



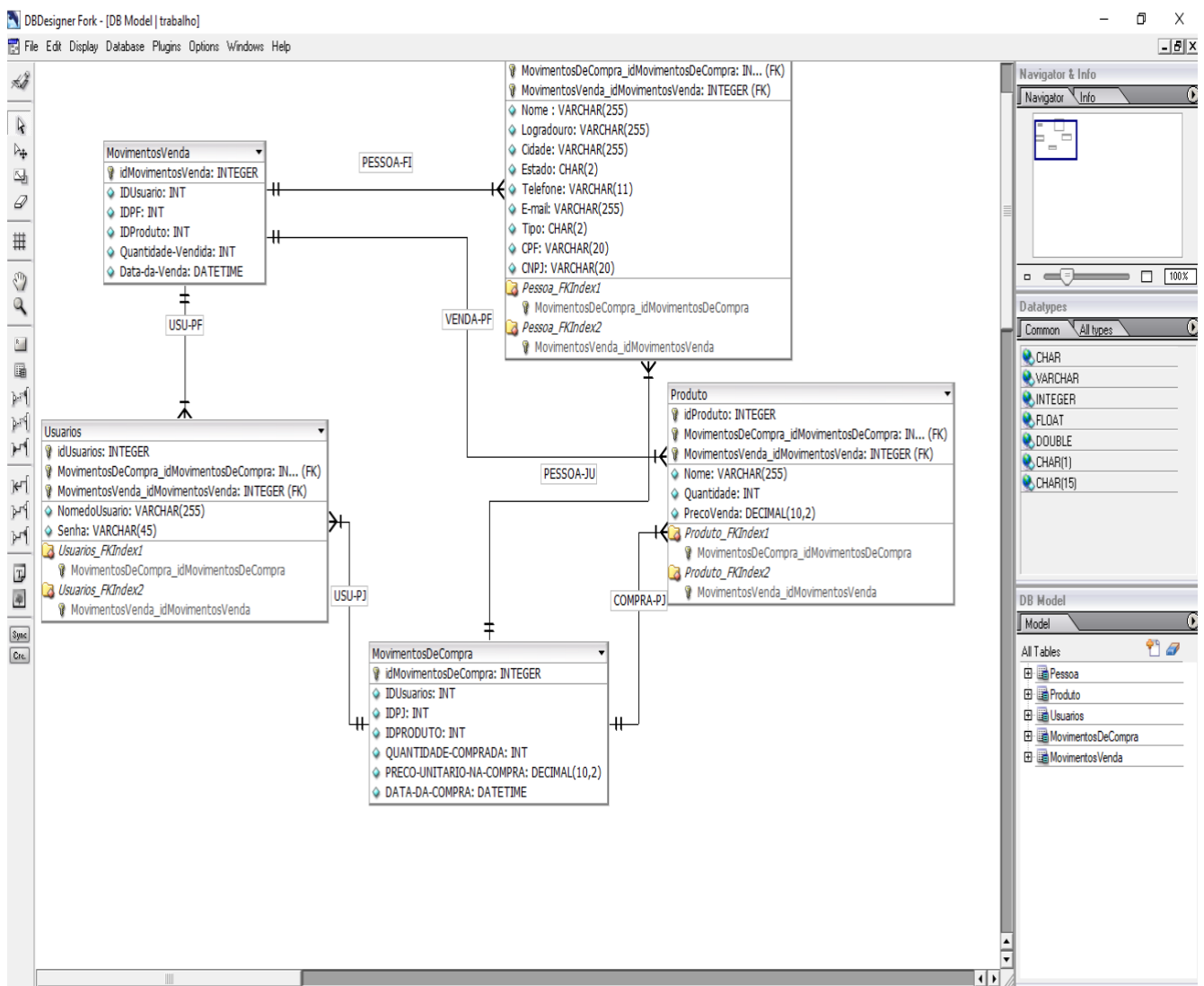
UNIVERSADE ESTÁCIO DE SÁ POLO IBIRITÉ

CURSO: DESENVOLVIMENTO FULL STACK
DISCIPLINA: INICIANDO O CAMINHO PELO JAVA

EVERTON GOMES COSTA
TURMA: 22.3
3º SEMESTRE

IBIRITÉ-MG
2023

PROCEDIMENTO 1:



Análise e Conclusão:

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

1x1 (Um-para-Um): Em um relacionamento 1x1, uma tabela está relacionada a outra, onde cada registro em uma tabela está associado a no máximo um registro em outra tabela e vice-versa. Isso é geralmente implementado através de chaves estrangeiras que são únicas em ambas as tabelas.

1xN (Um-para-Muitos): Neste relacionamento, um registro em uma tabela pode estar associado a vários registros em outra tabela, mas cada registro nesta segunda tabela está associado a no máximo um registro na primeira tabela. Isso é alcançado através da inclusão de uma chave estrangeira na tabela "muitos" que faz referência à chave primária da tabela "um".

NxN (Muitos-para-Muitos): Este relacionamento é implementado utilizando uma tabela de associação comunitária. Em vez de criar uma relação direta entre as duas tabelas, crie-se

uma tabela adicional que armazena pares de chaves estrangeiras, permitindo assim a representação de associações muitas-para-muitos.

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

Para representar herança em bancos de dados relacionais, você pode usar duas abordagens principais: a herança por tabela única (Herança de Tabela Única) e a herança por tabela múltipla (Herança de Tabela Múltipla).

Herança por Tabela Única (STI): Todas as classes relacionadas a mesma tabela. É utilizado um campo discriminador para identificar a qual classe cada registro pertence.

Herança por Tabela Múltipla (MTI): Cada classe derivada tem sua própria tabela. As tabelas derivadas contêm todos os campos da tabela base, além de seus próprios campos adicionais.

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

Ambiente Gráfico e Intuitivo: O SSMS fornece uma interface gráfica e intuitiva para gerenciar bancos de dados SQL Server, facilitando tarefas como criação, modificação e exclusão de objetos de banco de dados.

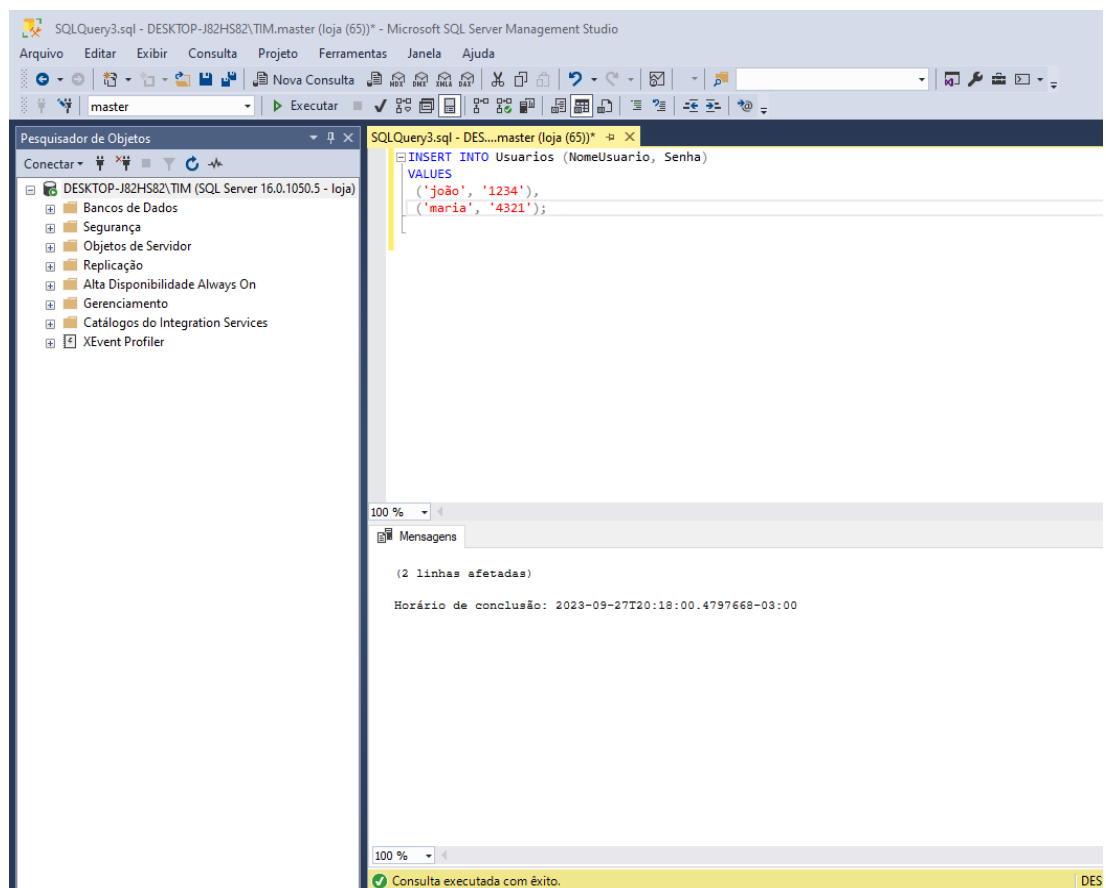
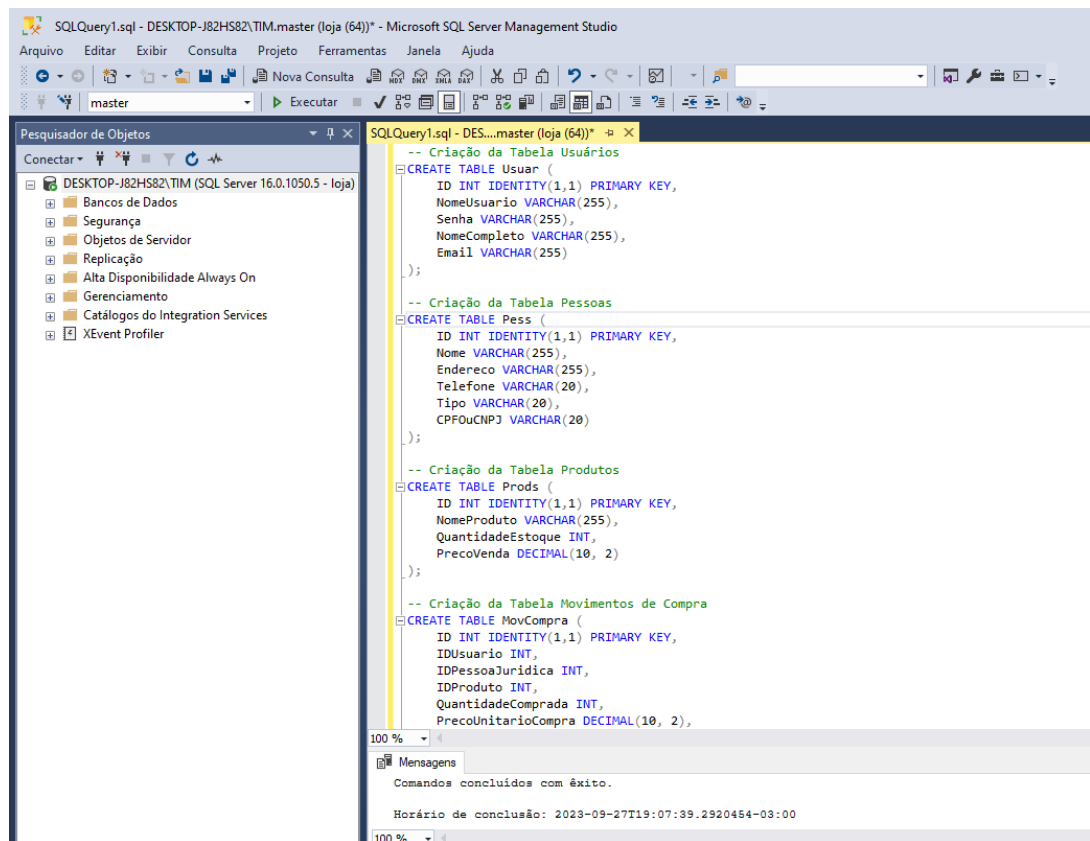
Editor SQL Integrado: O SSMS possui um editor SQL integrado que oferece destaque de sintaxe, autocompletar e outras funcionalidades que auxiliam na escrita e execução de consultas SQL.

Ferramentas de Desempenho e Otimização: O SSMS inclui ferramentas para monitorar o desempenho do banco de dados, melhorar consultas e índices, e analisar planos de execução.

Controle de Versão e Integração com o Controle de Fonte: O SSMS oferece integração com sistemas de controle de versão, permitindo que os desenvolvedores controlem e versionem seus scripts de banco de dados.

Automatização de Tarefas: O SSMS permite a criação e execução de scripts SQL, programação de tarefas e uso de procedimentos armazenados para automatizar tarefas de administração do banco de dados.

PROCEDIMENTO 2:



SQLQuery6.sql - DESKTOP-J82HS82\TIM.master (loja (61)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

master Executar

Pesquisador de Objetos

- DESKTOP-J82HS82\TIM (SQL Server 16.0.1050.5 - local)
- Bancos de Dados
 - Bancos de Dados do Sistema
 - master
 - Tabelas
 - Tabelas do Sistema
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.MovCompra
 - dbo.MovimentosCompra
 - dbo.MovimentosVenda
 - dbo.MoviVenda
 - dbo.Pess
 - dbo.Pessoas
 - Colunas
 - ID (PK, int, não nulo)
 - Nome (varchar(255), nulo)
 - Endereco (varchar(255), nulo)
 - Telefone (varchar(20), nulo)
 - Tipo (varchar(20), nulo)
 - CPFouCNPJ (varchar(20), nulo)
 - Chaves
 - Restrições
 - Gatilhos
 - Índices
 - PK_Pessoas__3214EC278
 - Estatísticas
 - dbo.Prods
 - dbo.Produtos
 - dbo.Usuar
 - dbo.Usuarios
 - Tabelas do Razão Descartadas
 - Exibições
 - Sinônimos
 - Programação
 - Service Broker

SQLQuery6.sql - DES...master (loja (61))

```
CREATE SEQUENCE pessoa_id_seq AS INT START WITH 1 INCREMENT BY 1;
```

SQLQuery3.sql - DES...master (loja (65))

100 %

Mensagens

Comandos concluídos com êxito.

Horário de conclusão: 2023-09-27T20:27:17.6697024-03:00

100 %

Consulta executada com êxito.

SQLQuery7.sql - DESKTOP-J82HS82\TIM.master (loja (72)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

master Executar

Pesquisador de Objetos

- DESKTOP-J82HS82\TIM (SQL Server 16.0.1050.5 - loja)
- Bancos de Dados
 - Bancos de Dados do Sistema
 - master
 - Tabelas
 - Tabelas do Sistema
 - Tabelas Externas
 - Tabelas de Grafo
 - dbo.MovCompra
 - dbo.MovimentosCompra
 - dbo.MovimentosVenda
 - dbo.MoviVenda
 - dbo.Pess
 - dbo.Pessoas
 - Colunas
 - ID (PK, int, não nulo)
 - Nome (varchar(255), nulo)
 - Endereco (varchar(255), nulo)
 - Telefone (varchar(20), nulo)
 - Tipo (varchar(20), nulo)
 - CPFouCNPJ (varchar(20), nulo)
 - Chaves
 - Restrições
 - Gatilhos
 - Índices
 - Estatísticas
 - dbo.Prods
 - dbo.Produtos
 - Colunas
 - ID (PK, int, não nulo)
 - NomeProduto (varchar(255), nulo)
 - QuantidadeEstoque (int, nulo)
 - PrecoVenda (decimal(10,2), nulo)
 - Chaves
 - Restrições
 - Gatilhos

SQLQuery7.sql - DES...master (loja (72))

```
SELECT TOP (1000) [ID]  
      ,[Nome]  
      ,[Endereco]  
      ,[Telefone]  
      ,[Tipo]  
      ,[CPFouCNPJ]  
FROM [master].[dbo].[Pessoas]
```

SQLQuery6.sql - DES...master (loja (58))

SQLQuery5.sql

100 %

Resultados

	ID	Nome	Endereco	Telefone	Tipo	CPFouCNPJ
1	1	João	rua 1	11111111111	pf	22222222222
2	2	Maria	rua 500	000000000	pj	11111111111
3	3	jorge	rua 800	44444444444	pf	88888888888

Consulta executada com êxito.

SQLQuery4.sql - DESKTOP-J82HS82\TIM.master (loja 56)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

master

Pesquisador de Objetos

Conectar

ESKTOP-J82HS82\TIM (SQL Server 16.0.1050.5 - loja)

Bancos de Dados

Bancos de Dados do Sistema

master

Tabelas

Tabelas do Sistema

Tabelas Externas

Tabelas de Grafo

dbo.MovCompra

dbo.MovimentosCompra

dbo.MovimentosVenda

dbo.MovVenda

dbo.Pess

dbo.Pessoas

dbo.Prods

dbo.Produtos

Colunas

ID (PK, int, não nulo)

NomeProduto (varchar(255), nulo)

QuantidadeEstoque (int, nulo)

PrecoVenda (decimal(10,2), nulo)

Chaves

Restrições

Gatilhos

Índices

Estatísticas

dbo.Usuario

dbo.Usuarios

Tabelas do Razão Descartadas

Exibições

Sinônimos

Programação

Service Broker

Armazenamento

Segurança

model

SQLQuery4.sql - DESKTOP-J82HS82\TIM.master (loja 56))

SQLQuery3.sql - DESKTOP-J82HS82\TIM.master (loja 52))

SQLQuery2.sql - DESKTOP-J82HS82\TIM.master (loja 51))

```
SELECT TOP (1000) [ID]
, [NomeProduto]
, [QuantidadeEstoque]
, [PrecoVenda]
FROM [master].[dbo].[Produtos]
```

100 %

Resultados Mensagens

	ID	NomeProduto	QuantidadeEstoque	PrecoVenda
1	1	Banana	100	5.00
2	2	Laranja	500	2.00
3	3	Manga	800	4.00

Consulta executada com êxito.

SQLQuery7.sql - DESKTOP-J82HS82\TIM.master (loja 62)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

master

Pesquisador de Objetos

Conectar

DESKTOP-J82HS82\TIM (SQL Server 16.0.1050.5 - loja)

Bancos de Dados

Bancos de Dados do Sistema

master

Tabelas

Tabelas do Sistema

Tabelas Externas

Tabelas de Grafo

dbo.MovCompra

dbo.MovimentosCompra

dbo.MovimentosVenda

dbo.MovVenda

dbo.Pess

dbo.Pessoas

Colunas

ID (PK, int, não nulo)

Nome (varchar(255), nulo)

Endereco (varchar(255), nulo)

Telefone (varchar(20), nulo)

Tipo (varchar(20), nulo)

CPFouCNPJ (varchar(20), nulo)

Chaves

Restrições

Gatilhos

Índices

PK_Pessoas_3214EC278

Estatísticas

dbo.Prods

dbo.Produtos

dbo.Usuario

dbo.Usuarios

Tabelas do Razão Descartadas

Exibições

Sinônimos

Programação

Service Broker

SQLQuery7.sql - DESKTOP-J82HS82\TIM.master (loja 62))

SQLQuery6.sql - DESKTOP-J82HS82\TIM.master (loja 61))

SQLQuery3.sql - DESKTOP-J82HS82\TIM.master (loja 51))

```
SELECT TOP (1000) [ID]
, [NomeUsuario]
, [Senha]
, [NomeCompleto]
, [Email]
FROM [master].[dbo].[Usuarios]
```

100 %

Resultados Mensagens

	ID	NomeUsuario	Senha	NomeCompleto	Email
1	1	op1	op1	NULL	NULL
2	2	op2	op2	NULL	NULL
3	3	joão	1234	NULL	NULL
4	4	maria	4321	NULL	NULL

Consulta executada com êxito.

Análise e Conclusão:

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

Sequência: No SQL Server, uma sequência é um objeto que gera uma sequência de valores únicos, geralmente usada para gerar chaves primárias. Pode ser usado em diferentes tabelas e não está vinculado a uma tabela específica.

Identidade: Uma identidade é uma propriedade que pode ser aplicada a uma coluna em uma tabela. Ela gera automaticamente valores incrementais exclusivos para essa coluna quando novas linhas são inseridas. Cada tabela pode ter no máximo uma identidade de coluna.

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

As chaves estrangeiras são fundamentais para manter a integridade referencial em um banco de dados relacional.

Eles garantem que as relações entre tabelas sejam válidas, impedindo que sejam criadas referências a registros que não existam na tabela referenciada.

Ao usar chaves estrangeiras, garanta que alterações em uma tabela não causem problemas nos relacionamentos com outras tabelas, mantendo a consistência dos dados.

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

Álgebra Relacional: Exemplos de operadores na álgebra relacional incluindo projeção (Π), seleção (σ), concentrada (\bowtie), união (\cup), interseção (\cap), renomeação (ρ), entre outros.

Cálculo Relacional: Não há uma correspondência direta entre os operadores da álgebra relacional e do design relacional. O cálculo relacional usa expressões mais descritivas, como $\{t \mid P(t)\}$ para representar as tuplas que satisfazem a condição P.

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

O agrupamento em consultas é realizado utilizando uma cláusula GROUP BY. Ele agrupa os resultados da consulta com base em uma ou mais colunas.

A cláusula GROUP BY é incluída por funções de agregação, como SUM, COUNT, AVG, etc., que operam nos grupos formados pelo GROUP BY.

Um requisito obrigatório ao usar GROUP BY é que todas as colunas na seleção sejam uma coluna no GROUP BY ou uma expressão envolvendo funções de agregação. Todas as colunas não agregadas devem ser incluídas no GROUP BY.