CURSO - Desenvolvimento full Stack
DISCIPLINA – Vamos manter as informações!
ALUNO (A) – Sfênia Mesquita da Silva Inacio
MATRÍCULA – 202208042341
CAMPUS - Polo Cohatrac/São Luis – MA
Mundo 3 Período 2023.2

1º Procedimento | Criando o Banco de Dados

1. **Título da Prática;**

Vamos manter as informações!

2. **Objetivos da prática;**
   - **a)** Identificar os requisitos de um sistema e transformá-los no modelo adequado.
   - **b)** Utilizar ferramentas de modelagem para bases de dados relacionais.
   - **c)** Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
   - **d)** Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
   - **e)** 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. **Todos os códigos solicitados neste roteiro de aula;**

https://github.com/sfenia/MP2-Estacio-Mundo3

4. Resultado da execução dos códigos;

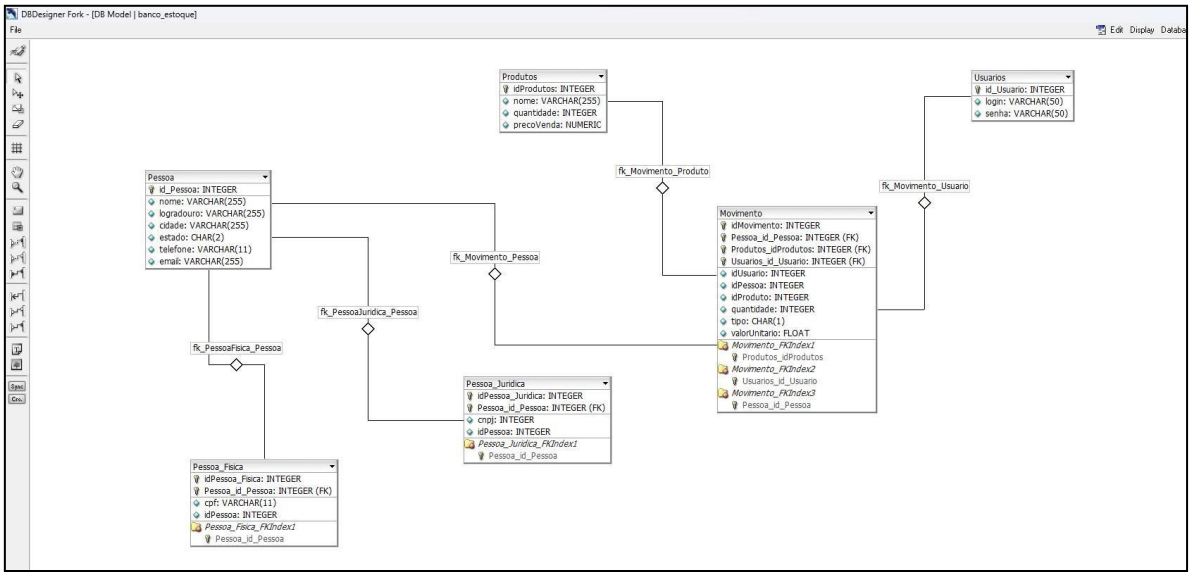


Figura 1 - Modelagem do banco via DBDesigner Fork

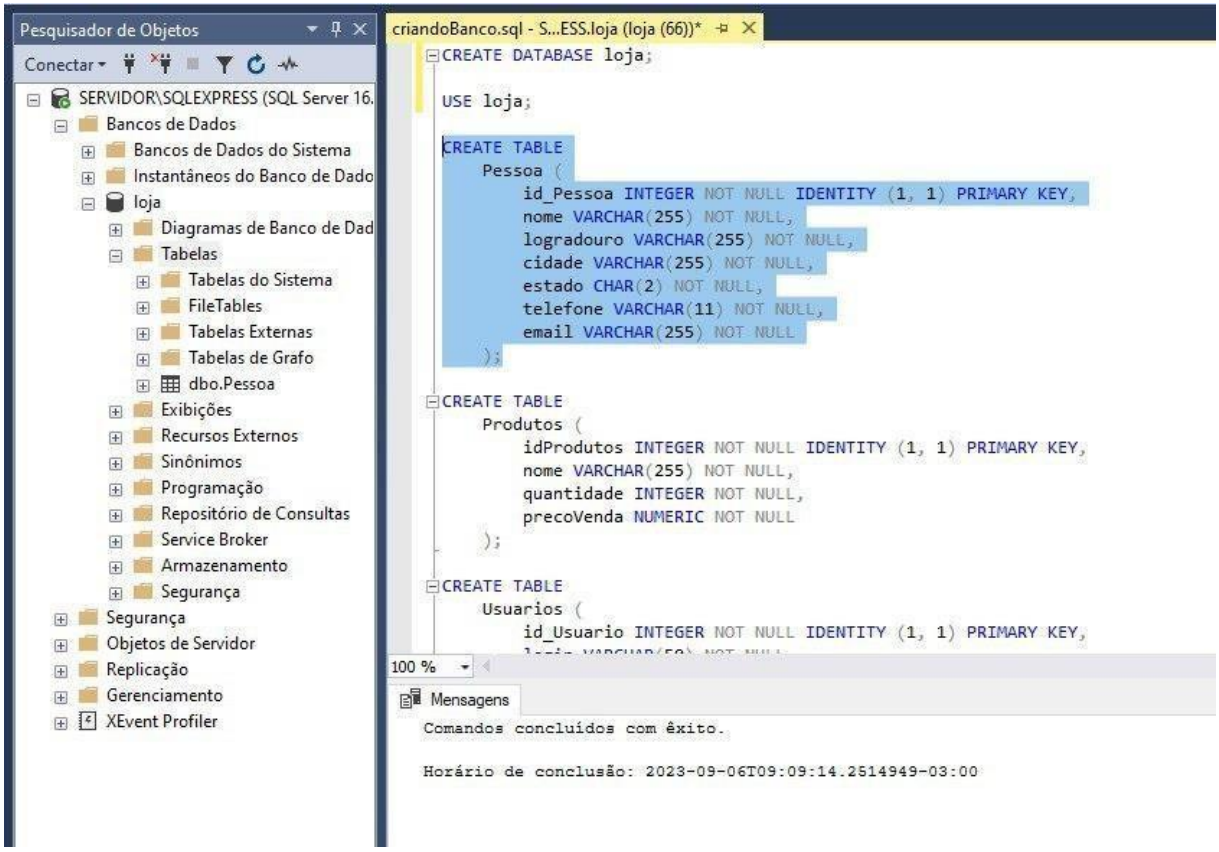


Figura 2 - Criando Tabela Pessoa

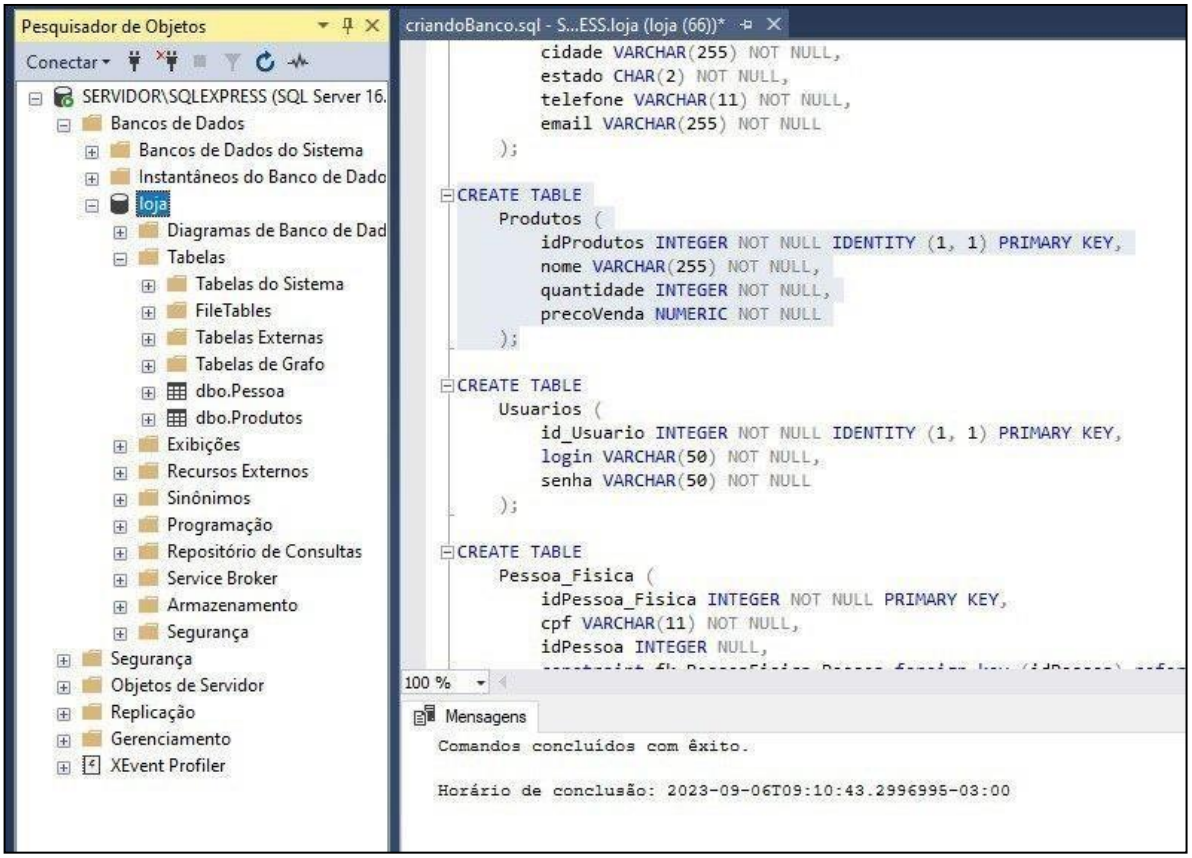


Figura 3 - Criando Tabela Produtos

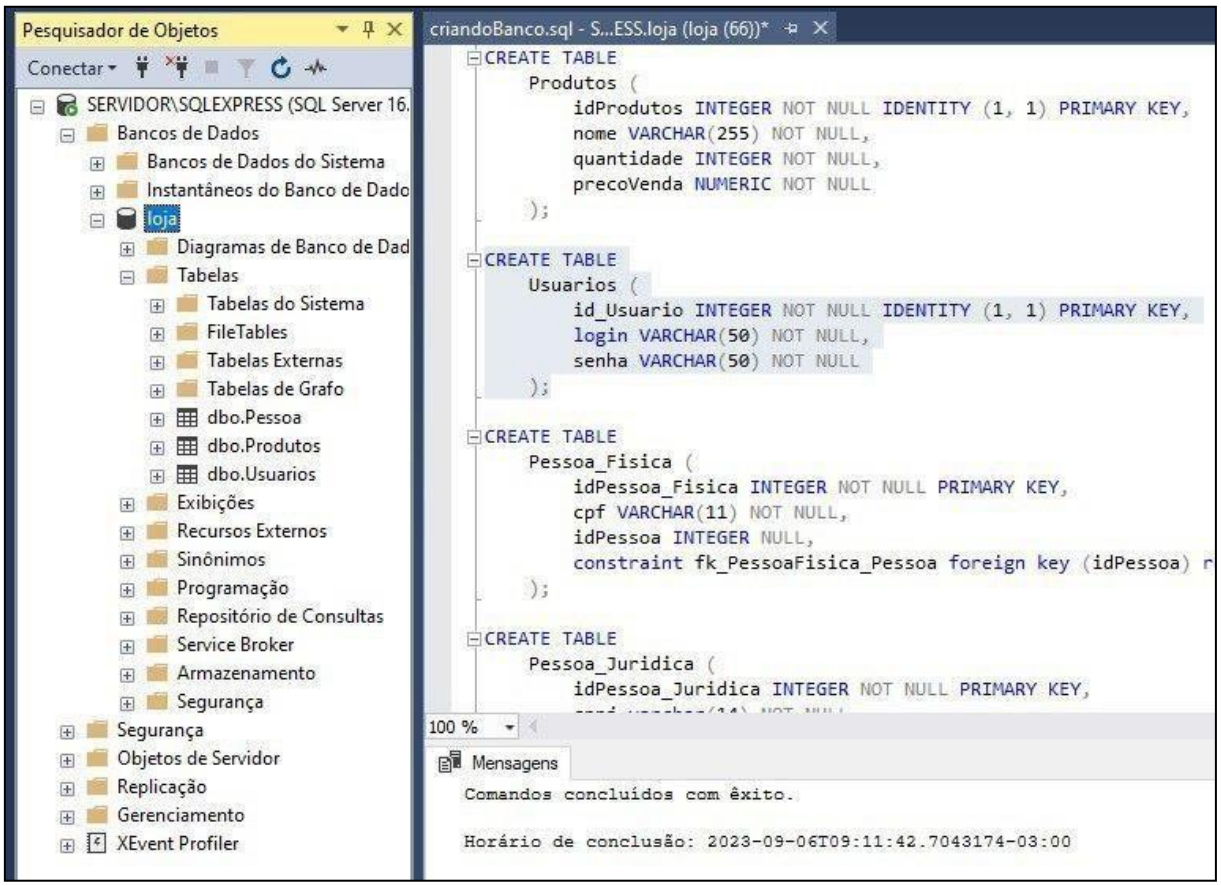


Figura 4 - Criando Tabela Usuários

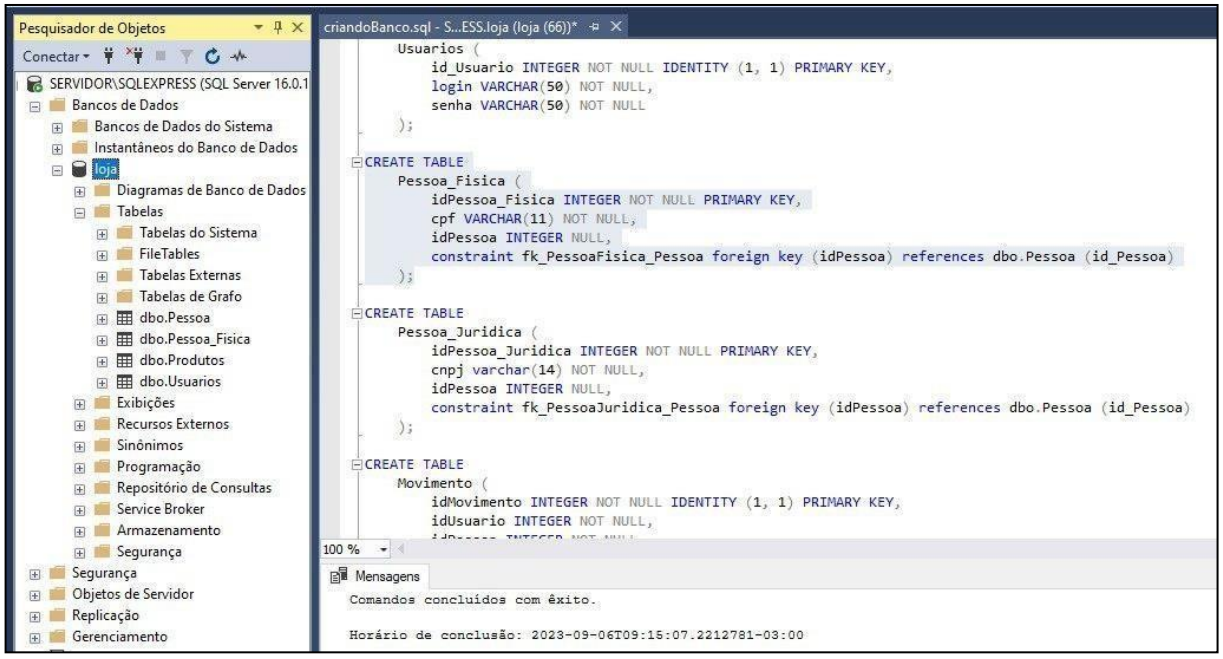


Figura 5 - Criando Tabela Pessoa_Fisica

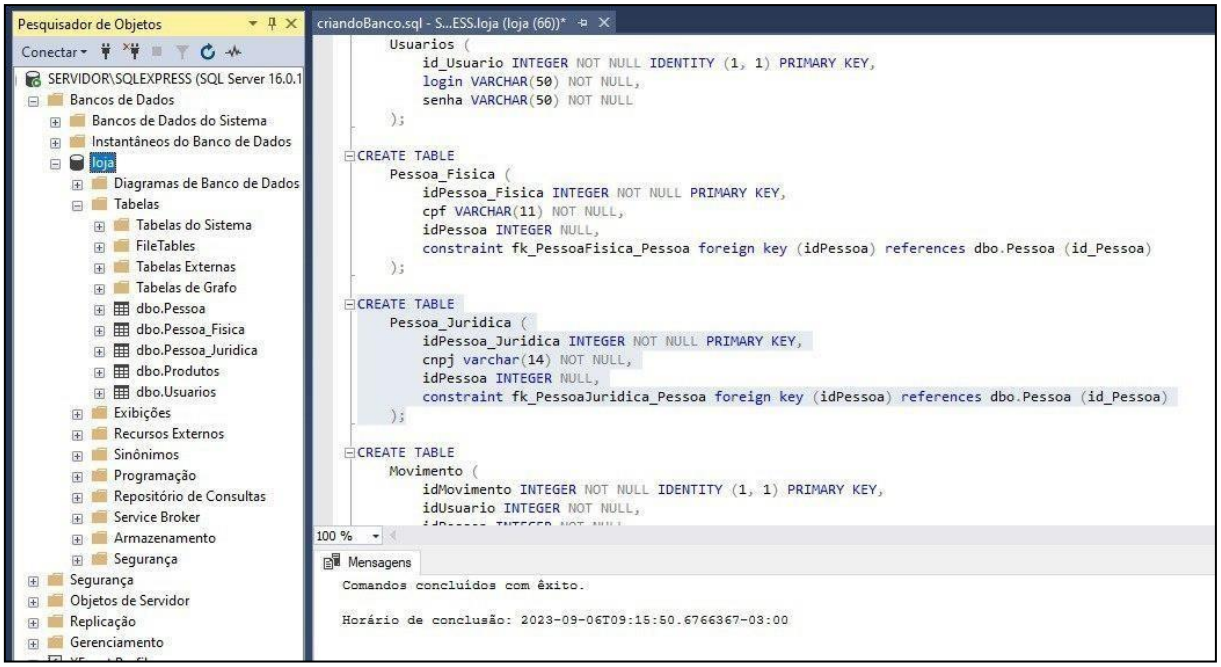


Figura 6 - Criando Tabela Pessoa_Juridica

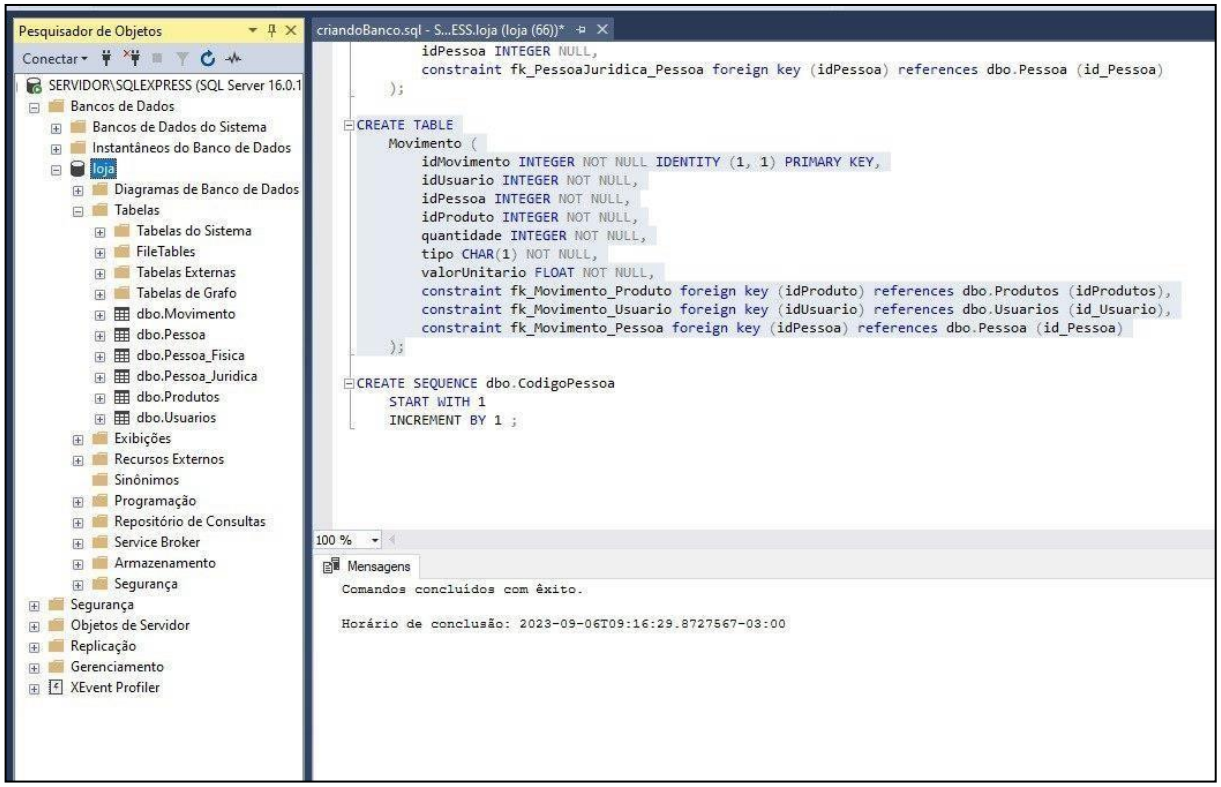


Figura 7 - Criando Tabela Movimento

5. Análise e Conclusão:

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

As diferentes cardinalidades são implementadas em um banco de dados relacional usando chaves primárias e chaves estrangeiras.

- Cardinalidade 1:1 (1 para 1)

Em um relacionamento 1:1, cada linha em uma tabela está relacionada a uma e apenas uma linha em outra tabela. Para implementar isso, as duas tabelas podem compartilhar uma chave primária comum.

Por exemplo, em uma tabela de produtos e uma tabela de fornecedores, temos a seguinte situação: cada produto tem um fornecedor específico, e cada fornecedor fornece um ou mais produtos. Podemos implementar esse relacionamento usando uma chave primária comum de ID do fornecedor nas duas tabelas.

- Cardinalidade 1:N (1 para vários)

Em um relacionamento 1:N, cada linha em uma tabela pode estar relacionada a várias linhas em outra tabela. Para implementar isso, a tabela com a cardinalidade mínima de 1 deve ter uma chave primária, e a tabela com a cardinalidade máxima de N deve ter uma chave estrangeira que faz referência à chave primária da tabela com a cardinalidade mínima de 1.

Considerando uma tabela de funcionários e uma tabela de projetos, cada funcionário pode trabalhar em vários projetos, mas cada projeto só pode ter um funcionário responsável. Podemos implementar esse relacionamento usando uma chave primária de ID do funcionário na tabela de funcionários e uma chave estrangeira de ID do funcionário na tabela de projetos.

- Cardinalidade N:N (vários para vários)

Em um relacionamento N:N, várias linhas em uma tabela podem estar relacionadas a várias linhas em outra tabela. Para implementar isso, é necessário criar uma terceira tabela que represente o relacionamento entre as duas tabelas originais. Essa tabela deve ter uma chave estrangeira de cada uma das tabelas originais.

Por exemplo, considere uma tabela de produtos e uma tabela de categorias. Um produto pode pertencer a várias categorias, e uma categoria pode incluir vários produtos. Podemos implementar esse relacionamento criando uma tabela de produto_categoria com uma chave estrangeira de ID do produto da tabela de produtos e uma chave estrangeira de ID da categoria da tabela de categorias.

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

O tipo de relacionamento que deve ser utilizado para representar o uso de herança em bancos de dados relacionais é o relacionamento 1:N. Nesse relacionamento, uma tabela pai pode ter muitos registros filhos, mas um registro filho só pode ter um registro pai.

Na herança, uma classe herda de outra classe. Isso significa que a classe herdeira tem todos os atributos e métodos da classe ancestral, além de seus próprios atributos e métodos.

Em um banco de dados relacional, a classe ancestral pode ser representada por uma tabela, e a classe herdeira pode ser representada por uma tabela filha que herda da tabela ancestral. A relação entre a tabela ancestral e a tabela filha é 1:N, pois cada linha na tabela ancestral pode ter várias linhas filhas relacionadas a ela.

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 (SSMS) é uma ferramenta gráfica que permite aos administradores de banco de dados realizar uma variedade de tarefas relacionadas ao gerenciamento de banco de dados. Oferecendo uma série de recursos que podem ajudar os administradores de banco de dados a melhorar sua produtividade. Alguns exemplos específicos

de como o SSMS pode ajudar a melhorar a produtividade nas tarefas relacionadas ao gerenciamento do banco de dados são:

- Criação de bancos de dados: O SSMS oferece um assistente que pode ajudar os administradores a criar bancos de dados com facilidade.
- Gerenciamento de consultas: O SSMS oferece um editor de consultas que pode ajudar os administradores a criar e depurar consultas de maneira mais eficiente.
- Monitoramento do desempenho: O SSMS oferece uma variedade de ferramentas que podem ajudar os administradores a monitorar o desempenho do banco de dados e identificar problemas potenciais.
- Gerenciamento de segurança: O SSMS oferece uma variedade de ferramentas que podem ajudar os administradores a gerenciar a segurança do banco de dados, incluindo o gerenciamento de usuários, grupos e políticas de acesso.

2º Procedimento | Alimentando a Base

1. **Título da Prática;**

   RPG0015 - Vamos manter as informações!

2. **Objetivo da Prática;**

   a) Identificar os requisitos de um sistema e transformá-los no modelo adequado.
   b) Utilizar ferramentas de modelagem para bases de dados relacionais.
   c) Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
   d) Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
   e) 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. **Todos os códigos solicitados neste roteiro de aula;**

   https://github.com/Thiagomoreira97/Java-MP2.git

4. **Os resultados da execução dos códigos também devem ser apresentados;**

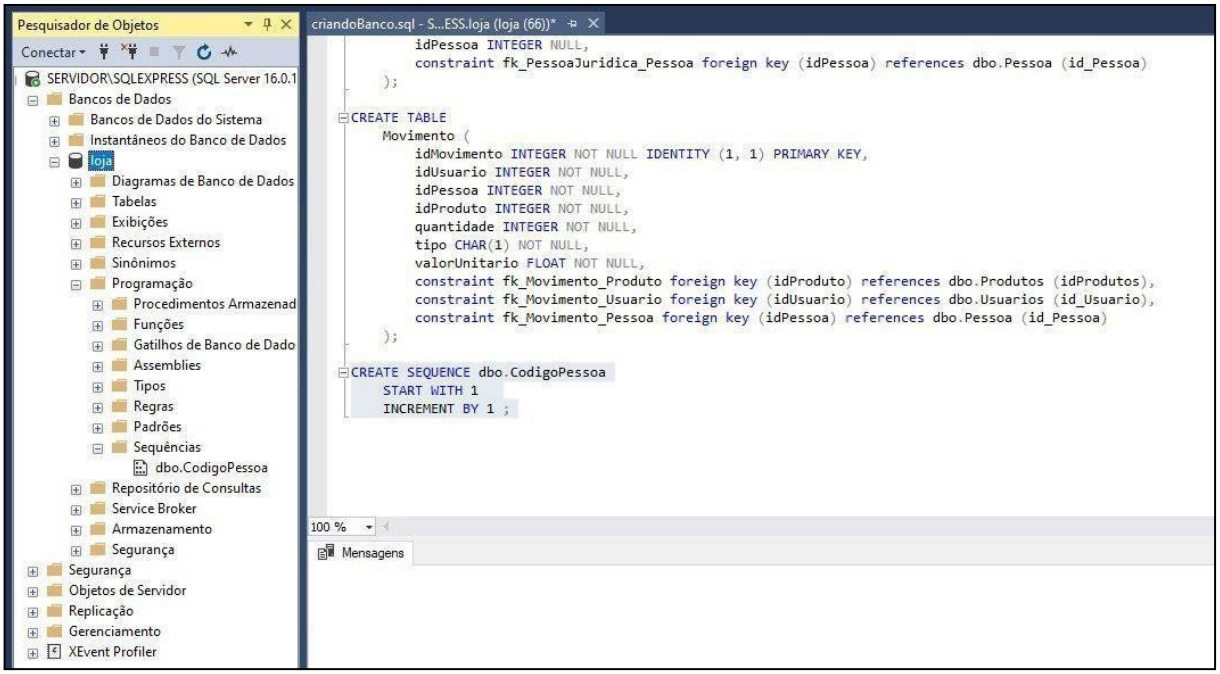


Figura 8 - Criando Sequence

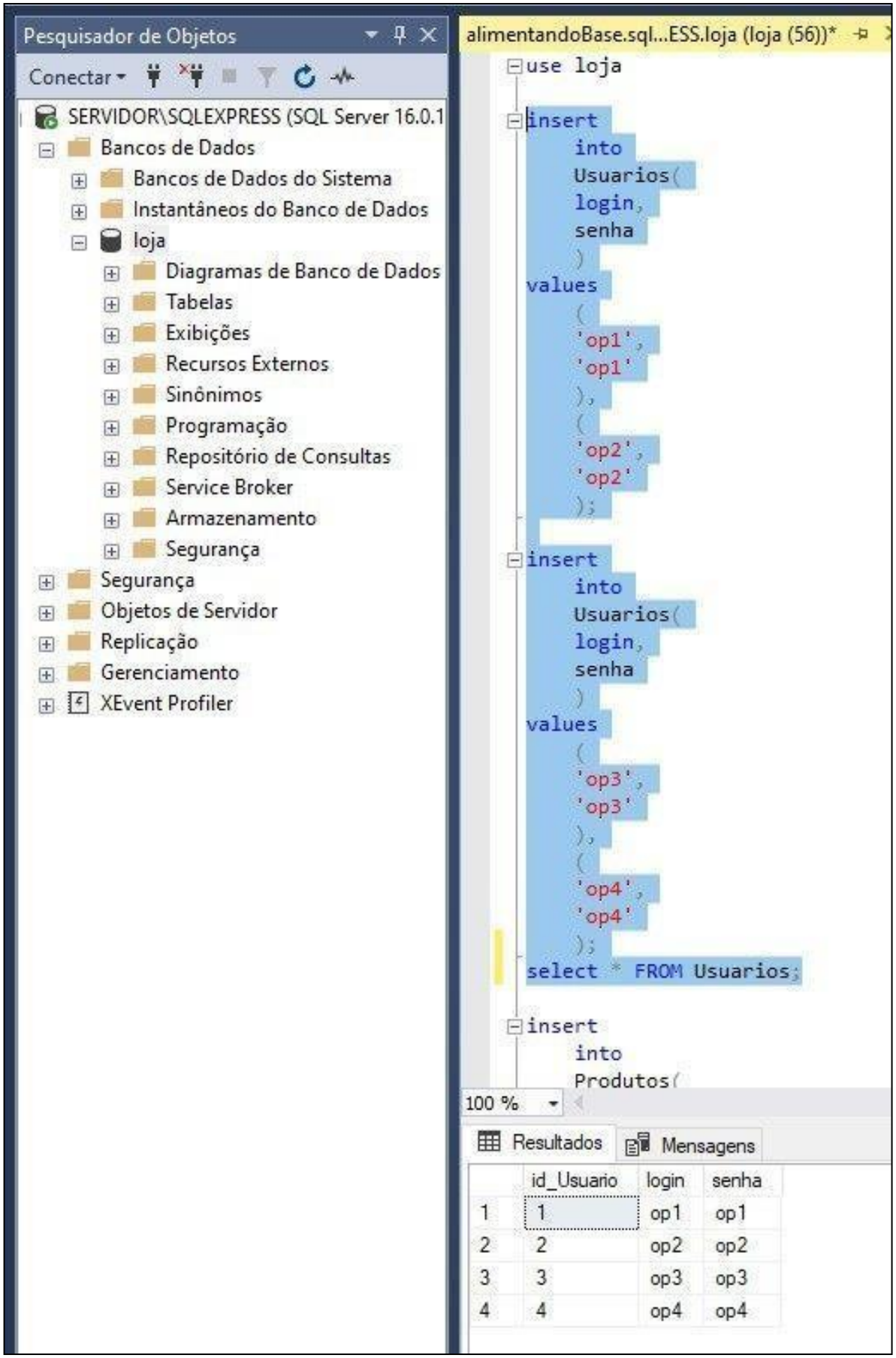


Figura 9 - Inserindo Usuários

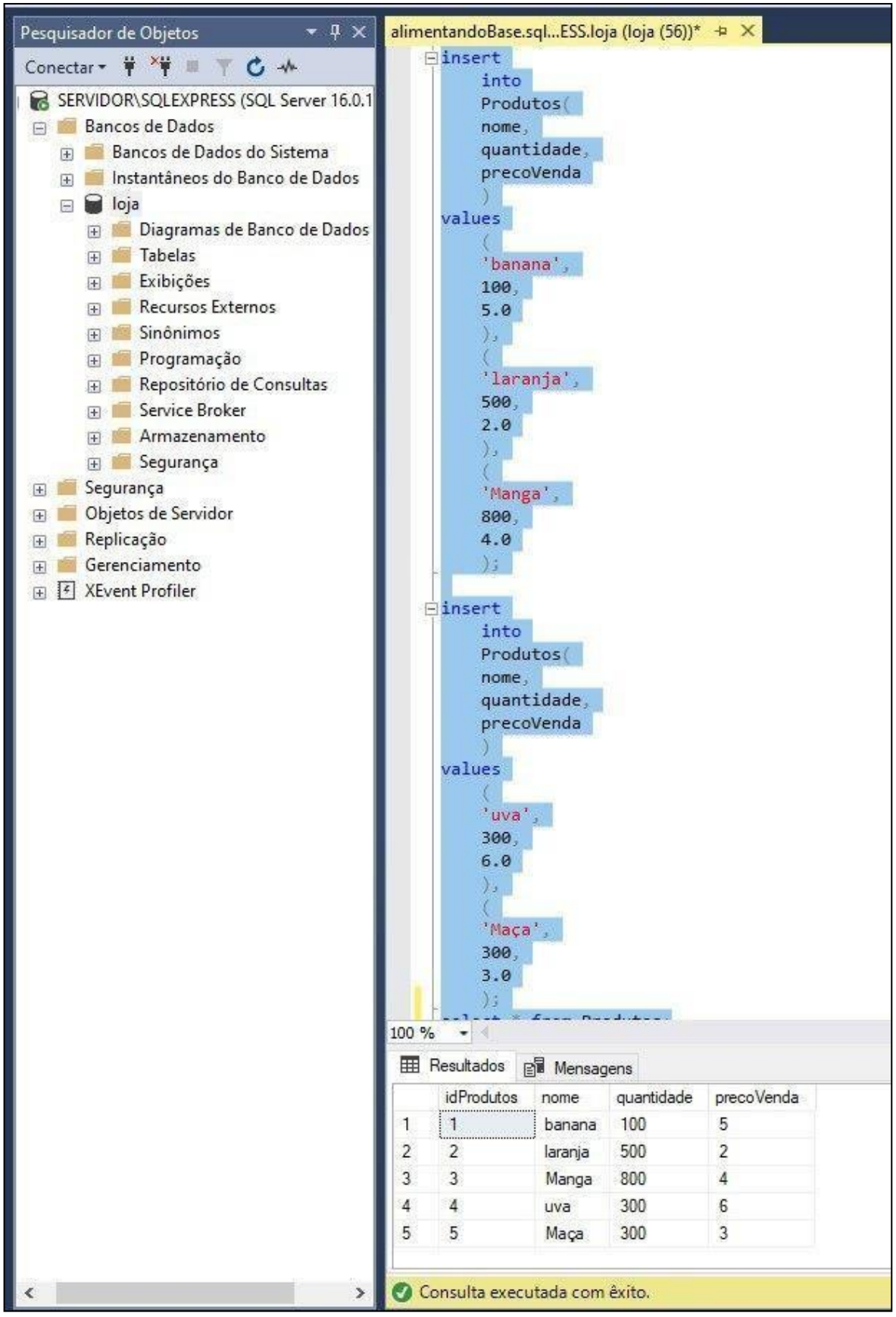


Figura 10 - Inserindo Produtos

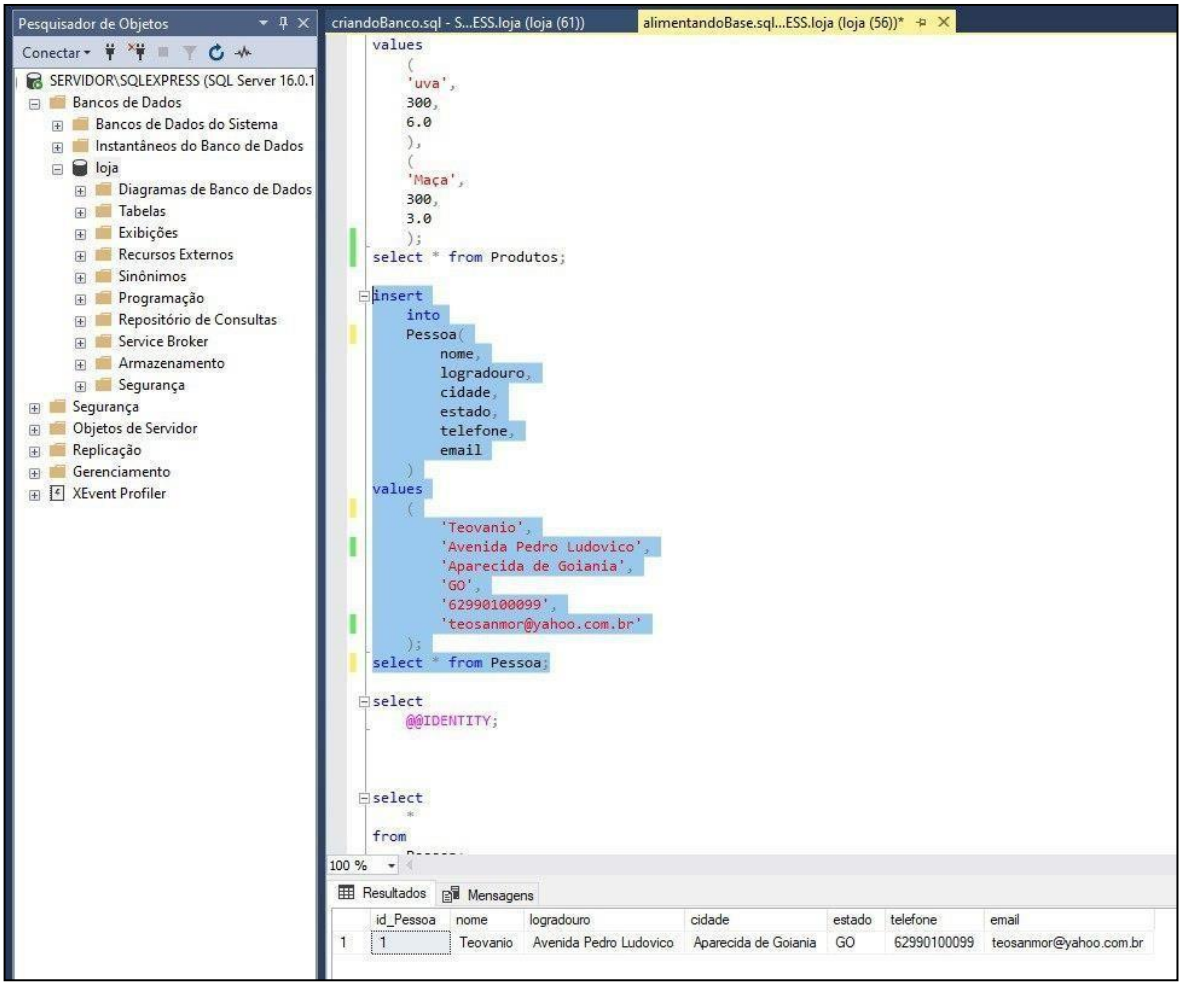


Figura 11 - Inserindo Pessoa.

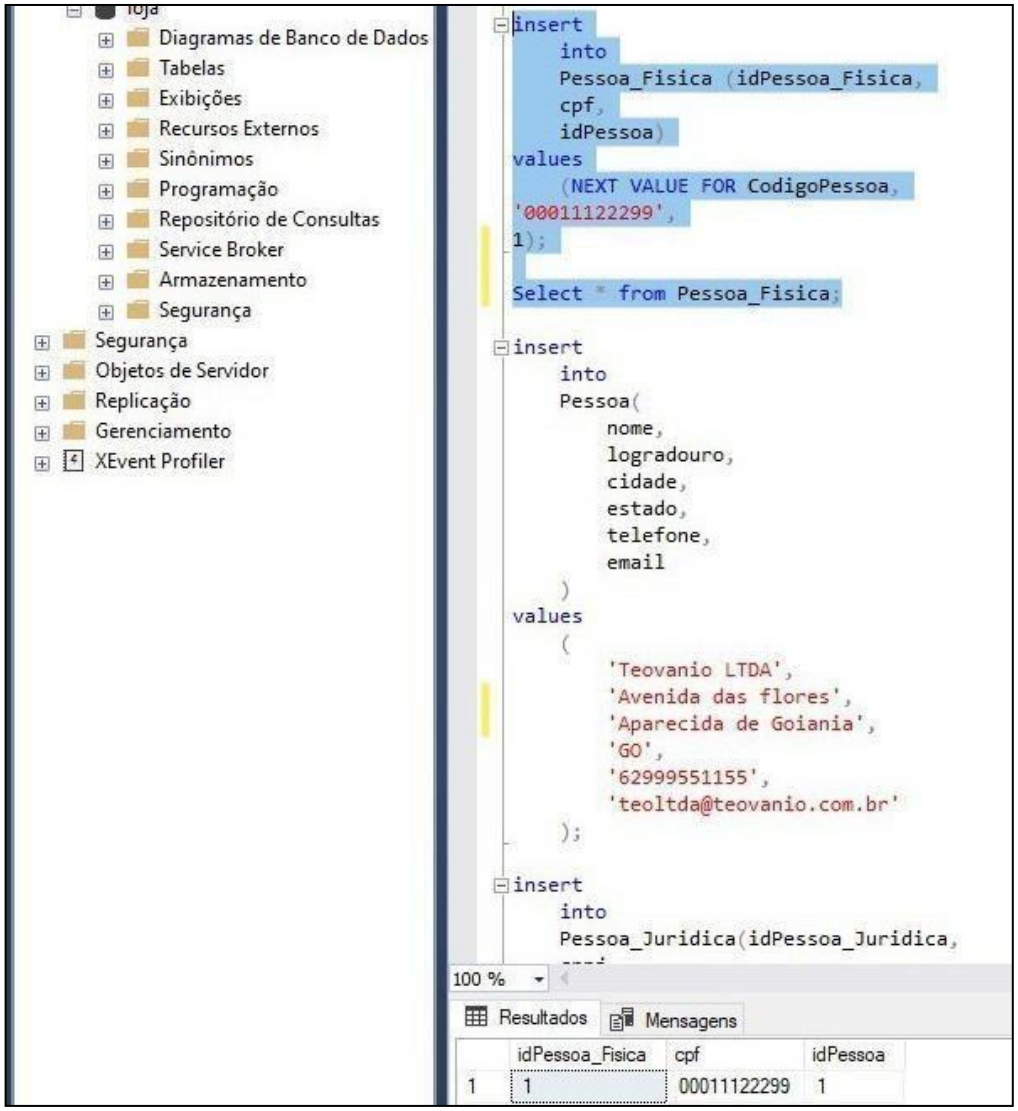


Figura 12 - Criação de Pessoa Fisica

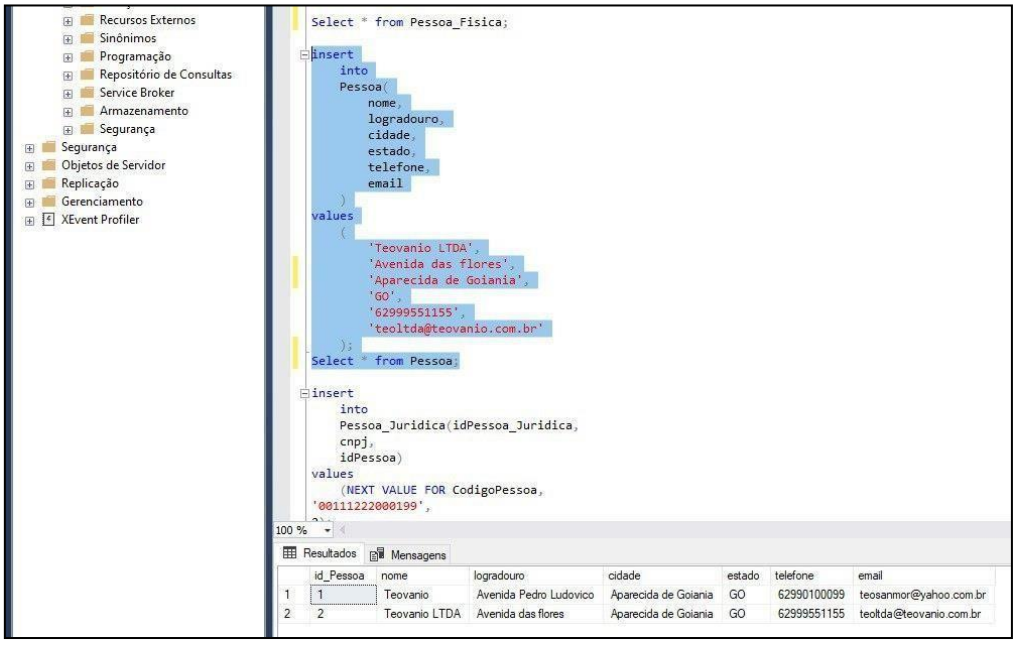


Figura 13 - Inclusão de nova Pessoa

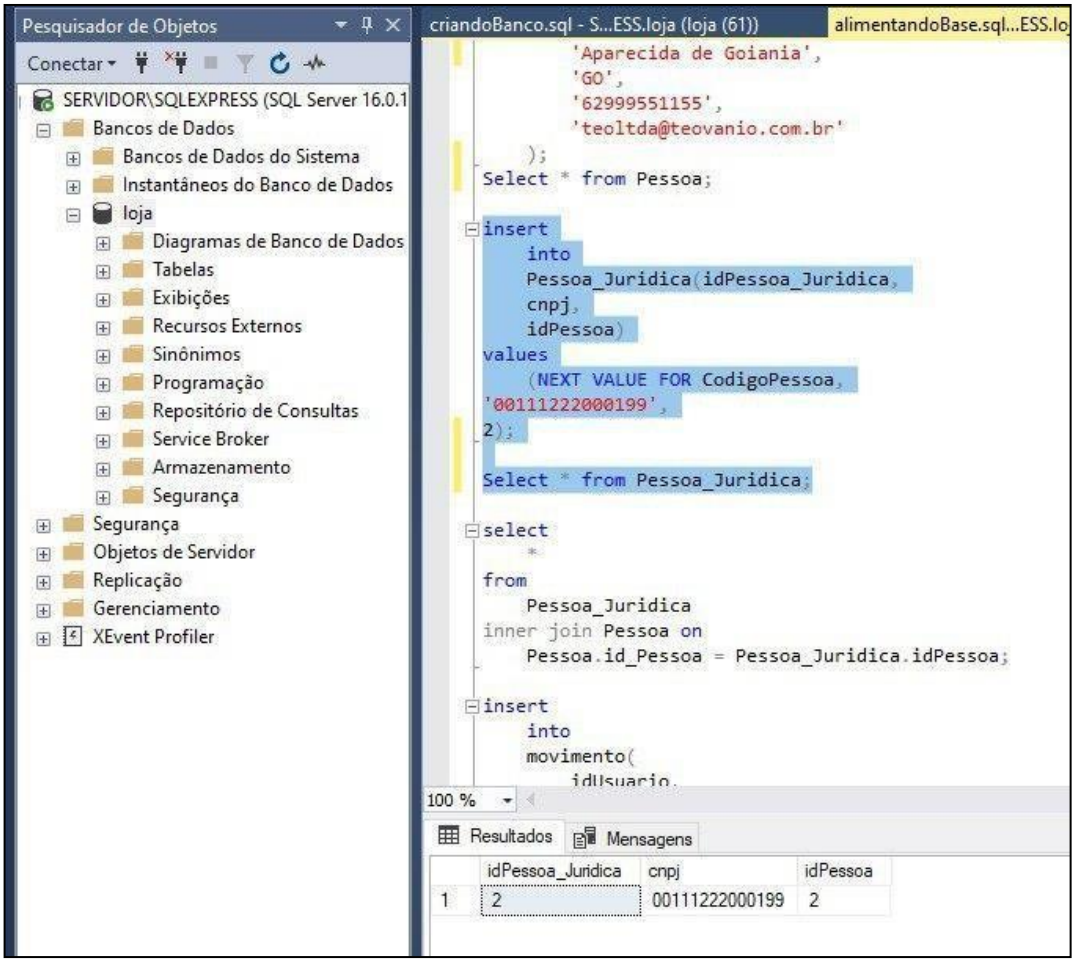
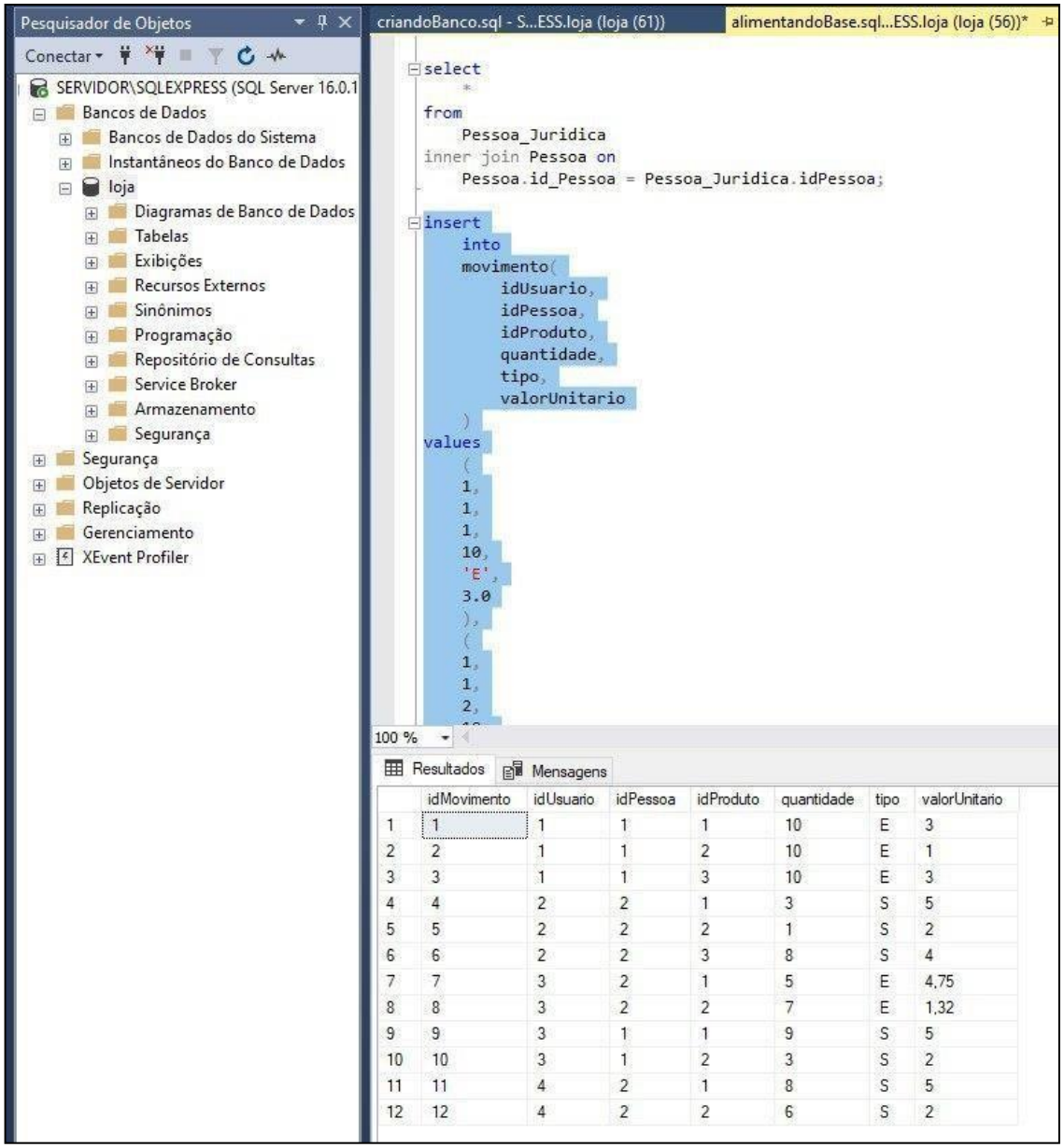


Figura 14 - Criação de Pessoa Jurídica



	idMovimento	idUsuario	idPessoa	idProduto	quantidade	tipo	valorUnitario
1	1	1	1	1	10	E	3
2	2	1	1	2	10	E	1
3	3	1	1	3	10	E	3
4	4	2	2	1	3	S	5
5	5	2	2	2	1	S	2
6	6	2	2	3	8	S	4
7	7	3	2	1	5	E	4,75
8	8	3	2	2	7	E	1,32
9	9	3	1	1	9	S	5
10	10	3	1	2	3	S	2
11	11	4	2	1	8	S	5
12	12	4	2	2	6	S	2

Figura 15 - Inclusão de movimento

5. Análise e Conclusão:

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

Sequence e identity são dois tipos de colunas de incremento automático em bancos de dados relacionais. No entanto, eles têm algumas diferenças importantes.

A principal diferença entre sequence e identity é que sequence é uma sequência de números inteiros que é gerenciada pelo banco de dados, enquanto identity é uma coluna de incremento automático que é armazenada na tabela.

Outra diferença importante é que o valor da sequência é independente do valor da coluna de incremento automático, enquanto o valor da coluna de incremento automático é igual ao valor da sequência.

Finalmente, sequence pode ser compartilhada por várias tabelas, enquanto identity só pode ser usada em uma tabela.

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

As chaves estrangerias são importantes para a consistência do banco porque ajudam a garantir que os dados em duas tabelas relacionadas sejam consistentes.

Uma chave estrangeira é um campo em uma tabela que faz referência a uma chave primária em outra tabela. Essa referência garante que um valor em uma tabela possa ser vinculado a um valor em outra tabela.

As chaves estrangeiras podem ajudar a evitar erros, como inserir um valor inválido em uma tabela. Isso ocorre porque a chave estrangeira restringe os valores que podem ser inseridos em uma tabela.

Podem ajudar a otimizar o desempenho: As chaves estrangeiras podem ajudar a otimizar o desempenho de consultas que envolvem várias tabelas. Isso ocorre porque as chaves estrangeiras permitem que o banco de dados use índices para acelerar as consultas.

Podem ajudar a garantir a integridade referencial: As chaves estrangeiras podem ajudar a garantir a integridade referencial. Isso significa que os dados nas tabelas relacionadas não podem ser inconsistentes.

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

Os operadores do SQL que pertencem à álgebra relacional são:

- Seleção: Retorna uma nova relação que contém apenas as tuplas que satisfazem uma condição especificada.
- Projeção: Retorna uma nova relação que contém apenas os atributos especificados de uma relação original.
- Junção: Retorna uma nova relação que combina as tuplas de duas ou mais relações, com base em uma condição especificada.
- Produto cartesiano: Retorna uma nova relação que contém todas as combinações possíveis de tuplas de duas ou mais relações.
- União: Retorna uma nova relação que contém todas as tuplas de duas ou mais relações, com base em uma condição especificada.
- Intersecção: Retorna uma nova relação que contém apenas as tuplas que estão em duas ou mais relações, com base em uma condição especificada.
- Diferença: Retorna uma nova relação que contém apenas as tuplas que estão em uma relação, mas não em outra relação.

Os operadores do SQL que são definidos no cálculo relacional são:

- Agregação: Operadores de agregação, como **COUNT()**, **SUM()**, **AVG()** e **MIN()**, são usados para calcular valores agregados para uma relação.
- Quantificadores: Os quantificadores **EXISTS()** e **FORALL()** são usados para expressar condições sobre conjuntos de tuplas.

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

O agrupamento em consultas é uma operação que agrupa os dados de uma tabela em grupos com base em um ou mais critérios. Os dados de cada grupo são então resumidos usando funções de agregação, como **COUNT()**, **SUM()**, **AVG()** e **MIN()**.

O agrupamento é realizado usando a cláusula **GROUP BY**. A cláusula **GROUP BY** especifica as colunas que serão usadas para agrupar os dados. Por exemplo, a seguinte consulta agrupa os dados da tabela **Produtos** por **Categoria** e **Fornecedor**:

SQL
```
SELECT Categoria, Fornecedor, COUNT(*) AS Quantidade
FROM Produtos
GROUP BY Categoria, Fornecedor;
```

Essa consulta retornará uma tabela com duas colunas: `Categoria` e `Fornecedor`. Cada linha da tabela representará um grupo de produtos com a mesma categoria e fornecedor. A coluna `Quantidade` contém o número de produtos em cada grupo.

O requisito obrigatório para o agrupamento é que a tabela deve conter pelo menos uma coluna que possa ser usada para agrupar os dados. Se a tabela não tiver nenhuma coluna que possa ser usada para agrupar os dados, a consulta retornará um erro.

Referências:

https://www.devmedia.com.br/modelagem-1-n-ou-n-n/38894

https://www.treinaweb.com.br/blog/relacionamento-1-1-1-n-e-n-n-com-django

https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/concepts/relational_databases_and_object_orientation_1C67069E.html

https://www.facom.ufu.br/~ilmerio/sbd20141/sbd2modeloER.pdf

https://learn.microsoft.com/pt-br/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16

https://awari.com.br/guia-completo-como-usar-o-sql-server-management-studio-para-otimizar-seu-trabalho/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Guia%20Completo:%20Como%20Usar%20o%20Sql%20Server%20Management%20Studio%20para%20Otimizar%20Seu%20Trabalho

http://www.linhadecodigo.com.br/artigo/3403/diferencas-entre-sequences-x-identity-no-microsoft-sql-server-2012.aspx

linkedin.com/pulse/explorando-os-diferentes-tipos-de-chaves-em-bancos-cordeiro-de-sousa/?originalSubdomain=pt

https://www.luis.blog.br/chave-primaria-chave-estrangeira-e-candidata.html

https://pt.wikipedia.org/wiki/%C3%81lgebra_relacional

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.facom.ufu.br/~ilmerio/sbd20141/sbd8algebraEcalculo.pdf