



Estácio

Atividade: Relatório de Prática

Nome: Jamison Queiroz

Matrícula: 202208101127

Curso: Desenvolvimento Full Stack

Turma: 2022.4

Disciplina: Vamos Manter as Informações

Professor: Simone Ingrid Monteiro Gama

Sumário

1. INTRODUÇÃO	3
2. OBJETIVO	3
3. SOFTWARE UTILIZADO.....	3
4. PROCEDIMENTOS.....	3
4.1 Criando o Banco de Dados	3
4.1.1 Análise e Conclusão	5
4.2 Alimentando a Base	5
4.2.1 Análise e Conclusão	9
5. CONCLUSÃO	9

1. INTRODUÇÃO

Esta prática visa Modelar e implementar um banco de dados simples, utilizando como base o SQL Server.

2. OBJETIVO

- 2.1. Identificar os requisitos de um sistema e transformá-los no modelo adequado.
- 2.2. Utilizar ferramentas de modelagem para bases de dados relacionais.
- 2.3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
- 2.4. Explorar a sintaxe SQL na consulta e manipulação de dados (DML).
- 2.5. 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.

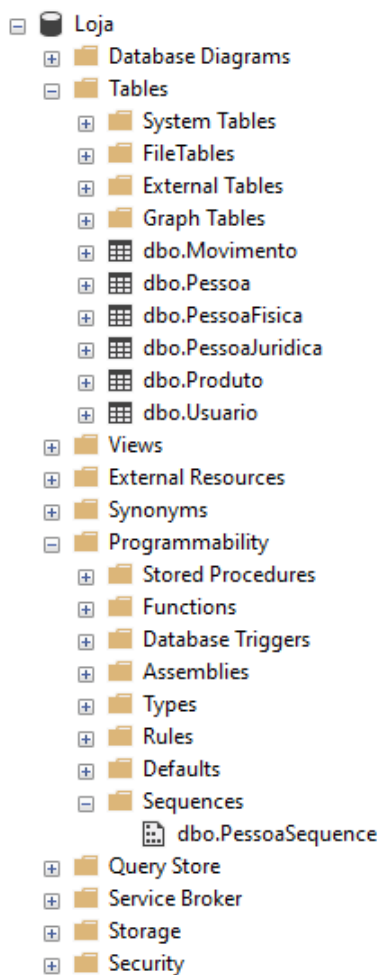
3. SOFTWARE UTILIZADO

- 3.1. SQL Server Management Studio 19
- 3.2. Microsoft SQL Server Express 2022
- 3.3. DBDesigner Fork






4. PROCEDIMENTOS

4.1. Criando o Banco de Dados

Nesta etapa foi criado a modelagem dos dados, no DBDesigner, e a sua implementação no Microsoft SQL Server Express 2022. Ressalto que o modelo pode ser visto nos arquivos **Modelagem_Dados.png** ou **Modelagem_Dados.xml** e o resultado dos procedimentos executados pode ser visualizado na imagem a seguir:



Os Scripts utilizados para gerar a implementação das tabelas, relacionamentos e outras funcionalidades no banco, estão disponibilizados dentro da pasta **ScriptsSQL**:

 CriandoRelacionamentos.sql	22/08/2023 10:15	Microsoft SQL Ser...	1 KB
 CriandoSequencia.sql	22/08/2023 10:30	Microsoft SQL Ser...	1 KB
 CriandoTabelas.sql	22/08/2023 10:05	Microsoft SQL Ser...	2 KB
 DBLoja.sql	22/08/2023 10:36	Microsoft SQL Ser...	15 KB
 Loja.bak	22/08/2023 10:35	BAK File	4.697 KB

4.1.1 Análise e Conclusão:

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

Utilizando relacionamentos entre as tabelas com Foreign Key (Chave Estrangeira) e Primary Key (Chave Primaria).

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

Relacionamento utilizando cardinalidade 1x1.

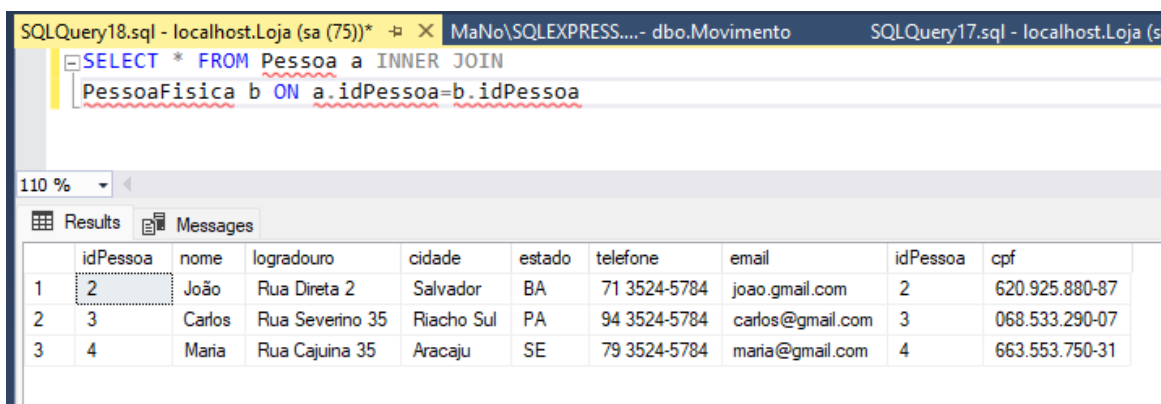
- c) Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?

Sendo um ambiente gráfico para gerenciamento de banco de dados ele fornece ferramentas que agilizam na construção de tabelas, bancos e outras funcionalidades de forma prática e rápida aumentando assim a produtividade das atividades relacionadas ao gerenciamento de banco de dados.

4.2. Alimentando a Base

Nesta etapa foi feito a alimentação do banco através do SQL Server Management Studio e efetuado os seguintes procedimentos:

- a) Dados completos de pessoas físicas.



The screenshot shows the SQL Server Enterprise Manager interface. The 'SQLQuery18.sql' window is active, displaying the following query:

```
SELECT * FROM Pessoa a INNER JOIN  
PessoaFisica b ON a.idPessoa=b.idPessoa
```

Below the query editor, the 'Results' tab is selected, showing the output of the query. The results are displayed in a table with 10 columns: idPessoa, nome, logradouro, cidade, estado, telefone, email, idPessoa, and cpf. The table contains 3 rows of data.

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoa	cpf
1	2	João	Rua Direta 2	Salvador	BA	71 3524-5784	joao@gmail.com	2	620.925.880-87
2	3	Carlos	Rua Severino 35	Riacho Sul	PA	94 3524-5784	carlos@gmail.com	3	068.533.290-07
3	4	Maria	Rua Cajuina 35	Aracaju	SE	79 3524-5784	maria@gmail.com	4	663.553.750-31

b) Dados completos de pessoas jurídicas.

SQLQuery18.sql - localhost.Loja (sa (75))* X MaNo\SQLEXPRESS....- dbo.Movimento SQLQuery17.sql - localhost.Loja (sa (62))*

```
SELECT * FROM Pessoa a INNER JOIN
PessoaJuridica b ON a.idPessoa=b.idPessoa
```

110 %

Results Messages

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoa	cnpj
1	7	Casa de Chá	Rua Cajuina 48	Aracaju	SE	79 3524-5784	ccha@gmail.com	7	75.124.487/0001-13
2	8	Mercadinho Barato	Rua Severino 56	Riacho Sul	PA	94 3524-5784	mbarato@gmail.com	8	63.418.364/0001-06
3	9	Atakadão	Rua Direta 23	Salvador	BA	71 3524-5784	atakadao@gmail.com	9	47.014.516/0001-59

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

SQLQuery18.sql - localhost.Loja (sa (75))* X MaNo\SQLEXPRESS....- dbo.Movimento SQLQuery17.sql - localhost.Loja (sa (62))*

```
SELECT b.nome AS Produto, c.nome AS Fornecedor, a.quantidade,
a.valorunitario, (a.quantidade*a.valorunitario) AS 'Valor Total' FROM Movimento a
INNER JOIN Produto b ON a.idProduto=b.idProduto
INNER JOIN Pessoa c ON a.idPessoa=c.idPessoa
WHERE a.tipo='E'
```

110 %

Results Messages

	Produto	Fornecedor	quantidade	valorunitario	Valor Total
1	Banana	Casa de Chá	15	5,00	75,00
2	Laranja	Casa de Chá	20	4,00	80,00
3	Manga	Atakadão	15	4,00	60,00

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

SQLQuery18.sql - localhost.Loja (sa (75))* X SQLQuery17.sql - localhost.Loja (sa (62))*

```
SELECT b.nome AS Produto, c.nome AS Comprador, a.quantidade,
b.precoventa, (a.quantidade*b.precoventa) AS 'Valor Total' FROM Movimento a
INNER JOIN Produto b ON a.idProduto=b.idProduto
INNER JOIN Pessoa c ON a.idPessoa=c.idPessoa
WHERE a.tipo='S'
```

110 %

Results Messages

	Produto	Comprador	quantidade	precoventa	Valor Total
1	Manga	Carlos	20	4,00	80,00
2	Banana	João	15	5,00	75,00
3	Banana	João	10	5,00	50,00

e) Valor total das entradas agrupadas por produto.

SQLQuery18.sql - localhost.Loja (sa (75))* SQLQuery17.sql - localhost.Loja (sa (62))*

```

SELECT b.nome AS Produto ,SUM(a.quantidade) AS 'Quantidade Total' FROM Movimento a
INNER JOIN Produto b ON a.idProduto=b.idProduto
INNER JOIN Pessoa c ON a.idPessoa=c.idPessoa
WHERE a.tipo='E'
GROUP BY b.nome
  
```

110 %

Results Messages

	Produto	Quantidade Total
1	Banana	15
2	Laranja	20
3	Manga	15

f) Valor total das saídas agrupadas por produto.

SQLQuery18.sql - localhost.Loja (sa (75))* SQLQuery17.sql - localhost.Loja (sa (62))*

```

SELECT b.nome AS Produto ,SUM(a.quantidade) AS 'Quantidade Total' FROM Movimento a
INNER JOIN Produto b ON a.idProduto=b.idProduto
INNER JOIN Pessoa c ON a.idPessoa=c.idPessoa
WHERE a.tipo='S'
GROUP BY b.nome
  
```

110 %

Results Messages

	Produto	Quantidade Total
1	Banana	25
2	Manga	20

g) Operadores que não efetuaram movimentações de entrada (compra).

SQLQuery18.sql - localhost.Loja (sa (75))* SQLQuery17.sql - localhost.Loja (sa (62))*

```

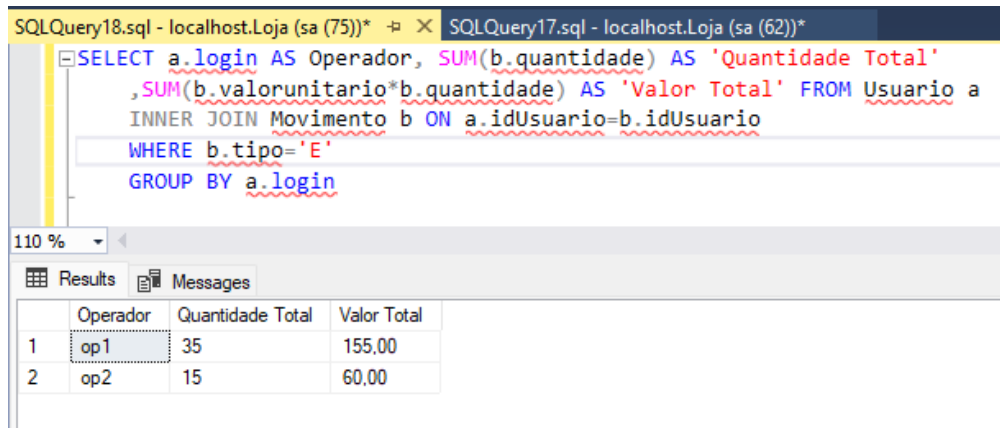
SELECT * FROM Usuario a
LEFT JOIN (SELECT * FROM Movimento WHERE tipo='E') b ON a.idUsuario=b.idUsuario
WHERE b.idMovimento IS NULL
  
```

110 %

Results Messages

	idUsuario	login	senha	idMovimento	idUsuario	idPessoa	idProduto	quantidade	tipo	valorUnitario
1	3	op3	op3	NULL	NULL	NULL	NULL	NULL	NULL	NULL

h) Valor total de entrada, agrupado por operador.



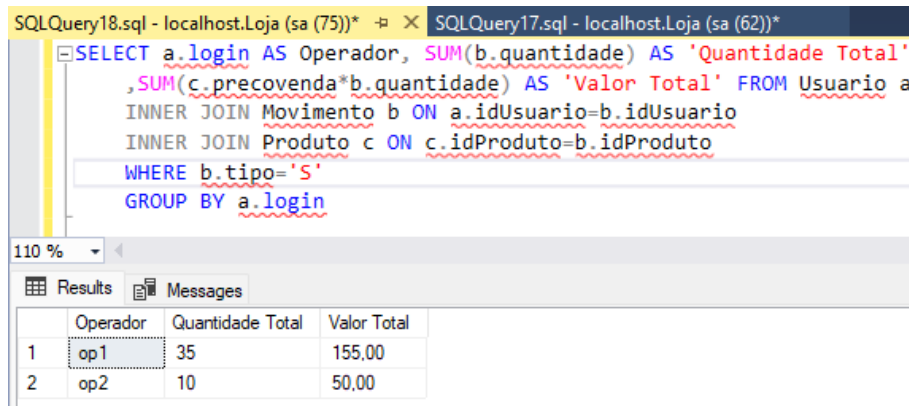
The screenshot shows a SQL query in SQLQuery18.sql. The query selects the login of the operator, the sum of quantities, and the sum of the product of unit price and quantity, grouped by operator login, for entry movements (tipo='E').

```
SELECT a.login AS Operador, SUM(b.quantidade) AS 'Quantidade Total',  
SUM(b.valorunitario*b.quantidade) AS 'Valor Total' FROM Usuario a  
INNER JOIN Movimento b ON a.idUsuario=b.idUsuario  
WHERE b.tipo='E'  
GROUP BY a.login
```

The results table shows two operators: op1 with a total quantity of 35 and a total value of 155.00, and op2 with a total quantity of 15 and a total value of 60.00.

	Operador	Quantidade Total	Valor Total
1	op1	35	155,00
2	op2	15	60,00

i) Valor total de saída, agrupado por operador.



The screenshot shows a SQL query in SQLQuery18.sql. The query selects the login of the operator, the sum of quantities, and the sum of the product of unit price and quantity, grouped by operator login, for sale movements (tipo='S').

```
SELECT a.login AS Operador, SUM(b.quantidade) AS 'Quantidade Total',  
SUM(c.precovenda*b.quantidade) AS 'Valor Total' FROM Usuario a  
INNER JOIN Movimento b ON a.idUsuario=b.idUsuario  
INNER JOIN Produto c ON c.idProduto=b.idProduto  
WHERE b.tipo='S'  
GROUP BY a.login
```

The results table shows two operators: op1 with a total quantity of 35 and a total value of 155.00, and op2 with a total quantity of 10 and a total value of 50.00.

	Operador	Quantidade Total	Valor Total
1	op1	35	155,00
2	op2	10	50,00

j) Valor médio de venda por produto, utilizando média ponderada.

Não foi possível efetuar o cálculo da média ponderada pois não foi especificado os pesos e um dos requisitos era “Os operadores (usuários) poderão efetuar movimentos de venda para um determinado produto, sempre para uma pessoa física, utilizando o preço de venda atualmente na base” o que seria um preço fixo de venda localizado na tabela Produto.

4.2.1 Análise e Conclusão:

- a) Quais as diferenças no uso de sequence e identity?

Ambos são usados para gerar incremento automaticamente sendo perfeitos para serem usados em Primary Key com autoincremento, porém a sequence é independente da tabela.

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

Através dela é implementada a integridade referencial no banco onde só será aceito valores referentes à chave primaria com a qual ela se relaciona.

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

Os operadores como União, Interseção, Diferença, Produto Cartesiano, Seleção, Projeção, Divisão, Junção, Renomear e Atribuição pertencem a álgebra relacional e os de comparação (<, ≤, =, ≠, >, ≥) e lógico (AND, OR, NOT) fazem parte do cálculo relacional.

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

Utilizando a cláusula GROUP BY sendo necessário informar a(s) coluna(s) que se deseja agrupar

5. CONCLUSÃO

Nesta prática foi possível treinar as habilidades de modelagem de dados bem como a implementação em um banco de dados relacional e os procedimentos necessários para efetuar o relacionamento entre as tabelas utilizando funcionalidades dos tipos de linguagem DDL, DML e DQL. A utilização do SQL Server Management Studio facilitou muito o trabalho de gerenciamento de banco de dados. Todo código utilizado nesta prática, bem como este relatório, estará disponível no GitHub.