



Relatório

UNIVERSADE ESTÁCIO DE SÁ POLO Paraíba
CURSO: DESENVOLVIMENTO FULL STACK
DISCIPLINA: INICIADO O CAMINHO PELO JAVA

Aluno: Yvo Murilo

Campus: Polo João Pessoa - Paraíba – PB

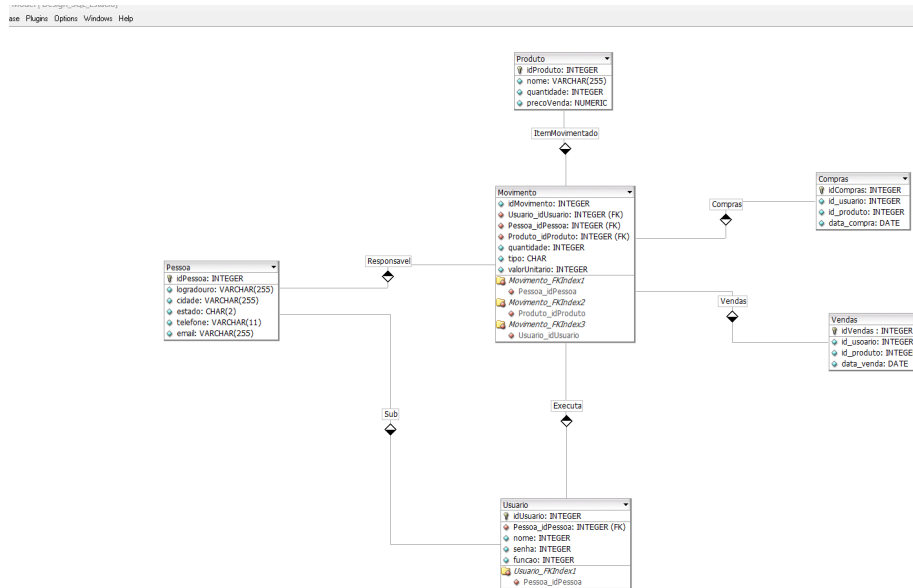
TURMA: 22.3

3º - SEMESTRE

Objetivos da prática

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

Procedimento -1



Análise e Conclusão

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

As diferentes cardinalidades em um banco de dados relacional, como 1X1, 1XN ou NxN, são implementadas por meio do uso de relacionamentos entre tabelas. Esses relacionamentos são estabelecidos por meio de chaves estrangeiras (foreign keys) e são fundamentais para a integridade referencial dos dados.

Cardinalidade 1X1:

.Para implementar uma relação 1X1 em um banco de dados relacional, você cria duas tabelas, cada uma com uma chave primária (primary key).

.Em uma das tabelas, você inclui uma chave estrangeira que faz referência à chave primária da outra tabela.

.Isso significa que cada registro na primeira tabela está relacionado a exatamente um registro na segunda tabela e vice-versa.

.Esse tipo de relacionamento é usado quando você tem informações separadas que estão fortemente relacionadas, mas não precisam estar na mesma tabela por motivos de organização ou de desempenho.

Cardinalidade 1XN

.Para implementar uma relação 1XN, você também cria duas tabelas com chaves primárias.

.Na tabela "um" (1), você inclui uma chave estrangeira que faz referência à tabela "muitos" (N).

.Isso significa que um registro na tabela "um" pode estar relacionado a muitos registros na tabela "muitos", mas um registro na tabela "muitos" está relacionado a apenas um registro na tabela "um".

.Esse tipo de relacionamento é comum quando você tem informações que são compartilhadas por vários registros em uma tabela relacionada.

Cardinalidade NxN

.A implementação de uma relação NxN é mais complexa e requer o uso de uma tabela de associação (tabela intermediária ou junção).

.Você cria três tabelas: duas tabelas principais com chaves primárias e uma terceira tabela que serve como tabela de associação.

.Cada tabela de associação contém chaves estrangeiras que fazem referência às tabelas principais, criando um relacionamento NxN.

Isso permite que muitos registros de uma tabela estejam relacionados a muitos registros da outra tabela.

.Esse tipo de relacionamento é usado quando você tem entidades que podem ter várias associações com outras entidades.

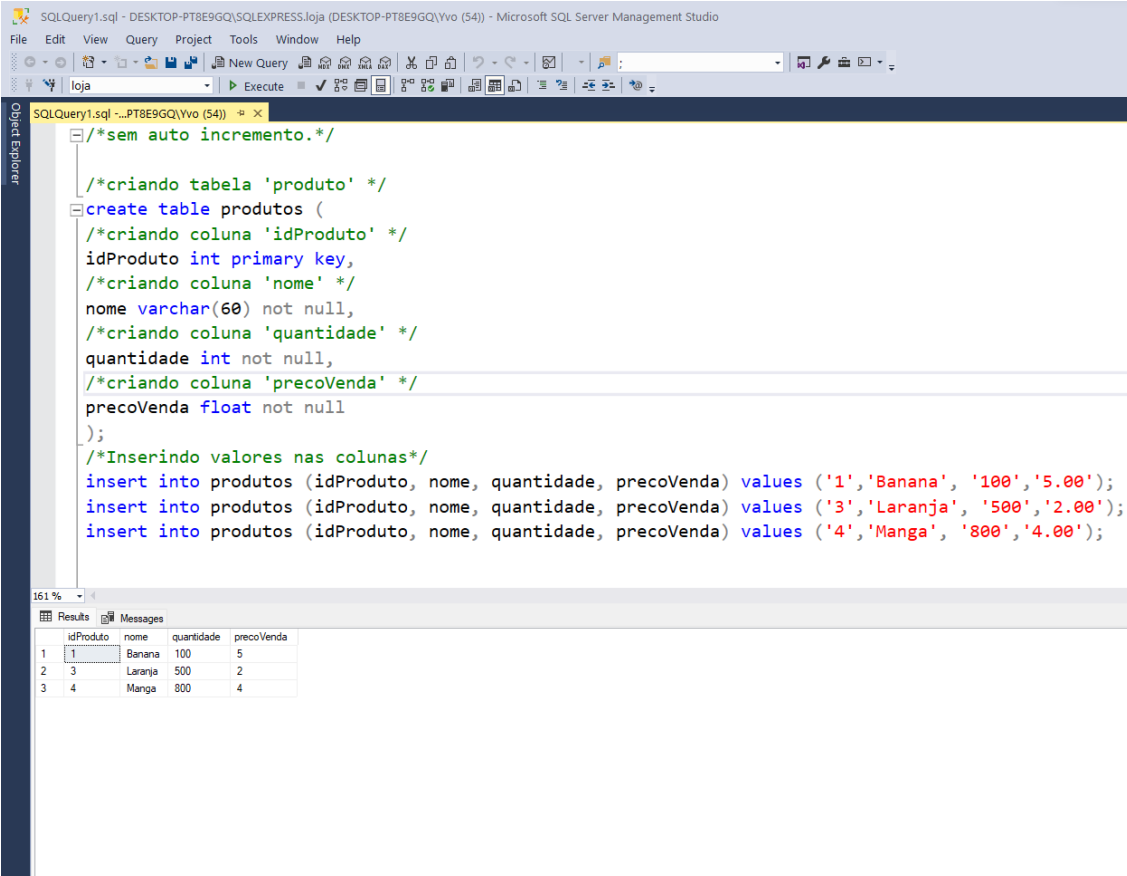
b)Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

Em bancos de dados relacionais, a herança (ou a relação "é-um") entre tabelas é geralmente representada usando um modelo de dados conhecido como "modelagem de herança" ou "modelagem de subclasses". Existem várias maneiras de implementar a herança em bancos de dados relacionais, e a abordagem exata que você deve escolher depende do sistema de gerenciamento de banco de dados (SGBD) que você está usando e dos requisitos específicos do seu aplicativo.

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

O SSMS permite que os usuários monitorem o desempenho do servidor de banco de dados, identificando gargalos e otimizando consultas. Ele fornece informações detalhadas sobre o uso de recursos, bloqueios, consultas em execução e muito mais.

Procedimento -2



The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The main window shows a SQL query in the 'SQLQuery1.sql' file. The query is a T-SQL script that creates a table named 'produtos' and inserts three rows of data. The query is as follows:

```
/*sem auto incremento.*/  
  
/*criando tabela 'produto' */  
create table produtos (  
/*criando coluna 'idProduto' */  
idProduto int primary key,  
/*criando coluna 'nome' */  
nome varchar(60) not null,  
/*criando coluna 'quantidade' */  
quantidade int not null,  
/*criando coluna 'precoVenda' */  
precoVenda float not null  
);  
  
/*Inserindo valores nas colunas*/  
insert into produtos (idProduto, nome, quantidade, precoVenda) values ('1', 'Banana', '100', '5.00');  
insert into produtos (idProduto, nome, quantidade, precoVenda) values ('3', 'Laranja', '500', '2.00');  
insert into produtos (idProduto, nome, quantidade, precoVenda) values ('4', 'Manga', '800', '4.00');
```

Below the query editor, the 'Results' pane shows the output of the query. It displays a table with four columns: 'idProduto', 'nome', 'quantidade', and 'precoVenda'. The table contains three rows of data:

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

tabela_usuarios.sql - DESKTOP-PT8E9GQ\SQLEXPRESS.loja (DESKTOP-PT8E9GQ\Yvo (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

loja Execute

Object Explorer

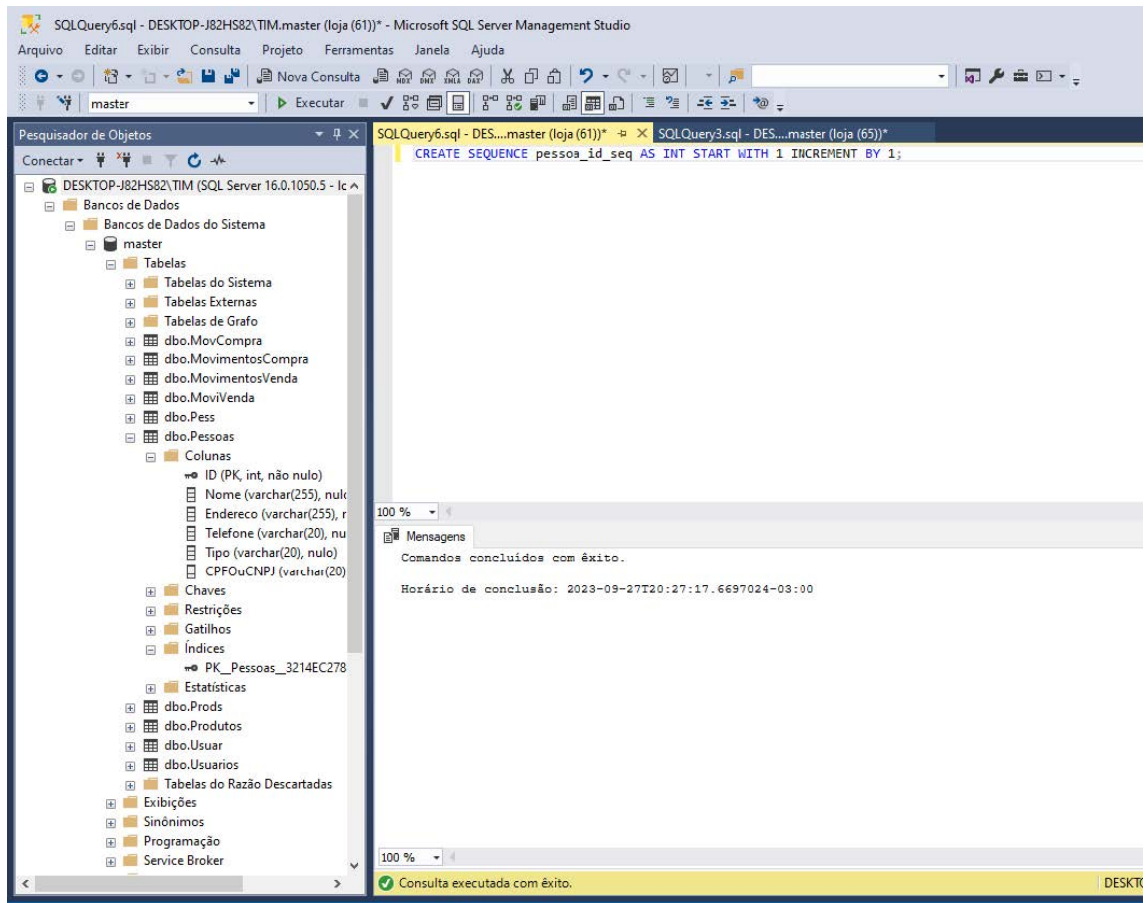
tabela_usuarios....PT8E9GQ\Yvo (52))

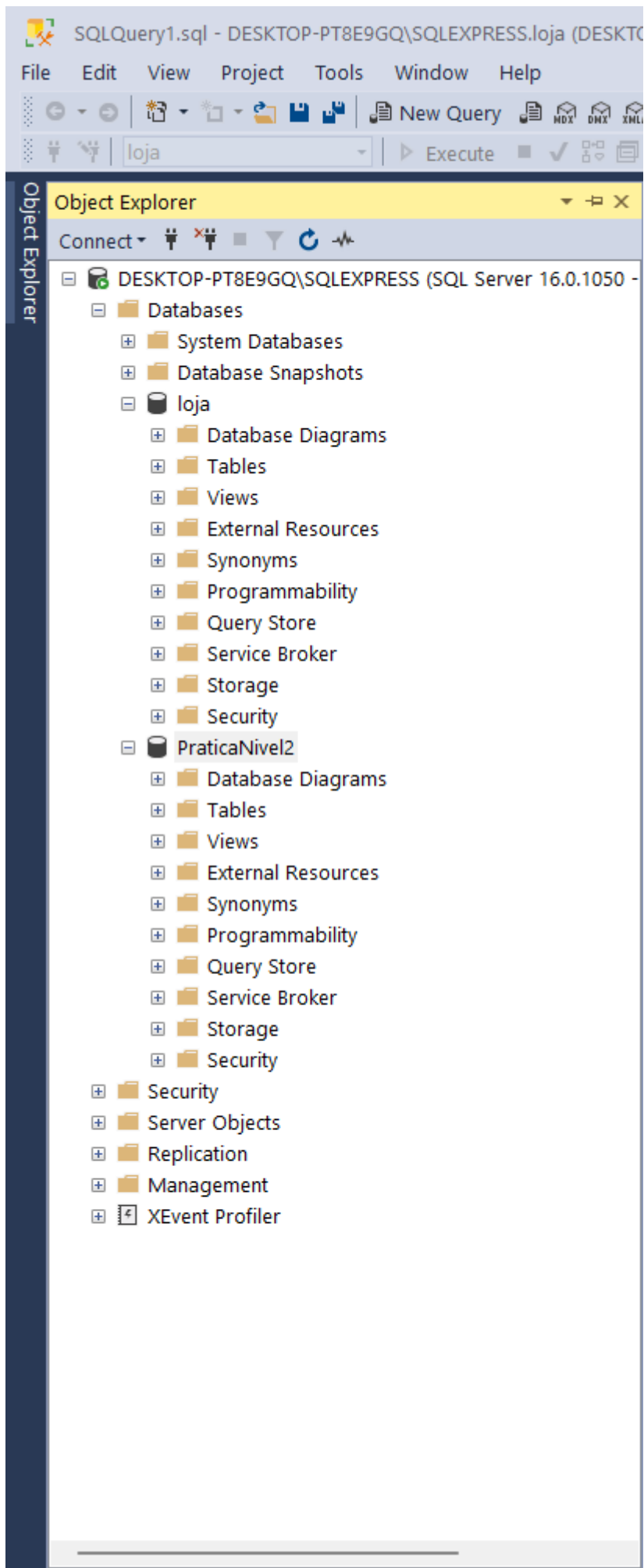
```
/*sem auto incremento.*/
create table usuario (
  idUsuario int primary key,
  login varchar(60) not null,
  senha varchar(60) not null
);
insert into usuarios (idUsuario, login, senha) values ('1', 'op1', 'op1');
insert into usuarios (idUsuario, login, senha) values ('2', 'op2', 'op2');
select * from usuarios;
```

161 %

Results Messages

	idUsuario	login	senha
1	1	op1	op1
2	2	op2	op2





Análise e Conclusão

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

Em um contexto de banco de dados e alinhamento de sequências, "sequence" e "identity" são termos comumente usados para descrever diferentes aspectos das semelhanças entre sequências biológicas, como sequências de DNA, RNA ou proteínas.

Sequence

"Sequence" refere-se à própria cadeia de nucleotídeos (no caso de DNA ou RNA) ou aminoácidos (no caso de proteínas). Quando você fala sobre a sequência, está se referindo à ordem específica dos componentes individuais na cadeia, ou seja, as letras que compõem a sequência (A, T, C, G para DNA, A, U, C, G para RNA, e aminoácidos no caso de proteínas).

Identity

"Identity" refere-se à medida de quão semelhantes duas sequências são em termos de correspondência de nucleotídeos ou aminoácidos. É uma medida de quão idênticas duas sequências são em relação uma à outra. A identidade é geralmente expressa como uma porcentagem e representa a proporção de nucleotídeos ou aminoácidos idênticos entre as duas sequências em comparação.

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

As chaves estrangeiras desempenham um papel fundamental na manutenção da consistência de um banco de dados. Elas são um componente-chave da integridade referencial, que é um conjunto de regras e restrições que garantem que os dados em um banco de dados sejam consistentes e precisos.

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

A álgebra relacional e o cálculo relacional são duas abordagens distintas para consultar dados em bancos de dados relacionais. Ambas são linguagens formais usadas em sistemas de gerenciamento de banco de dados (DBMS) e são usadas para recuperar informações de um banco de dados, mas têm abordagens diferentes.

Operadores da Álgebra Relacional em SQL:

Projeção (SELECT): O operador SELECT é usado para recuperar colunas específicas de uma tabela.

Restrição (WHERE): O operador WHERE é usado para filtrar linhas com base em uma condição.

União (UNION): O operador UNION combina resultados de duas consultas em uma única saída.

Interseção (INTERSECT): O operador INTERSECT retorna linhas que estão presentes em ambas as consultas.

Diferença (EXCEPT ou MINUS): O operador EXCEPT ou MINUS retorna linhas presentes na primeira consulta, mas não na segunda.

Operadores do Cálculo Relacional em SQL:

SELEÇÃO (σ): A operação de seleção é usada para escolher linhas que atendam a um determinado predicado.

PROJEÇÃO (π): A operação de projeção é usada para selecionar colunas específicas de uma tabela.

UNIÃO (\cup): A operação de união retorna todas as linhas que estão presentes em qualquer uma das tabelas.

INTERSEÇÃO (\cap): A operação de interseção retorna todas as linhas que estão presentes em ambas as tabelas.

DIFERENÇA ($-$): A operação de diferença retorna todas as linhas presentes em uma tabela, mas não na outra.

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

O agrupamento em consultas, frequentemente referido como clustering em consulta, é uma técnica utilizada para organizar e agrupar consultas de pesquisa em categorias ou clusters semelhantes com base em critérios específicos. Isso pode ser útil em sistemas de recuperação de informações, motores de busca e em outras aplicações onde é necessário entender e gerenciar um grande volume de consultas de pesquisa.