO Usuario (idUserio, loginUsuario, senha) VALUES (1, 'op1', 'op1');  
INSERT INTO Usuario (idUserio, loginUsuario, senha) VALUES (2, 'op2', 'op2');
```

```
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (1, 'Banana', 100, 5.00);  
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (2, 'Laranja', 500, 2.00);  
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (3, 'Manga', 800, 4.00);
```

```
INSERT INTO Pessoa (idPessoa, nome, locadouro, cidade, estado, telefone, email) VALUES (7, 'Joao', 'Rua 12,
casa 3, Quitanda', 'Riacho do Sul', 'PA', '1111-1111', 'email@example.com');
INSERT INTO PessoaFisica (idPessoaFisica, cpf) VALUES (7, '1111111111');
```

```
INSERT INTO Pessoa (idPessoa, nome, locadouro, cidade, estado, telefone, email) VALUES (15, 'JJC', 'Rua 11,
Centro', 'Riacho do Norte', 'PA', '2222-2222', 'email@example.com');
INSERT INTO PessoaJuridica (idPessoaJuridica, cnpj) VALUES (15, '222222222222');
```

```
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade, tipo,
valorUnitario) VALUES (1, 1, 7, 1, 20, 'S', 4.00);
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade, tipo,
valorUnitario) VALUES (4, 1, 7, 3, 15, 'S', 2.00);
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade, tipo,
valorUnitario) VALUES (5, 2, 7, 3, 10, 'S', 3.00);
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade, tipo,
valorUnitario) VALUES (7, 1, 15, 3, 15, 'E', 5.00);
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade, tipo,
valorUnitario) VALUES (8, 1, 15, 4, 20, 'E', 4.00);
```

```
-----

SELECT p.*, pf.cpf
FROM Pessoa p
INNER JOIN PessoaFisica pf ON p.idPessoa = pf.idPessoaFisica;
```

```
SELECT p.*, pj.cnpj
FROM Pessoa p
INNER JOIN PessoaJuridica pj ON p.idPessoa = pj.idPessoaJuridica;
```

```
SELECT m.*, p.nome as fornecedor, pr.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade *
m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.Pessoa_idPessoa
INNER JOIN Produto pr ON pr.idProduto = m.Produto_idProduto
WHERE m.tipo = 'E';
```

```
SELECT m.*, p.nome as comprador, pr.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade *
m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.Pessoa_idPessoa = p.idPessoa
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S';
```

```
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as compras
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'E'
GROUP BY pr.nome;
```

```
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as vendas
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;
```

```
SELECT u.*
```

```

FROM Usuario u
LEFT JOIN Movimento m ON u.idUsuario = m.Usuario_idUsuario AND m.tipo = 'E'
WHERE m.idMovimento IS NULL;

```

```

SELECT u.loginUsuario, SUM(m.valorUnitario * m.quantidade) as compras
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'E'
GROUP BY u.loginUsuario;

```

```

SELECT u.loginUsuario, SUM(m.valorUnitario * m.quantidade) as vendas
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'S'
GROUP BY u.loginUsuario;

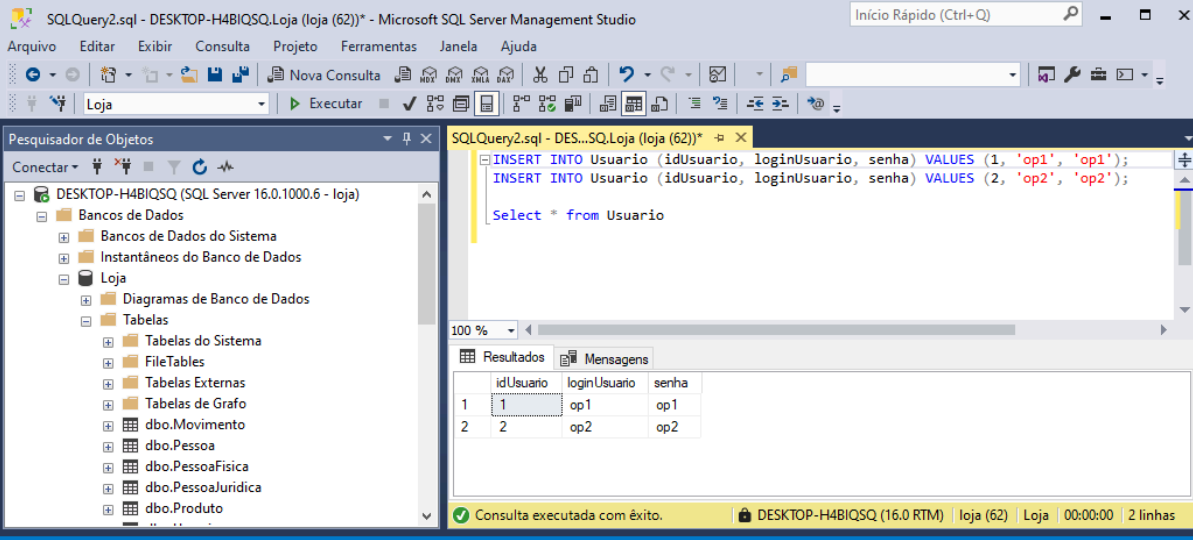
```

```

SELECT pr.nome, SUM(m.valorUnitario * m.quantidade) / SUM(m.quantidade) as media
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;

```

4. Resultado da execução do código:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following SQL code:

```

INSERT INTO Usuario (idUsuario, loginUsuario, senha) VALUES (1, 'op1', 'op1');
INSERT INTO Usuario (idUsuario, loginUsuario, senha) VALUES (2, 'op2', 'op2');

Select * from Usuario

```

The Results pane displays the following data:

	idUsuario	loginUsuario	senha
1	1	op1	op1
2	2	op2	op2

The status bar at the bottom indicates: "Consulta executada com êxito. DESKTOP-H4BIQSQ (16.0 RTM) loja (62) Loja 00:00:00 2 linhas".

SQLQuery3.sql - DESKTOP-H4BIQSQ.Loja (loja (63))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Executar

Pesquisador de Objetos

Conectar

Tabelas

- Tabelas do Sistema
- FileTables
- Tabelas Externas
- Tabelas de Grafo
- dbo.Movimento
- dbo.Pessoa
- dbo.PessoaFisica
- dbo.PessoaJuridica
- dbo.Produto
- Colunas
 - idProduto (PK, int, não nulo)
 - nome (varchar(255), nulo)
 - quantidade (int, nulo)
 - precoVenda (numeric(18,0), nulo)
- Chaves

SQLQuery3.sql - DES...SQ.Loja (loja (63))*

```
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (1, 'Banana', 100, 5);
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (2, 'Laranja', 500, 2);
INSERT INTO Produto (idProduto, nome, quantidade, precoVenda) VALUES (3, 'Manga', 800, 4);
SELECT * FROM Produto
```

SQLQuery2.sql - DES...SQ.Loja (loja (62))*

100 %

Resultados Mensagens

	idProduto	nome	quantidade	precoVenda
1	1	Banana	100	5
2	2	Laranja	500	2
3	3	Manga	800	4

Consulta executada com êxito. DESKTOP-H4BIQSQ (16.0 RTM) loja (63) Loja 00:00:00 3 linhas

Pronto Li 5 Col 22 Car 22 INS

SQLQuery3.sql - DESKTOP-H4BIQSQ.Loja (loja (63))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Executar

Pesquisador de Objetos

Conectar

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

dbo.Pessoa

dbo.PessoaFisica

Colunas

- idPessoaFisica (PK, FK, int, não nulo)
- cpf (varchar(255), nulo)

Chaves

Restrições

Gatilhos

Índices

Estadísticas

dbo.PessoaJuridica

dbo.Produto

dbo.Usuario

SQLQuery3.sql - DES...SQ.Loja (loja (63))*

```
INSERT INTO Pessoa (idPessoa, nome, locadouro, cidade, estado, telefone, email) VALUES (7, 'Joao', 'Rua 12, casa 3, Quitanda', 'Riachão do Sul', 'PA', '1111-1111', 'email@example.com');
INSERT INTO PessoaFisica (idPessoaFisica, cpf) VALUES (7, '11111111111');
SELECT * FROM Pessoa INNER JOIN PessoaFisica ON Pessoa.idPessoa = PessoaFisica.idPessoaFisica;
```

SQLQuery2.sql - DES...SQ.Loja (loja (62))*

100 %

Resultados Mensagens

	idPessoa	nome	locadouro	cidade	estado	telefone	email	idPessoaFisica	cpf
1	7	Joao	Rua 12, casa 3, Quitanda	Riachão do Sul	PA	1111-1111	email@example.com	7	11111111111

Consulta executada com êxito. DESKTOP-H4BIQSQ (16.0 RTM) loja (63) Loja 00:00:00 1 linhas

Pronto Li 2 Col 74 Car 74 INS

SQLQuery3.sql - DESKTOP-H4BIQSQ.Loja (loja (63))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Executar

Pesquisador de Objetos

Conectar

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

dbo.Pessoa

dbo.PessoaFisica

Colunas

- idPessoaFisica (PK, FK, int, não nulo)
- cpf (varchar(255), nulo)

Chaves

Restrições

Gatilhos

Índices

Estadísticas

dbo.PessoaJuridica

dbo.Produto

dbo.Usuario

SQLQuery3.sql - DES...SQ.Loja (loja (63))*

```
INSERT INTO Pessoa (idPessoa, nome, locadouro, cidade, estado, telefone, email) VALUES (15, 'JJC', 'Rua 11, Centro', 'Riachão do Norte', 'PA', '2222-2222', 'email@example.com');
INSERT INTO PessoaJuridica (idPessoaJuridica, cnpj) VALUES (15, '22222222222222');
SELECT * FROM Pessoa INNER JOIN PessoaJuridica ON Pessoa.idPessoa = PessoaJuridica.idPessoaJuridica
```

SQLQuery2.sql - DES...SQ.Loja (loja (62))*

100 %

Resultados Mensagens

	idPessoa	nome	locadouro	cidade	estado	telefone	email	idPessoaJuridica	cnpj
1	15	JJC	Rua 11, Centro	Riachão do Norte	PA	2222-2222	email@example.com	15	22222222222222

Consulta executada com êxito. DESKTOP-H4BIQSQ (16.0 RTM) loja (63) Loja 00:00:00 1 linhas

Pronto Li 2 Col 83 Car 83 INS

SQLQuery3.sql - DESKTOP-H4BIQSQ.Loja (loja (63))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Pesquisador de Objetos

Conectar

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

Colunas

- idMovimento (PK, int, não nulo)
- Usuario_idUsuario (FK, int, não nulo)
- Pessoa_idPessoa (FK, int, não nulo)
- Produto_idProduto (FK, int, não nulo)
- quantidade (int, nulo)
- tipo (char(15), nulo)
- valorUnitario (numeric(18,0), nulo)

Chaves

Restrições

Gatilhos

Índices

Estatísticas

SQLQuery3.sql - DESKTOP-H4BIQSQ.Loja (loja (63))*

```

INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade)
VALUES (1, 1, 7, 1, 20)
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade)
VALUES (2, 4, 1, 7, 3)
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade)
VALUES (3, 5, 2, 7, 3)
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade)
VALUES (4, 6, 1, 15, 3)
INSERT INTO Movimento (idMovimento, Usuario_idUsuario, Pessoa_idPessoa, Produto_idProduto, quantidade)
VALUES (5, 7, 1, 15, 3)
Select * from Movimento

```

100 %

Resultados Mensagens

idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produto_idProduto	quantidade	tipo	valorUnitario
1	1	7	1	20	S	4
2	4	1	7	3	S	2
3	5	2	7	3	S	3
4	6	1	15	3	E	5
5	7	1	15	3	E	5

Consulta executada com êxito.

DESKTOP-H4BIQSQ (16.0 RTM) | loja (63) | Loja | 00:00:00 | 5 linhas

Pronto Lin. 1 Col. 1 INS

4.a:

SQLQuery4.sql - DESKTOP-H4BIQSQ.Loja (loja (72))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Pesquisador de Objetos

Conectar

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

Colunas

- idMovimento (PK, int, não nulo)
- Usuario_idUsuario (FK, int, não nulo)
- Pessoa_idPessoa (FK, int, não nulo)
- Produto_idProduto (FK, int, não nulo)
- quantidade (int, nulo)
- tipo (char(15), nulo)
- valorUnitario (numeric(18,0), nulo)

Chaves

Restrições

Gatilhos

Índices

Estatísticas

SQLQuery4.sql - DESKTOP-H4BIQSQ.Loja (loja (72))*

```

SELECT p.*, pf.cpf
FROM Pessoa p
INNER JOIN PessoaFisica pf ON p.idPessoa = pf.idPessoaFisica;

```

100 %

Resultados Mensagens

idPessoa	nome	localidade	cidade	estado	telefone	email	cpf
7	Joao	Rua 12, casa 3, Quitanda	Riachão do Sul	PA	1111-1111	email@example.com	111111111111

Consulta executada com êxito.

DESKTOP-H4BIQSQ (16.0 RTM) | loja (72) | Loja | 00:00:00 | 1 linhas

Pronto Li 1 Col. 1 Car. 1 INS

4.b:

SQLQuery4.sql - DESKTOP-H4BIQSQ.Loja (loja (72))* - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

Loja

Pesquisador de Objetos

Conectar

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

Colunas

- idMovimento (PK, int, não nulo)
- Usuario_idUsuario (FK, int, não nulo)
- Pessoa_idPessoa (FK, int, não nulo)
- Produto_idProduto (FK, int, não nulo)
- quantidade (int, nulo)
- tipo (char(15), nulo)
- valorUnitario (numeric(18,0), nulo)

Chaves

Restrições

Gatilhos

Índices

Estatísticas

SQLQuery4.sql - DESKTOP-H4BIQSQ.Loja (loja (72))*

```

SELECT p.*, pj.cnpj
FROM Pessoa p
INNER JOIN PessoaJuridica pj ON p.idPessoa = pj.idPessoaJuridica;

```

100 %

Resultados Mensagens

idPessoa	nome	localidade	cidade	estado	telefone	email	cnpj
15	JJC	Rua 11, Centro	Riachão do Norte	PA	2222-2222	email@example.com	22222222222222

Consulta executada com êxito.

DESKTOP-H4BIQSQ (16.0 RTM) | loja (72) | Loja | 00:00:00 | 1 linhas

Pronto Li 3 Col. 66 Car. 66 INS

4.c:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Pesquisador de Objetos' (Object Explorer) with the database structure. The central pane shows the SQL query editor with the following query:

```
SELECT m.*, p.nome as fornecedor, pr.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.Pessoa_idPessoa
INNER JOIN Produto pr ON pr.idProduto = m.Produto_idProduto
WHERE m.tipo = 'E';
```

The right pane shows the 'Resultados' (Results) tab with the following data:

	idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produto_idProduto	quantidade	tipo	valorUnitario	fornecedor	Produto	quantidade	valorUnitario	total
1	6	1	15	3	15	E	5	JJC	Manga	15	5	75
2	7	1	15	3	15	E	5	JJC	Manga	15	5	75

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully.)

4.d:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Pesquisador de Objetos' (Object Explorer) with the database structure. The central pane shows the SQL query editor with the following query:

```
SELECT m.*, p.nome as comprador, pr.nome as Produto, m.quantidade, m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.Pessoa_idPessoa = p.idPessoa
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S';
```

The right pane shows the 'Resultados' (Results) tab with the following data:

	idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produto_idProduto	quantidade	tipo	valorUnitario	comprador	Produto	quantidade	valorUnitario	total
1	1	1	7	1	20	S	4	Joao	Banana	20	4	80
2	4	1	7	3	15	S	2	Joao	Manga	15	2	30
3	5	2	7	3	10	S	3	Joao	Manga	10	3	30

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully.)

4.e:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Pesquisador de Objetos' (Object Explorer) with the database structure. The central pane shows the SQL query editor with the following query:

```
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as compras
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'E'
GROUP BY pr.nome;
```

The right pane shows the 'Resultados' (Results) tab with the following data:

	nome	compras
1	Manga	150

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully.)

4.f:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Pesquisador de Objetos' (Object Explorer) on the left displays the database structure, including tables like 'Movimento' and 'Produto'. The main query window contains the following SQL code:

```
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as vendas
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;
```

The 'Resultados' (Results) pane shows the output of the query:

	nome	vendas
1	Banana	80
2	Manga	60

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

4.g:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Pesquisador de Objetos' (Object Explorer) on the left displays the database structure. The main query window contains the following SQL code:

```
SELECT u.*
FROM Usuario u
LEFT JOIN Movimento m ON u.idUsuario = m.Usuario_idUsuario AND m.tipo = 'E'
WHERE m.idMovimento IS NULL;
```

The 'Resultados' (Results) pane shows the output of the query:

	idUsuario	loginUsuario	senha
1	2	op2	op2

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

4.h:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Pesquisador de Objetos' (Object Explorer) on the left displays the database structure. The main query window contains the following SQL code:

```
SELECT u.loginUsuario, SUM(m.valorUnitario * m.quantidade) as compras
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'E'
GROUP BY u.loginUsuario;
```

The 'Resultados' (Results) pane shows the output of the query:

	loginUsuario	compras
1	op1	150

The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

4.i:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Pesquisador de Objetos' (Object Explorer) on the left displays the database structure, including tables like 'Movimento' and 'Usuario'. The central pane shows a SQL query in 'SQLQuery4.sql':

```
SELECT u.loginUsuario, SUM(m.valorUnitario * m.quantidade) as vendas
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'S'
GROUP BY u.loginUsuario;
```

The 'Resultados' (Results) pane at the bottom shows the output of the query:

	loginUsuario	vendas
1	op1	110
2	op2	30

The status bar at the bottom indicates the query was executed successfully ('Consulta executada com êxito').

4.j:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Pesquisador de Objetos' (Object Explorer) on the left displays the database structure. The central pane shows a SQL query in 'SQLQuery5.sql':

```
SELECT pr.nome, SUM(m.valorUnitario * m.quantidade) / SUM(m.quantidade) as media
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;
```

The 'Resultados' (Results) pane at the bottom shows the output of the query:

	nome	media
1	Banana	4.000000
2	Manga	2.400000

The status bar at the bottom indicates the query was executed successfully ('Consulta executada com êxito').

5. Análise e Conclusão:

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

Apesar de ambos serem usados para a geração de numeração automática, identity é restrito a uma tabela, já sequence, não.

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

A chave estrangeira habilita o relacionamento entre tabelas, sem ela, a integridade referencial, que possibilita a implementação da integridade de dados diretamente no banco de dados, não aconteceria.

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

Álgebra relacional é uma forma de cálculo sobre conjuntos ou relações, muitas das características originais foram inspiradas nela. Dentre as operações estão:

Seleção (σ): Seleciona linhas específicas que satisfazem certo predicado.

Projeção (π): Elimina colunas não desejadas.

Produto Cartesiano(\times): Habilita a combinação de duas relações.

União (\cup): Combina os resultados de tabelas diferentes.

Diferença ($-$): Retorna as linhas diferentes entre as tabelas.

Junção (\bowtie): Combina as linhas das tabelas.

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

O agrupamento em consultas SQL é realizado através do comando “group by”, utilizado para combinar linhas com os mesmos valores em colunas específicas.

