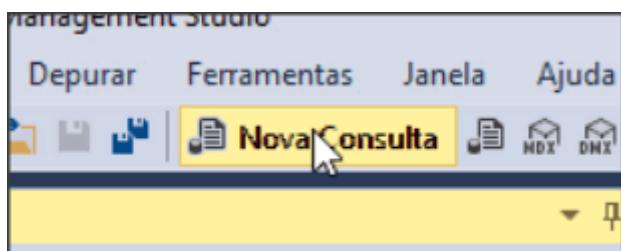


---

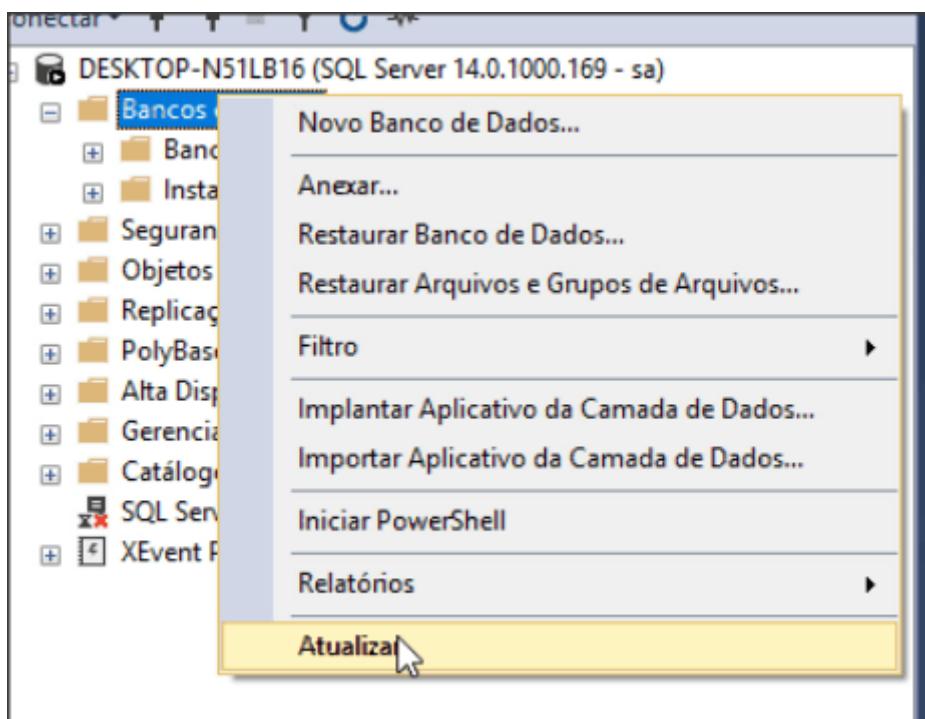
1) No **SQL Server Management Studio 2017**, conecte-se ao seu servidor, e clique sobre **Nova Consulta**:



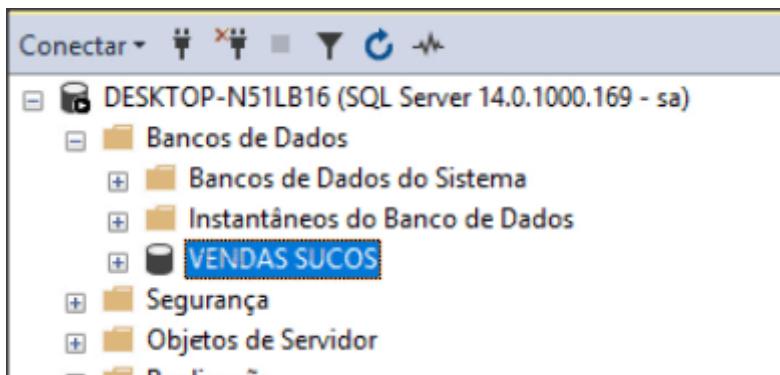
2) Digite o comando abaixo e execute-o:

```
CREATE DATABASE [VENDAS_SUCOS]
```

3) Volte à pasta **Banco de Dados** e clique com o botão da direita do mouse sobre ela, escolhendo a opção **Atualiza**:



4) Você pode ver o banco criado:



5) Execute comandos mais complexos para criação da base de dados. Digite o exemplo abaixo, mas antes certifique-se que você tenha criado em sua máquina o subdiretório **C:\TEMP\DATA**:

```
CREATE DATABASE [VENDAS SUCOS 2]
ON (NAME = VENDAS_SUCOS_2_DAT,
FILENAME = 'C:\TEMP\DATA\VENDAS_SUCOS_02.MDF',
SIZE = 10,
MAXSIZE = 50,
FILEGROWTH = 5)
LOG ON (NAME = VENDAS_SUCOS_2_LOG,
FILENAME = 'C:\TEMP\DATA\VENDAS_SUCOS_02.LDF',
SIZE = 10,
MAXSIZE = 50,
FILEGROWTH = 5)
```

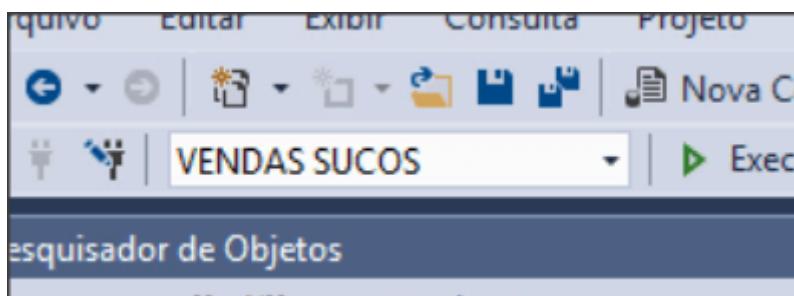
6) Verifique os arquivos criados em **C:\TEMP\DATA**:

Nome	Data de modificaç...	Tipo	Tamanho
VENDAS_SUCOS_02.LDF	27/04/2018 13:00	SQL Server Database...	10.240 KB
VENDAS_SUCOS_02.MDF	27/04/2018 13:00	SQL Server Database...	10.240 KB

7) Para apagar o banco de dados, basta executar o comando **DROP**, como mostrado abaixo:

```
DROP DATABASE [VENDAS SUCOS 2]
```

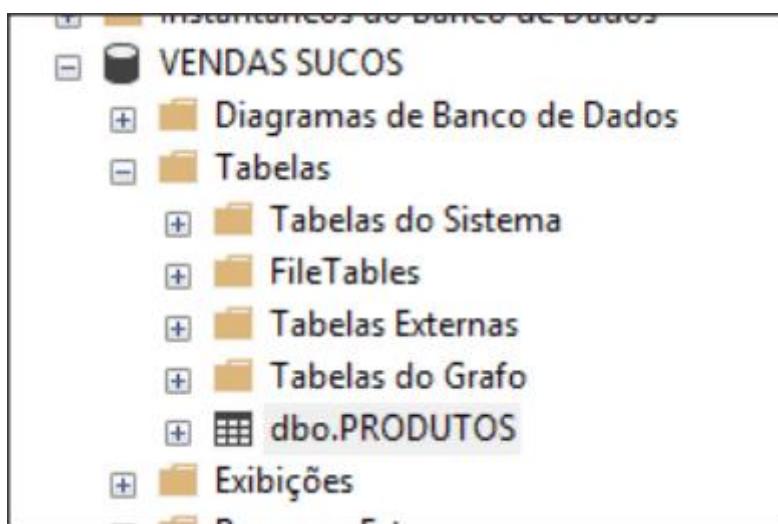
8) Certifique-se que você esteja acessando a base **VENDAS SUCOS**:



9) Para criar a tabela de produtos, conforme a especificação do diagrama de entidades e relacionamentos digite:

```
CREATE TABLE [PRODUTOS]
([CÓDIGO] VARCHAR(10) NOT NULL,
[DESCRITOR] VARCHAR(100) NULL,
[SABOR] VARCHAR(50) NULL,
[TAMANHO] VARCHAR(50) NULL,
[EMBALAGEM] VARCHAR(50) NULL,
[PREÇO LISTA] FLOAT NULL,
CONSTRAINT [PK_PRODUTOS]
PRIMARY KEY CLUSTERED ([CÓDIGO]))
```

10) Execute o comando e atualize a lista de dados da árvore à esquerda no Management Studio. Veja se a tabela foi criada:

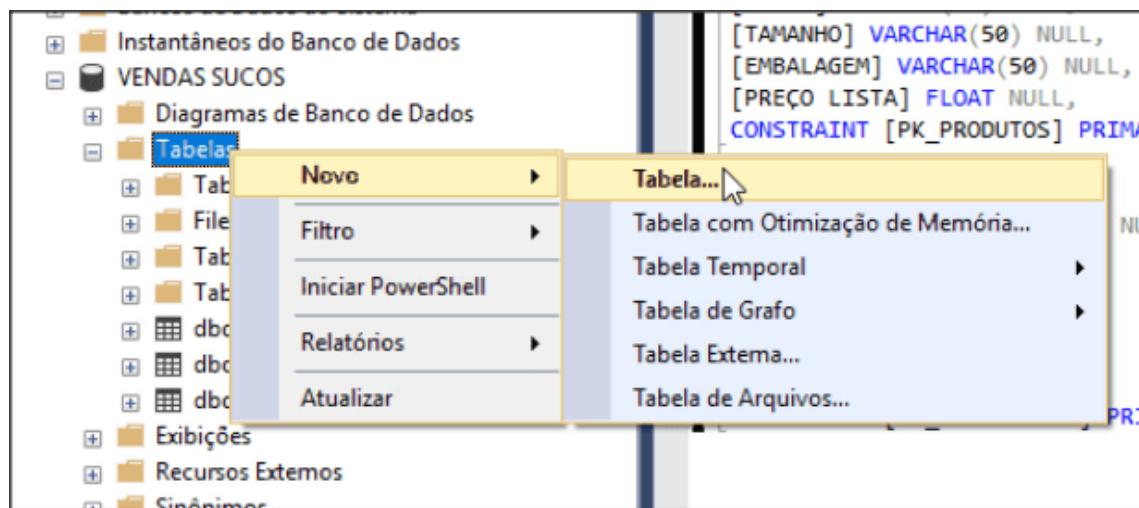


11) Repita o processo, criando a tabela de vendedores, digitando o comando abaixo:

```
CREATE TABLE [VENDEDORES]
([MATRÍCULA] VARCHAR(5) NOT NULL,
[NOME] VARCHAR(100) NULL,
[BAIRRO] VARCHAR(50) NULL,
[COMISSÃO] FLOAT NULL,
[DATA ADMISSÃO] DATE NULL,
[FÉRIAS] BIT NULL,
CONSTRAINT [PK_VENDEDORES]
```

```
PRIMARY KEY CLUSTERED ([MATRÍCULA]) )
```

- 12) Para criar a tabela de clientes pelo assistente, clique com o botão da direita do mouse sobre **Tabelas** e escolha as opções **Novo --> Tabela**:



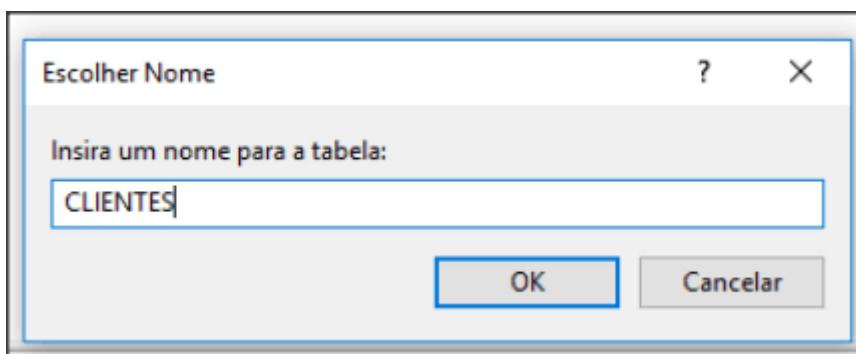
- 13) No grid apresentado, digite os campos da nova tabela de clientes:

Nome da Coluna	Tipo de Dados	Permitir Nul...
CPF	varchar(11)	<input type="checkbox"/>
NOME	varchar(100)	<input checked="" type="checkbox"/>
ENDERECO	varchar(150)	<input checked="" type="checkbox"/>
BAIRRO	varchar(50)	<input checked="" type="checkbox"/>
CIDADE	varchar(50)	<input checked="" type="checkbox"/>
ESTADO	varchar(50)	<input checked="" type="checkbox"/>
CEP	varchar(8)	<input checked="" type="checkbox"/>
[DATA NASCIMENTO]	date	<input checked="" type="checkbox"/>
IDADE	int	<input checked="" type="checkbox"/>
SEXO	varchar(1)	<input checked="" type="checkbox"/>
[LIMITE CRÉDITO]	float	<input checked="" type="checkbox"/>
[VOLUME COMPRA]	float	<input checked="" type="checkbox"/>
[PRIMEIRA COMPRA]	bit	<input checked="" type="checkbox"/>

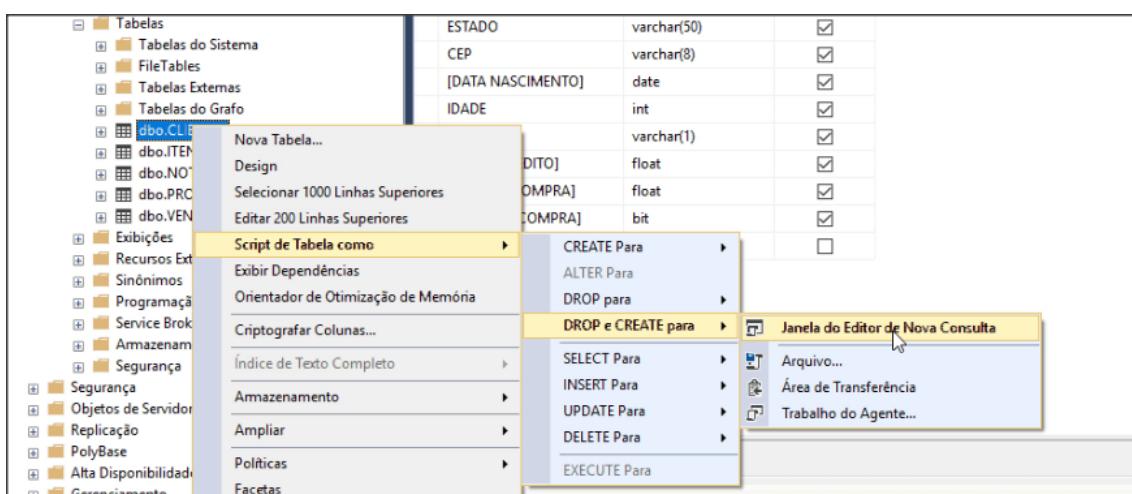
- 14) Clique com o botão da direita do mouse sobre o primeiro campo **CPF** e selecione a opção do menu **Definir Chave Primária**:

The screenshot shows the SSMS interface with the title bar 'DESKTOP-N51LB16....COS - dbo.Table\_1\*'. In the center, there's a table definition grid with columns 'Nome da Coluna', 'Tipo de Dados', and 'Permitir Nul...'. The first column has the value 'CPF'. A context menu is open at the top of this column, with the 'Definir Chave Primária' option highlighted.

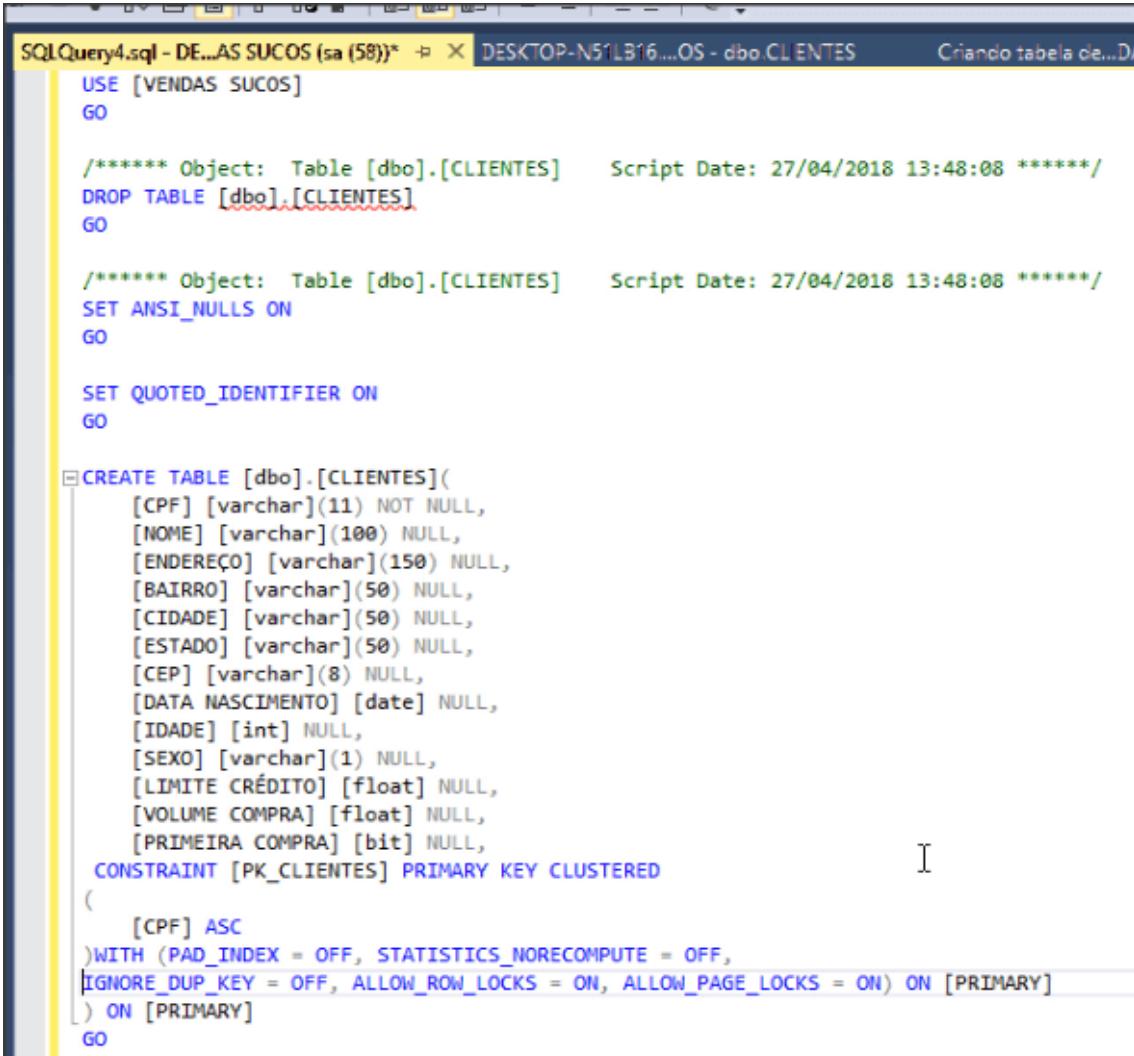
15) Clique no botão **Salvar** e digite o nome da tabela (**CLIENTES**):



16) Clique com o botão da direita do mouse sobre o nome da tabela **CLIENTES** e escolha a opção **Script de Tabela como --> DROP e CREATE para --> Janela do Editor de Nova Consulta**:



17) Haverá uma janela com o script gerado pelo SQL Server, para criação de uma tabela com as propriedades da tabela **CLIENTES**:



```

SQLQuery4.sql - DE...AS SUCOS (sa (58)) * DESKTOP-N51LB16...OS - dbo.CLIENTES Criando tabela de...
USE [VENDAS_SUCOS]
GO

/******** Object: Table [dbo].[CLIENTES] Script Date: 27/04/2018 13:48:08 *****/
DROP TABLE [dbo].[CLIENTES]
GO

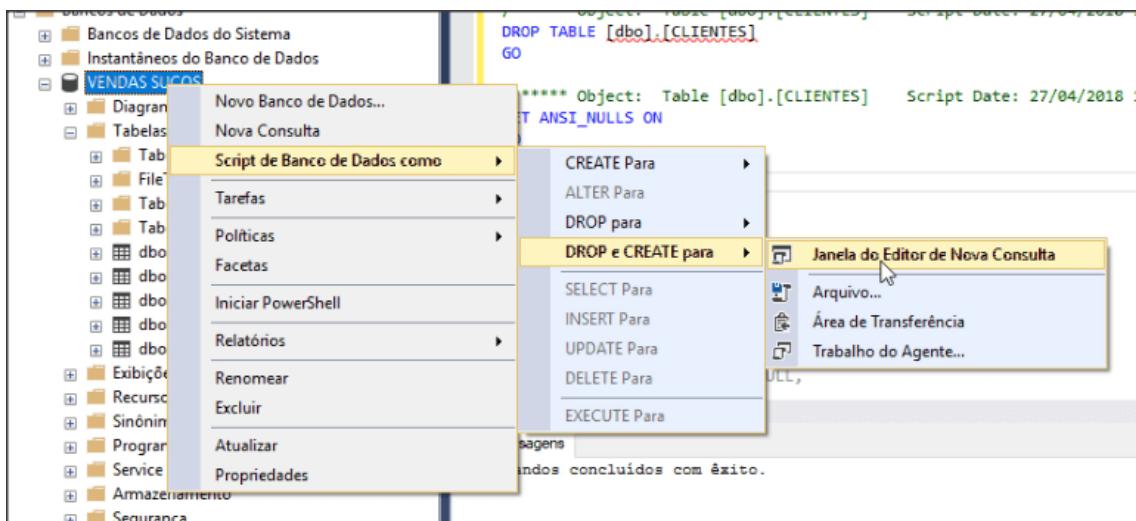
/******** Object: Table [dbo].[CLIENTES] Script Date: 27/04/2018 13:48:08 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[CLIENTES](
    [CPF] [varchar](11) NOT NULL,
    [NOME] [varchar](100) NULL,
    [ENDERECO] [varchar](150) NULL,
    [BAIRRO] [varchar](50) NULL,
    [CIDADE] [varchar](50) NULL,
    [ESTADO] [varchar](50) NULL,
    [CEP] [varchar](8) NULL,
    [DATA NASCIMENTO] [date] NULL,
    [IDADE] [int] NULL,
    [SEXO] [varchar](1) NULL,
    [LIMITE CRÉDITO] [float] NULL,
    [VOLUME COMPRA] [float] NULL,
    [PRIMEIRA COMPRA] [bit] NULL,
    CONSTRAINT [PK_CLIENTES] PRIMARY KEY CLUSTERED
(
    [CPF] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

18) Você pode gerar o script de criação do banco de dados clicando com o botão da direita do mouse sobre a base de dados **VENDA\_SUCOS** e selecionando **Script de Banco de Dados** como --> **DROP e CREATE** para -> **Janela do Editor de Nova Consulta**:



19) Você terá:

```

USE [master]
GO

/********* Object: Database [VENDAS SUCOS]    Script Date: 27/04/2018 13:51:30 *****/
DROP DATABASE [VENDAS SUCOS]
GO

/********* Object: Database [VENDAS SUCOS]    Script Date: 27/04/2018 13:51:30 *****/
CREATE DATABASE [VENDAS SUCOS]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'VENDAS SUCOS', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\VENDAS SUCOS.mdf' , SIZE
LOG ON
( NAME = N'VENDAS SUCOS_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\VENDAS SUCOS_log.ldf'
GO

ALTER DATABASE [VENDAS SUCOS] SET COMPATIBILITY_LEVEL = 140
GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [VENDAS SUCOS].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO

ALTER DATABASE [VENDAS SUCOS] SET ANSI_NULL_DEFAULT OFF
GO

ALTER DATABASE [VENDAS SUCOS] SET ANSI_NULLS OFF
GO

ALTER DATABASE [VENDAS SUCOS] SET ANSI_PADDING OFF
GO

ALTER DATABASE [VENDAS SUCOS] SET ANSI_WARNINGS OFF
GO

ALTER DATABASE [VENDAS SUCOS] SET ARITHABORT OFF
GO

ALTER DATABASE [VENDAS SUCOS] SET AUTO_CLOSE OFF
GO

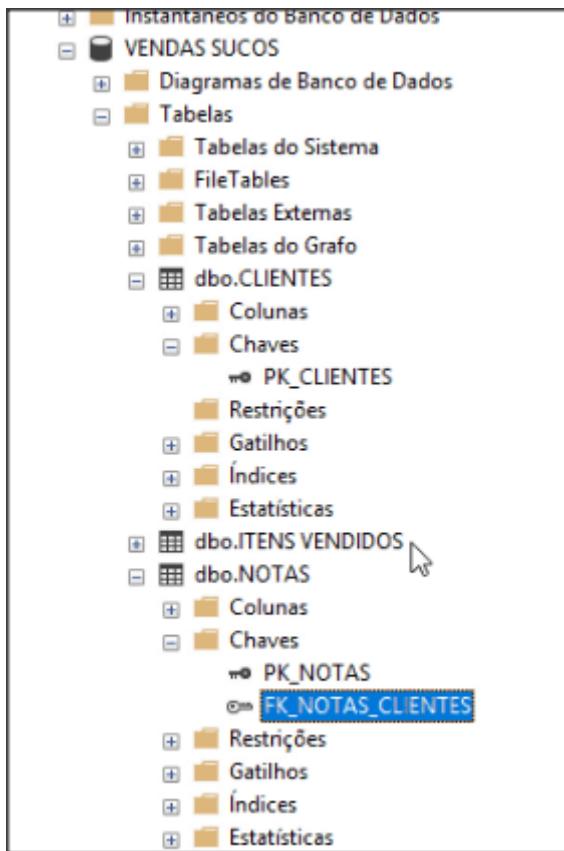
```

Não execute estes comandos para não perder o seu banco de dados.

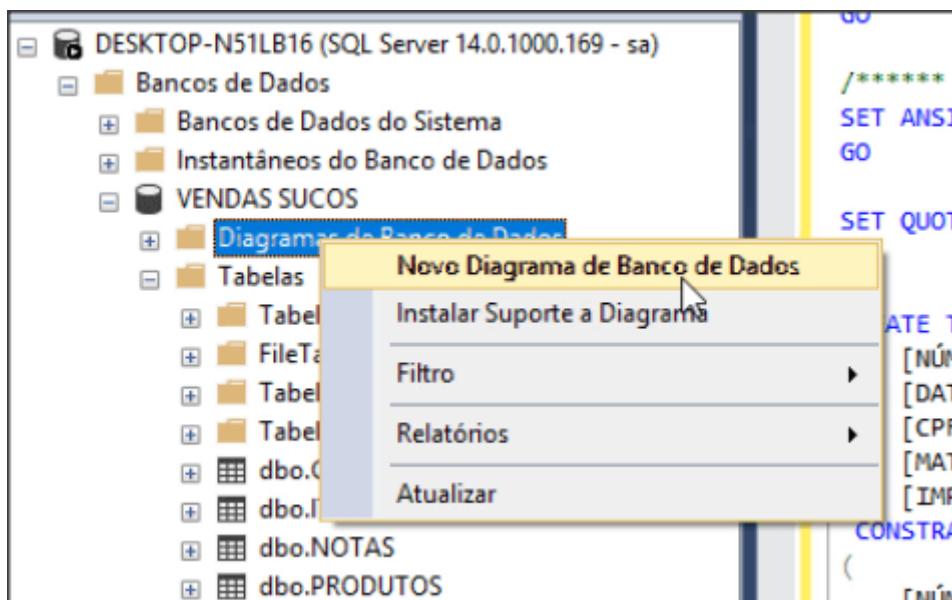
20) Através do campo **CPF**, crie um relacionamento entre a tabela de clientes e a de itens de notas fiscais. Para isso, abra uma nova consulta no **SQL Server Management Studio**, certificando-se que o banco **VENDAS SUCOS** esteja selecionado, e digite o comando abaixo:

```
ALTER TABLE [NOTAS] WITH CHECK ADD CONSTRAINT [FK_NOTAS_CLIENTES]
FOREIGN KEY([CPF]) REFERENCES [CLIENTES] ([CPF])
```

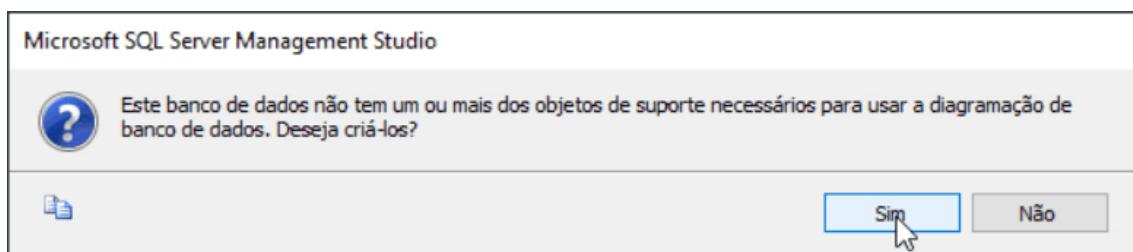
21) Execute este comando. Depois verifique as restrições criadas nas tabelas **CLIENTES** e **NOTAS**, no item da árvore de objetos chamado **Chaves**:



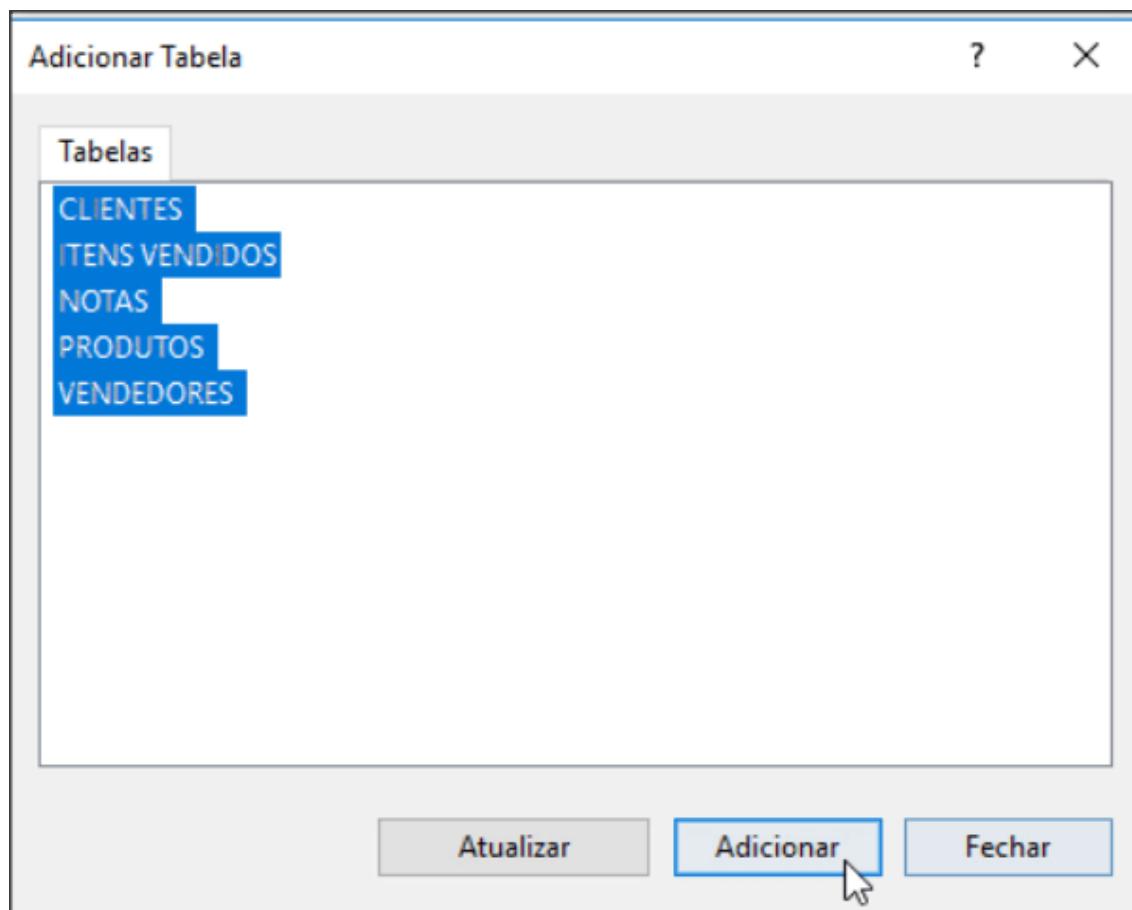
22) Crie um novo relacionamento, agora usando o diagrama do SQL Server 2017. Para isso, clique com o botão da direita do mouse sobre a pasta **Diagramas de Banco de Dados**, abaixo do banco de dados **VENDAS SUCOS**, e selecione a opção **Novo Diagrama de Banco de Dados**:



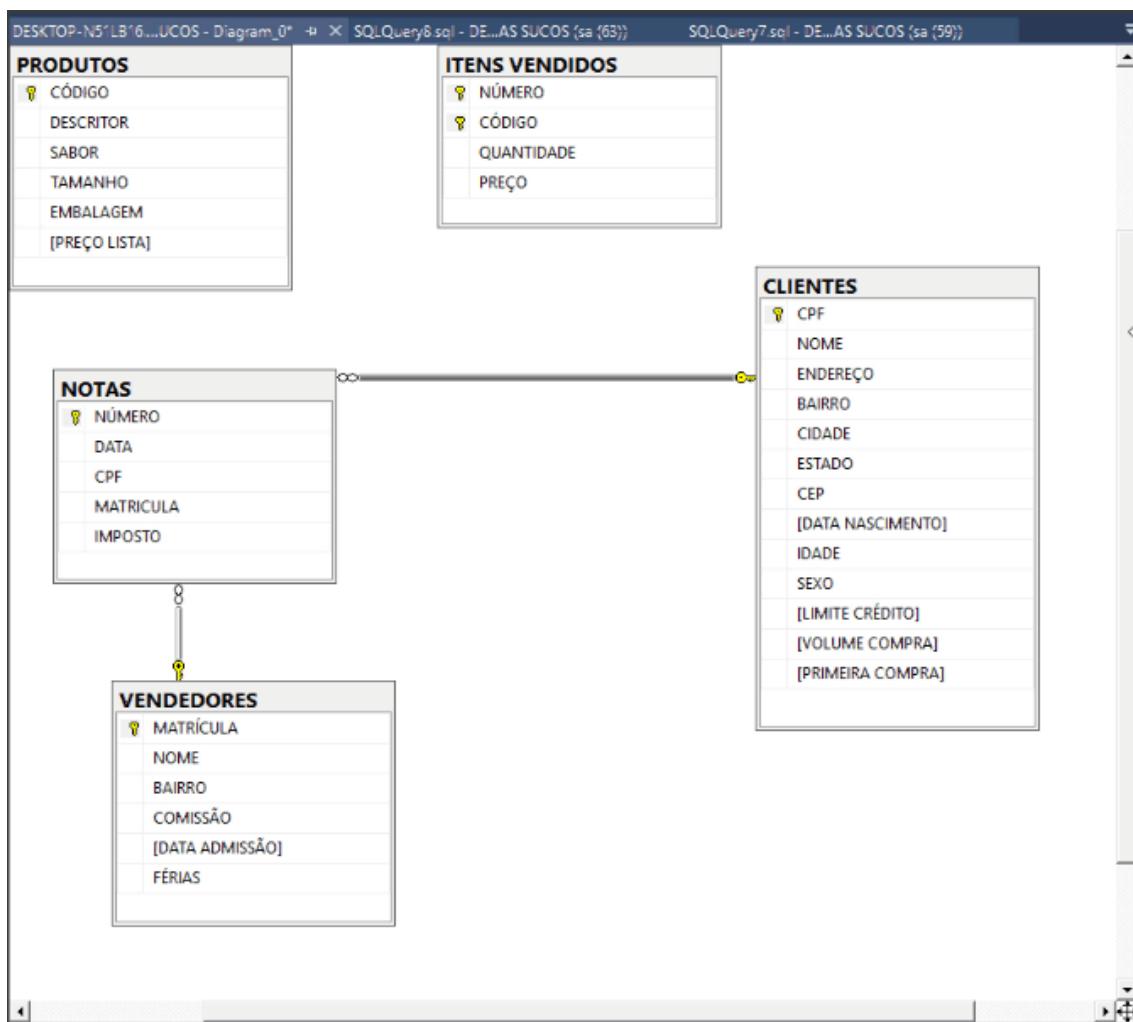
23) Clique em **Sim** na caixa de diálogo abaixo:



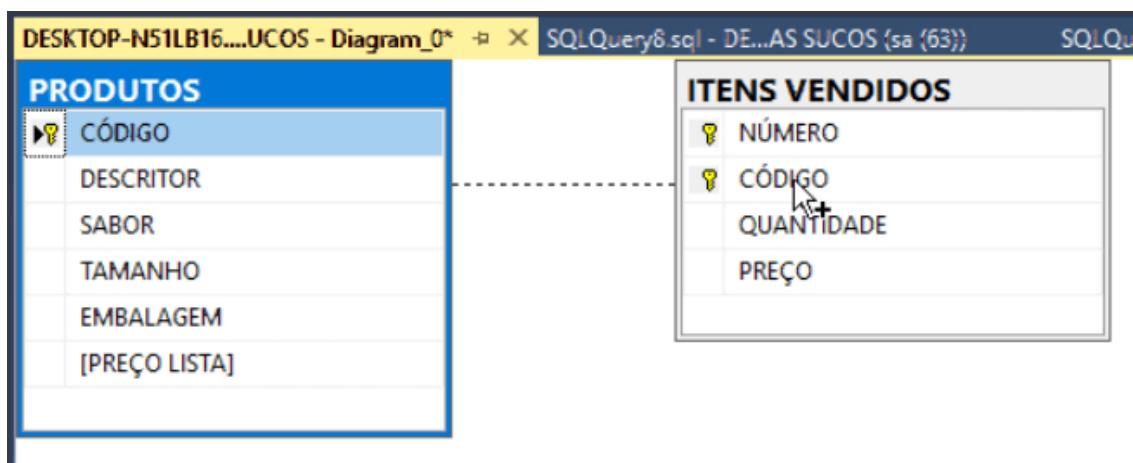
24) Selecione todas as tabelas e depois clique no botão **Adicionar**:



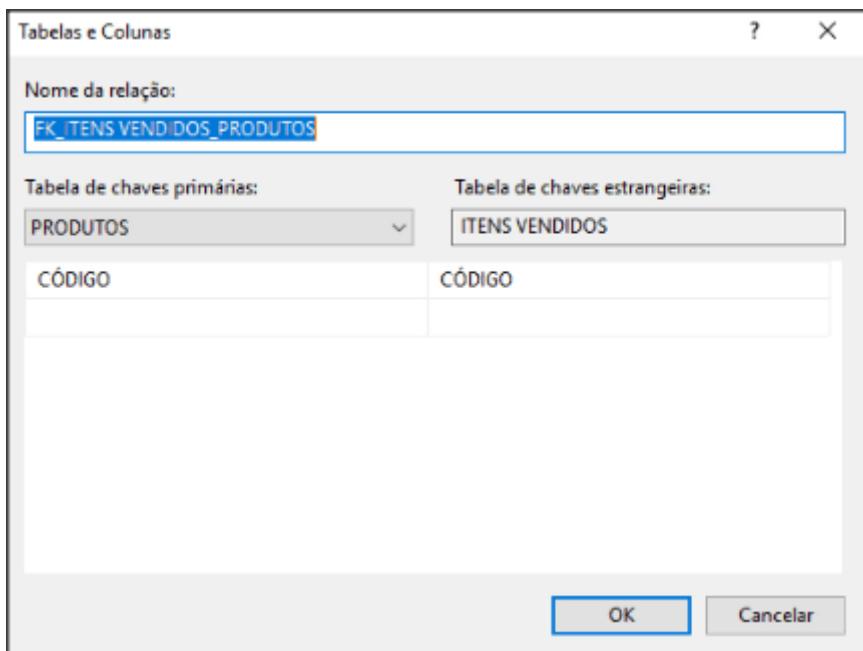
25) Organize o esquema visual das tabelas a fim de poder observar em um único diagrama:



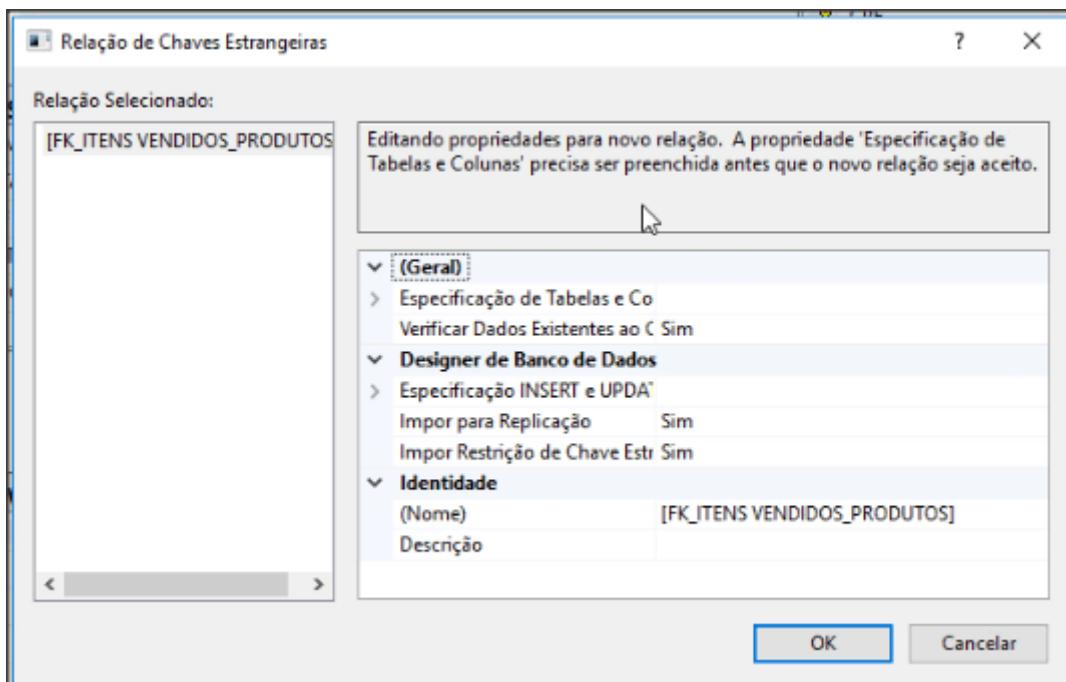
26) Com o mouse, clique no campo **Código**, da tabela **PRODUTOS** e, mantendo o dedo no clique no mouse, sem soltá-lo, arraste-o para o campo **Código**, da tabela **ITENS VENDIDOS**:



27) Após a associação, ao soltar o dedo do botão do mouse, uma caixa de diálogo será mostrada, onde você pode ver a associação que será construída. Depois clique em **OK**:



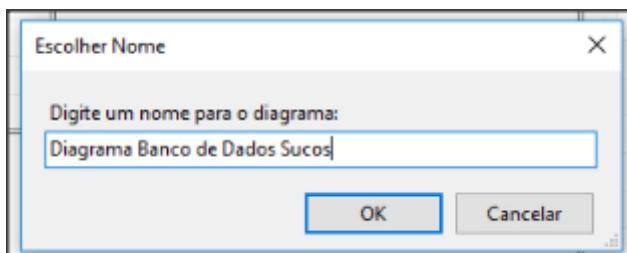
28) Na próxima tela, você pode ver as propriedades do relacionamento. Clique novamente em **OK**:



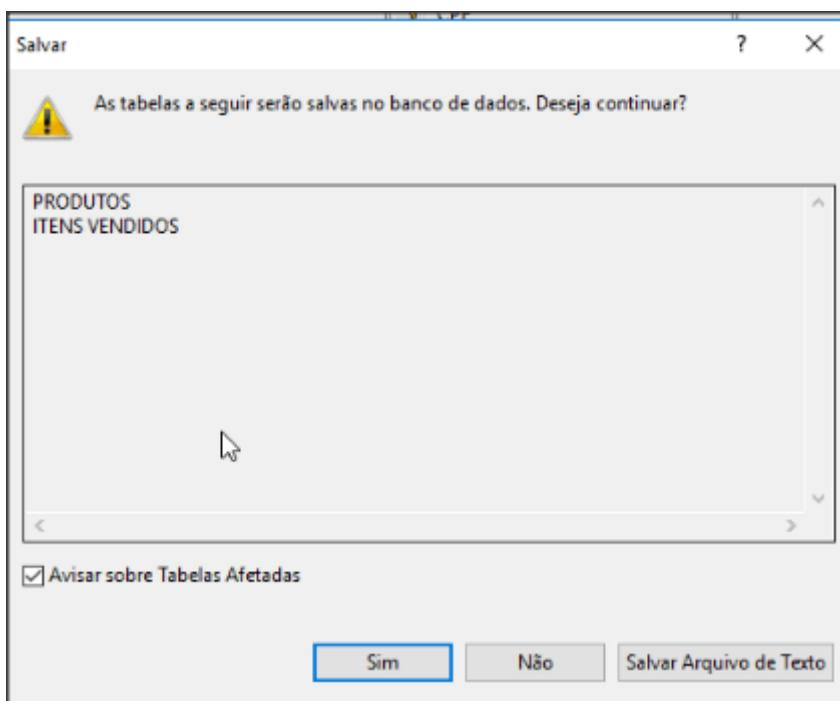
29) Para salvar a criação do relacionamento, clique no botão **Salvar**:



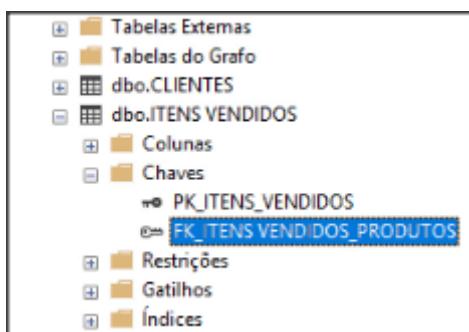
30) O **SQL Server Management Studio** irá perguntar o nome do diagrama, coloque um nome de sua preferência e clique em **OK**:



31) Confirme a alteração que será executada na tabela:



32) Por fim, veja o relacionamento criado:



Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

---

- 1) Vá no SQL Server Management Studio e crie uma nova consulta associada à base de dados **VENDAS\_SUCOS**. Digite o comando abaixo:

```
INSERT INTO [PRODUTOS]
([CÓDIGO], [DESCRITOR], [SABOR],
[TAMANHO], [EMBALAGEM], [PREÇO LISTA])
VALUES ('1040107', 'Light - 350 ml - Melancia',
'Melancia', '350 ml', 'Lata', 4.56)
```

- 2) Execute mais de um comando de inclusão:

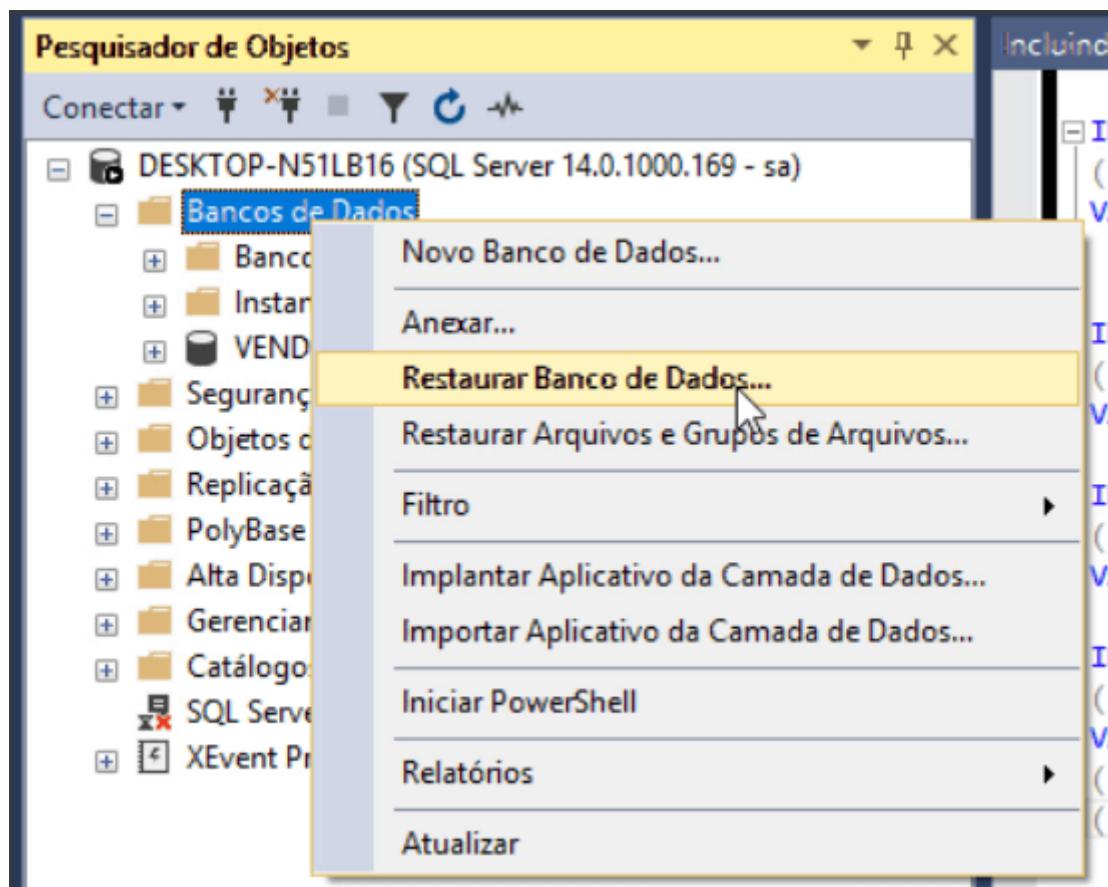
```
INSERT INTO [PRODUTOS]
([CÓDIGO], [DESCRITOR], [SABOR],
[TAMANHO], [EMBALAGEM], [PREÇO LISTA])
VALUES ('1040108', 'Light - 350 ml - Graviola',
'Graviola', '350 ml', 'Lata', 4.00)
INSERT INTO [PRODUTOS]
([CÓDIGO], [DESCRITOR], [SABOR], [TAMANHO],
[EMBALAGEM], [PREÇO LISTA])
VALUES ('1040109', 'Light - 350 ml - Açaí',
'Açaí', '350 ml', 'Lata', 5.60)
```

Depois de selecionar estes dois comandos e executá-los, as duas inclusões serão efetuadas.

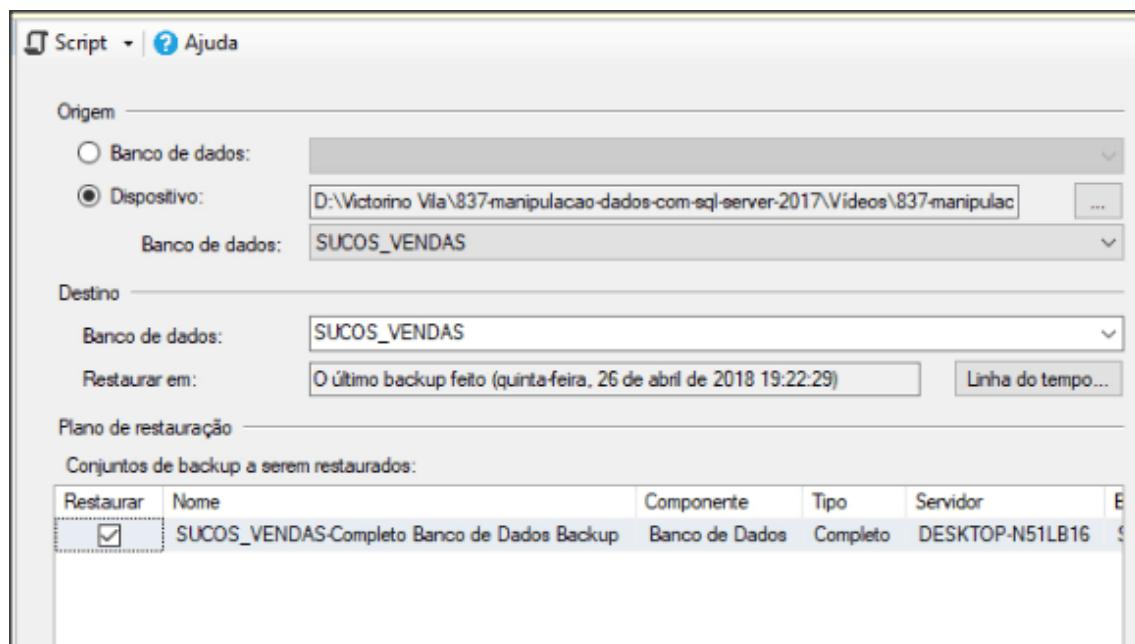
- 3) Você também pode incluir mais de um registro em um mesmo comando, como mostrado abaixo:

```
INSERT INTO [PRODUTOS]
([CÓDIGO], [DESCRITOR], [SABOR], [TAMANHO],
[EMBALAGEM], [PREÇO LISTA])
VALUES
('1040110', 'Light - 350 ml - Jaca', 'Jaca',
'350 ml', 'Lata', 6.00),
('1040111', 'Light - 350 ml - Manga', 'Manga',
'350 ml', 'Lata', 3.50)
```

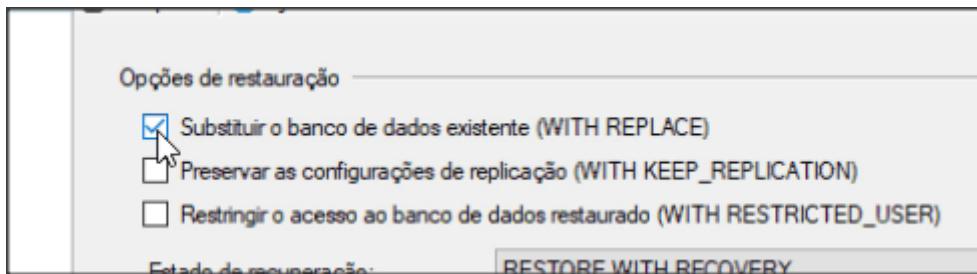
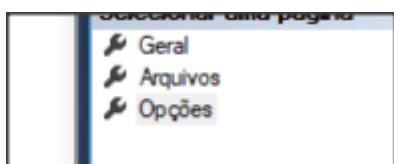
- 4) Você pode mesclar o comando **INSERT** com o comando **SELECT** para incluir diversos registros de uma só vez. Para isso, carregue o backup de um outro banco de dados, que será usado como fonte. Clique com o botão da direita do mouse sobre a pasta **Banco de Dados** e selecione a opção **Restaurar Banco de Dados**:



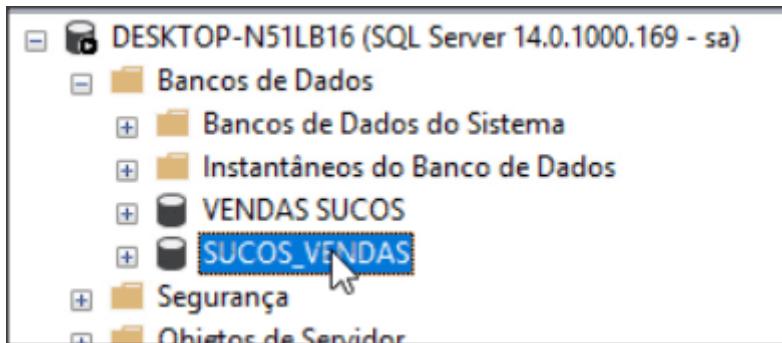
5) Clique na opção **Dispositivo** e selecione o arquivo **SUCOS\_VENDAS.BAK**, que você pode baixar [aqui](#):



6) Na seção **Opções**, selecione **Substituir o banco de dados existente (WITH REPLACE)**. Depois clique em **OK**:



7) Com isso, você terá o banco de dados recuperado com o nome **SUCOS\_VENDAS**. O banco **VENDAS\_SUCOS** é o banco de dados que você está construindo e o **SUCOS\_VENDAS** fará o papel de banco de dados fonte:



8) Crie uma nova consulta associada à base **VENDAS\_SUCOS**:



9) Para se referir à tabela que está em outro banco de dados, basta, no nome da tabela, colocar o nome do banco de dados, como mostrado abaixo:

```
SELECT * FROM [SUCOS_VENDAS] . [dbo] . [TABELA DE PRODUTOS]
```

Assim você pode fazer uma seleção na tabela **Tabela de Produtos**, que faz parte do banco fonte.

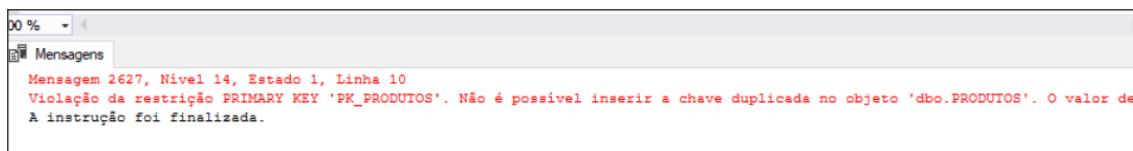
10) Você pode selecionar os dados da tabela fonte (**Tabela de Produtos**) para incluir na tabela destino (**Produtos**). Mas, para isso, sua seleção na origem tem que resultar em nomes de campos iguais aos contidos na tabela destino:

```
SELECT [CODIGO DO PRODUTO] AS [CÓDIGO],
       [NOME DO PRODUTO] AS [DESCRIPTOR], [SABOR], [TAMANHO],
       [EMBALAGEM], [PREÇO DE LISTA] AS [PREÇO LISTA]
  FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
```

11) Assim você pode executar o comando abaixo:

```
INSERT INTO [PRODUTOS] ([CÓDIGO], [DESCRIPTOR], [SABOR],
                      [TAMANHO], [EMBALAGEM], [PREÇO LISTA])
SELECT [CODIGO DO PRODUTO] AS [CÓDIGO],
       [NOME DO PRODUTO] AS [DESCRIPTOR], [SABOR], [TAMANHO],
       [EMBALAGEM], [PREÇO DE LISTA] AS [PREÇO LISTA]
  FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
```

12) A execução deste comando acarretará em um erro de chave primária, pois o produto **1040107** já havia sido inserido na tabela destino em outra aula:



13) Quando um bloco completo é lido e um dos registros deste bloco apresentar um erro, todo o bloco é recusado no momento da gravação. Para isso, retire o produto **1040107** de nossa seleção:

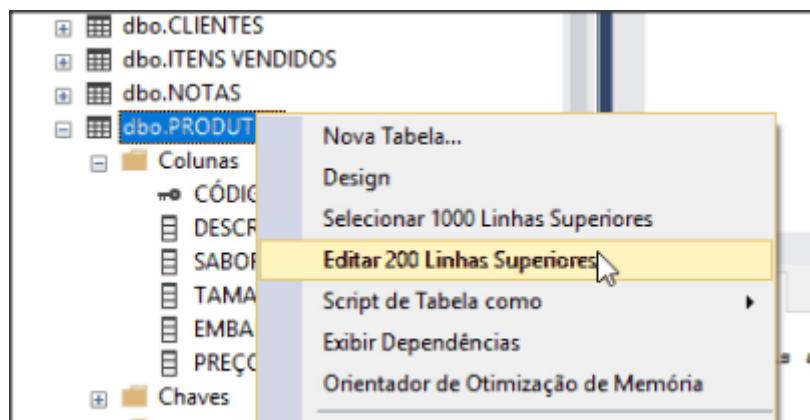
```
SELECT [CODIGO DO PRODUTO] AS [CÓDIGO],
       [NOME DO PRODUTO] AS [DESCRIPTOR], [SABOR], [TAMANHO],
       [EMBALAGEM], [PREÇO DE LISTA] AS [PREÇO LISTA]
  FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
 WHERE [CODIGO DO PRODUTO] <> '1040107'
```

14) Assim, executando o comando abaixo:

```
INSERT INTO [PRODUTOS] ([CÓDIGO], [DESCRIPTOR], [SABOR],
                      [TAMANHO], [EMBALAGEM], [PREÇO LISTA])
SELECT [CODIGO DO PRODUTO] AS [CÓDIGO],
       [NOME DO PRODUTO] AS [DESCRIPTOR], [SABOR], [TAMANHO],
       [EMBALAGEM], [PREÇO DE LISTA] AS [PREÇO LISTA]
  FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
 WHERE [CODIGO DO PRODUTO] <> '1040107'
```

Você terá todos os dados transferidos da tabela fonte para a tabela alvo.

15) Você pode digitar os dados diretamente na tabela. Para isso, clique com o botão da direita do mouse sobre a tabela **PRODUTOS** e selecione a opção **Editar 200 Linhas Superiores**:



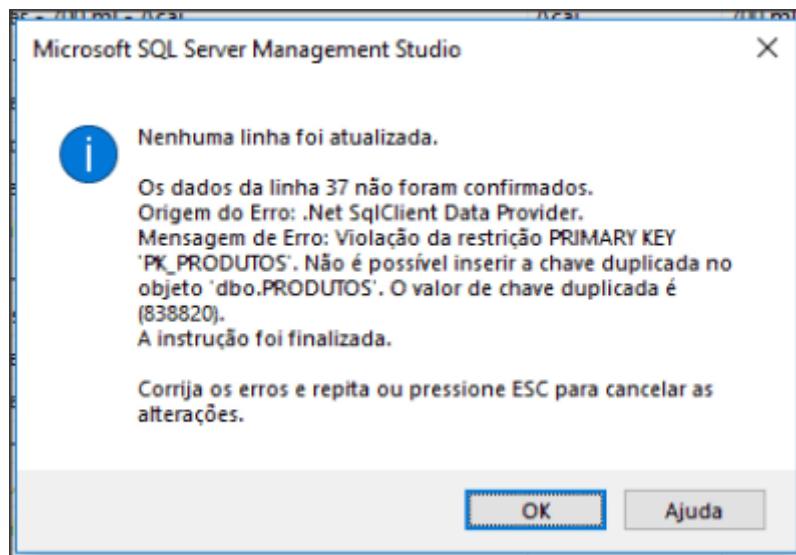
16) Você terá:

DESKTOP-N51LB16....S - dbo.PRODUTOS						
	CÓDIGO	DESCRITOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
▶	100089	Sabor da Mont...	Uva	700 ml	Garrafa	6,309
	1002334	Linha Citros - 1 ...	Lima/Limão	1 Litro	PET	7,004
	1002767	Videira do Cam...	Cereja/Maça	700 ml	Garrafa	8,41
	1004327	Videira do Cam...	Melância	1,5 Litros	PET	19,51
	1013793	Videira do Cam...	Cereja/Maça	2 Litros	PET	24,01
	1022450	Festival de Sab...	Açaí	2 Litros	PET	38,012
	1037797	Clean - 2 Litros ...	Laranja	2 Litros	PET	16,008
	1040107	Light - 350 ml - ...	Melância	350 ml	Lata	4,56
	1040108	Light - 350 ml - ...	Graviola	350 ml	Lata	4
	1040109	Light - 350 ml - ...	Açaí	350 ml	Lata	5,6
	1040110	Light - 350 ml - ...	Jaca	350 ml	Lata	6
	1040111	Light - 350 ml - ...	Manga	350 ml	Lata	3,5
	1041119	Linha Citros - 7...	Lima/Limão	700 ml	Garrafa	4,904
	1078680	Frescor do Verã...	Manga	470 ml	Garrafa	5,1795
	1086543	Linha Refrescan...	Manga	1 Litro	PET	11,0105
	1096818	Linha Refrescan...	Manga	700 ml	Garrafa	7,7105
	1101035	Linha Refrescan...	Morango/Limão	1 Litro	PET	9,0105
	229900	Pedaços de Fru...	Maça	350 ml	Lata	4,211
	231776	Festival de Sab...	Açaí	700 ml	Garrafa	13,312
	235653	Frescor do Verã...	Manga	350 ml	Lata	3,8595
	243083	Festival de Sab...	Maracujá	1,5 Litros	PET	10,512
	290478	Videira do Cam...	Melância	350 ml	Lata	4,56

17) Você pode digitar um registro diretamente na tabela. E os problemas de restrições (chaves primárias ou estrangeiras) serão alertados durante a digitação. Para ver isso, digite a seguinte informação:

```
CÓDIGO: 838820
DESCRITOR: Clean - 1,5 Litros - Jaca
SABOR: Jaca
TAMANHO: 1,5 Litros
EMBALAGEM: PET
PREÇO LISTA: 15,5
```

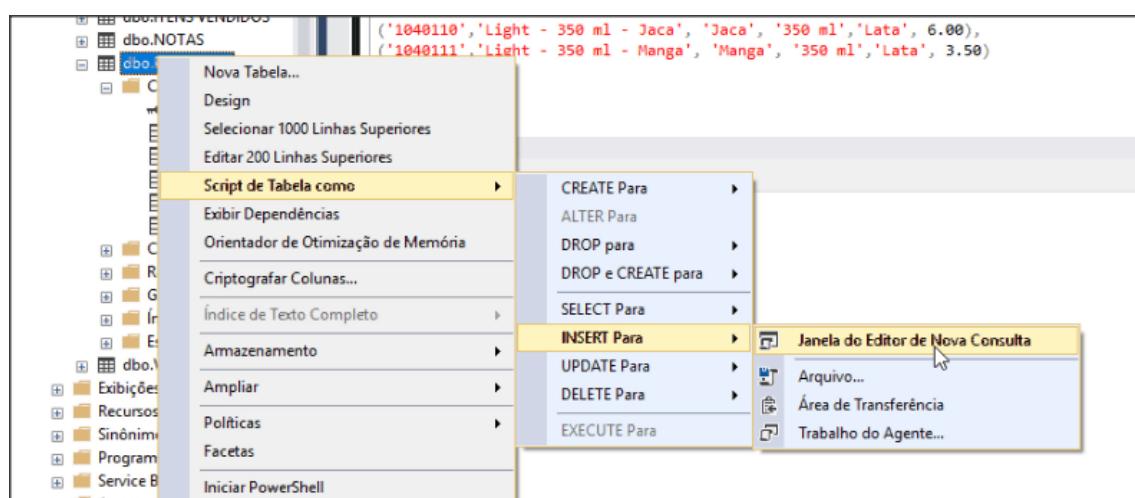
Ao confirmar, você terá o erro:



18) Corrija para o código **838821** e inclua o registro:

838820	Clean - 1,5 Litros - Açaí	Açaí	1,5 Litros	PET	13
838821	Clean - 1,5 Litros - Jaca	Jaca	1,5 Litros	PET	13,5
NULL	NULL	NULL	NULL	NULL	NULL

19) Você pode também pedir para que o **SQL Server Management Studio** te ajude a escrever o comando **INSERT** de uma tabela. Para isso, clique com o botão da direita do mouse sobre a tabela **PRODUTOS** e escolha a opção **Script de Tabela como --> INSERT Para Janela do Editor de Nova Consulta**:



20) Você o comando mostrado abaixo:

```

USE [VENDAS SUCOS]
GO

INSERT INTO [dbo].[PRODUTOS]
([CÓDIGO]
,[DESCRITOR]
,[SABOR]
,[TAMANHO]
,[EMBALAGEM]
,[PREÇO LISTA])
VALUES
(,'CÓDIGO, varchar(10),'
,'DESCRITOR, varchar(100),'
,'SABOR, varchar(50),'
,'TAMANHO, varchar(50),'
,'EMBALAGEM, varchar(50),'
,'PREÇO LISTA, float,')

GO

```

21) Agora basta substituir os dados após **VALUES** para incluir um novo registro na tabela:

```

USE [VENDAS SUCOS]
GO

INSERT INTO [dbo].[PRODUTOS]
([CÓDIGO]
,[DESCRITOR]
,[SABOR]
,[TAMANHO]
,[EMBALAGEM]
,[PREÇO LISTA])
VALUES
('1040120'
,'Light 350 ml - Tangerina'
,'Tangerina'
,'350 ml'
,'Lata'
,5.50)

GO

```

22) Faça o download do arquivo **Vendedores.csv** [aqui](#) e verifique seu conteúdo através de um editor de texto:

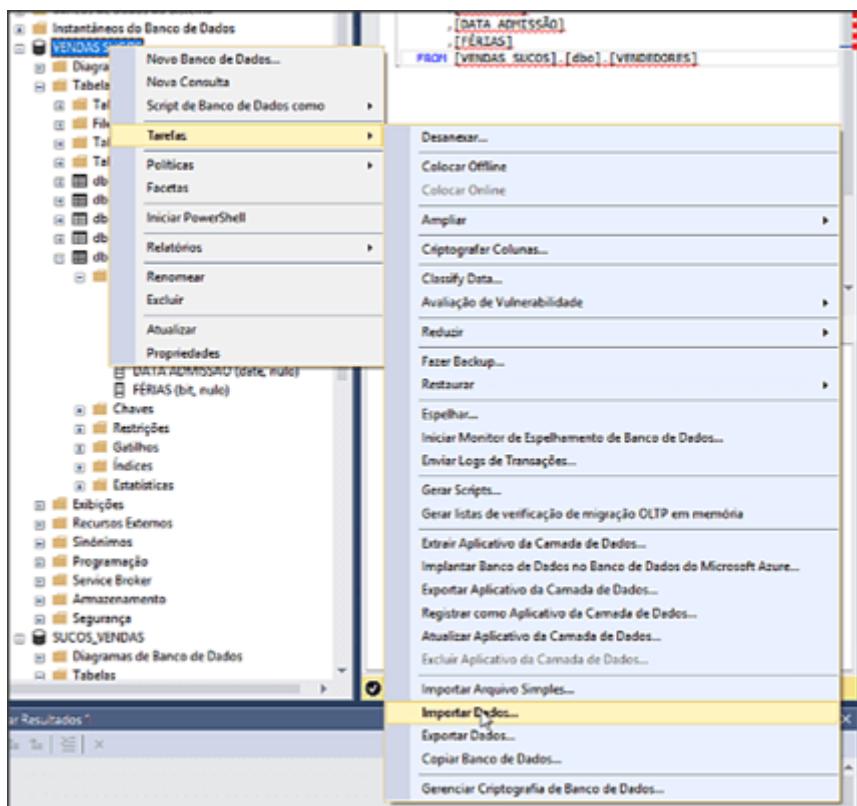
Vendedores.csv - Bloco de notas						
Arquivo Editar Formatar Exibir Ajuda						
MATRICULA;NOME;COMISSÃO;DATA ADMISSÃO;FÉRIAS;BAIRRO						
235;Márcio Almeida Silva;0,08;15/08/2014;0;Tijuca						
236;Cláudia Moraes;0,08;17/09/2013;1;Jardins						
237;Roberta Martins;0,11;18/03/2017;1;Copacabana						
238;Péricles Alves;0,11;21/08/2016;0;Santo Amaro						

23) Volte ao **SQL Server Management Studio** e clique com o botão da direita do mouse sobre a tabela de vendedores, selecione a opção **Selecionar as 1000 Primeiras Linhas Superiores**. Veja que a tabela está vazia:

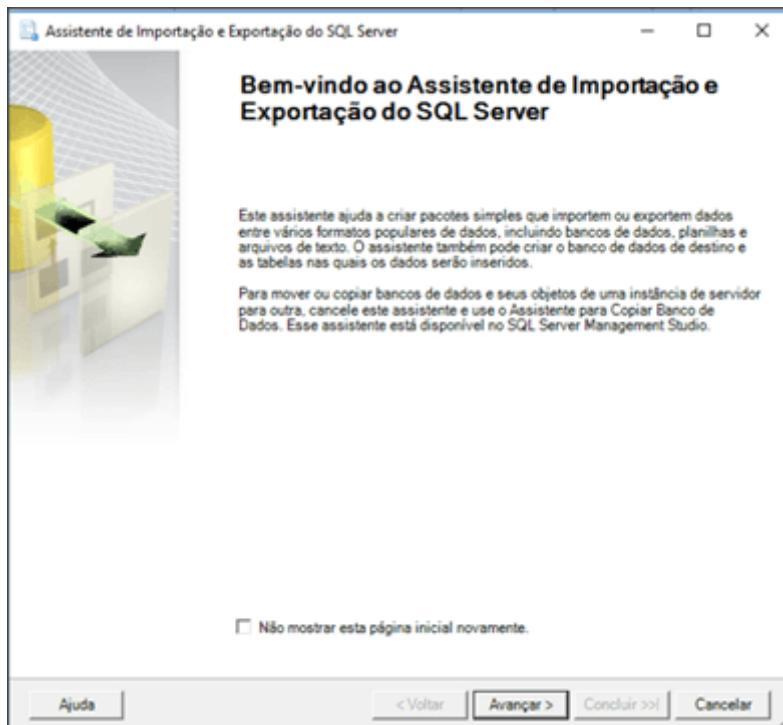
The screenshot shows the 'Resultados' tab of the SQL Server Management Studio results pane. At the top, there are tabs for 'Resultados' and 'Mensagens'. Below the tabs, there is a header row with columns labeled: MATRÍCULA, NOME, BAIRRO, COMISSÃO, DATA ADMISSÃO, and FÉRIAS. The main body of the table is completely empty, indicating no data has been returned.

MATRÍCULA	NOME	BAIRRO	COMISSÃO	DATA ADMISSÃO	FÉRIAS

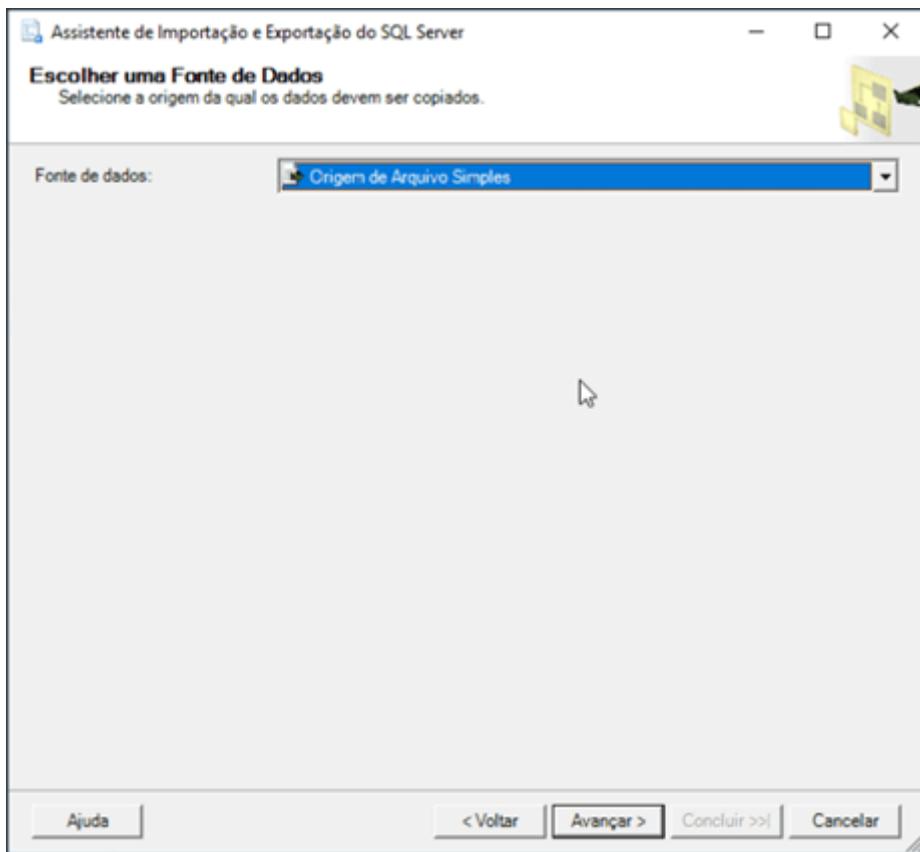
24) Clique com o botão da direita do mouse sobre o banco de dados **VENDAS SUCOS** e escolha a opção **Tarefas --> Importar dados**:



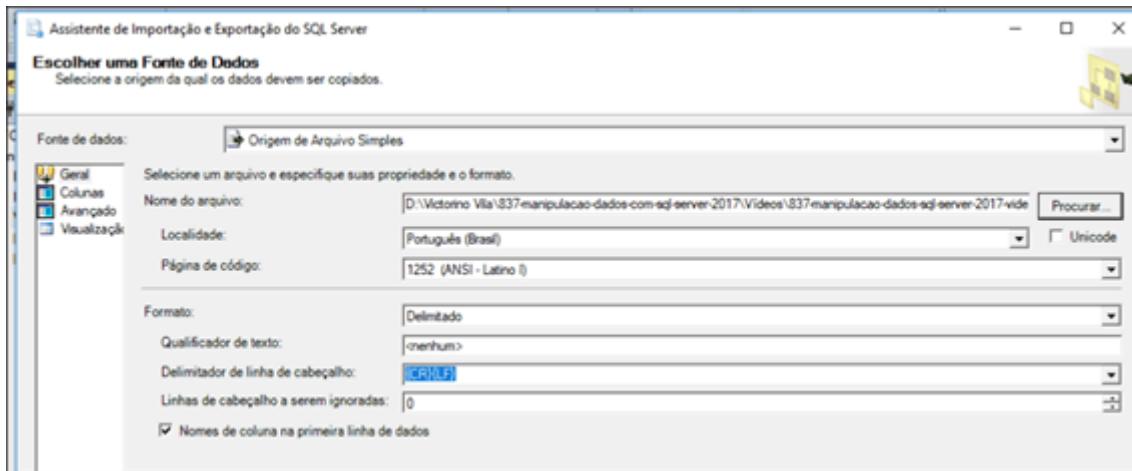
25) Clique em Avançar:



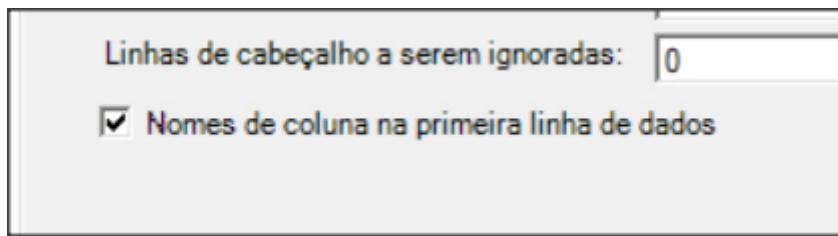
26) Em fonte de dados, selecione Origem de Arquivo Simples:



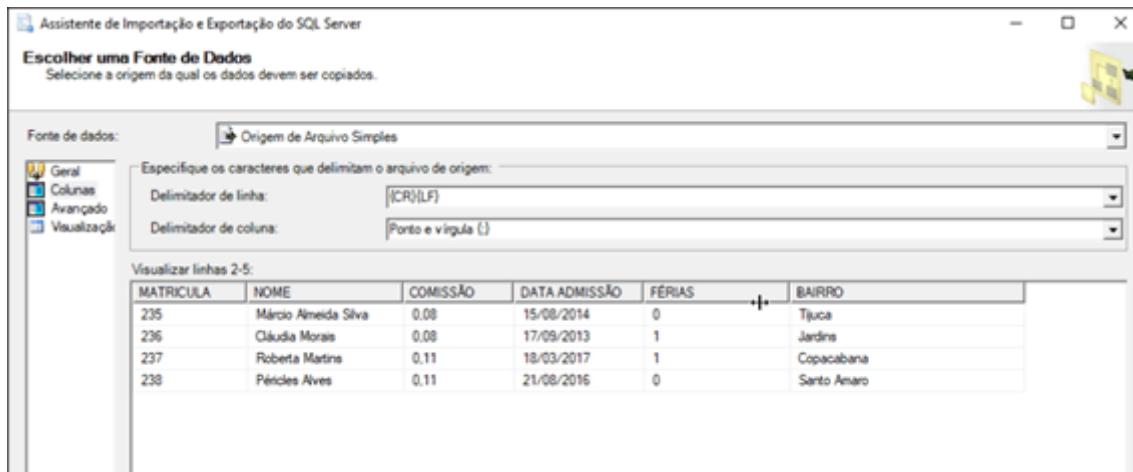
27) Após a escolha da origem de dados, a caixa de diálogo será mostrada como a abaixo. Clique em **Procurar...** e selecione o arquivo que foi baixado na sua máquina (o arquivo **Vendedores.csv**):



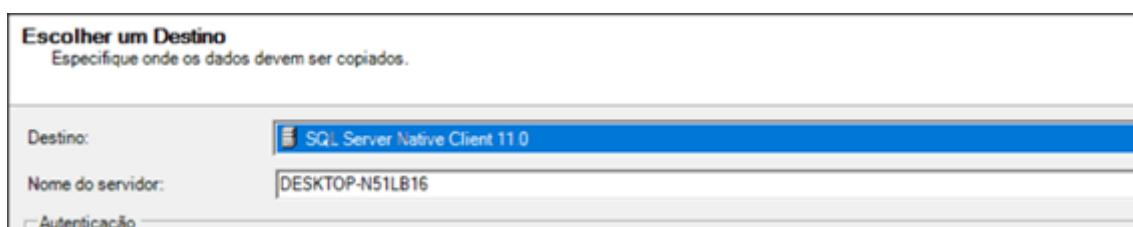
28) Certifique-se que a opção **Nomes de coluna na primeira linha de dados** esteja marcada. Depois clique em **Avançar**:



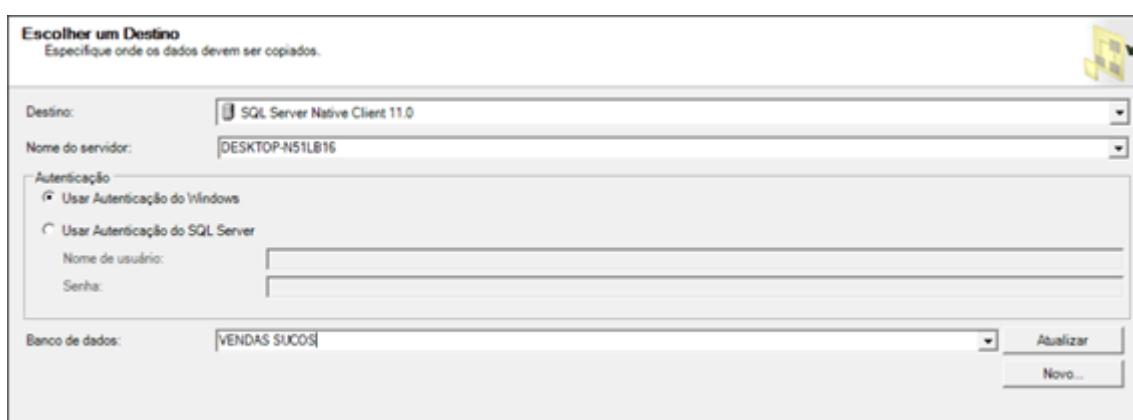
29) Note que você pode visualizar uma amostra dos dados que serão importados do arquivo texto. Depois de visualizá-los, clique em **Avançar**:



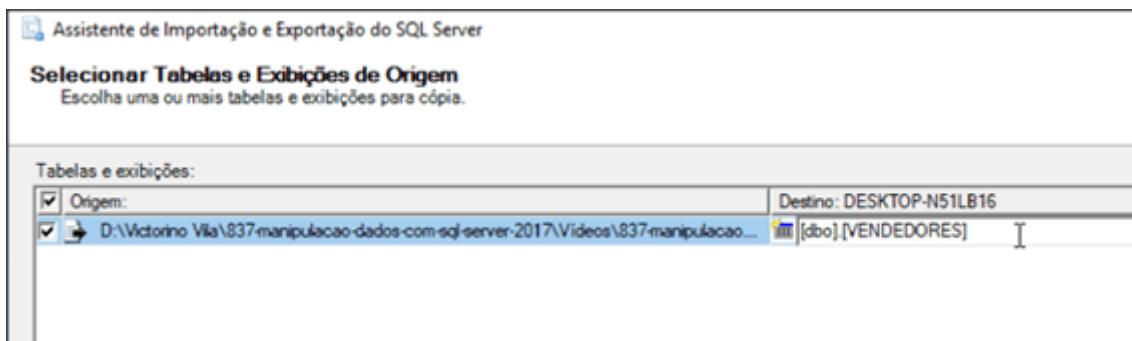
30) Defina o destino, escolhendo **SQL Server Native Client 11.0**:



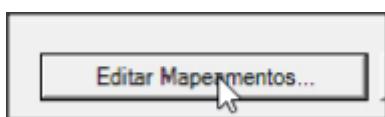
31) Inclua também as propriedades de conexão na base de dados **VENDAS SUCOS**. Depois clique em **Avançar**:



32) Do lado direito, selecione o nome da tabela destino (que será **VENDEDORES**):



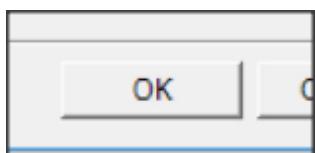
33) Clique no botão **Editar Mapeamentos** para observar se os campos do arquivo de origem estão corretamente associados aos campos da tabela de destino:



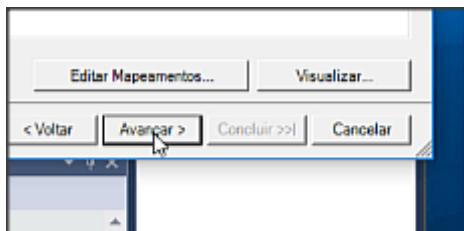
34) A associação será sugerida pelos nomes dos campos. Alguns campos podem não ter esta associação automática configurada. Faça a associação manualmente, conforme mostrado abaixo:

Origem	Destino	Tipo	Permit...	Taman...	Precisão	Esc...
MATRÍCULA	MATRÍCULA	varchar	<input type="checkbox"/>	5		
NOME	NOME	varchar	<input checked="" type="checkbox"/>	100		
COMISSÃO	COMISSÃO	float	<input checked="" type="checkbox"/>			
DATA ADMISS...	DATA ADMISS...	date	<input checked="" type="checkbox"/>			
FÉRIAS	FÉRIAS	bit	<input checked="" type="checkbox"/>			
BAIRRO	BAIRRO	varchar	<input checked="" type="checkbox"/>	50		

35) Clique em **OK** após confirmar as associações:



36) Ao voltar na tela onde você associou o arquivo de origem à tabela de destino, clique em **Avançar**:



37) Você verá um resumo das associações a serem utilizadas na importação dos dados. Clique novamente em **Avançar**:

Mapeamento de tipo de dados:

Coluna de Origem	Tipo de Origem	Coluna de Destino	Tipo de Destino	Conversão	Se Houver Erro	Se Houver Truncamento
● MATRÍCULA	cadeia de car...	MATRÍCULA	varchar			
● NOME	cadeia de car...	NOME	varchar			
⚠ COMISSÃO	cadeia de car...	COMISSÃO	float	<input checked="" type="checkbox"/>	User Global	User Global
⚠ DATA ADMIS...	cadeia de car...	DATA ADMIS...	date	<input checked="" type="checkbox"/>	User Global	User Global
⚠ FÉRIAS	cadeia de car...	FÉRIAS	bit	<input checked="" type="checkbox"/>	User Global	User Global
● BAIRRO	cadeia de car...	BAIRRO	varchar			

Para exibir os detalhes da conversão, clique duas vezes na linha que contém o tipo de origem da coluna a ser convertida.

Se Houver Erro (global):  Se Houver Truncamento (global):

Ajuda | < Voltar | Avançar > | Concluir >> | Cancelar |

38) Veja se a opção **Executar imediatamente** está marcada e clique novamente em **Avançar**:

Assistente de Importação e Exportação do SQL Server

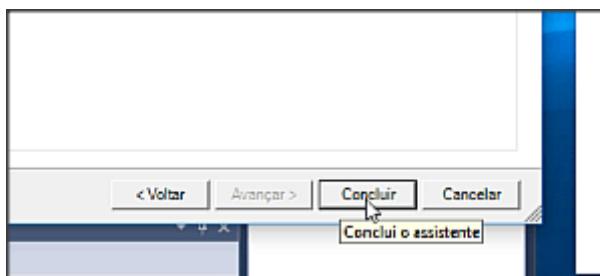
**Salvar e Executar Pacote**  
Indica se o pacote SSIS deve ser salvo ou não.

Executar imediatamente  
 Salvar Pacote SSIS  
 SQL Server  
 Sistema de arquivos

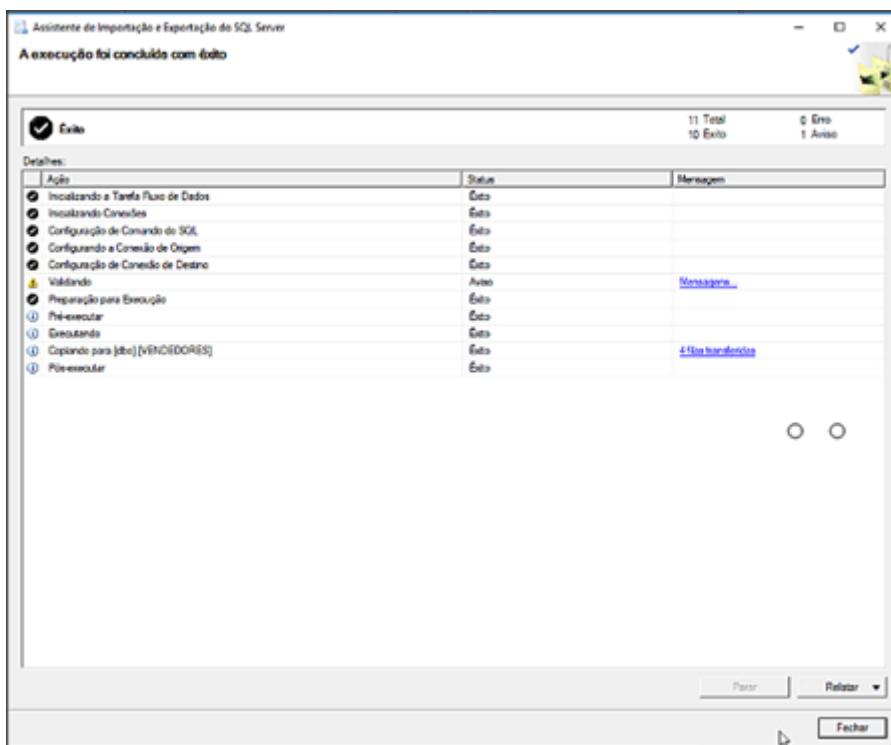
Nível de proteção do pacote:

Senha:   
 Digite a senha novamente:

39) Clique em Concluir:



40) Veja o processo sendo executado:



41) Clique em Fechar e veja novamente o conteúdo da tabela **VENDEDORES**, agora com dados:

A screenshot of a results viewer window with a 'Resultados' tab selected. The table has the following data:

- 1) Observe os valores dos dados referentes ao produto **1040107**. Para tal, crie uma nova consulta associada à base **VENDAS\_SUCOS**, executando o comando abaixo:

```
SELECT * FROM PRODUTOS
```

Podemos localizá-lo no *grid* com os resultados:

7	1037797	Clean - 2 Litros - Laranja	Laranja	2 Litros	PET	16,008
8	1040107	Light - 350 ml - Melância	Melância	350 ml	Lata	4,56
9	1040108	Light - 350 ml - Graviola	Graviola	350 ml	Lata	4

- 2) Altere o preço da tabela, executando:

```
UPDATE [PRODUTOS] SET [PREÇO LISTA] = 5
WHERE [CÓDIGO] = '1040107'
```

Após a execução, você terá:

7	1037797	Clean - 2 Litros - Laranja	Laranja	2 Litros	PET	16,008
8	1040107	Light - 350 ml - Melância	Melância	350 ml	Lata	5
9	1040108	Light - 350 ml - Graviola	Graviola	350 ml	Lata	4

- 2) No mesmo comando **UPDATE**, altere mais de um campo. Digite:

```
UPDATE [PRODUTOS] SET [SABOR] = 'Laranja', [EMBALAGEM] = 'PET'
WHERE [CÓDIGO] = '1040107'
```

Você terá:

7	1037797	Clean - 2 Litros - Laranja	Laranja	2 Litros	PET	16,008
8	1040107	Light - 350 ml - Melância	Laranja	350 ml	PET	5
9	1040108	Light - 350 ml - Graviola	Graviola	350 ml	Lata	4

- 3) O comando **UPDATE** pode ser feito em mais de um registro ao mesmo tempo. Se você executar o comando:

```
SELECT * FROM PRODUTOS WHERE [SABOR] = 'Maracujá'
```

Terá vários produtos com sabor **Maracujá**:

CÓDIGO	DESCRIPTOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
1	Festival de Sabores - 1,5 Litros - Maracujá	Maracujá	1,5 Litros	PET	10,512
2	Festival de Sabores - 700 ml - Maracujá	Maracujá	700 ml	Garrafa	4,912

5) Se você executar:

```
UPDATE [PRODUTOS] SET [PREÇO LISTA] = [PREÇO LISTA] * 1.10
WHERE [SABOR] = 'Maracujá'
```

Estará aumentando em 10% os preços de todos os produtos de sabor **Maracujá**.

6) Você pode usar o comando **UPDATE** junto com o **FROM** (como se fosse efetuar um **JOIN** entre tabelas) para alterar dados de uma tabela, baseado no conteúdo de outra. Para isso, mude o valor do preço de um produto na tabela **TABELA DE PRODUTOS**, da base **SUCOS\_VENDAS** que é a tabela fonte, através de uma nova consulta associada à base **VENDAS SUCOS**. Execute:

```
UPDATE [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
SET [PREÇO DE LISTA] = 7
WHERE [CÓDIGO DO PRODUTO] = '1000889'
```

7) Se você executar estes dois comandos comparando os preços dos mesmos produtos nas duas tabelas (fonte e alvo), você terá:

```
SELECT * FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
WHERE [CÓDIGO DO PRODUTO] = '1000889'
SELECT * FROM [PRODUTOS] WHERE [CÓDIGO] = '1000889'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	7,00

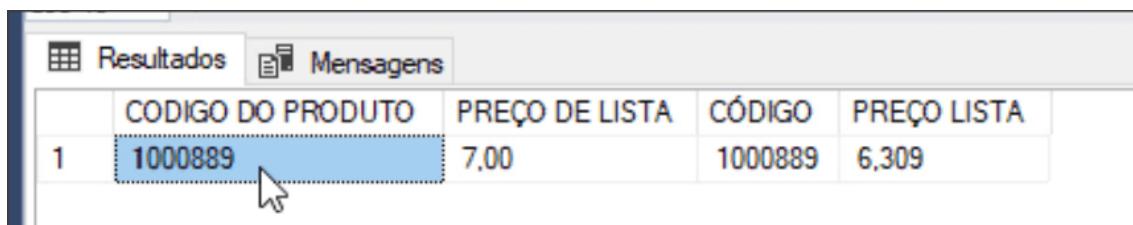
	CÓDIGO	DESCRITOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Uva	700 ml	Garrafa	6,309

Na tabela origem, o preço é 7 e na destino o preço é 6,309.

8) Sincronize o preço. Para isso, use o comando **UPDATE** com **FROM**. Mas antes execute o comando abaixo:

```
SELECT B.[CÓDIGO DO PRODUTO], B.[PREÇO DE LISTA],
       A.[CÓDIGO], A.[PREÇO LISTA] FROM [PRODUTOS] A
INNER JOIN [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
ON A.[CÓDIGO] = B.[CÓDIGO DO PRODUTO]
WHERE A.[CÓDIGO] = '1000889'
```

Este comando faz um **JOIN** entre as duas tabelas (origem e destino) mostrando os dois preços lado a lado:



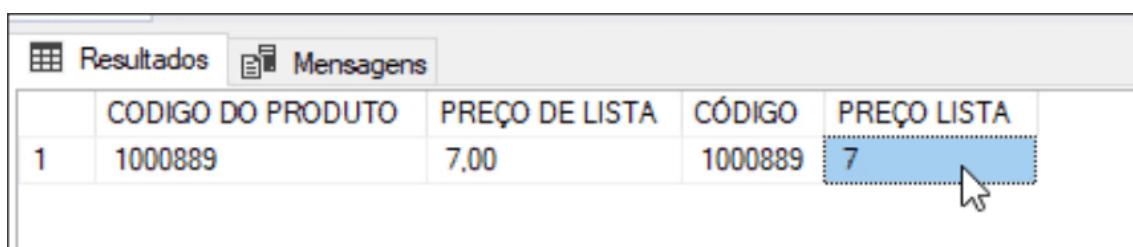
	CODIGO DO PRODUTO	PREÇO DE LISTA	CÓDIGO	PREÇO LISTA
1	1000889	7,00	1000889	6,309

- 9) Para haver a sincronização, execute o **UPDATE** usando os mesmos parâmetros do **JOIN** acima:

```
UPDATE A SET A.[PREÇO LISTA] = B.[PREÇO DE LISTA] FROM [PRODUTOS] A
INNER JOIN [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
    ON A.[CÓDIGO] = B.[CODIGO DO PRODUTO]
WHERE A.[CÓDIGO] = '1000889'
```

- 10) Após a execução do comando acima, você pode comparar novamente o preço nas duas tabelas e verá que eles estão sincronizados:

```
SELECT B.[CODIGO DO PRODUTO], B.[PREÇO DE LISTA],
       A.[CÓDIGO], A.[PREÇO LISTA] FROM [PRODUTOS] A
INNER JOIN [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
    ON A.[CÓDIGO] = B.[CODIGO DO PRODUTO]
WHERE A.[CÓDIGO] = '1000889'
```



	CODIGO DO PRODUTO	PREÇO DE LISTA	CÓDIGO	PREÇO LISTA
1	1000889	7,00	1000889	7

- 11) O comando **MERGE** faz a mesma coisa que o **UPDATE FROM**. Veja: altere o preço de origem de outro produto, executando o comando abaixo:

```
UPDATE [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS]
    SET [PREÇO DE LISTA] = 8
WHERE [CODIGO DO PRODUTO] = '1002334'
```

- 12) Comparando este produto na tabela origem e destino, você terá:

```
SELECT B.[CODIGO DO PRODUTO], B.[PREÇO DE LISTA],
       A.[CÓDIGO], A.[PREÇO LISTA] FROM [PRODUTOS] A
INNER JOIN [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
    ON A.[CÓDIGO] = B.[CODIGO DO PRODUTO]
WHERE [CÓDIGO] = '1002334'
```

	CODIGO DO PRODUTO	PREÇO DE LISTA	CÓDIGO	PREÇO LISTA
1	1002334	8,00	1002334	7,00

13) Vamos executar o **MERGE**:

```
MERGE INTO [PRODUTOS] A
USING [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
ON A.[CÓDIGO] = B.[CODIGO DO PRODUTO]
AND A.[CÓDIGO] = '1002334'
WHEN MATCHED THEN
UPDATE SET A.[PREÇO LISTA] = B.[PREÇO DE LISTA];
```

14) Após a execução, se você comparar novamente o produto nas duas tabelas, você:

```
SELECT B.[CODIGO DO PRODUTO], B.[PREÇO DE LISTA],
       A.[CÓDIGO], A.[PREÇO LISTA] FROM [PRODUTOS] A
INNER JOIN [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS] B
      ON A.[CÓDIGO] = B.[CODIGO DO PRODUTO]
     WHERE [CÓDIGO] = '1002334'
```

	CODIGO DO PRODUTO	PREÇO DE LISTA	CÓDIGO	PREÇO LISTA
1	1002334	8,00	1002334	8

15) Baixe o arquivo **Inclusão de registros.sql** [aqui](#) e abra-o. Execute os comandos de inclusão de dados conforme mostrados abaixo. Certifique-se que esteja na base **VENDAS SUCOS**:

```
INSERT INTO [dbo].[PRODUTOS] ([CÓDIGO], [DESCRITOR],
    [SABOR], [TAMANHO], [EMBALAGEM], [PREÇO LISTA])
VALUES ('1001001', 'Sabor dos Alpes 700 ml - Manga',
        'Manga', '700 ml', 'Garrafa', 7.50),
       ('1001000', 'Sabor dos Alpes 700 ml - Melão',
        'Melão', '700 ml', 'Garrafa', 7.50),
       ('1001002', 'Sabor dos Alpes 700 ml - Graviola',
        'Graviola', '700 ml', 'Garrafa', 7.50),
       ('1001003', 'Sabor dos Alpes 700 ml - Tangerina',
        'Tangerina', '700 ml', 'Garrafa', 7.50),
       ('1001004', 'Sabor dos Alpes 700 ml - Abacate',
        'Abacate', '700 ml', 'Garrafa', 7.50),
       ('1001005', 'Sabor dos Alpes 700 ml - Açaí',
        'Açaí', '700 ml', 'Garrafa', 7.50),
       ('1001006', 'Sabor dos Alpes 1 Litro - Manga',
        'Manga', '1 Litro', 'Garrafa', 7.50),
```

```
('1001007', 'Sabor dos Alpes 1 Litro - Melão',
     'Melão', '1 Litro', 'Garrafa', 7.50),
('1001008', 'Sabor dos Alpes 1 Litro - Graviola',
     'Graviola', '1 Litro', 'Garrafa', 7.50),
('1001009', 'Sabor dos Alpes 1 Litro - Tangerina',
     'Tangerina', '1 Litro', 'Garrafa', 7.50),
('1001010', 'Sabor dos Alpes 1 Litro - Abacate',
     'Abacate', '1 Litro', 'Garrafa', 7.50),
('1001011', 'Sabor dos Alpes 1 Litro - Açaí',
     'Açaí', '1 Litro', 'Garrafa', 7.50)
```

- 16) Execute o comando abaixo para listar todos os produtos com o texto **Sabor dos Alpes**:

```
SELECT * FROM [PRODUTOS]
WHERE SUBSTRING([DESCRITOR], 1, 15) = 'Sabor dos Alpes'
```

- 17) Exclua o registro **1001000**:

```
DELETE FROM [PRODUTOS] WHERE [CÓDIGO] = '1001000'
```

- 18) Selecione todos os produtos com tamanho igual a **1 Litro** e que seja **Sabor dos Alpes**:

```
SELECT * FROM [PRODUTOS] WHERE [TAMANHO] = '1 Litro'
AND SUBSTRING([DESCRITOR], 1, 15) = 'Sabor dos Alpes'
```

	CÓDIGO	DESCRIPTOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
1	1001006	Sabor dos Alpes 1 Litro - Manga	Manga	1 Litro	Garrafa	7,5
2	1001007	Sabor dos Alpes 1 Litro - Melão	Melão	1 Litro	Garrafa	7,5
3	1001008	Sabor dos Alpes 1 Litro - Graviola	Graviola	1 Litro	Garrafa	7,5
4	1001009	Sabor dos Alpes 1 Litro - Tangerina	Tangerina	1 Litro	Garrafa	7,5
5	1001010	Sabor dos Alpes 1 Litro - Abacate	Abacate	1 Litro	Garrafa	7,5
6	1001011	Sabor dos Alpes 1 Litro - Açaí	Açaí	1 Litro	Garrafa	7,5

- 19) Para apagá-los, use o comando **DELETE** com a mesma cláusula **WHERE**:

```
DELETE FROM [PRODUTOS] WHERE [TAMANHO] = '1 Litro'
AND SUBSTRING([DESCRITOR], 1, 15) = 'Sabor dos Alpes'
```

- 20) Liste todos os produtos que existam na tabela **PRODUTO**, do banco **VENDAS\_SUCOS**, e que não existam na tabela **CODIGO DO PRODUTO** do banco de origem (**SUCOS\_VENDAS**):

```
SELECT * FROM [PRODUTOS] WHERE [CÓDIGO]
NOT IN (SELECT [CODIGO DO PRODUTO]
FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS])
```

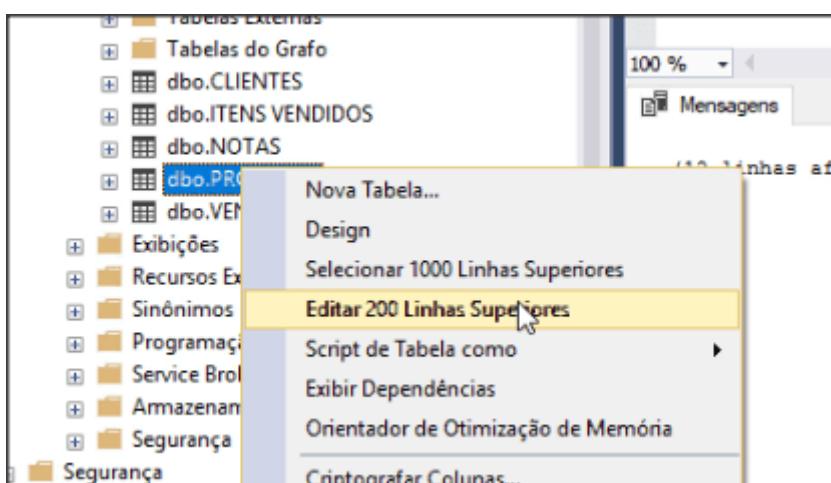
	CÓDIGO	DESCRIPTOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA	O
1	1000900	Sabor da Montanha 700 ml - Jaca	Jaca	700 ml	Garrafa	8,5	
2	1000901	Sabor da Montanha 700 ml - Manga	Manga	700 ml	Garrafa	7,5	
3	1000902	Sabor da Montanha 700 ml - Melão	Melão	700 ml	Garrafa	7,5	
4	1000903	Sabor da Montanha 700 ml - Graviola	Graviola	700 ml	Garrafa	7,5	
5	1000904	Sabor da Montanha 700 ml - Tangerina	Tangerina	700 ml	Garrafa	7,5	
6	1000905	Sabor da Montanha 700 ml - Abacate	Abacate	700 ml	Garrafa	7,5	
7	1000906	Sabor da Montanha 700 ml - Açaí	Açaí	700 ml	Garrafa	7,5	
8	1000907	Sabor da Montanha 1 Litro - Manga	Manga	1 Litro	Garrafa	7,5	
9	1000908	Sabor da Montanha 1 Litro - Melão	Melão	1 Litro	Garrafa	7,5	
10	1000909	Sabor da Montanha 1 Litro - Graviola	Graviola	1 Litro	Garrafa	7,5	
11	1000910	Sabor da Montanha 1 Litro - Tangerina	Tangerina	1 Litro	Garrafa	7,5	
12	1000911	Sabor da Montanha 1 Litro - Abacate	Abacate	1 Litro	Garrafa	7,5	
13	1000912	Sabor da Montanha 1 Litro - Açaí	Açaí	1 Litro	Garrafa	7,5	
14	1001001	Sabor dos Alpes 700 ml - Manga	Manga	700 ml	Garrafa	7,5	
15	1001002	Sabor dos Alpes 700 ml - Graviola	Graviola	700 ml	Garrafa	7,5	
16	1001003	Sabor dos Alpes 700 ml - Tangerina	Tangerina	700 ml	Garrafa	7,5	

Existem 25 linhas.

- 21) Para apagar os registros que existam em **PRODUTO** e não existam em **CÓDIGO DO PRODUTO**, basta colocar a condição de teste na cláusula **WHERE**:

```
DELETE FROM [PRODUTOS] WHERE [CÓDIGO]
NOT IN (SELECT [CÓDIGO DO PRODUTO]
        FROM [SUCOS_VENDAS].[dbo].[TABELA DE PRODUTOS])
```

- 22) Clique com o botão da direita do mouse sobre a tabela **PRODUTOS** e selecione a opção **Editar 200 Linhas Superiores**:



- 23) Se você for no final da lista, verá o ponto de onde você pode editar novos registros:

	826490	Linha Refrescante - 700 ml - Morango/L...	Morango/Limão	700 ml	Garrafa	6,3105
	838819	Clean - 1,5 Litros - Laranja	Laranja	1,5 Litros	PET	12,008
*	NULL	NULL	NULL	NULL	NULL	NULL
	1	de 43		▶	▶	

24) Faça uma série de edições diretamente no *grid* para alterar os dados, por exemplo, para alterar os preços:

EMBALAGEM	PREÇO LISTA
Garrafa	7
Garrafa	8
Garrafa	9
Garrafa	10
Garrafa	12
Garrafa	11
Garrafa	9
Garrafa	7,5

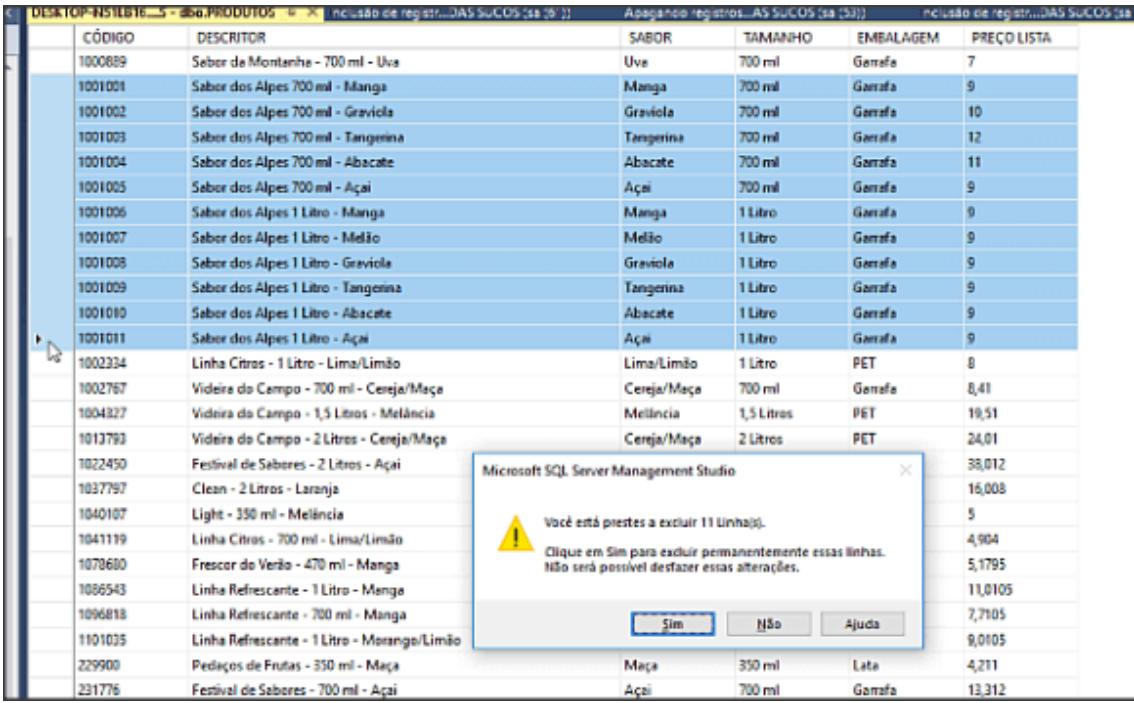
25) A alteração efetuada no *grid* automaticamente reflete no resultado da tabela. Se você executar:

```
SELECT * from [PRODUTOS]
WHERE SUBSTRING([DESCRITOR], 1, 15) = 'Sabor dos Alpes'
```

Verá os valores já alterados:

CÓDIGO	DESCRITOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
1	Sabor dos Alpes 700 ml - Melão	Melão	700 ml	Garrafa	8
2	Sabor dos Alpes 700 ml - Manga	Manga	700 ml	Garrafa	9
3	Sabor dos Alpes 700 ml - Graviola	Graviola	700 ml	Garrafa	10
4	Sabor dos Alpes 700 ml - Tangerina	Tangerina	700 ml	Garrafa	12
5	Sabor dos Alpes 700 ml - Abacate	Abacate	700 ml	Garrafa	11
6	Sabor dos Alpes 700 ml - Açaí	Açaí	700 ml	Garrafa	9
7	Sabor dos Alpes 1 Litro - Manga	Manga	1 Litro	Garrafa	9
8	Sabor dos Alpes 1 Litro - Melão	Melão	1 Litro	Garrafa	9
9	Sabor dos Alpes 1 Litro - Graviola	Graviola	1 Litro	Garrafa	9
10	Sabor dos Alpes 1 Litro - Tangerina	Tangerina	1 Litro	Garrafa	9
11	Sabor dos Alpes 1 Litro - Abacate	Abacate	1 Litro	Garrafa	9
12	Sabor dos Alpes 1 Litro - Açaí	Açaí	1 Litro	Garrafa	9

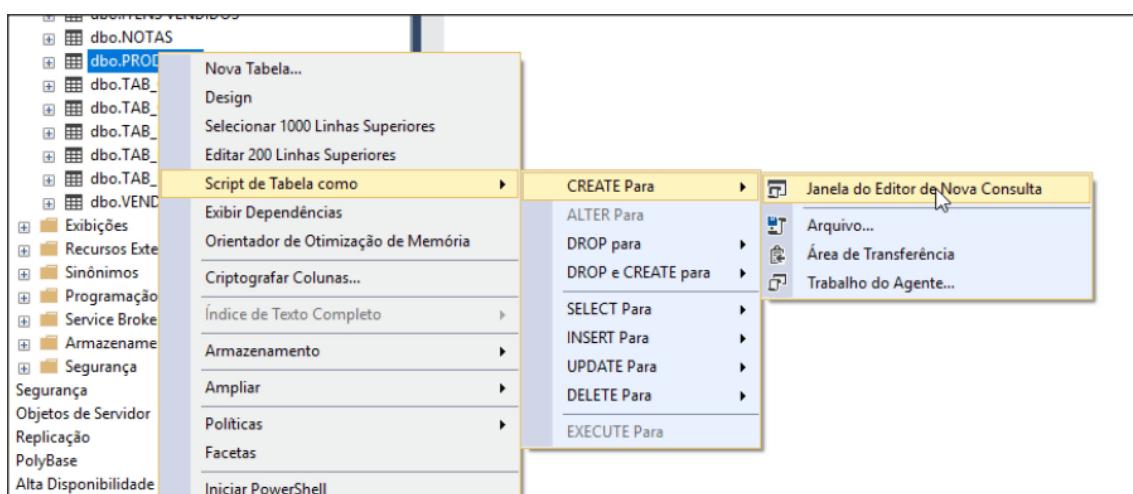
26) Você pode selecionar um bloco no *grid* e depois exclui-lo teclando **DEL**:



A screenshot of Microsoft SQL Server Management Studio showing a table named 'PRODUTOS'. A context menu is open over the table, with the 'Script Tabela como' option selected. A sub-menu for 'CREATE Para' is open, and the 'Janela do Editor de Nova Consulta' option is highlighted.

CÓDIGO	DESCRIPTOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
100089	Saber da Montanha - 700 ml - Uva	Uva	700 ml	Garrafa	7
1001001	Saber dos Alpes 700 ml - Manga	Manga	700 ml	Garrafa	9
1001002	Saber dos Alpes 700 ml - Graviola	Graviola	700 ml	Garrafa	10
1001003	Saber dos Alpes 700 ml - Tangerina	Tangerina	700 ml	Garrafa	12
1001004	Saber dos Alpes 700 ml - Abacate	Abacate	700 ml	Garrafa	11
1001005	Saber dos Alpes 700 ml - Açaí	Açaí	700 ml	Garrafa	9
1001006	Saber dos Alpes 1 Litro - Manga	Manga	1 Litro	Garrafa	9
1001007	Saber dos Alpes 1 Litro - Melão	Melão	1 Litro	Garrafa	9
1001008	Saber dos Alpes 1 Litro - Graviola	Graviola	1 Litro	Garrafa	9
1001009	Saber dos Alpes 1 Litro - Tangerina	Tangerina	1 Litro	Garrafa	9
1001010	Saber dos Alpes 1 Litro - Abacate	Abacate	1 Litro	Garrafa	9
1001011	Saber dos Alpes 1 Litro - Açaí	Açaí	1 Litro	Garrafa	9
1002334	Linha Citros - 1 Litro - Lima/Limão	Lima/Limão	1 Litro	PET	8
1002767	Videira do Campo - 700 ml - Cereja/Maçã	Cereja/Maçã	700 ml	Garrafa	8,41
1004327	Vidreira do Campo - 1,5 Litros - Melancia	Melancia	1,5 Litros	PET	19,51
1013793	Videira do Campo - 2 Litros - Cereja/Maçã	Cereja/Maçã	2 Litros	PET	24,01
1022450	Festival de Sabores - 2 Litros - Açaí				38,012
1037797	Clean - 2 Litros - Laranja				16,008
1040107	Light - 350 ml - Melancia				5
1041119	Linha Citros - 700 ml - Lima/Limão				4,904
1070680	Frescor do Verão - 470 ml - Manga				5,1795
1086543	Linha Refrescante - 1 Litro - Manga				11,0105
1096818	Linha Refrescante - 700 ml - Manga				7,7105
1101035	Linha Refrescante - 1 Litro - Morango/Limão				9,0105
229900	Pedaços de Frutas - 350 ml - Maçã	Maçã	350 ml	Lata	4,211
231776	Festival de Sabores - 700 ml - Açaí	Açaí	700 ml	Garrafa	13,312

27) Clique com botão da direita do mouse sobre a tabela de produtos e escolha a opção **Script da Tabela como** --> **CREATE Para** --> Janela do Editor de Nova Consulta:



28) Altere o nome da tabela para **Produtos\_2** e o nome da **PK** para **PK\_Produtos\_2**. Depois execute a criação da nova tabela:

```
USE [VENDAS_SUCOS]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PRODUTOS_2] (
    [CÓDIGO] [varchar](10) NOT NULL,
    [DESCRIPTOR] [varchar](100) NULL,
    [SABOR] [varchar](50) NULL,
```

```

[TAMANHO] [varchar] (50) NULL,
[EMBALAGEM] [varchar] (50) NULL,
[PREÇO LISTA] [float] NULL,
CONSTRAINT [PK_PRODUTOS_2] PRIMARY KEY CLUSTERED (
    [CÓDIGO] ASC
) WITH (
    PAD_INDEX = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON
) ON [PRIMARY]) ON [PRIMARY]
GO

```

29) Inclua os dados contidos na tabela original (**Produtos**) para esta nova, executando o comando abaixo:

```

INSERT INTO PRODUTOS_2
SELECT * FROM PRODUTOS

```

30) Agora você tem produtos na tabela nova:

```
SELECT * FROM PRODUTOS_2
```

	CÓDIGO	DESCRIPTOR	SABOR	TAMANHO	EMBALAGEM	PREÇO LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Uva	700 ml	Garrafa	7
2	1002334	Linha Citros - 1 Litro - Lima/Limão	Lima/Limão	1 Litro	PET	8
3	1002767	Videira do Campo - 700 ml - Cereja/Maça	Cereja/Maça	700 ml	Garrafa	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	Melância	1,5 Litros	PET	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maça	Cereja/Maça	2 Litros	PET	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	Açaí	2 Litros	PET	38,012
7	1037797	Clean - 2 Litros - Laranja	Laranja	2 Litros	PET	16,008
8	1040107	Light - 350 ml - Melância	Laranja	350 ml	PET	5
9	1041119	Linha Citros - 700 ml - Lima/Limão	Lima/Limão	700 ml	Garrafa	4,904
10	1078680	Frescor do Verão - 470 ml - Manga	Manga	470 ml	Garrafa	5,1795
11	1086543	Linha Refrescante - 1 Litro - Manga	Manga	1 Litro	PET	11,0105
12	1096818	Linha Refrescante - 700 ml - Manga	Manga	700 ml	Garrafa	7,7105
13	1101035	Linha Refrescante - 1 Litro - Morango/Limão	Morango/Limão	1 Litro	PET	9,0105
14	229900	Pedaços de Frutas - 350 ml - Maça	Maça	350 ml	Lata	4,211

31) Você pode fazer uma alteração em todos os registros da tabela. No exemplo abaixo, aumente o preço de todos os produtos em 10%:

```

UPDATE PRODUTOS_2
SET [PREÇO LISTA] = [PREÇO LISTA] * 1.10

```

32) Você também pode apagar todos os registros da tabela, executando:

```
DELETE FROM PRODUTOS_2
```

33) Verifique o conteúdo da tabela de vendedores, executando:

```
SELECT * FROM [VENDEDORES]
```

	MATRÍCULA	NOME	BAIRRO	COMISSÃO	DATA ADMISSÃO	FÉRIAS
1	235	Márcio Almeida Silva	Tijuca	0,08	2014-08-15	0
2	236	Cláudia Morais	Jardins	0,08	2013-09-17	1
3	237	Roberta Martins	Copacabana	0,11	2017-03-18	1
4	238	Péricles Alves	Santo Amaro	0,11	2016-08-21	0

34) Execute o comando abaixo para iniciar uma nova transação:

```
BEGIN TRANSACTION
```

35) Efetue uma alteração da comissão dos vendedores, aumentando-a em 15%:

```
UPDATE [VENDEDORES] SET [COMISSÃO] = [COMISSÃO] * 1.15
```

36) Inclua um novo vendedor:

```
INSERT INTO [VENDEDORES] ([MATRÍCULA], [NOME], [BAIRRO],  
[COMISSÃO], [DATA ADMISSÃO], [FÉRIAS])  
VALUES ('99999', 'João da Silva', 'Icaraí', 0.08, '2014-09-01', 0)
```

37) Agora, se você verificar a tabela de vendedores, terá:

```
SELECT * FROM [VENDEDORES]
```

	MATRÍCULA	NOME	BAIRRO	COMISSÃO	DATA ADMISSÃO	FÉRIAS
1	235	Márcio Almeida Silva	Tijuca	0,092	2014-08-15	0
2	236	Cláudia Morais	Jardins	0,092	2013-09-17	1
3	237	Roberta Martins	Copacabana	0,1265	2017-03-18	1
4	238	Péricles Alves	Santo Amaro	0,1265	2016-08-21	0
5	99999	João da Silva	Icaraí	0,08	2014-09-01	0

38) Agora, executando o comando abaixo:

```
ROLLBACK
```

Você volta ao status antes do `BEGIN TRANSACTION`.

39) Verifique agora a tabela de vendedores:

```
SELECT * FROM [VENDEDORES]
```

	MATRÍCULA	NOME	BAIRRO	COMISSÃO	DATA ADMISSÃO	FÉRIAS
1	235	Márcio Almeida Silva	Tijuca	0,08	2014-08-15	0
2	236	Cláudia Morais	Jardins	0,08	2013-09-17	1
3	237	Roberta Martins	Copacabana	0,11	2017-03-18	1
4	238	Péricles Alves	Santo Amaro	0,11	2016-08-21	0

40) Se você fizer novamente o início de uma nova transação:

```
BEGIN TRANSACTION
```

41) E executar novamente a alteração de comissão e inclusão de novo vendedor:

```
UPDATE [VENDEDORES] SET [COMISSÃO] = [COMISSÃO] * 1.15
INSERT INTO [VENDEDORES] ([MATRÍCULA], [NOME], [BAIRRO],
[COMISSÃO], [DATA ADMISSÃO], [FÉRIAS])
VALUES ('99999', 'João da Silva', 'Icarai', 0.08, '2014-09-01', 0)
```

42) Ao executar o comando abaixo, as alterações são confirmadas:

```
COMMIT
```

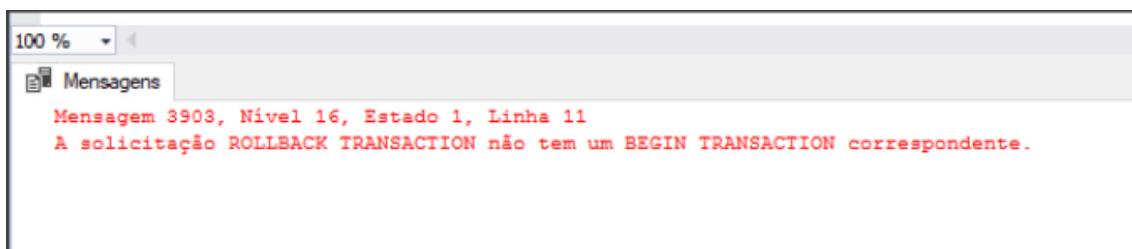
43) Verificando agora a tabela:

```
SELECT * FROM [VENDEDORES]
```

Você terá:

	MATRÍCULA	NOME	BAIRRO	COMISSÃO	DATA ADMISSÃO	FÉRIAS
1	235	Márcio Almeida Silva	Tijuca	0,092	2014-08-15	0
2	236	Cláudia Morais	Jardins	0,092	2013-09-17	1
3	237	Roberta Martins	Copacabana	0,1265	2017-03-18	1
4	238	Péricles Alves	Santo Amaro	0,1265	2016-08-21	0
5	99999	João da Silva	Icarai	0,08	2014-09-01	0

44) Se você executar o comando de **ROLLBACK**, terá um erro, porque o comando **COMMIT** encerrou a última transação:



1) Crie uma tabela com o campo auto-incremento, conforme mostrado abaixo:

```
CREATE TABLE TAB_IDENTITY (
    ID INT IDENTITY (1,1) NOT NULL,
    DESCRIPTOR VARCHAR(20) NULL
)
```

2) Insira alguns registros, note que nos comandos de **INSERT** abaixo, não foi especificado o valor para o campo de auto-incremento:

```
INSERT INTO TAB_IDENTITY (DESCRIPTOR) VALUES ('CLIENTE X')
INSERT INTO TAB_IDENTITY (DESCRIPTOR) VALUES ('CLIENTE Y')
INSERT INTO TAB_IDENTITY (DESCRIPTOR) VALUES ('CLIENTE Z')
INSERT INTO TAB_IDENTITY (DESCRIPTOR) VALUES ('CLIENTE W')
INSERT INTO TAB_IDENTITY (DESCRIPTOR) VALUES ('CLIENTE A')
```

3) Se você selecionar dados da tabela, verá o valor do campo auto-incremento que foi incluído automaticamente pelo SQL Server:

```
SELECT * FROM TAB_IDENTITY
```

	ID	DESCRIPTOR
1	1	CLIENTE X
2	2	CLIENTE Y
3	3	CLIENTE Z
4	4	CLIENTE W

4) Exclua um dos registros da tabela acima:

```
DELETE FROM TAB_IDENTITY WHERE ID = 1
```

5) Se você incluir um novo registro:

```
INSERT INTO TAB_IDENTITY (DESCRITOR) VALUES ('CLIENTE A')
```

6) O auto-incremento continuará a crescer seu valor:

```
SELECT * FROM TAB_IDENTITY
```

	ID	DESCRITOR
1	5	CLIENTE A
2	2	CLIENTE Y
3	3	CLIENTE Z
4	4	CLIENTE W

7) Você pode criar um campo de auto-incremento com valor inicial e intervalo de crescimento diferente de **(1,1)**. Para isso, elimine a tabela:

```
DROP TABLE TAB_IDENTITY
```

8) Crie novamente a tabela, mas agora o campo auto-incremento com as propriedades **(100,5)**:

```
CREATE TABLE TAB_IDENTITY (
    ID INT IDENTITY (100,5) NOT NULL,
    DESCRITOR VARCHAR(20) NULL
)
```

9) Inclua novamente os registros:

```
INSERT INTO TAB_IDENTITY (DESCRITOR)
VALUES ('CLIENTE X')
INSERT INTO TAB_IDENTITY (DESCRITOR)
VALUES ('CLIENTE Y')
INSERT INTO TAB_IDENTITY (DESCRITOR)
VALUES ('CLIENTE Z')
INSERT INTO TAB_IDENTITY (DESCRITOR)
VALUES ('CLIENTE W')
INSERT INTO TAB_IDENTITY (DESCRITOR)
VALUES ('CLIENTE A')
```

10) Selecionando os dados:

```
SELECT * FROM TAB_IDENTITY
```

Você terá:

	ID	DESCRITOR
1	100	CLIENTE X
2	105	CLIENTE Y
3	110	CLIENTE Z
4	115	CLIENTE W
5	120	CLIENTE A

11) Crie uma nova consulta no Management Studio e digite os comandos abaixo para criar uma nova tabela. Execute o comando:

```
CREATE TABLE TAB_PADRAO (
    ID INT IDENTITY (1,1) NOT NULL,
    DESCRITOR VARCHAR(20) NULL,
    ENDERECO VARCHAR(200) NULL,
    CIDADE VARCHAR(50) DEFAULT 'Cidade não definida',
    DATA_CRIACAO DATE DEFAULT GETDATE()
)
```

12) Agora digite um novo comando de inclusão de registros:

```
INSERT INTO TAB_PADRAO (DESCRITOR, ENDERECO, CIDADE, DATA_CRIACAO)
VALUES ('CLIENTE X', 'RUA PROJETADA A', 'SÃO PAULO', '2018-04-30')
```

13) Fazendo a seleção dos registros, você terá:

```
SELECT * FROM TAB_PADRAO
```

	ID	DESCRITOR	ENDERECO	CIDADE	DATA_CRIACAO
1	1	CLIENTE X	RUA PROJETADA A	SÃO PAULO	2018-04-30

14) Agora, inclua novo registro, omitindo uma série de valores:

```
INSERT INTO TAB_PADRAO (DESCRITOR) VALUES ('CLIENTE X')
```

15) Se você observar a tabela, verá uma série de valores nulos para os campos omitidos, exceto no campo **CIDADE**, que possui um valor padrão definido na criação da tabela:

	ID	DESCRIPTOR	ENDERECO	CIDADE	DATA_CRIACAO
1	1	CLIENTE X	RUA PROJETADA A	SAO PAULO	2018-04-30
2	2	CLIENTE X	NULL	Cidade não definida	2018-04-30

16) Crie uma nova consulta no Management Studio e digite os comandos abaixo para criar uma nova tabela::

```
CREATE TABLE TAB_FATURAMENTO (
    DATA_VENDA DATE NULL,
    TOTAL_VENDA FLOAT
)
```

17) Agora, crie uma **TRIGGER** que irá apagar todo o conteúdo da tabela criada anteriormente e calcular o faturamento total de todas as notas fiscais por dia:

```
CREATE TRIGGER TG_ITENS_VENDIDOS
ON [ITENS VENDIDOS]
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
DELETE FROM TAB_FATURAMENTO;
INSERT INTO TAB_FATURAMENTO (DATA_VENDA, TOTAL_VENDA)
SELECT A.DATA AS DATA_VENDA,
       SUM(B.QUANTIDADE * B.[PREÇO]) AS TOTAL_VENDA
FROM NOTAS A INNER JOIN [ITENS VENDIDOS] B
ON A.NÚMERO = B.NÚMERO
GROUP BY A.DATA;
END;
```

18) insira uma nota fiscal e alguns itens:

```
INSERT INTO [NOTAS] ([NÚMERO], DATA, CPF, MATRICULA, IMPOSTO)
VALUES ('0100', '2018-05-15', '1471156710', '235', 0.18)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0100', '1000889', 100, 1)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0100', '1002334', 100, 1)
```

19) Após a inclusão desta nota fiscal, verifique a tabela com os faturamentos consolidados:

```
SELECT * FROM TAB_FATURAMENTO
```

	DATA_VENDA	TOTAL_VENDA
1	2018-04-30	90
2	2018-05-01	90
3	2018-05-15	200

20) Insira uma nova nota fiscal com seus respectivos itens:

```
INSERT INTO [NOTAS] ([NÚMERO], DATA, CPF, MATRICULA, IMPOSTO)
VALUES ('0101', '2018-05-15', '1471156710', '235', 0.18)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0101', '1000889', 100, 1)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0101', '1002334', 100, 1)
```

21) Verificando a tabela de faturamentos consolidados, você verá novos valores:

```
SELECT * FROM TAB_FATURAMENTO
```

	DATA_VENDA	TOTAL_VENDA
1	2018-04-30	90
2	2018-05-01	90
3	2018-05-15	400

22) Mais uma nota, agora para o dia 16, e você terá novos valores:

```
INSERT INTO [NOTAS] ([NÚMERO], DATA, CPF, MATRICULA, IMPOSTO)
VALUES ('0102', '2018-05-16', '1471156710', '235', 0.18)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0102', '1000889', 100, 1.5)
INSERT INTO [ITENS VENDIDOS] ([NÚMERO], [CÓDIGO], [QUANTIDADE], [PREÇO])
VALUES ('0102', '1002334', 200, 1)
```

Depois:

```
SELECT * FROM TAB_FATURAMENTO
```

	DATA_VENDA	TOTAL_VENDA
1	2018-04-30	90
2	2018-05-01	90
3	2018-05-15	400
4	2018-05-16	350

23) Qualquer alteração na tabela irá mudar a tabela de faturamentos, já que você colocou a propriedade **AFTER INSERT, UPDATE, DELETE**. Se você executar o comando de exclusão de um item de nota:

```
DELETE FROM [ITENS VENDIDOS] WHERE [NÚMERO] = '0102' AND [CÓDIGO] = '1002334'
```

Terá novos valores na tabela de faturamento consolidado:

	DATA_VENDA	TOTAL_VENDA
1	2018-04-30	90
2	2018-05-01	90
3	2018-05-15	400
4	2018-05-16	350

24) Também haverá mudanças ao executar um **UPDATE**:

```
UPDATE [ITENS VENDIDOS] SET [QUANTIDADE] = 300
WHERE [NÚMERO] = '0102' AND [CÓDIGO] = '1000889'
```

	DATA_VENDA	TOTAL_VENDA
1	2018-04-30	90
2	2018-05-01	90
3	2018-05-15	400
4	2018-05-16	450

25) Crie uma nova consulta no Management Studio e digite os comandos abaixo para criar uma nova tabela:

```
CREATE TABLE TAB_CHECK (
    ID INT NOT NULL,
    NOME VARCHAR(20) NULL,
    IDADE INT NULL,
    CIDADE VARCHAR(20) NULL,
    CONSTRAINT CHK_PESSOA CHECK (IDADE >= 18)
```

)

26) Na tabela acima, a idade foi restringida para maior de 18 anos. Se você executar o comando:

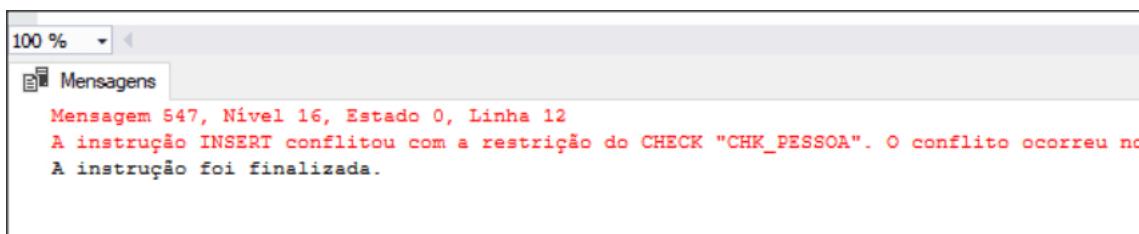
```
INSERT INTO TAB_CHECK (ID, NOME, IDADE, CIDADE)
VALUES (1, 'JOÃO', 19, 'RIO DE JANEIRO')
```

Ele será incluído na tabela.

27) Mas se você executar outro **INSERT**, conforme mostrado abaixo:

```
INSERT INTO TAB_CHECK (ID, NOME, IDADE, CIDADE)
VALUES (2, 'PEDRO', 17, 'SÃO PAULO')
```

Terá a mensagem:



28) Se corrigir para:

```
INSERT INTO TAB_CHECK (ID, NOME, IDADE, CIDADE)
VALUES (2, 'PEDRO', 20, 'SÃO PAULO')
```

Agora sim a informação é incluída.

29) A condição do **CHECK** pode ser composta com **AND** ou **OR**, por exemplo.

Crie outra tabela:

```
CREATE TABLE TAB_CHECK2 (
    ID INT NOT NULL,
    NOME VARCHAR(20) NULL,
    IDADE INT NULL,
    CIDADE VARCHAR(20) NULL,
    CONSTRAINT CHK_PESSOA2
        CHECK (IDADE >= 18 AND CIDADE = 'RIO DE JANEIRO')
)
```

30) Ao incluir um novo registro:

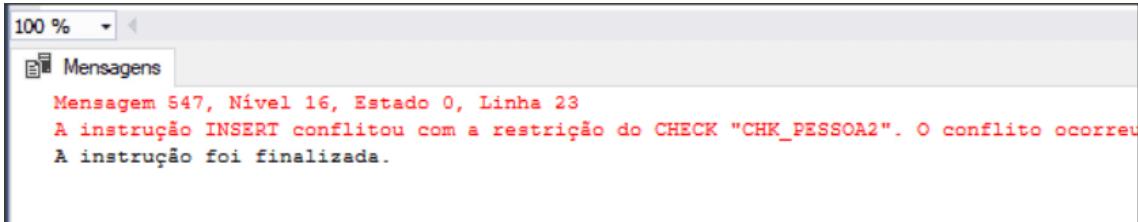
```
INSERT INTO TAB_CHECK2 (ID, NOME, IDADE, CIDADE)
VALUES (1, 'JOÃO', 19, 'RIO DE JANEIRO')
```

Não haverá problemas.

31) Mas se você digitar:

```
INSERT INTO TAB_CHECK2 (ID, NOME, IDADE, CIDADE)
VALUES (2, 'PEDRO', 20, 'RIO DE JANEIRO')
```

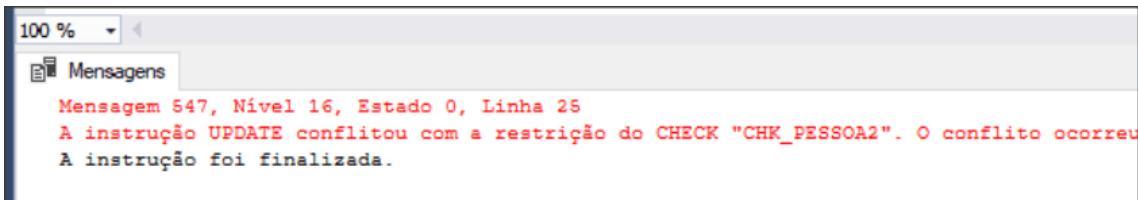
O problema de integridade será apresentado, já que a idade está correta, mas a cidade não:



The screenshot shows the 'Mensagens' (Messages) window in SSMS. The message is: 'Mensagem 547, Nível 16, Estado 0, Linha 23 A instrução INSERT conflitou com a restrição do CHECK "CHK\_PESSOA2". O conflito ocorreu. A instrução foi finalizada.' (Message 547, Level 16, State 0, Line 23. The INSERT statement conflicted with the CHECK constraint "CHK\_PESSOA2". The conflict occurred. The statement has been terminated.)

32) A restrição é testada não somente durante o **INSERT**, mas também durante o **UPDATE**:

```
UPDATE TAB_CHECK2 SET CIDADE = 'SÃO PAULO' WHERE ID = 2
```



The screenshot shows the 'Mensagens' (Messages) window in SSMS. The message is: 'Mensagem 547, Nível 16, Estado 0, Linha 25 A instrução UPDATE conflitou com a restrição do CHECK "CHK\_PESSOA2". O conflito ocorreu. A instrução foi finalizada.' (Message 547, Level 16, State 0, Line 25. The UPDATE statement conflicted with the CHECK constraint "CHK\_PESSOA2". The conflict occurred. The statement has been terminated.)

[00:00] Já sabemos criar banco de dados. O próximo passo é criar tabelas. Como vimos na definição de "tabelas", que foi colocada no início do curso, sabemos que para definirmos uma tabela, precisamos basicamente dizer quantos campos essa tabela vai ter e qual é o tipo de cada campo.

[00:25] Um dado dentro de um campo, não pode ter tipos diferentes. Então, óbvio, para criar tabela, a primeira coisa que temos que aprender é: quais são os tipos disponíveis para usarmos na criação da tabela no SQL Server?

[00:44] Nós temos vários tipos, que são organizados em grupos. O primeiro grupo que eu estou mostrando aqui em cima, são os numéricos exatos. Ele tem esse nome "numéricos exatos", porque eu defino muito bem o tamanho e o número de casas decimais que esses números vão ter.

[01:04] A gente começa pelo Bigint. O Bigint, como o próprio nome em inglês diz, é um inteiro grande, ou seja, é um número que não tem casa decimal e possui um limite, inferior e superior, que está sendo mostrado aqui em cima. Ele vai de  $-2^{63}$  a  $+2^{63}-1$ . Então, é assim o limite do Bigint.

[01:34] O Numeric, é um número com casas decimais, porém, com casas decimais fixas. Se eu digo que um Numeric tem cinco decimais e se eu representar, por exemplo o número 1 dentro do banco de dados, ele vai ser representado com 1,00000, ou seja, cinco casas decimais. Independente do número ser inteiro ou possuir casas decimais realmente.

[02:02] No Numeric, eu posso definir números de até 38 casas decimais. O tamanho desse campo vai variar pelo número de casas decimais que eu definir.

[02:14] O Bit, é um inteiro que me dá o valor 1 ou 0. Ele é usado normalmente para definir campos lógicos. Normalmente utilizamos o 1 como verdadeiro e o 0 como falso, ou vice-versa, não lembro direito, mas acho que é isso mesmo: o 1 é verdadeiro e o 0, falso. Ou seja, é um campo que dá um status por exemplo, do que ele significa.

[02:44] O Smallint, é um inteiro menor que tem as mesmas definições do Bigint e cujo os limites são um pouco menores do que o Bigint. Na verdade, são bem menores, praticamente quatro vezes menores.

[03:00] O Decimal é igual ao Numeric em sua definição. Tanto faz você definir o campo como decimal ou numeric, eles são a mesma coisa: números com casas decimais fixas.

[03:12] Smallmoney. Normalmente um campo money representa valores financeiros, e além de ter casas decimais, esse valores financeiros são representados dentro do banco de dados com um prefixo, que é o símbolo da moeda que ele representa. Pode ser o símbolo do dólar, do real, do euro, e dependendo do idioma ou da propriedade específica do idioma em que o SQL Server está rodando, você já vai ter automaticamente esses símbolos.

[03:50] Claro que ele é Smallmoney, porque o limite de valores dele é menor, vai de (-214.000 a +214.000 e uns quebrados).

[04:03] O Int, é o inteiro cujo o tamanho está entre o Bit e o Smallint. Ele vai de -2^31 a +2^31-1.

[04:15] O Tinyint, é um inteiro bem pequeno, eu o represento de 0 a 255.

[04:23] O Money, já é a representação de valores financeiros um pouco maior. Na verdade, muito maior, vai de -922.337.203.685.477. Então, o menor valor e o maior valor, é o mesmo menos um.

[04:53] Claro, o Money também tem o símbolo da moeda representado na frente. Esses são os numéricos exatos. Partiremos para o segundo grupo: os numéricos aproximados.

[05:04] Eles têm esse nome, porque na verdade, fazem um arredondamento, são o que chamamos de "números com ponto flutuante". Nós temos dois tipos: o Float e o Real.

[05:19] O Float é o número que tem um range de tamanho, muito maior, ou seja, a sua representação é muito maior. Enquanto o Real, é esse número com um limite um pouco menor.

[05:37] Data e hora: Os campos do tipo data e hora, representam algum período do tempo. Então, nós temos o Data, que me dá o dia, mês e ano. Na verdade, normalmente eu represento no SQL Server ano, mês e dia.

[05:55] Depois, nós vamos ver em outros tópicos do curso, que nós podemos por exemplo, exibir o resultado em determinados formatos diferentes.

[06:05] O DateTimeOffset, é o ano/mês/dia, com hora/minuto/segundo, porém, com a informação do fuso horário. Dependendo de como estou manipulando o meu dado, vou ter além da hora, qual é o deslocamento em relação ao ponto de Greenwich que aquele horário representa.

[06:32] Eu acho que aqui, por exemplo, Rio de Janeiro, São Paulo ou sudeste do Brasil, se eu não me engano... Eu posso estar falando besteira, porque eu não conferi essa informação antes, mas acho que é -4 em relação à Greenwich.

[06:49] DateTime2: eu vejo a data, porém só com hora e minuto, o segundo sempre vai vir 00.

[07:02] O Smalldate, é a data com a hora apenas, não tem minutos e não tem segundos.

[07:10] O Datetime, nessa sim tenho a data, com a hora, minuto e segundo, porém, não tem a representação do fuso horário. Quem tem isso é o DateTimeOffset.

[07:22] E com o Time, eu só vejo a representação da hora.

[07:31] Cadeia de caracteres: nós temos dois grupos, as unicodes e as não unicodes. Já vou explicar o que significa unicode caso você não saiba. Primeiro, eu tenho o Char, o Varchar e o Texto. Os três são cadeias de caracteres não unicode. Sendo que o Char tem o valor fixo de 1 a 8000. O Varchar, também.

[07:58] A diferença do Char e do Varchar, é a seguinte: o Varchar tem o tamanho variável, ou seja, se eu digo que o meu Varchar é tamanho 10, mas eu salvo no campo uma palavra com dois caracteres, ele só vai armazenar aqueles dois caracteres.

[08:17] Se o campo é um Char de 10, e eu gravo o mesmo campo com dois caracteres, ele vai ter que salvar dois caracteres e oito caracteres vazios, ou seja, ele vai ocupar os dez caracteres. Então, a diferença do Char para o Varchar, é que o Char é fixo, e o Varchar é variável.

[08:38] Tem uma outra coisa. O Varchar tem um parâmetro chamado Max, que é para eu representar caracteres muito grandes. Se o campo for um Varchar de Max, ele pode receber até dois gigabytes de tamanho.

[08:58] E o campo Texto, é uma cadeia de caracteres não unicode, mas que representa dados maiores do que o Varchar pode armazenar, ou dados binários. Bem, nChar, nVarchar e nTexto, têm as mesmas definições que o Char, o Varchar e Texto, porém, agora, são caracteres unicode.

[09:26] Bem, é aquela pergunta: o que significa "unicode"? Unicode significa uma cadeia de caracteres que fornece num único conjunto de dados, todos os caracteres de todos os idiomas do mundo.

[09:42] Nós, ocidentais, português e línguas latinas, por exemplo, usamos o abecedário normal, que vai de A a Z. Porém, existem outras línguas como o russo, grego, hebraico, árabe. As línguas orientais, como por exemplo, chinês, coreano, japonês, que utilizam uma cadeia de caracteres diferentes, são outros símbolos, outras formas de representar as letras.

[10:12] Então, se eu vou armazenar no meu banco de dados uma informação que possui caracteres diferentes dos normais, tenho que usar o nVarchar ou nChar ou nTexto, porque ele vai acertar cadeias de caracteres unicode.

[10:33] Finalmente, caracteres binários, que representam dados de, por exemplo: eu tenho lá no meu diretório um arquivo-texto salvo. Eu posso pegar aquele arquivo-texto e salvar num campo do meu SQL Server. Ele será salvo como "binário".

[10:55] A mesma coisa vai acontecer com o VarBinário. A diferença, está no tamanho. O Binário armazena até 8.000 caracteres de tamanho, enquanto o VarBinário, chega até q  $2^{31}-1$  caracteres.

Ex:

```
CREATE TABLE [TABELA DE CLIENTES]
(
[CPF] [VARCHAR] (11) ,
[NOME] [VARCHAR] (100) ,
[ENDERECO 1] [VARCHAR] (150) ,
[ENDERECO 2] [VARCHAR] (150) ,
[BAIRRO] [VARCHAR] (50) ,
[CIDADE] [VARCHAR] (50) ,
[ESTADO] [VARCHAR] (2) ,
[CEP] [VARCHAR] (8) ,
[DATA DE NASCIMENTO] [DATE],
[IDADE] [SMALLINT],
[SEXO] [VARCHAR] (1) ,
[LIMITE DE CREDITO] [MONEY] ,
[VOLUME DE COMPRA] [FLOAT] ,
[PRIMEIRA COMPRA] [BIT]
)
```

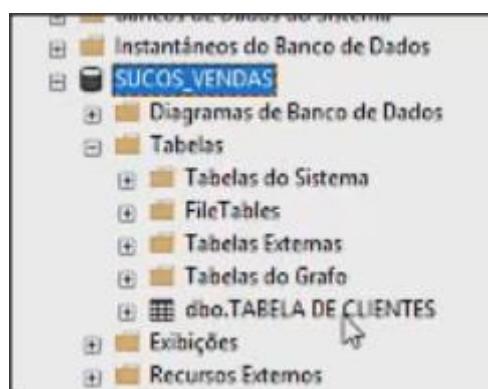
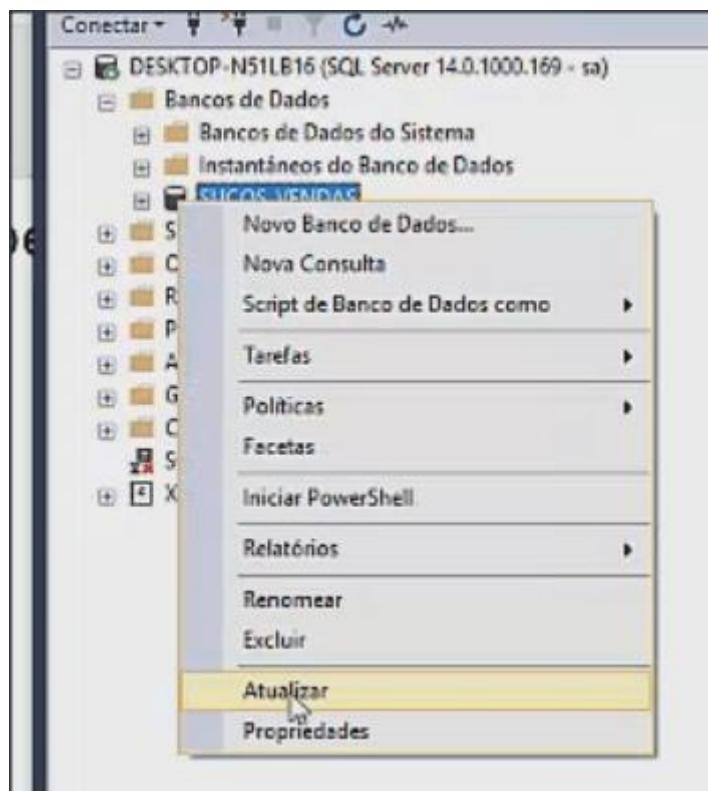
## Criando a primeira tabela

1) Acesse o **Management Studio**.

2) Crie a tabela de cliente, digitando o comando abaixo:

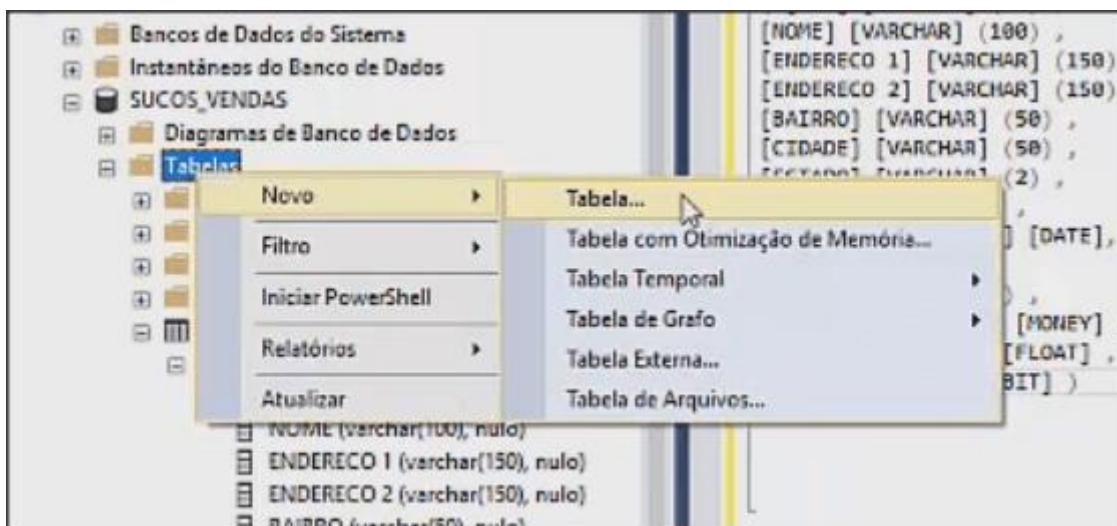
```
CREATE TABLE [TABELA DE CLIENTES] (
[CPF] [VARCHAR] (11) ,
[NOME] [VARCHAR] (100) ,
[ENDERECO 1] [VARCHAR] (150) ,
[ENDERECO 2] [VARCHAR] (150) ,
[BAIRRO] [VARCHAR] (50) ,
[CIDADE] [VARCHAR] (50) ,
[ESTADO] [VARCHAR] (2) ,
[CEP] [VARCHAR] (8) ,
[DATA DE NASCIMENTO] [DATE],
[IDADE] [SMALLINT],
[SEXO] [VARCHAR] (1) ,
[LIMITE DE CREDITO] [MONEY] ,
[VOLUME DE COMPRA] [FLOAT] ,
[PRIMEIRA COMPRA] [BIT]
)
```

3) Execute o comando e depois atualize a árvore de objetos do SQL Server para observar a nova tabela criada:



## Criando uma tabela através de assistente

4) Crie agora uma outra tabela, porém usando o **Assistente do Management Studio**. Clique com o botão da direita do mouse sobre o nome do banco de dados e selecione **Novo --> Tabela**:

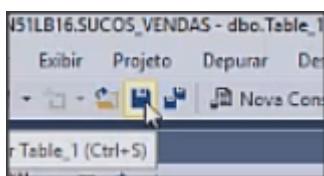


5) No *grid*, preencha com os dados da nova tabela, conforme mostrado abaixo:

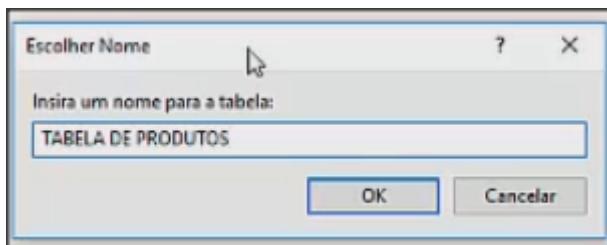
- [CÓDIGO DO PRODUTO] Varchar (10)
- [NOME DO PRODUTO] Varchar (50)
- [EMBALAGEM] Varchar (20)
- [TAMANHO] Varchar (10)
- [SABOR] Varchar (20)
- [PREÇO DE LISTA] Smallmoney

	Nome da Coluna	Tipo de Dados	Permitir Nul...
	[CÓDIGO DO PRODUTO]	varchar(10)	<input checked="" type="checkbox"/>
	[NOME DO PRODUTO]	varchar(50)	<input checked="" type="checkbox"/>
	EMBALAGEM	varchar(20)	<input checked="" type="checkbox"/>
	TAMANHO	varchar(10)	<input checked="" type="checkbox"/>
▶	SABOR	varchar(20)	<input checked="" type="checkbox"/>
	[PREÇO DE LISTA]	smallmoney	<input type="checkbox"/>

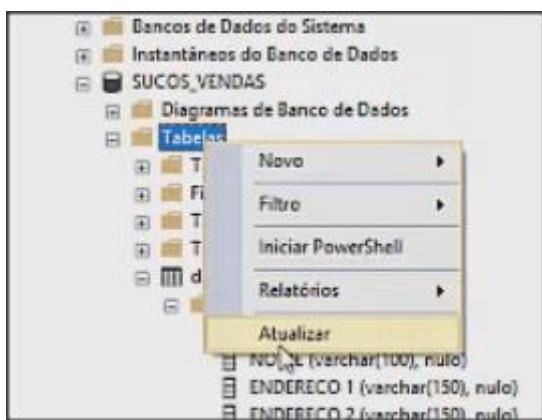
6) Clique no botão **Salvar**:



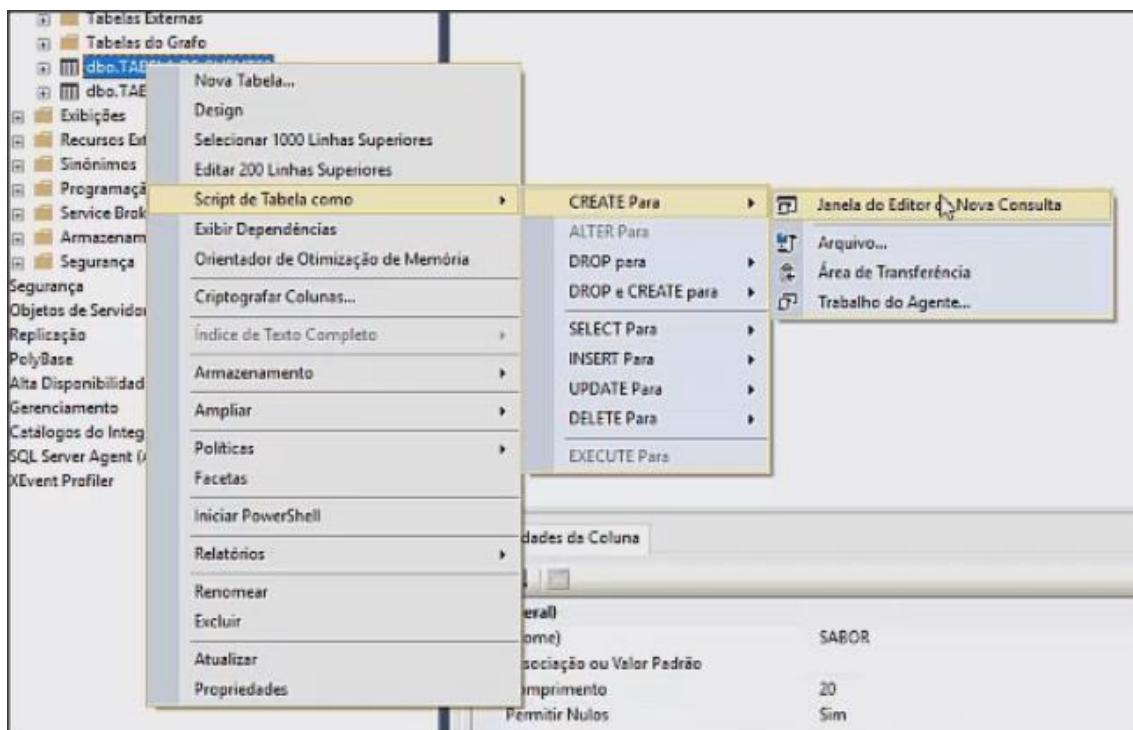
7) Inclua o nome da tabela:



8) Clique em **OK**, depois clique com o botão da direita do mouse sobre **Tabelas** e selecione a opção **Atualizar**. Você verá a nova tabela no banco de dados:



9) Clique com o botão da direita do mouse sobre a tabela de clientes e selecione **Script de Tabela como --> CREATE Para --> Janela do Editor da Nova Consulta**:



10) Você terá:

```

SQLQuery4.sql - DE_S_VENDAS (sa (55))  DESKTOP-N5ILB16\ELA DE PRODUÇÃO
USE [SUCOS_VENDAS]
GO

/****** Object: Table [dbo].[TABELA DE CLIENTES] Script
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[TABELA DE CLIENTES](
    [CPF] [varchar](11) NULL,
    [NOME] [varchar](100) NULL,
    [ENDERECO 1] [varchar](150) NULL,
    [ENDERECO 2] [varchar](150) NULL,
    [BAIRRO] [varchar](50) NULL,
    [CIDADE] [varchar](50) NULL,
    [ESTADO] [varchar](2) NULL,
    [CEP] [varchar](8) NULL,
    [DATA DE NASCIMENTO] [date] NULL,
    [IDADE] [smallint] NULL,
    [SEXO] [varchar](1) NULL,
    [LIMITE DE CREDITO] [money] NULL,
    [VOLUME DE COMPRA] [float] NULL,
    [PRIMEIRA COMPRA] [bit] NULL
) ON [PRIMARY]
GO

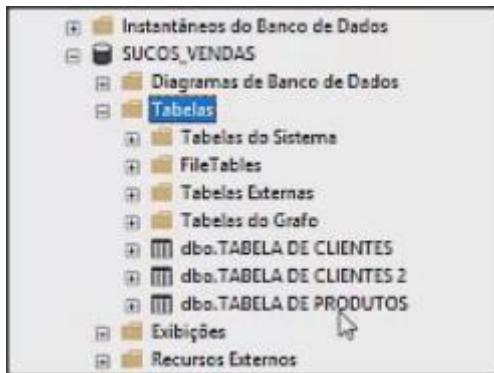
```

11) Modifique o nome da tabela no script para **TABELA DE CLIENTES 2**:

```
SET QUOTED_IDENTIFIER ON
GO

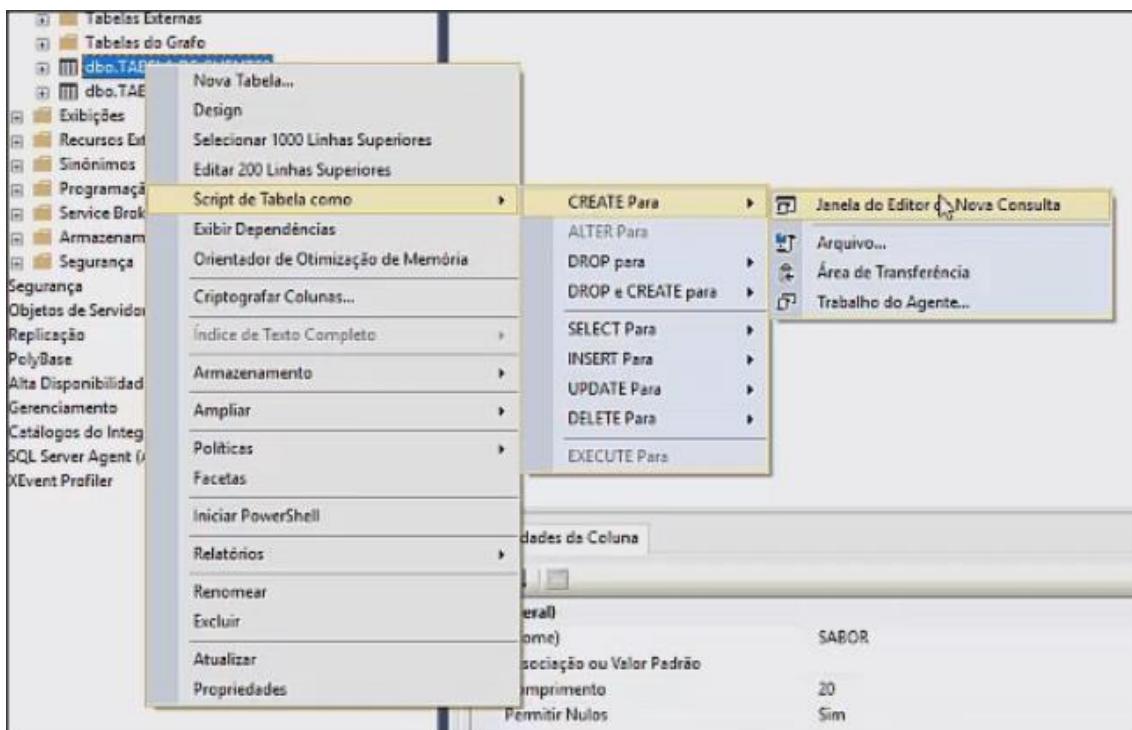
CREATE TABLE [dbo].[TABELA DE CLIENTES 2](
    [CPF] [varchar](11) NULL,
    [NOME] [varchar](100) NULL,
    [ENDERECO 1] [varchar](150) NULL,
```

12) Execute. Depois de atualizar os objetos você terá a nova tabela listada:



### Apagando uma tabela

13) Clique com o botão da direita do mouse sobre **Tabela de Clientes** e selecione **Script de Tabela como --> CREATE Para --> Janela do Editor da Nova Consulta**:



14) Edite o nome da tabela para **Tabela de Clientes 3**:

```

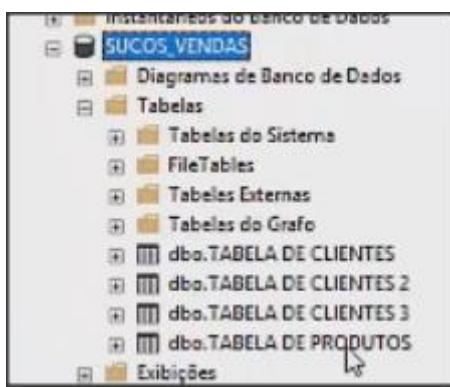
    SET ANSI_NULLS ON
    GO

    SET QUOTED_IDENTIFIER ON
    GO

    CREATE TABLE [dbo].[TABELA DE CLIENTES 3](
        [CPF] [varchar](11) NULL,
        [NOME] [varchar](100) NULL,
        [ENDERECO 1] [varchar](150) NULL,
        [ENDERECO 2] [varchar](150) NULL,
        [BAIRRO] [varchar](50) NULL,
    )

```

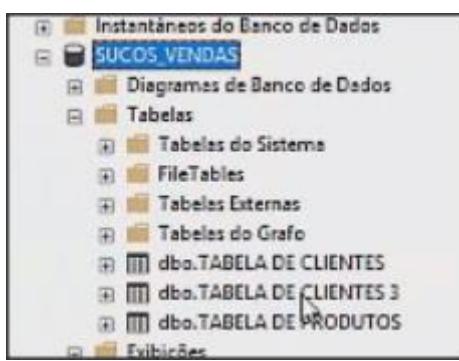
15) Execute, atualize e verifique se a tabela foi criada:



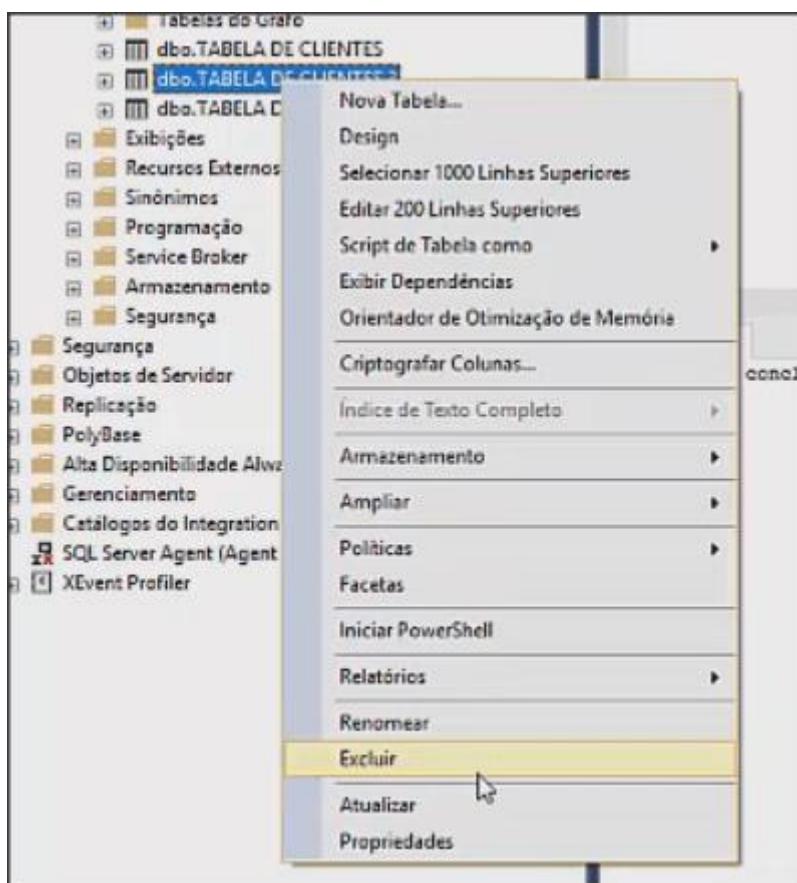
16) Apague a **Tabela de Clientes 2**. Para isso, crie uma nova consulta e digite o comando:

```
DROP TABLE [dbo].[TABELA DE CLIENTES 2]
```

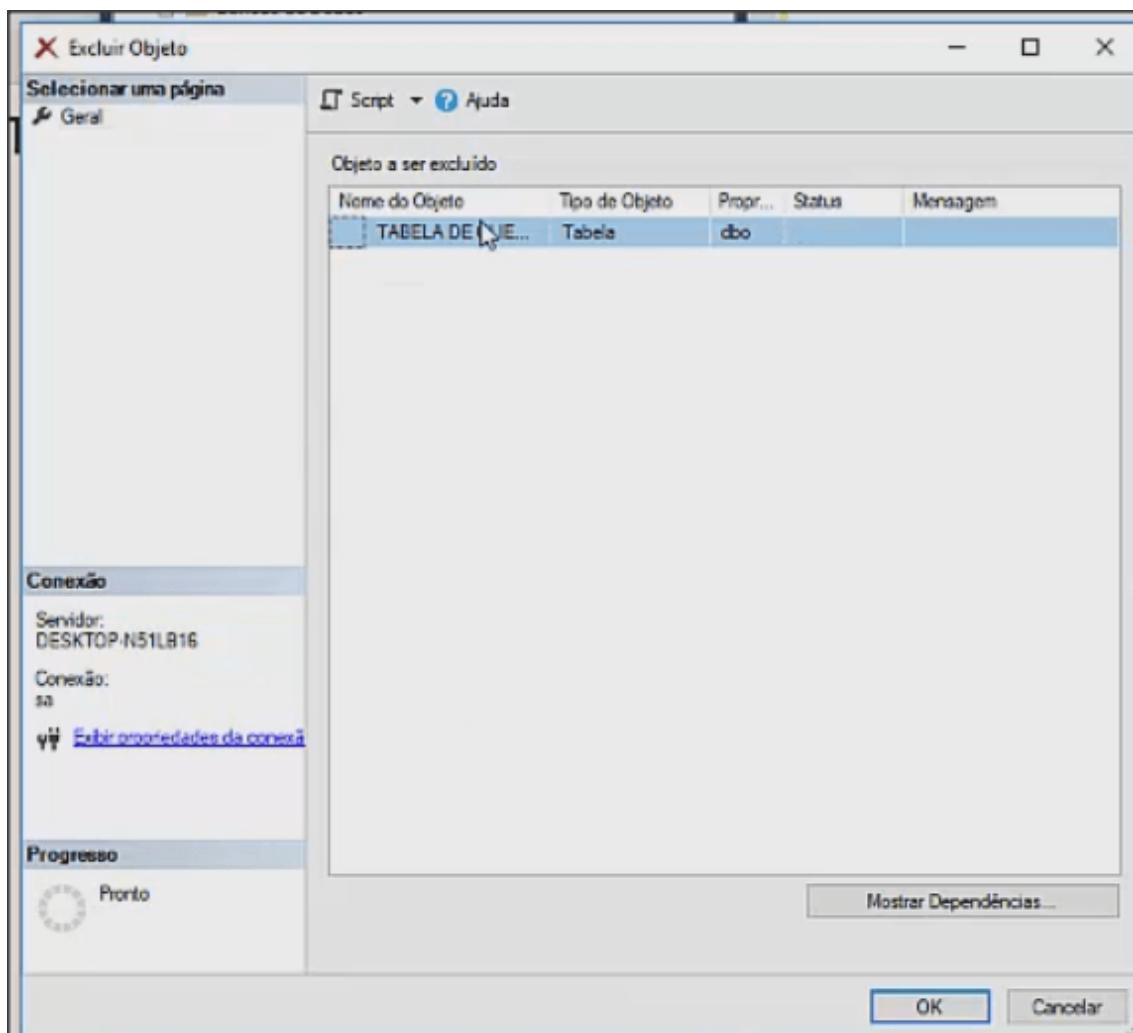
17) Execute o comando, atualize as tabelas e verifique que a tabela não existe mais:



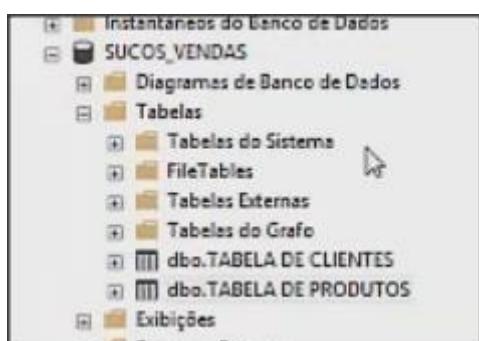
18) Exclua a **Tabela de Clientes 3** através do assistente. Para isso, clique com o botão da direita do mouse sobre ela e escolha a opção **Excluir**:



19) Você pode ver a caixa de diálogo abaixo:



20) Clique em **OK**. A tabela foi excluída:



```
use [SUCOS_VENDAS]

INSERT INTO [TABELA DE PRODUTOS]
([CÓDIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO], [SABOR], [PREÇO DE LISTA])
VALUES
('1040107', 'Light - 350 ml - Melancia', 'Lata', '350 ml', 'Melancia', 4.56),
('1037797', 'Clean - 2 Litros - Laranja', 'PET', '2 - Litros', 'Laranja', 16.01),
('1000889', 'Sabor da Montanha - 700 ml - Uva', 'Garrafa', '700 ml', 'Uva', 6.31),
```

```

('1004327','Videira do Campo - 1,5 Litros - Melancia', 'PET', '1,5
Litros', 'Melancia', 19.51)

select * from [TABELA DE PRODUTOS]

UPDATE [TABELA DE PRODUTOS] SET
[EMBALAGEM] = 'Lata',
[PREÇO DE LISTA] = 2.46
WHERE [CODIGO DO PRODUTO] = '544931'

UPDATE [TABELA DE PRODUTOS] SET
[EMBALAGEM] = 'Garrafa'
WHERE [CODIGO DO PRODUTO] = '1078680'

DELETE FROM [TABELA DE PRODUTOS]
WHERE [CODIGO DO PRODUTO] = '1088126'

ALTER TABLE [TABELA DE PRODUTOS]
ADD CONSTRAINT PK_PRODUTOS
PRIMARY KEY CLUSTERED ([CODIGO DO PRODUTO])

ALTER TABLE [TABELA DE PRODUTOS]
ALTER COLUMN [CODIGO DO PRODUTO]
VARCHAR(10) NOT NULL

```

Dica para inclusão de data (ano-mês-dia) >> '2021-11-02'

## Incluindo dados na tabela

1) Acesse o Management Studio.

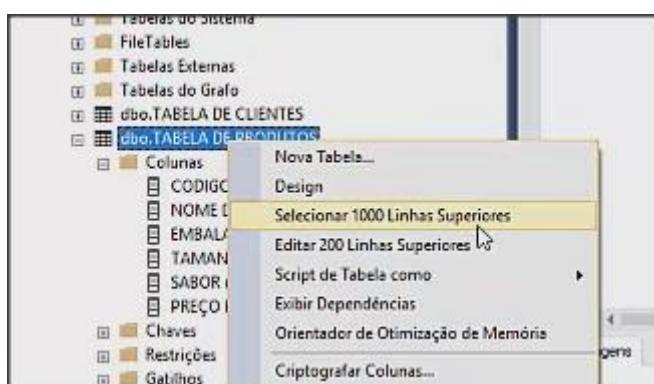
2) Crie uma nova consulta e digite o comando abaixo:

```

INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('1040107','Light - 350 ml - Melancia', 'Lata', '350 ml', 'Melancia',
4.56)

```

3) Clique com o botão da direita do mouse sobre a tabela **Tabela de Produtos** e escolha a opção **Selecionar 1000 Linhas Superiores**:



4) Você verá que o novo registro foi incluído na tabela:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350ml - Melância	Lata	350 ml	Melância	4.56

### Incluindo registros ao mesmo tempo

5) Você pode incluir vários registros. Crie uma nova consulta e digite:

```
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('1037797', 'Clean - 2 Litros - Laranja', 'PET', '2 Litros', 'Laranja',
16.01)

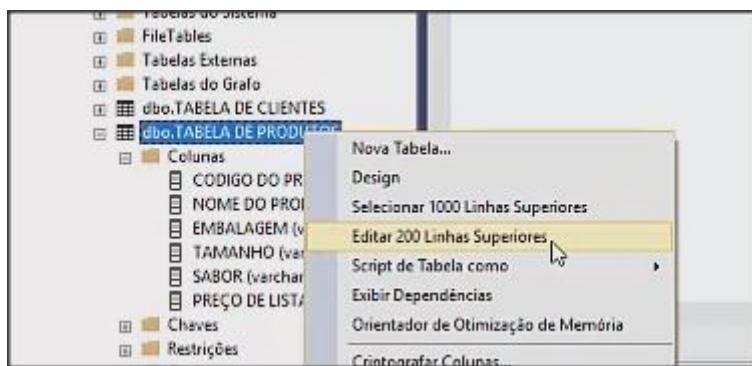
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('1000889', 'Sabor da Montanha - 700 ml - Uva', 'Garrafa', '700 ml',
'Uva', 6.31)
```

6) Execute o comando e verifique o conteúdo da tabela, escolhendo a opção **Selecionar 1000 Linhas Superiores**:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350ml - Melância	Lata	350 ml	Melância	4.56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16.01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.31

### Incluindo um registro direto na tabela

7) Clique com o botão da direita do mouse sobre a tabela **Tabela de Produtos** e selecione **Editar 200 Linhas Superiores**:



8) Digite o novo campo no *grid*:

- **1004327**
- **Videira do Campo 1,5 Litros Melancia**
- **PET**
- **1,5 Litros**
- **Melancia**
- **19,51**

	1037797	Clean - 2 Litros ...	PET	2 Litros	Laranja	16,0100
	1000889	Sabor da Mont...	Garrafa	700 ml	Uva	6,3100
»	1004327	Videira do Cam...	PET	1,5 Litros	Melância	19,5100
»	NULL	NULL	NULL	NULL	NULL	NULL

9) Inclua mais uma linha com dados:

- **1088216**
- **Linha Citros 1 Litro Limão**
- **PET**
- **1 Litro**
- **Limão**
- **7,00**

	1000889	Sabor da Mont...	Garrafa	700 ml	Uva	6,3100
	1004327	Videira do Cam...	PET	1,5 Litros	Melância	19,5100
»	1008126	Linha Citros - 1 ...	PET	1 Litro	Limão	7,0000
»	NULL	NULL	NULL	NULL	NULL	NULL

10) Verifique o conteúdo da tabela selecionando as 1000 primeiras linhas superiores:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4.56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16.01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.31
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19.51
5	1088126	Linha Citrus - 1 Litro - Limão	PET	1 Litro	Limão	7.00

## Alterando registros

- 11) Crie uma nova consulta. Edite os comandos abaixo:

```
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('544931', 'Frescor do Verão - 350 ml - Limão', 'PET', '350
ml', 'Limão', 3.20)

INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('1078680', 'Frescor do Verão - 470 ml - Manga', 'Lata', '470
ml', 'Manga', 5.18)
```

- 12) Execute e confira se estes novos registros foram incluídos na tabela:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4.56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16.01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.31
4	544931	Frescor do Verão - 350 ml - Limão	PET	350 ml	Limão	3.20
5	1078680	Frescor do Verão - 470 ml - Manga	Lata	470 ml	Manga	5.18
6	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19.51
7	1088126	Linha Citrus - 1 Litro - Limão	PET	1 Litro	Limão	7.00

- 13) Altere estes dois registros novos. Para isto, crie uma nova consulta e digite:

```
UPDATE [TABELA DE PRODUTOS] SET
[EMBALAGEM] = 'Lata',
[PREÇO DE LISTA] = 2.46
WHERE [CODIGO DO PRODUTO] = '544931'

UPDATE [TABELA DE PRODUTOS] SET
[EMBALAGEM] = 'Garrafa'
```

```
WHERE [CODIGO DO PRODUTO] = '1078680'
```

- 14) Execute e confira se estes novos registros foram alterados na tabela:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,31
4	544931	Frescor do Verão - 350 ml - Limão	Lata	350 ml	Limão	2,46
5	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,18
6	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
7	1088126	Linha Citros - 1 Litro - Limão	PET	1 Litro	Limão	7,00

- 15) Você pode alterar o registros através da edição diretamente na tabela pelo *grid*. Para isso, clicando com o botão da direita do mouse sobre **Tabela de Produtos** e selecionando a opção **Editar 200 Linhas Superiores**, você pode modificar os registros:

CODIGO DO P...	NOME DO PR...	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,5600
1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,0100
1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,3100
544931	Frescor do Verão - 350 ml - Limão	Lata	350 ml	Limão	2,4600
1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1800
1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,5100
1088126	Linha Citros - 1 Litro - Limão	Garrafa	1 Litro	Limão	7,0000
**	NULL	NULL	NULL	NULL	NULL

## Excluindo registros

- 16) Crie uma nova consulta. Edite os comandos abaixo:

```
DELETE FROM [TABELA DE PRODUTOS]
WHERE [CODIGO DO PRODUTO] = '1088216'
```

- 17) Execute e verifique que o registro escolhido foi removido:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melancia	Lata	350 ml	Melancia	4,56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,31
4	544931	Frescor do Verão - 350 ml - Limão	Lata	350 ml	Limão	2,46
5	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,18
6	1004327	Vidella do Campo - 1,5 Litros - Melancia	PET	1,5 Litros	Melancia	19,51

## Chave primária

18) Crie uma nova consulta. Edite os comandos abaixo:

```
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('788975', 'Pedaços de Frutas - 1,5 Litros - Maça', 'PET', '1,5
Litros', 'Maça', 18.01)

INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('788975', 'Pedaços de Frutas - 1,5 Litros - Maça', 'PET', '1,5
Litros', 'Maça', 18.01)
```

19) Ao executar, você irá iremos observar que dois registros foram gravados na tabela com códigos repetidos:

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melancia	Lata	350 ml	Melancia	4,56
2	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,01
3	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,31
4	544931	Frescor do Verão - 350 ml - Limão	Lata	350 ml	Limão	2,46
5	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,18
6	788975	Pedaços de Frutas - 1,5 Litros - Maça	PET	1,5 Litros	Maça	18,01
7	788975	Pedaços de Frutas - 1,5 Litros - Maça	PET	1,5 Litros	Maça	18,01

20) Crie a chave primária para evitar que isso acima aconteça. Mas antes, exclua estes registros repetidos, digitando e executando o comando:

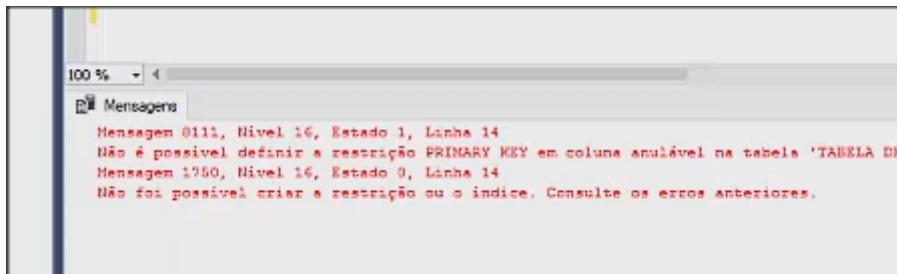
```
DELETE FROM [TABELA DE PRODUTOS] WHERE [CODIGO DO PRODUTO] = '788975'
```

21) Digite o comando de criação da chave primária. Digite:

```
ALTER TABLE [TABELA DE PRODUTOS]
```

```
ADD CONSTRAINT PK_PRODUTOS
PRIMARY KEY CLUSTERED ([CODIGO DO PRODUTO])
```

22) Execute o comando. Você terá uma mensagem de erro:



Isto acontece por causa da propriedade **NULL** dos campos que estão virando chaves primárias.

23) Digite o comando abaixo:

```
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[PREÇO DE LISTA])
VALUES
('788975', 'Pedaços de Frutas - 1,5 Litros - Maça', 'PET', '1,5
Litros', 18.01)
```

Nele, você não irá incluir dados para a coluna **SABOR**.

24) Consulte agora a tabela. Veja que a coluna **SABOR** para este novo campo está preenchido com **NULL**:

1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,01
1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,31
544931	Frescor do Verão - 350 ml - Limão	Lata	350 ml	Limão	2,40
1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,18
788975	Pedaços de Frutas - 1,5 Litros - Maça	PET	1,5 Litros	NULL	18,01

## Resolvendo o problema da chave primária

25) Crie uma nova consulta. Edite os comandos abaixo:

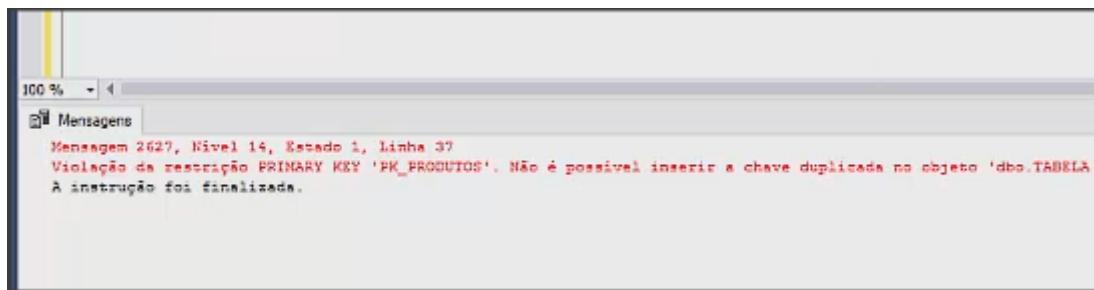
```
ALTER TABLE [TABELA DE PRODUTOS]
ALTER COLUMN [CODIGO DO PRODUTO]
VARCHAR(10) NOT NULL
```

E execute o passo 21 novamente.

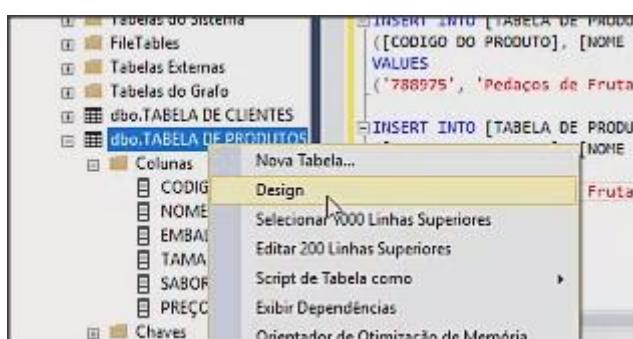
26) Execute o comando de inclusão novamente:

```
INSERT INTO [TABELA DE PRODUTOS]
([CODIGO DO PRODUTO], [NOME DO PRODUTO], [EMBALAGEM], [TAMANHO],
[SABOR], [PREÇO DE LISTA])
VALUES
('788975', 'Pedaços de Frutas - 1,5 Litros - Maça', 'PET', '1,5
Litros', 'Maça', 18.01)
```

27) Execute-o novamente. Você terá o erro por causa da chave primária, não permitindo dois produtos com o mesmo código:



28) Clique com o botão da direita do mouse sobre a tabela **Tabela de Produtos** e escolha a opção **Designer**:



29) Você pode ver a representação da chave primária com a propriedade **Permitir Nulos** desmarcada e um símbolo de uma chave ao lado do nome do campo **[CODIGO DO PRODUTO]**:

	Nome da Coluna	Tipo de Dados	Permitir Nulos
000.1	[CODIGO DO PRODUTO]	varchar(10)	☒
os	[NOME DO PRODUTO]	varchar(50)	☑
dos	EMBALAGEM	varchar(20)	☑
	TAMANHO	varchar(10)	☑
	SABOR	varchar(20)	☑
	[PREÇO DE LISTA]	smallmoney	☐

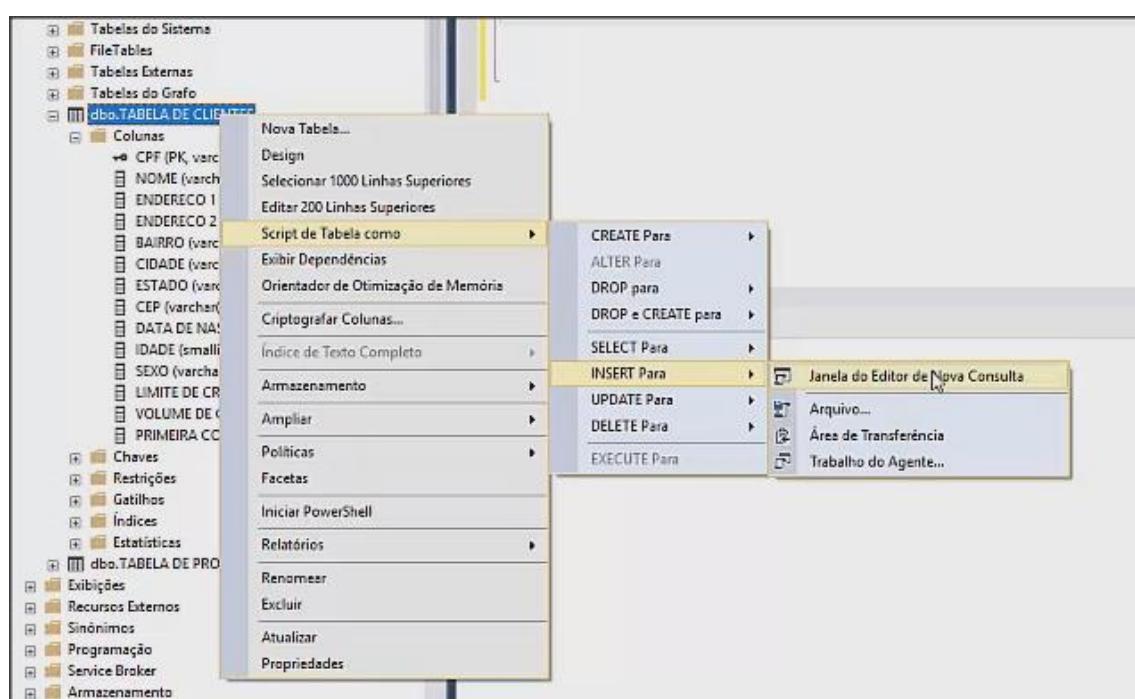
## Manipulando datas e campos lógicos

30) Crie uma nova consulta. Edite os comandos abaixo para criar a chave primária da tabela de clientes para o campo CPF:

```
ALTER TABLE [TABELA DE CLIENTES]
ALTER COLUMN [CPF]
VARCHAR(11) NOT NULL
```

```
ALTER TABLE [TABELA DE CLIENTES]
ADD CONSTRAINT PK_CLIENTES
PRIMARY KEY CLUSTERED ([CPF])
```

31) Insira registros nesta tabela. Mas, para facilitar a digitação do comando de inserção, clique com o botão da direita do mouse sobre a **Tabela de Clientes** e escolha a opção **Script de Tabela como --> INSERT Para --> Janela do Editor de Nova Consulta**:



32) O comando de inclusão de registro é preenchido, bastando apenas que você complete com os dados:

```

USE [SUCOS_VENDAS]
GO

INSERT INTO [dbo].[TABELA DE CLIENTES]
([CPF]
,[NOME]
,[ENDERECO 1]
,[ENDERECO 2]
,[BAIRRO]
,[CIDADE]
,[ESTADO]
,[CEP]
,[DATA DE NASCIMENTO]
,[IDADE]
,[SEXO]
,[LIMITE DE CREDITO]
,[VOLUME DE COMPRA]
,[PRIMEIRA COMPRA])
VALUES
(<CPF, varchar(11),>
,<NOME, varchar(100),>
,<ENDERECO 1, varchar(150),>
,<ENDERECO 2, varchar(150),>
,<BAIRRO, varchar(50),>
,<CIDADE, varchar(50),>
,<ESTADO, varchar(2),>
,<CEP, varchar(8),>
,<DATA DE NASCIMENTO, date,>
,<IDADE, smallint,>
,<SEXO, varchar(1),>
,<LIMITE DE CREDITO, money,>
,<VOLUME DE COMPRA, float,>
,<PRIMEIRA COMPRA, bit,>)
GO

```

33) Altere o comando digitando:

```

INSERT INTO [dbo] . [TABELA DE CLIENTES]
( [CPF]
,[NOME]
,[ENDERECO 1]
,[ENDERECO 2]
,[BAIRRO]
,[CIDADE]
,[ESTADO]
,[CEP]
,[DATA DE NASCIMENTO]
,[IDADE]
,[SEXO]
,[LIMITE DE CREDITO]
,[VOLUME DE COMPRA]
,[PRIMEIRA COMPRA] )
VALUES
('00300000001'
,'João da Silva'
,'Rua Projetada A número 10'
,''
,'Centro'
,'Rio de Janeiro'
,'RJ'
,'20000000'
,'1990-10-25'
,27
,'M'
,120000.50
,1000
,1)

```

Veja neste comando a forma com que o campo do tipo **Data** e **Bit** foram incluídos.

34) Execute o comando e verifique o conteúdo da tabela:

Resultados Mensagens												
	ENDEREÇO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO	IDADE	SEXO	LIMITE DE CREDITO	VOLUME DE COMPRA	PRIMEIRA COMPRA
1	Projetada A número 10		Centro	Rio de Janeiro	RJ	2000000	1990-10-25	27	M	120000,50	1000	1

## Consultando os dados

1) Acesse o **Management Studio**.

2) Abra o arquivo **Incluindo dados tabela produtos e clientes.sql** e execute o script. Diversos registros serão incluídos na tabela de produtos e de clientes.

3) Se você verificar a tabela de produtos, terá:

Resultados Mensagens						
	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	100089	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Líma/Limão	PET	1 Litro	Líma/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
9	1041119	Unha Citros - 700 ml - Líma/Limão	Garrafa	700 ml	Líma/Limão	4,904
10	1042712	Unha Citros - 700 ml - Limão	Garrafa	700 ml	Limão	4,904
11	1051510	Fresco do Verão - 470 ml - Limão	Garrafa	470 ml	Limão	3,2995
12	1078680	Fresco do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
13	1086543	Unha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
14	1080126	Unha Citros - 1 Litro - Limão	PET	1 Litro	Limão	7,004
15	1096818	Unha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
16	1101035	Unha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão	9,0105
17	229900	Fedêzias de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211

4) O mesmo para a tabela de clientes:

100 %

Resultados Mensagens

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO	IDADE	SEXO	LIMITE DE CREDITO
1	1471156710	Erica Carvalho	R. Inquia		Jardins	São Paulo	SP	00012212	1990-09-01	27	F	170000,00
2	19290952743	Fernando Cavalcante	R. Dois de Fevereiro		Água Santa	Rio de Janeiro	RJ	22000000	2000-02-12	18	M	100000,00
3	2600506709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12	18	M	120000,00
4	3623344710	Marcos Nogueira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ	22002012	1995-01-13	23	M	110000,00
5	492472710	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19	23	M	75000,00
6	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11	22	M	170000,00
7	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP	88192029	1995-11-14	22	F	70000,00
8	5640641702	Paulo César Matos	Rua Hélio Betrão		Tijuca	Rio de Janeiro	RJ	21002020	1991-08-30	26	M	120000,00
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16	32	M	140000,00
10	7771579779	Marcelo Mattos	R. Eduardo Luís Lopes		Brás	São Paulo	SP	88200212	1992-03-25	25	M	120000,00
11	0502602733	Valdeci da Silva	R. Srg. Edson de Oliveira		Jardins	São Paulo	SP	81152002	1995-10-07	22	M	110000,00
12	8719655770	Carlos Eduardo	Av. Gen. Guedes da Fontoura		Jardins	São Paulo	SP	81152002	1983-12-20	34	M	200000,00
13	9203760794	Edson Melo dea	R. Pinto de Azevedo		Cidade Nova	Rio de Janeiro	RJ	22002002	1995-10-07	22	M	150000,00
14	94307575700	Welber Lontro	R. Cel. Almeida		Fazenda	Rio de Janeiro	RJ	22000201	1999-06-20	23	M	60000,00
15	95939180787	Fábio Carvalho	R. das Jacarandas da Península		Bonsucesso	Rio de Janeiro	RJ	22002020	1992-01-05	15	M	90000,00

5) Quando listamos o conteúdo das tabelas, o SQL Server Management Studio escreveu justamente o comando para consulta de dados:

SQLQuery21.sql - D\_S\_VENDAS (sa (54)) SQLQuery20.sql - D\_S\_VENDAS (sa (54))

```
***** Script do comando SelectTopNRows de SSMS *****
SELECT TOP (1000) [CPF]
,[NOME]
,[ENDERECO 1]
,[ENDERECO 2]
,[BAIRRO]
,[CIDADE]
,[ESTADO]
,[CEP]
,[DATA DE NASCIMENTO]
,[IDADE]
,[SEXO]
,[LIMITE DE CREDITO]
,[VOLUME DE COMPRA]
,[PRIMEIRA COMPRA]
FROM [SUCOS_VENDAS].[dbo].[TABELA DE CLIENTES]
```

6) Se editarmos o comando acima e, no lugar de TOP 1000 você modificar para TOP 5, verá apenas 5 registros:

SQLQuery21.sql - D\_S\_VENDAS (sa (54)) SQLQuery2

```
***** Script do comando SelectTopNRows
SELECT top 5 [CPF]
,[NOME]
,[ENDERECO 1]
,[ENDERECO 2]
,[BAIRRO]
,[CIDADE]
,[ESTADO]
```

100 %

Resultados Mensagens

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO
1	1471156710	Erica Carvalho	R. Inquia		Jardins	São Paulo	SP	00012212	1990-09-01
2	19290952743	Fernando Cavalcante	R. Dois de Fevereiro		Água Santa	Rio de Janeiro	RJ	22000000	2000-02-12
3	2600506709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12
4	3623344710	Marcos Nogueira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ	22002012	1995-01-13
5	492472710	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19

7) Se, no lugar dos campos, colocar um asterisco (como abaixo) verá a mesma coisa se estivesse digitado todos os campos.

```
SELECT * FROM [TABELA DE CLIENTES]
```

100 %

	CPF	NOME	ENDEREÇO 1	ENDEREÇO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO	IDADE	SEXO
1	1471156710	Erica Carvalho	R. Inquia		Jardim	São Paulo	SP	80012212	1990-09-01	27	F
2	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Aquá Santa	Rio de Janeiro	RJ	22000000	2000-02-12	18	M
3	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12	18	M
4	3623344710	Marcos Nougueira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ	22002012	1995-01-13	23	M
5	492472710	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19	23	M
6	50534475787	Abel Síva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11	22	M
7	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP	88192029	1995-11-14	22	F
8	5640641702	Paulo César Mattos	Rua Hélio Bibrão		Tijuca	Rio de Janeiro	RJ	21002020	1991-08-30	26	M
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16	32	M
10	7771579779	Marcelo Mattos	R. Eduardo Luís Lopes		Braço	São Paulo	SP	88202912	1992-03-25	25	M
11	8502602733	Voldeci da Silva	R. Srg. Edson de Oliveira		Jardim	São Paulo	SP	82122020	1995-10-07	22	M
12	8719655770	Carlos Eduardo	Av. Gen. Guedes da Fontoura		Jardim	São Paulo	SP	81192002	1983-12-20	34	M
13	9283760794	Edson Melélieis	R. Pinto de Azevedo		Cidade Nova	Rio de Janeiro	RJ	22002002	1995-10-07	22	M
14	94307575700	Walber Lontra	R. Cel. Almeida		Piedade	Rio de Janeiro	RJ	22002021	1989-06-20	28	M
15	95939180787	Fábio Carvalho	R. das Jacarandas da Península		Barna da Tijuca	Rio de Janeiro	RJ	22002020	1992-01-05	16	M

8) Você pode restringir as colunas a serem exibidas. Digite o comando abaixo e execute-o.

```
SELECT [CPF], [NOME] FROM [TABELA DE CLIENTES]
```

100 %

	CPF	NOME
1	1471156710	Erica Carvalho
2	19290992743	Fernando Cavalcante
3	2600586709	César Teixeira
4	3623344710	Marcos Nougueira
5	492472710	Eduardo Jorge
6	50534475787	Abel Síva
7	5576228758	Petra Oliveira
8	5640641702	Paulo César Mattos
9	5840119709	Gabriel Araújo
10	7771579779	Marcelo Mattos
11	8502602733	Voldeci da Silva
12	8719655770	Carlos Eduardo
13	9283760794	Edson Melélieis
14	94307575700	Walber Lontra
15	95939180787	Fábio Carvalho

9) Você pode mudar o nome das colunas criando o que é chamado de ALIAS:

```
SELECT [CPF] AS IDENTIFICADOR, [NOME] AS CLIENTE
FROM [TABELA DE CLIENTES]
```

	IDENTIFICADOR	CLIENTE
1	1471156710	Erica Carvalho
2	19290992743	Fernando Cavalcante
3	2600586709	César Teixeira
4	3623344710	Marcos Nogueira
5	492472710	Eduardo Jorge
6	50534475787	Abel Silva
7	5576228758	Petra Oliveira
8	5640641702	Paulo César Mattos
9	5840119709	Gabriel Araújo
10	7771579779	Marcelo Mattos
11	0502602733	Valdeci da Silva
12	8719655770	Carlos Eduardo
13	9283760794	Edson Melo
14	94307575700	Walber Lomira
15	95939180787	Fábio Carvalho

10) Ou também modificar a ordem de exibição das colunas no resultado da consulta:

```
SELECT [NOME] AS CLIENTE, [CPF] AS IDENTIFICADOR
FROM [TABELA DE CLIENTES]
```

	CLIENTE	IDENTIFICADOR
1	Erica Carvalho	1471156710
2	Fernando Cavalcante	19290992743
3	César Teixeira	2600586709
4	Marcos Nogueira	3623344710
5	Eduardo Jorge	492472710
6	Abel Silva	50534475787
7	Petra Oliveira	5576228758
8	Paulo César Mattos	5640641702
9	Gabriel Araújo	5840119709
10	Marcelo Mattos	7771579779
11	Valdeci da Silva	0502602733
12	Carlos Eduardo	8719655770
13	Edson Melo	9283760794
14	Walber Lomira	94307575700
15	Fábio Carvalho	95939180787

## Filtrando registros

11) Para limitar o resultado da consulta, inclua uma condição, muito parecida quando você aprendeu a excluir e alterar registros de uma tabela. Digite:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [CÓDIGO DO PRODUTO] =
'1000889'
```

12) Você verá apenas o registro selecionado:

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.309

- 13) Você pode limitar a seleção usando como critério qualquer outro campo.  
Digite:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Uva'
```

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.309

- 14) Se você limitar para verificar somente sabores **Limão**, verá mais registros.  
Digite:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Limão'
```

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1042712	Linha Citros - 700 ml - Limão	Garrafa	700 ml	Limão	4.904
1051510	Frescor do Verão - 470 ml - Limão	Garrafa	470 ml	Limão	3,2995
1088126	Linha Citros - 1 Litro - Limão	PET	1 Litro	Limão	7.004
544931	Freacordo Verão - 350 ml - Limão	Lata	350 ml	Limão	2,4595

- 15) E qualquer condição de **WHERE** vale para os comandos **UPDATE** e **DELETE**.  
Digite:

```
DELETE FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Limão'
```

Você verá na mensagem de retorno que 4 registros foram afetados.

- 16) Se você executar novamente o comando de seleção, nada será  
retornado:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Limão'
```

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA

## Filtrando usando maior, menor, diferente

17) Você pode fazer filtros envolvendo números. Digite:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] = 4.555
```

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1 1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4.555

18) Você pode verificar os produtos que custam mais que 10. Digite:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] > 10
```

CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1 1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
2 1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
3 1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
4 1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
5 1006543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
6 231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
7 243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
8 326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	15,5105
9 520380	Pedaços de Frutas - 1 Litro - Maçã	PET	1 Litro	Maçã	12,011
10 695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,512
11 746596	Light - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,505
12 788975	Pedaços de Frutas - 1,5 Litros - Maçã	PET	1,5 Litros	Maçã	18,011
13 838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008

19) Ou menores que 10:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] < 10
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
2	1002334	Linha Citros - 1 Litro - Líma/Limão	PET	1 Litro	Líma/Limão 7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã 8,41
4	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância 4,555
5	1041119	Linha Citros - 700 ml - Líma/Limão	Garrafa	700 ml	Líma/Limão 4,904
6	1070600	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga 5,1795
7	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga 7,7105
8	1101035	Linha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão 9,0105
9	229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã 4,211
10	235653	Fresco do Verão - 350 ml - Manga	Lata	350 ml	Manga 3,8595
11	290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância 4,56
12	394479	Sabor da Montanha - 700 ml - Cereja	Garrafa	700 ml	Cereja 8,409
13	479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja 3,768
14	723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá 4,912
15	773912	Clean - 1 Litro - Laranja	PET	1 Litro	Laranja 8,000
16	783663	Sabor da Montanha - 700 ml - Morango	Garrafa	700 ml	Morango 7,709

20) Você pode usar o maior e igual, ou menor ou igual para incluir o critério de seleção no resultado:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] >= 4.555
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva 6,309
2	1002334	Linha Citros - 1 Litro - Líma/Limão	PET	1 Litro	Líma/Limão 7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã 8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância 19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã 24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí 38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja 16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância 4,555
9	1041119	Linha Citros - 700 ml - Líma/Limão	Garrafa	700 ml	Líma/Limão 4,904
10	1070600	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga 5,1795
11	1096543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga 11,0105
12	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga 7,7105

21) Você pode usar o maior e menor para condições que envolvam strings. O critério de determinar quem é maior ou menor será o de ordem alfabética:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE EMBALAGEM <= 'Lata'
```

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	100089	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
3	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
4	1041119	Linha Outros - 700 ml - Líma/Limão	Garrafa	700 ml	Líma/Limão	4,904
5	1070600	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
6	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
7	229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211
8	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
9	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
10	290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56
11	394479	Sabor da Montanha - 700 ml - Cereja	Garrafa	700 ml	Cereja	8,409
12	479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3,768
13	723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá	4,912
14	783663	Sabor da Montanha - 700 ml - Morango	Garrafa	700 ml	Morango	7,709
15	812829	Clean - 350 ml - Laranja	Lata	350 ml	Laranja	2,808
16	826490	Linha Refrescante - 700 ml - Morango/Limão	Garrafa	700 ml	Morango/Limão	6,3105

22) Há também o critério de selecionar alguém diferente de um valor. Para isso, use o símbolo <>:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] <> 4.555
```

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	100089	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Outros - 1 Litro - Líma/Limão	PET	1 Litro	Líma/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1041119	Linha Outros - 700 ml - Líma/Limão	Garrafa	700 ml	Líma/Limão	4,904
9	1078880	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
10	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
11	1096910	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
12	1101035	Linha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão	9,0105
13	229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211
14	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
15	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
16	243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
17	290470	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56

## Filtrando por datas

23) Você pode usar as datas como critério de seleção. O SQL Server irá usar a ordem definida no calendário para determinar quem é maior ou menor.

Digite:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE [DATA DE NASCIMENTO] = '1995-09-11'
```

24) Ou então:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE [DATA DE NASCIMENTO] <=
'1995-09-11'
```

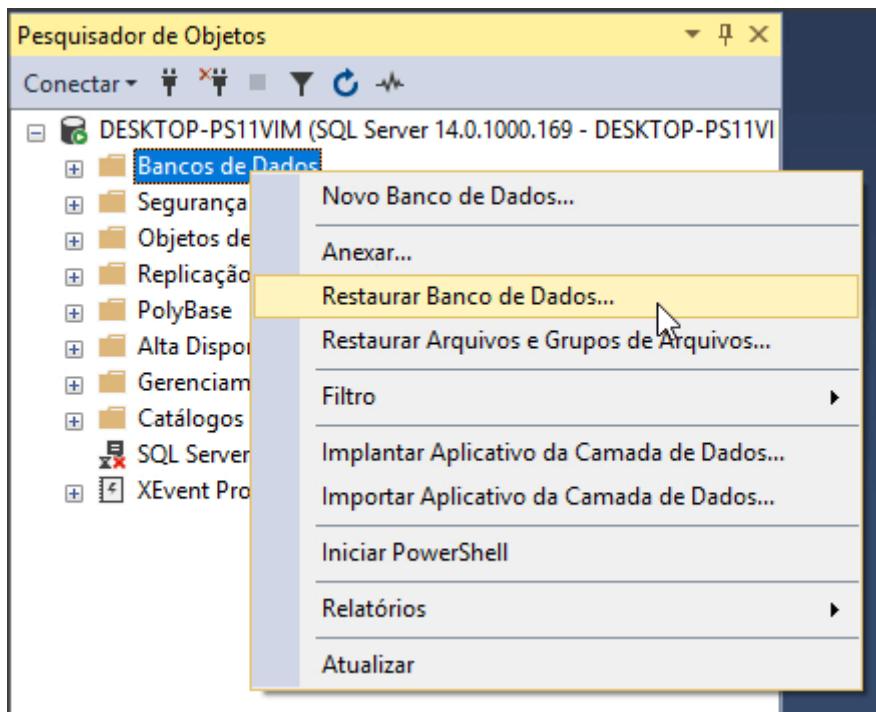
25) Você pode usar algumas funções aplicadas à data que determinam condições sobre o dia, mês ou ano. Veja o exemplo abaixo:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE YEAR([DATA DE NASCIMENTO]) <
1995
```

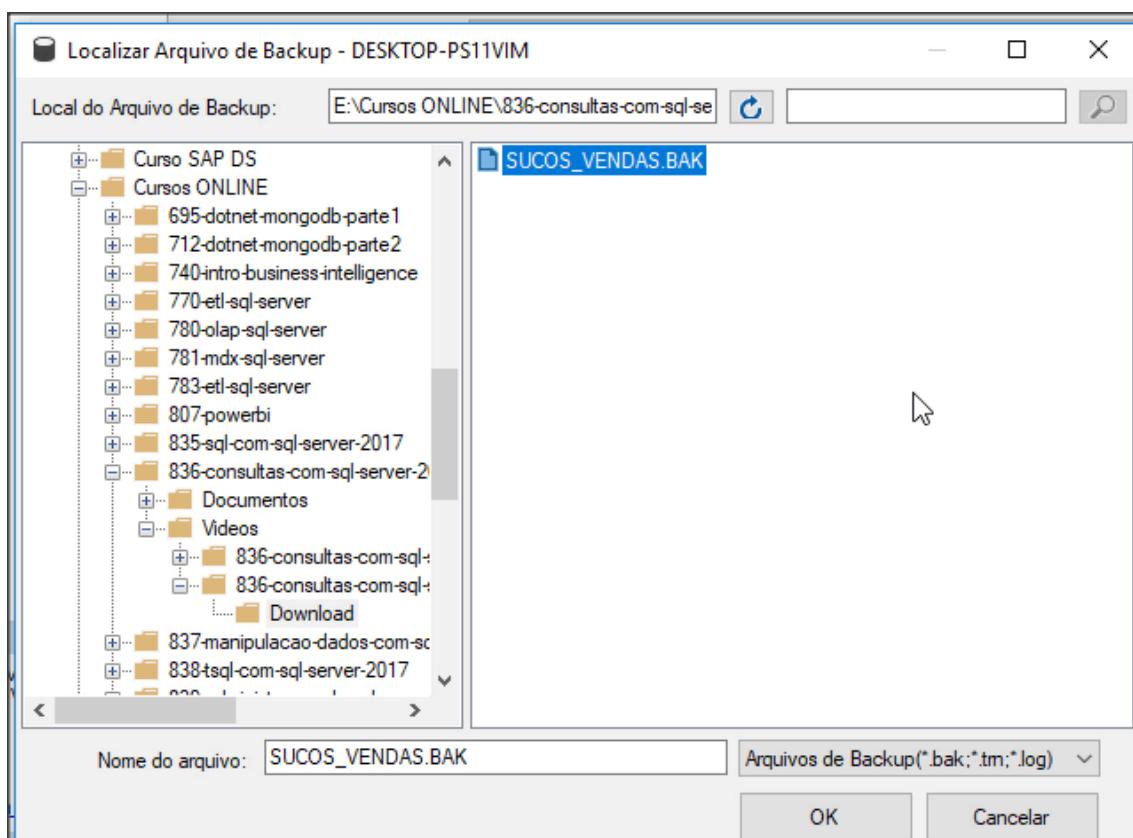
Nesta atividade, você verá como preparar o seu ambiente, para poder realizar as tarefas e prosseguir com o treinamento. Para isso, baixe [aqui](#) os arquivos necessários e siga os passos abaixo.

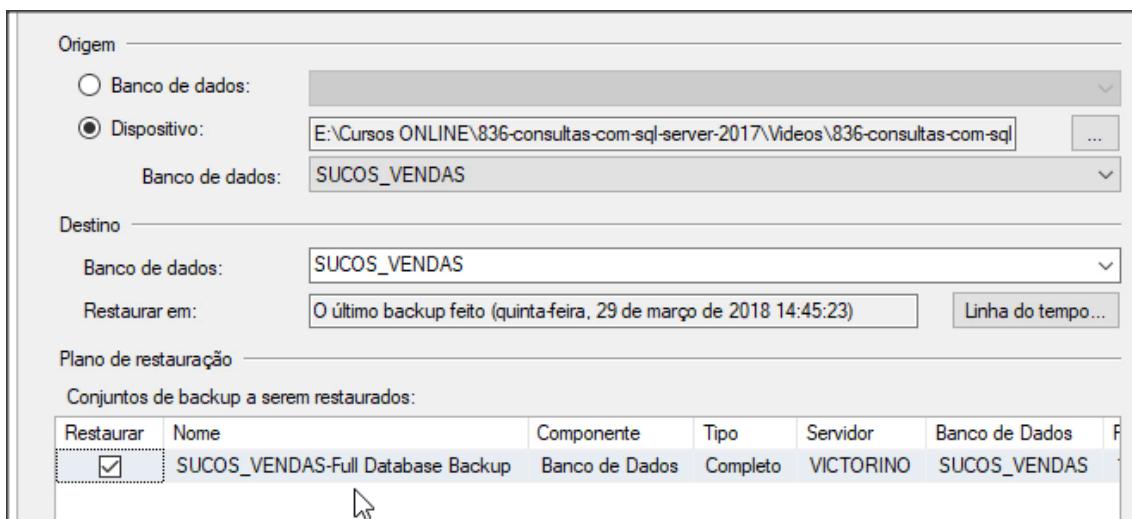
---

1) Vá no **SQL Server Management Studio 2017** e conecte-se ao seu servidor. Após isso, na árvore à esquerda, clique com o botão da direita do mouse sobre **Banco de Dados** e selecione a opção **Restaurar Banco de Dados**:

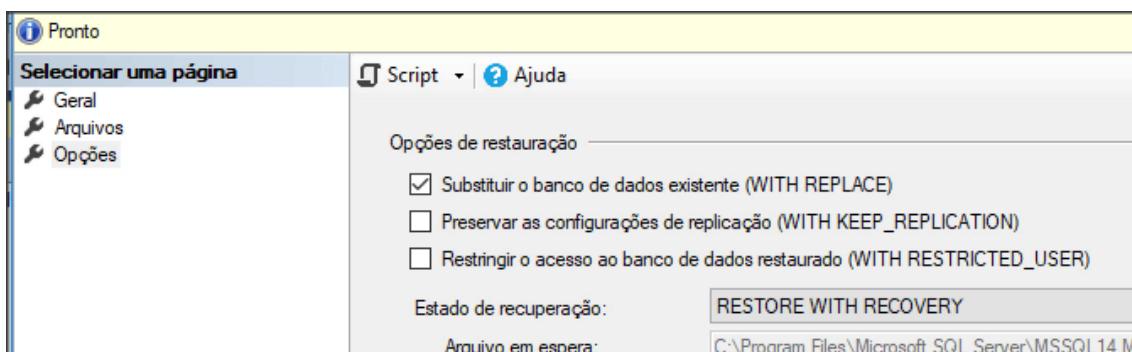


2) Selecione o arquivo a ser usado para recuperação do backup:

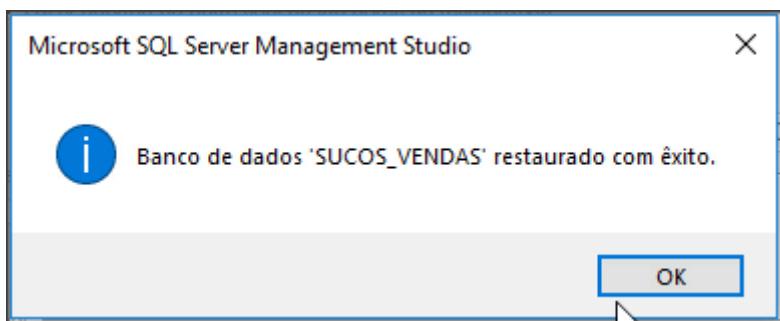




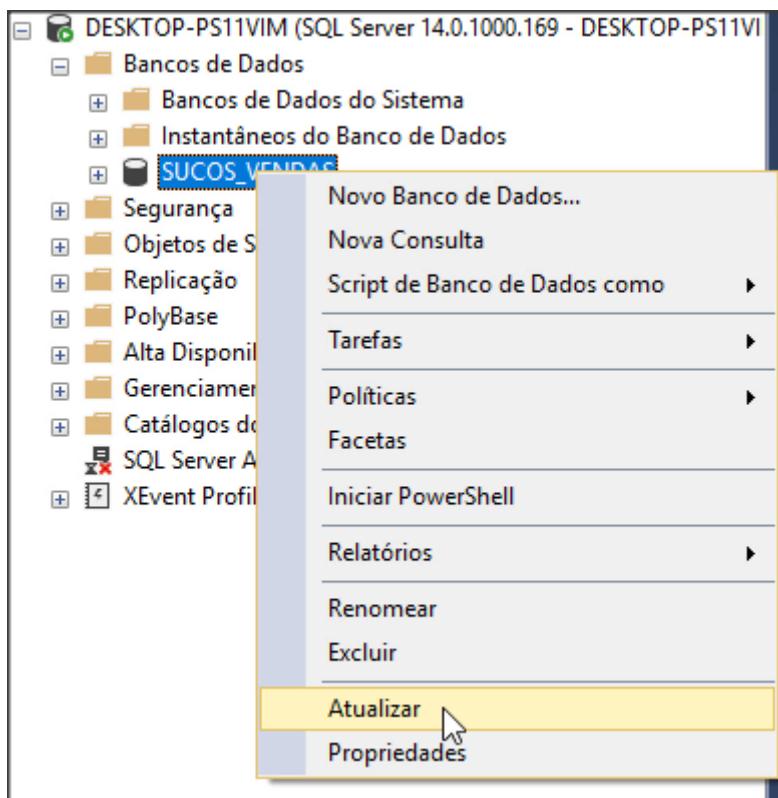
3) Selecione a opção **Substituir banco de dados existente**:



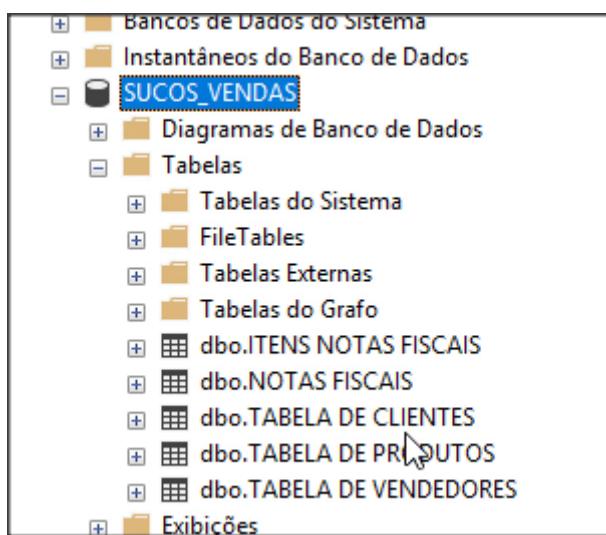
4) Clique em **OK**. O banco de dados será restaurado:



5) Para você se certificar que a restauração funcionou, clique em **Atualizar**, após clicar com o botão da direita do mouse sobre o banco de dados:

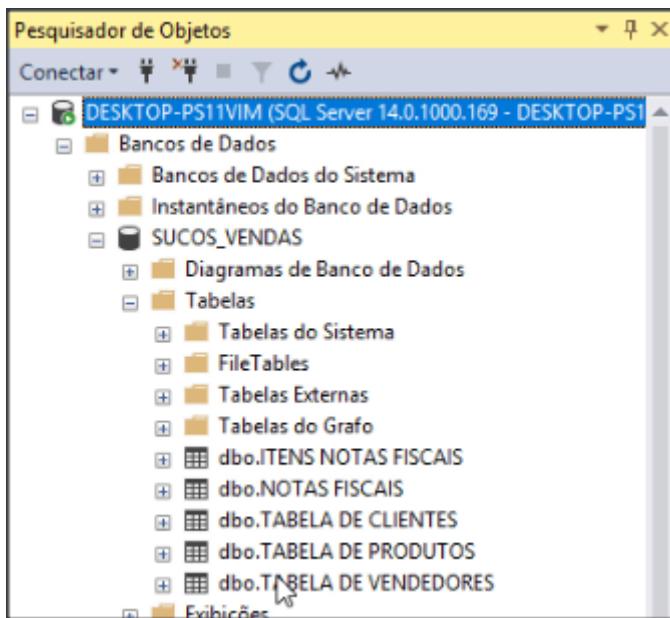


6) E confira se as tabelas do banco de dados são exibidas:



## Conhecendo o banco de dados

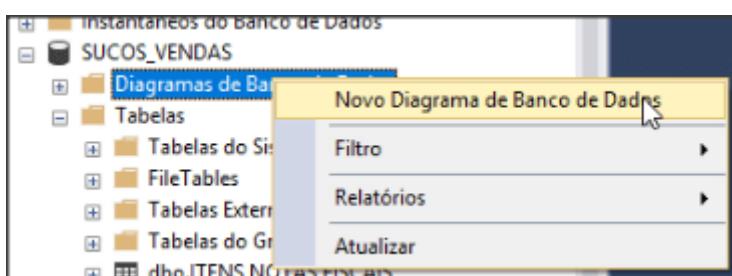
1) Acesse o **SQL Server Management Studio 2017** e verifique as tabelas do banco de dados de vendas da empresa de suco de frutas:

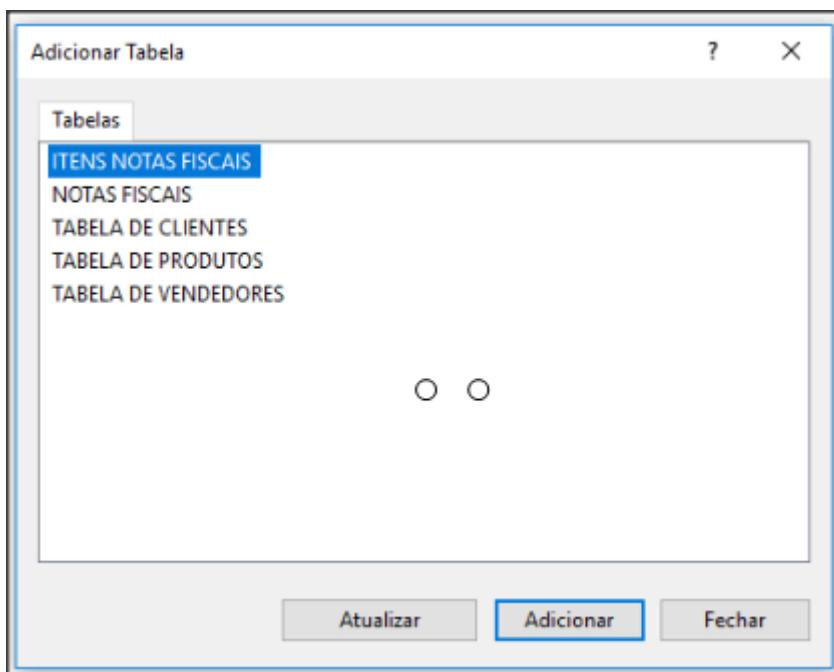


2) Há as tabelas:

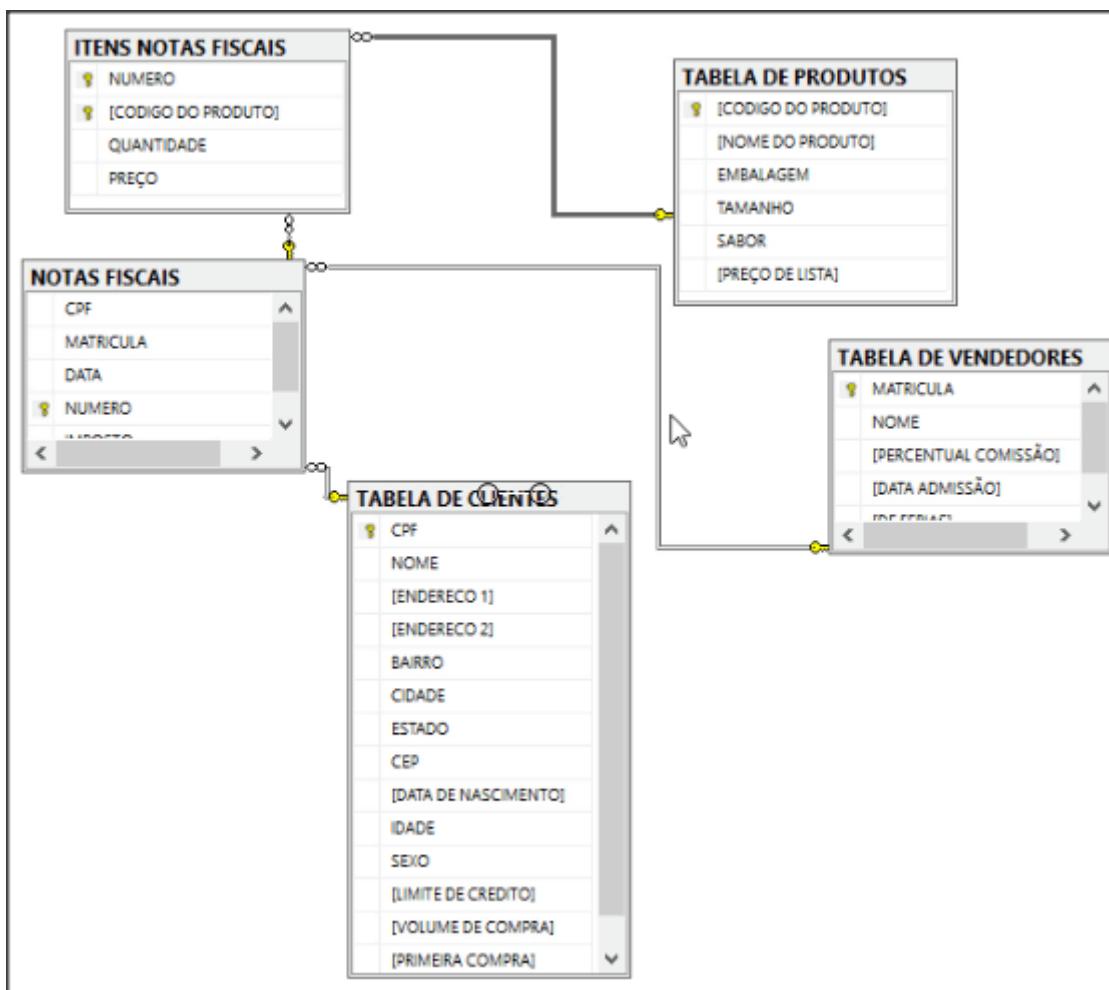
- **TABELA DE VENDEDORES** - Lista de vendedores que efetuaram as vendas.
- **TABELA DE PRODUTOS** - Lista dos produtos oferecidos pela empresa.
- **TABELA DE CLIENTES** - Lista de clientes que compram os produtos da empresa.
- **NOTAS FISCAIS** - Notas fiscais das vendas feitas pela empresa.
- **ITEMS NOTAS FISCAIS** - Itens das notas fiscais.

3) Uma forma rápida de visualizar as tabelas e seus relacionamentos é indo em **Diagrama de Banco de Dados** e, clicando com o botão da direita do mouse sobre esta opção, escolha a opção **Novo Diagrama de Banco de Dados**. Você terá a lista de todas as tabelas:





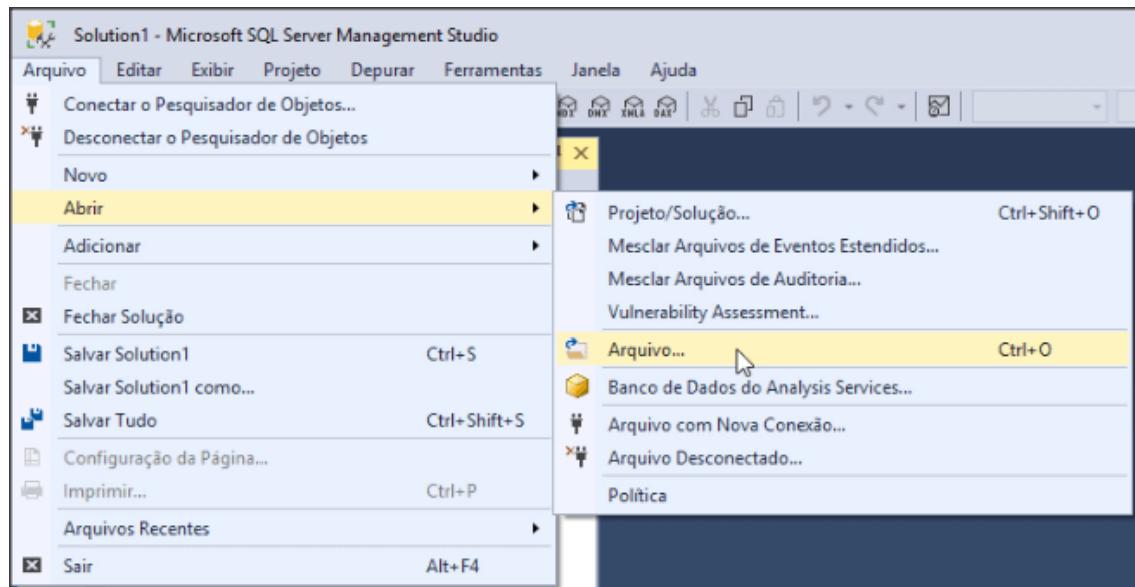
4) Selecione todas as tabelas e clique em **Adicionar**. Você terá a representação das tabelas graficamente:



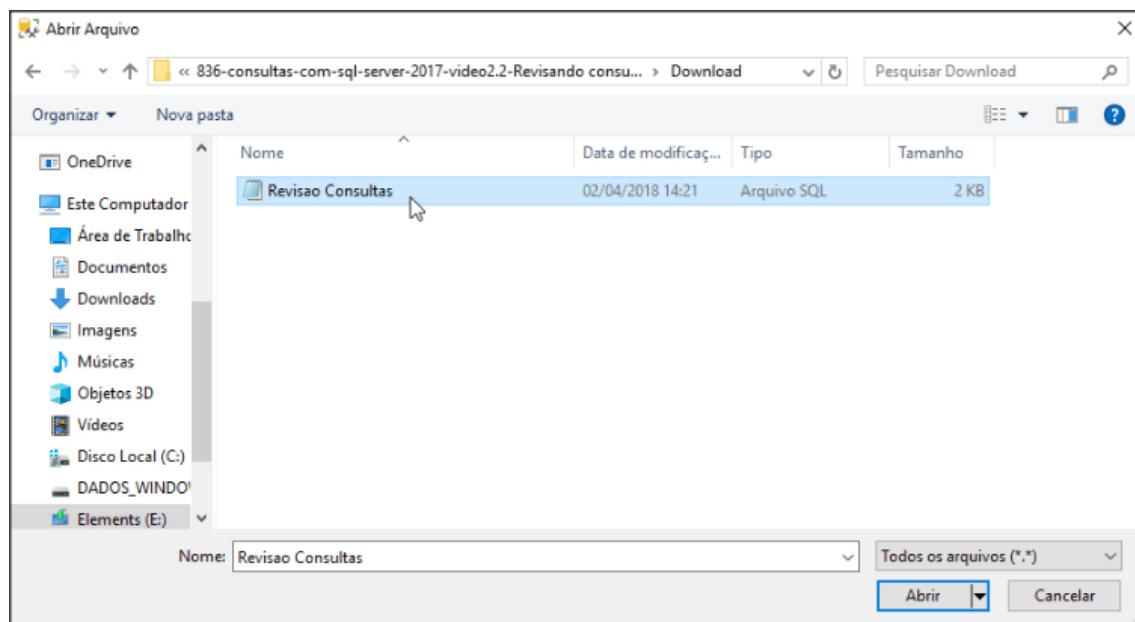
5) Observe os campos de cada tabela com os seus respectivos tipos e as relações entre as tabelas.

## Revisando consultas

6) Vá no **SQL Server Management Studio 2017** e selecione no menu: **Arquivo --> Abrir --> Arquivo...**:



7) Selecione o arquivo **Revisao Consultas.sql**:



8) Reveja as consultas já vistas no curso de **Introdução ao SQL Server**.

Digite:

```
SELECT [CPF]
```

```

    ,[NOME]
    ,[ENDERECO 1]
    ,[ENDERECO 2]
    ,[BAIRRO]
    ,[CIDADE]
    ,[ESTADO]
    ,[CEP]
    ,[DATA DE NASCIMENTO]
    ,[IDADE]
    ,[SEXO]
    ,[LIMITE DE CREDITO]
    ,[VOLUME DE COMPRA]
    ,[PRIMEIRA COMPRA]
FROM [SUCOS_VENDAS].[dbo].[TABELA DE CLIENTES]

```

Você terá a lista de clientes da **TABELA DE CLIENTES**:

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE
1	1471156710	Érica Carvalho	R. Iniquia		Jardins	São Paulo
2	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Áqua Santa	Rio de Janeiro
3	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro
4	3623344710	Marcos Nougueira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro
5	492472718	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro
6	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro
7	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo
8	5648641702	Paulo César Mattos	Rua Hélio Beltrão		Tijuca	Rio de Janeiro
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo
10	7771579779	Marcelo Mattos	R. Eduardo Luis Lopes		Brás	São Paulo

9) Você pode substituir a lista dos campos por um asterisco:

```
SELECT * FROM [TABELA DE CLIENTES]
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE
1	1471156710	Érica Carvalho	R. Iniquia		Jardins	São Paulo
2	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Áqua Santa	Rio de Janeiro
3	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro
4	3623344710	Marcos Nougueira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro
5	492472718	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro
6	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro
7	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo
8	5648641702	Paulo César Mattos	Rua Hélio Beltrão		Tijuca	Rio de Janeiro
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo
10	7771579779	Marcelo Mattos	R. Eduardo Luis Lopes		Brás	São Paulo

10) Você pode escolher os campos a serem visualizados:

```

SELECT [CPF]
      ,[NOME]
  FROM [TABELA DE CLIENTES]

```

	CPF	NOME
1	1471156710	Érica Carvalho
2	19290992743	Fernando Cavalcante
3	2600586709	César Teixeira
4	3623344710	Marcos Nougeuira
5	492472718	Eduardo Jorge
6	50534475787	Abel Silva
7	5576228758	Petra Oliveira
8	5648641702	Paulo César Mattos
9	5840119709	Gabriel Araujo
10	7771579779	Marcelo Mattos
11	8502682733	Valdeci da Silva

11) Você também pode criar um *alias* para cada coluna:

```
SELECT [CPF] AS IDENTIFICADOR
      , [NOME] AS CLIENTE
    FROM [TABELA DE CLIENTES]
```

	IDENTIFICADOR	CLIENTE
1	1471156710	Érica Carvalho
2	19290992743	Fernando Cavalcante
3	2600586709	César Teixeira
4	3623344710	Marcos Nougeuira
5	492472718	Eduardo Jorge
6	50534475787	Abel Silva
7	5576228758	Petra Oliveira
8	5648641702	Paulo César Mattos
9	5840119709	Gabriel Araujo
10	7771579779	Marcelo Mattos
11	8502682733	Valdeci da Silva

12) E pode trocar a ordem de exibição do resultado:

```
SELECT [NOME] AS CLIENTE
      , [CPF] AS IDENTIFICADOR
    FROM [TABELA DE CLIENTES]
```

	CLIENTE	IDENTIFICADOR
1	Érica Carvalho	1471156710
2	Fernando Cavalcante	19290992743
3	César Teixeira	2600586709
4	Marcos Nougeuira	3623344710
5	Eduardo Jorge	492472718
6	Abel Silva	50534475787
7	Petra Oliveira	5576228758
8	Paulo César Mattos	5648641702
9	Gabriel Araujo	5840119709
10	Marcelo Mattos	7771579779
11	Valdeci da Silva	8502682733

13) Você pode filtrar os dados exibidos pela consulta, usando o campo primário:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [CODIGO DO PRODUTO] =
'1000889'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309

14) Mas também pode filtrar por qualquer outro campo:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Uva'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309

15) No caso de filtros usando valores, você não usa aspas simples no critério de busca:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] = 4.555
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555

16) Use os símbolos de >, >=, < ou <= para filtrar os dados:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [PREÇO DE LISTA] > 10
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO
1	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
2	1013793	Videira do Campo - 2 Litros - Cereja/Maça	PET	2 Litros	Cereja/Maça	24,01
3	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,01
4	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,00
5	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,01
6	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,31
7	243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,51
8	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,51
9	520380	Pedaços de Frutas - 1 Litro - Maça	PET	1 Litro	Maça	12,01
10	695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,51

17) O critério de `>`, `>=`, `<` ou `<=` pode ser aplicado a um campo string. O SQL Server irá usar o critério de ordenação alfabética:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE EMBALAGEM <> 'Lata'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002767	Videira do Campo - 700 ml - Cereja/Maça	Garrafa	700 ml	Cereja/Maça	8,41
3	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
4	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
5	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
6	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
7	229900	Pedaços de Frutas - 350 ml - Maça	Lata	350 ml	Maça	4,211
8	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
9	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
10	290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56

18) Você também pode aplicar filtro de datas. Para evitar problemas de critérios de representação das datas, use, no filtro, o formato **ANO-MÊS-DIA**:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE [DATA DE NASCIMENTO] = '1995-09-11'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maça	Garrafa	700 ml	Cereja/Maça	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maça	PET	2 Litros	Cereja/Maça	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,01
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,00
8	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
9	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
10	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,01

19) Você pode usar **>**, **>=**, **<** ou **<=** para as datas:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE [DATA DE NASCIMENTO] >=
'1995-09-11'
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO
1	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Água Santa	Rio de Janeiro	RJ
2	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ
3	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ
4	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP
5	8502682733	Valdeci da Silva	R. Srg. Édison de Oliveira		Jardins	São Paulo	SP
6	9283760794	Edson Meilleles	R. Pinto de Azevedo		Cidade Nova	Rio de Janeiro	RJ

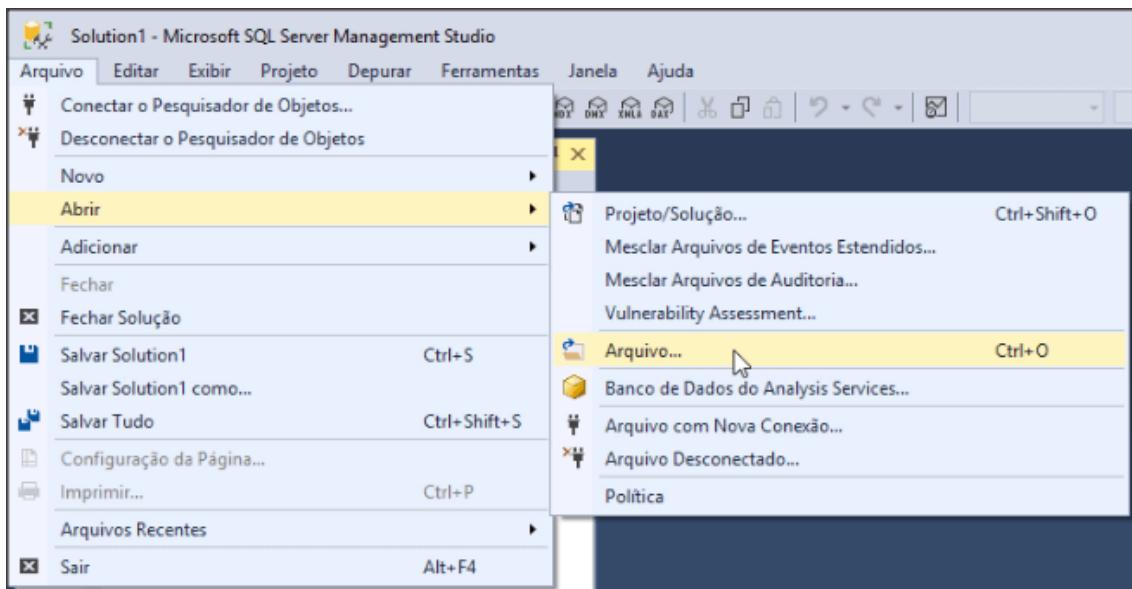
20) Nos critérios de filtro, você pode usar funções do SQL Server. Por exemplo, use a função **YEAR** aplicada ao campo **DATA DE NASCIMENTO**:

```
SELECT * FROM [TABELA DE CLIENTES] WHERE YEAR([DATA DE NASCIMENTO]) = 1995
```

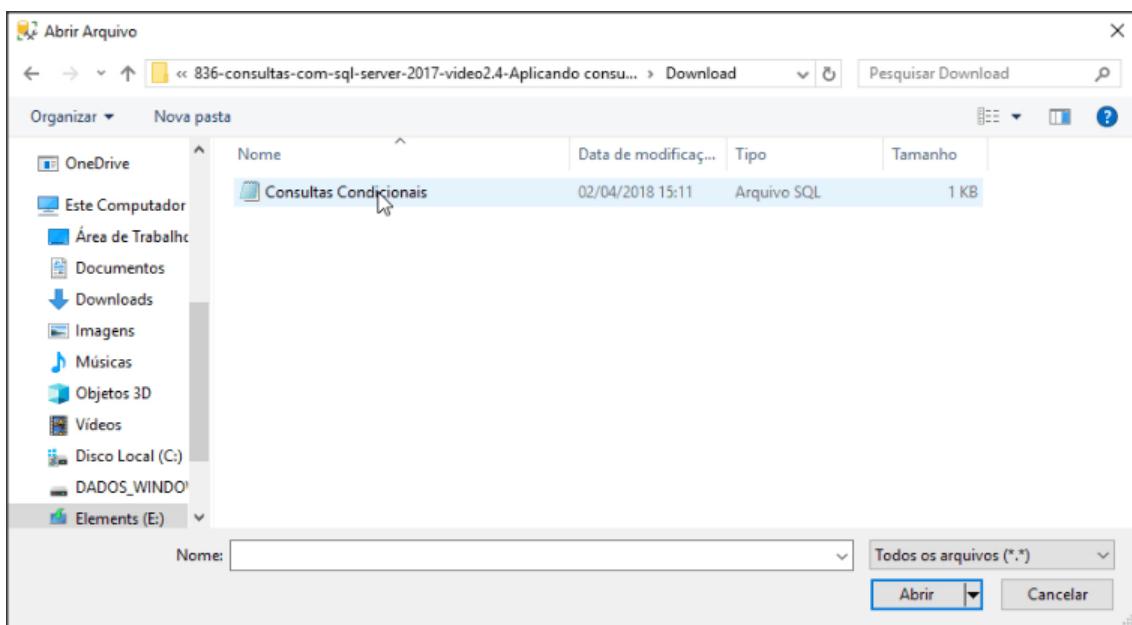
	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO
1	3623344710	Marcos Nougeira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ
2	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ
3	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP
4	8502682733	Valdeci da Silva	R. Srg. Édison de Oliveira		Jardins	São Paulo	SP
5	9283760794	Edson Meilleles	R. Pinto de Azevedo		Cidade Nova	Rio de Janeiro	RJ

### Aplicando consultas condicionais no SQL Server

21) Vá no **SQL Server Management Studio 2017** e selecione no menu: **Arquivo --> Abrir --> Arquivo...**:



22) Selecione o arquivo **Consultas Condicionais.sql**:



23) Use o **OR** para selecionar produtos que sejam de um determinado sabor ou de um determinado tamanho:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] = 'Manga' OR [TAMANHO] = '700 ml'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
3	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
4	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
5	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
6	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
7	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
8	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
9	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
10	394479	Sabor da Montanha - 700 ml - Cereja	Garrafa	700 ml	Cereja	8,409
11	723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá	4,912

24) Já o comando **AND** listará os produtos que sejam de um determinado sabor e também de um determinado tamanho:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] = 'Manga' AND [TAMANHO] = '700 ml'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105

25) O comando **NOT**, na frente da condição, retorna o contrário do que a condição busca:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
NOT ([SABOR] = 'Manga' AND [TAMANHO] = '700 ml')
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
9	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
10	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
11	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
12	1101035	Linha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão	9,0105
13	229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211
14	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
15	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
16	243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
17	290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56
18	326779	Linha Refrescante - 1,5 Litros - Melância	PET	1,5 Litros	Melância	10,512

26) O **NOT** pode ser aplicado somente a uma parte da condição de filtro:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] = 'Manga' AND NOT ([TAMANHO] = '700 ml')
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
2	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
3	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
4	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105

27) O comando **IN** busca o registro através de um conjunto de valores:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] IN ('Manga', 'Laranja')
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
2	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
3	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
4	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
5	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
6	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
7	479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3,768
8	773912	Clean - 1 Litro - Laranja	PET	1 Litro	Laranja	8,008
9	812829	Clean - 350 ml - Laranja	Lata	350 ml	Laranja	2,808
10	838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008

Esse comando seria equivalente a:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] = 'Manga' OR [SABOR] = 'Laranja'
```

28) Você pode aplicar o **NOT** sobre a condição com **IN**:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] NOT IN ('Manga', 'Laranja')
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
8	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
9	1101035	Linha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão	9,0105
10	229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211
11	231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
12	243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
13	290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56
14	394479	Sabor da Montanha - 700 ml - Cereja	Garrafa	700 ml	Cereja	8,409
15	520380	Pedaços de Frutas - 1 Litro - Maçã	PET	1 Litro	Maçã	12,011
16	695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,512
17	723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá	4,912
18	746600	Festival de Sabores - 1,5 Litros - Maçã	PET	1,5 Litros	Maçã	10,505

29) É possível colocar junto com o `IN` condições usando `>`, `>=`, `<` ou `<=`:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] IN ('Manga', 'Laranja') AND [PREÇO DE LISTA] > 10
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
2	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
3	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
4	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7,7105
5	235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
6	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
7	479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3,768
8	773912	Clean - 1 Litro - Laranja	PET	1 Litro	Laranja	8,008
9	812829	Clean - 350 ml - Laranja	Lata	350 ml	Laranja	2,808
10	838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008

30) Há o comando `BETWEEN`, onde você pode selecionar entre dois valores:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] IN ('Manga', 'Laranja') AND [PREÇO DE LISTA] BETWEEN 10 AND 13
```

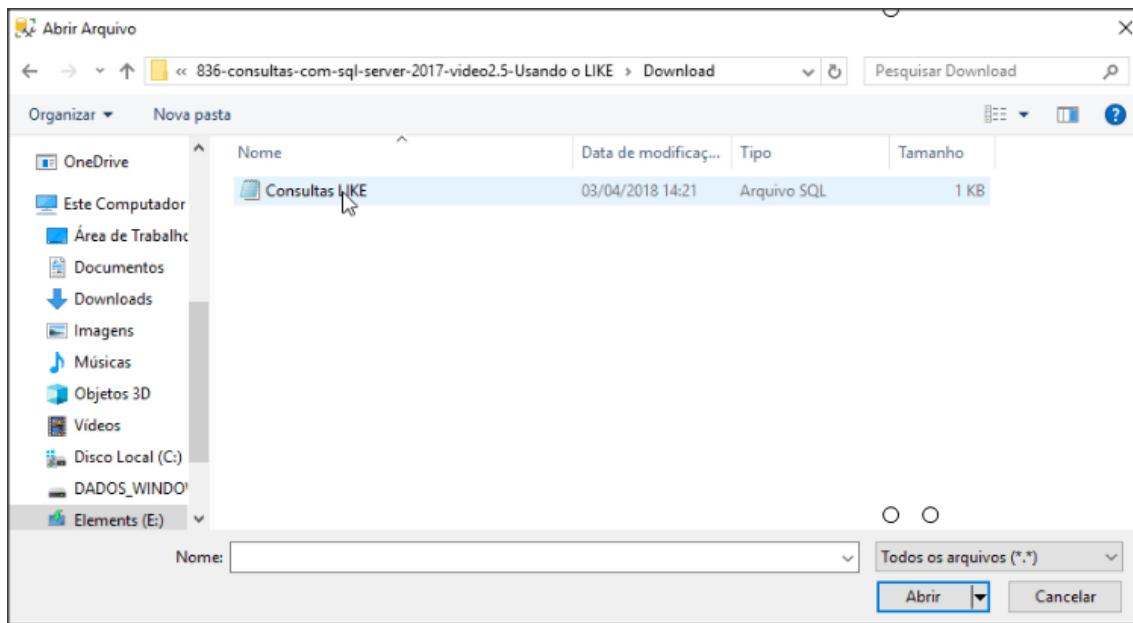
	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
2	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
3	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
4	838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008

Que seria equivalente ao comando:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE
[SABOR] IN ('Manga', 'Laranja') AND [PREÇO DE LISTA] >= 10 AND [PREÇO DE LISTA] <= 13
```

## Usando o LIKE

31) Vá no **SQL Server Management Studio 2017** e selecione no menu: **Arquivo --> Abrir --> Arquivo....** Em seguida, selecione o arquivo **Consultas LIKE.sql**:



32) Execute o comando abaixo:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [NOME DO PRODUTO] LIKE
'%Litros%'
```

Aqui, você irá achar todos os registros cujo campo **[NOME DO PRODUTO]** possua dentro do seu descritor a palavra **Litros**, seja no início, no fim ou no meio do descritor:

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
2	1013793	Videira do Campo - 2 Litros - Cereja/Maça	PET	2 Litros	Cereja/Maça	24,01
3	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
4	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
5	243083	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
6	326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
7	695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,512
8	746596	Light - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,505
9	788975	Pedaços de Frutas - 1,5 Litros - Maça	PET	1,5 Litros	Maça	18,011
10	838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008

33) Você pode buscar apenas os campos usando o critério de apenas o campo começar com a palavra **Litros**:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [NOME DO PRODUTO] LIKE
'Litros%'
```

34) Também podemos mesclar a condição **LIKE** com outras condições:

```
SELECT * FROM [TABELA DE PRODUTOS] WHERE [NOME DO PRODUTO] LIKE
'%Litros%'  
AND [SABOR] = 'Laranja'
```

	CODIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16.008
2	838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12.008

## Usando DISTINCT para visualizar a tabela

1) Abra o arquivo **Consultas DISTINCT 1.sql** no **SQL Server Management Studio 2017**.

2) Execute a consulta:

```
SELECT EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS]
```

	EMBALAGEM	TAMANHO
1	Garrafa	700 ml
2	PET	1 Litro
3	Garrafa	700 ml
4	PET	1,5 Litros
5	PET	2 Litros
6	PET	2 Litros
7	PET	2 Litros
8	Lata	350 ml
9	Garrafa	700 ml
10	Garrafa	470 ml
11	PET	1 Litro

Note que há diversos registros com **EMBALAGEM** e **TAMANHO** repetidos.

3) Agora, execute o mesmo comando com a cláusula **DISTINCT**:

```
SELECT DISTINCT EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS]
```

	EMBALAGEM	TAMANHO
1	Garrafa	470 ml
2	Garrafa	700 ml
3	Lata	350 ml
4	PET	1 Litro
5	PET	1,5 Litros
6	PET	2 Litros

4) Os filtros aplicados até agora podem ser executados em conjunto com os comandos SQL com a cláusula **DISTINCT**, mesmo que o campo usado para a condição não esteja sendo exibido no resultado da consulta:

```
SELECT DISTINCT EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS] WHERE
[SABOR] = 'Laranja'
```

	EMBALAGEM	TAMANHO
1	Garrafa	470 ml
2	Lata	350 ml
3	PET	1 Litro
4	PET	1,5 Litros
5	PET	2 Litros

### Limitando a saída da consulta

5) Abra o arquivo **Consultas DISTINCT 2.sql**.

6) Execute a seleção de todos os produtos:

```
SELECT * FROM [TABELA DE PRODUTOS]
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
9	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
10	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
11	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105

7) Limite e olhe os primeiros 3 produtos desta lista:

```
SELECT TOP 3 * FROM [TABELA DE PRODUTOS]
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41

8) Limite a consulta, utilizando filtros, como mostrado abaixo:

```
SELECT TOP 3 * FROM [TABELA DE PRODUTOS] WHERE [SABOR] = 'Laranja'
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
2	479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3.768
3	773912	Clean - 1 Litro - Laranja	PET	1 Litro	Laranja	8.008

9) Também é possível mesclar o **TOP** com o **DISTINCT** como mostrado abaixo:

```
SELECT DISTINCT TOP 3 EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS]
```

	EMBALAGEM	TAMANHO
1	Garrafa	470 ml
2	Garrafa	700 ml
3	Lata	350 ml

## Ordenando a saída da consulta

10) Abra o arquivo **Ordenando a saída da consulta.sql**.

11) Execute o comando abaixo para listar todos os produtos:

```
SELECT * FROM [dbo].[TABELA DE PRODUTOS]
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6.309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7.004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8.41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4.555
9	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4.904
10	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5.1795
11	1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
12	1096818	Linha Refrescante - 700 ml - Manga	Garrafa	700 ml	Manga	7.7105
13	1101035	Linha Refrescante - 1 Litro - Morango/Li...	PET	1 Litro	Morango/L...	9.0105
14	229900	Pedacinho de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4.211

12) Visualize esta mesma lista, mas ordenada pelo preço:

```
SELECT * FROM [TABELA DE PRODUTOS] ORDER BY [PREÇO DE LISTA]
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1 812829	Clean - 350 ml - Laranja	Lata	350 ml	Laranja	2,808
2 479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3,768
3 235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
4 229900	Pedaços de Frutas - 350 ml - Maçã	Lata	350 ml	Maçã	4,211
5 1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
6 290478	Videira do Campo - 350 ml - Melância	Lata	350 ml	Melância	4,56
7 1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
8 723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá	4,912
9 1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
10 1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
11 826490	Linha Refrescante - 700 ml - Morango/Limão	Garrafa	700 ml	Morango/Limão	6,3105
12 1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
13 783663	Sabor da Montanha - 700 ml - Morango	Garrafa	700 ml	Morango	7,709
14 1096818	Linha Refrescante - 700 ml - Manna	Garrafa	700 ml	Manna	7,7105

13) Na consulta acima, você vê os dados ordenados do menor para o maior.

Se quiser ao contrário, use a cláusula **DESC**:

```
SELECT * FROM [TABELA DE PRODUTOS] ORDER BY [PREÇO DE LISTA] DESC
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1 1022450	Festival de Sabores - 2 Litros	PET	2 Litros	Açaí	38,012
2 695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,512
3 1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
4 1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5 746596	Light - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,505
6 788975	Pedaços de Frutas - 1,5 Litros - Maçã	PET	1,5 Litros	Maçã	18,011
7 326779	Linha Refrescante - 1,5 Litros - Manga	PET	1,5 Litros	Manga	16,5105
8 1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
9 231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
10 520380	Pedaços de Frutas - 1 Litro - Maçã	PET	1 Litro	Maçã	12,011
11 838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008
12 1086543	Linha Refrescante - 1 Litro - Manga	PET	1 Litro	Manga	11,0105
13 243003	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
14 1101035	Linha Refrescante - 1 Litro - Morango/Limão	PET	1 Litro	Morango/Limão	9,0105

Consulta executada com êxito.

| DESKTOP-N51LB16 (14.0 RTM) | sa (55) | SUCOS\_VENDAS | 00:00:00 | 31 linhas

14) Você pode usar como critério de ordenação um campo string. Neste caso, o SQL Server usará como critério de ordenação a ordem alfabética:

```
SELECT * FROM [TABELA DE PRODUTOS] ORDER BY [NOME DO PRODUTO]
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1 773912	Clear - 1 Litro - Laranja	PET	1 Litro	Laranja	8,008
2 838819	Clean - 1,5 Litros - Laranja	PET	1,5 Litros	Laranja	12,008
3 1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
4 812829	Clean - 350 ml - Laranja	Lata	350 ml	Laranja	2,808
5 479745	Clean - 470 ml - Laranja	Garrafa	470 ml	Laranja	3,768
6 695594	Festival de Sabores - 1,5 Litros - Açaí	PET	1,5 Litros	Açaí	28,512
7 243003	Festival de Sabores - 1,5 Litros - Maracujá	PET	1,5 Litros	Maracujá	10,512
8 1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
9 231776	Festival de Sabores - 700 ml - Açaí	Garrafa	700 ml	Açaí	13,312
10 723457	Festival de Sabores - 700 ml - Maracujá	Garrafa	700 ml	Maracujá	4,912
11 235653	Frescor do Verão - 350 ml - Manga	Lata	350 ml	Manga	3,8595
12 1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795
13 746596	Light - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,505
14 1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555

15) Bem como ordenar usando critérios em conjunto:

```
SELECT [EMBALAGEM], [NOME DO PRODUTO] FROM [TABELA DE PRODUTOS]
ORDER BY [EMBALAGEM], [NOME DO PRODUTO]
```

	EMBALAGEM	NOME DO PRODUTO
1	Garrafa	Clean - 470 ml - Laranja
2	Garrafa	Festival de Sabores - 700 ml - Açaí
3	Garrafa	Festival de Sabores - 700 ml - Maracujá
4	Garrafa	Frescor do Verão - 470 ml - Manga
5	Garrafa	Linha Citros - 700 ml - Líma/Limão
6	Garrafa	Linha Refrescante - 700 ml - Manga
7	Garrafa	Linha Refrescante - 700 ml - Morango/Limão
8	Garrafa	Sabor da Montanha - 700 ml - Cereja
9	Garrafa	Sabor da Montanha - 700 ml - Morango
10	Garrafa	Sabor da Montanha - 700 ml - Uva
11	Garrafa	Videira do Campo - 700 ml - Cereja/Maçã
12	Lata	Clean - 350 ml - Laranja
13	Lata	Frescor do Verão - 350 ml - Manga
14	Lata	Líght - 350 ml - Melância

Consulta executada com êxito.

DESKTOP-N51LB16 (14.0 RTM) | sa (55) | SUCOS\_VENDAS | 00:00:00 | 31 linhas

16) Inclusive a direção da ordenação pode ser diferente. Uma para cada tipo de campo:

```
SELECT [EMBALAGEM], [NOME DO PRODUTO] FROM [TABELA DE PRODUTOS]
ORDER BY [EMBALAGEM] DESC, [NOME DO PRODUTO] ASC
```

	EMBALAGEM	NOME DO PRODUTO
1	PET	Clean - 1 Litro - Laranja
2	PET	Clean - 1,5 Litros - Laranja
3	PET	Clean - 2 Litros - Laranja
4	PET	Festival de Sabores - 1,5 Litros - Açaí
5	PET	Festival de Sabores - 1,5 Litros - Maracujá
6	PET	Festival de Sabores - 2 Litros - Açaí
7	PET	Light - 1,5 Litros - Melância
8	PET	Linha Citros - 1 Litro - Líma/Limão
9	PET	Linha Refrescante - 1 Litro - Manga
10	PET	Linha Refrescante - 1 Litro - Morango/Limão
11	PET	Linha Refrescante - 1,5 Litros - Manga
12	PET	Pedaços de Frutas - 1 Litro - Maçã
13	PET	Pedaços de Frutas - 1,5 Litros - Maçã
14	PET	Videira do Campo - 1,5 Litros - Melância

## Agrupando os resultados

17) Abra o arquivo **Agrupando os resultados.sql**.

18) Liste todos os clientes:

```
SELECT * FROM [dbo].[TABELA DE CLIENTES]
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NAS
1	1471156710	Érica Carvalho	R. Irajá		Jardins	São Paulo	SP	80012212	1990-09-01
2	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Águas Santa	Rio de Janeiro	RJ	22000000	2000-02-12
3	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12
4	3623344710	Marcos Nougeura	Av. Pastor Martin Luther King Junior		Irinhauma	Rio de Janeiro	RJ	22002012	1995-01-13
5	492472718	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19
6	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11
7	5576220758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP	08192029	1995-11-14
8	5648641702	Paulo César Mattos	Rua Hélio Beltrão		Tijuca	Rio de Janeiro	RJ	21002020	1991-08-30
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16

19) Agora, selecione somente o estado e o limite de crédito:

```
SELECT ESTADO, [LIMITE DE CREDITO] FROM [dbo].[TABELA DE CLIENTES]
```

	ESTADO	LIMITE DE CREDITO
1	SP	170000,00
2	RJ	100000,00
3	RJ	120000,00
4	RJ	110000,00
5	RJ	75000,00
6	RJ	170000,00
7	SP	70000,00
8	RJ	120000,00
9	SP	140000,00
10	SP	120000,00

20) Calcule o total de limite de crédito por estado. Neste caso, use o comando **GROUP BY** para obter este resultado:

```
SELECT ESTADO, SUM([LIMITE DE CREDITO]) FROM [TABELA DE CLIENTES]
GROUP BY ESTADO
```

	ESTADO	(Nenhum nome de coluna)
1	RJ	995000,00
2	SP	810000,00

21) Você pode obter também a média de crédito em cada estado:

```
SELECT ESTADO, AVG([LIMITE DE CREDITO]) FROM [TABELA DE CLIENTES]
GROUP BY ESTADO
```

ESTADO	(Nenhum nome de coluna)
1 RJ	110555,5555
2 SP	135000.00

22) Do preço de lista dos produtos, obtenha os mais caros por embalagem:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) FROM [TABELA DE PRODUTOS]
GROUP BY EMBALAGEM
```

EMBALAGEM	(Nenhum nome de coluna)
1 Garrafa	13,312
2 Lata	4,56
3 PET	38,012

23) Ou então o menor:

```
SELECT EMBALAGEM, MIN([PREÇO DE LISTA]) FROM [TABELA DE PRODUTOS]
GROUP BY EMBALAGEM
```

EMBALAGEM	(Nenhum nome de coluna)
1 Garrafa	3,768
2 Lata	2,808
3 PET	7,004

24) O comando **COUNT (\*)** conta o número de ocorrências. Por exemplo, para saber quantos produtos há por cada tipo de embalagem, execute:

```
SELECT EMBALAGEM, COUNT(*) FROM [TABELA DE PRODUTOS] GROUP BY
EMBALAGEM
```

	EMBALAGEM	(Nenhum nome de coluna)
1	Garrafa	11
2	Lata	5
3	PET	15

25) E você pode juntar os comandos de **GROUP BY** com os filtros já apresentados anteriormente:

```
SELECT BAIRRO, ESTADO, SUM([LIMITE DE CREDITO]) FROM [TABELA DE CLIENTES]
WHERE [ESTADO] = 'SP' GROUP BY BAIRRO, ESTADO
```

	BAIRRO	(Nenhum nome de coluna)
1	Santo Amaro	140000.00
2	Lapa	70000.00
3	Jardins	480000.00
4	Brás	120000.00

## Usando a condição HAVING

26) Abra o arquivo **Usando Having.sql**.

27) Execute o seguinte comando:

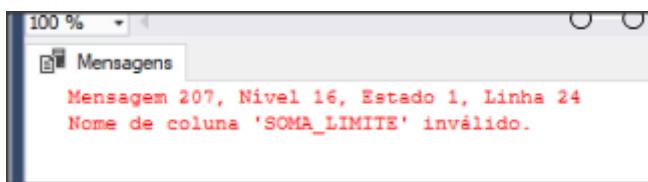
```
SELECT BAIRRO, ESTADO, SUM([LIMITE DE CREDITO]) FROM [TABELA DE CLIENTES]
WHERE [ESTADO] = 'SP' GROUP BY BAIRRO, ESTADO
```

	BAIRRO	ESTADO	(Nenhum nome de coluna)
1	Brás	SP	120000.00
2	Jardins	SP	480000.00
3	Lapa	SP	70000.00
4	Santo Amaro	SP	140000.00

28) Agora, limite esta consulta aos valores maiores que 140000. O natural seria aplicar um filtro sobre o **SUM([LIMITE DE CREDITO])**:

```
SELECT BAIRRO, ESTADO, SUM([LIMITE DE CREDITO]) AS SOMA_LIMITE FROM [TABELA DE CLIENTES]
WHERE [ESTADO] = 'SP' AND SOMA_LIMITE >= 140000 GROUP BY BAIRRO, ESTADO
```

Nesta consulta, foi aplicado um *alias* para `SUM([LIMITE DE CREDITO])`, que foi utilizado no filtro `SOMA_LIMITE >= 140000`. Mas, ao rodar esta consulta, você terá:



29) Para isso, use o **HAVING**. Digite o seguinte comando:

```
SELECT BAIRRO, ESTADO, SUM([LIMITE DE CREDITO]) AS SOMA_LIMITE FROM
[TABELA DE CLIENTES]
WHERE [ESTADO] = 'SP' GROUP BY BAIRRO, ESTADO HAVING SUM([LIMITE DE
CREDITO]) >= 140000
```

Você terá:

	BAIRRO	ESTADO	SOMA_LIMITE
1	Jardins	SP	480000,00
2	Santo Amaro	SP	140000,00

30) Olhe outra consulta, agora associada à lista de clientes:

```
SELECT * FROM [dbo].[TABELA DE CLIENTES]
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NAS
1	1471156710	Érica Carvalho	R. Iracília		Jardins	São Paulo	SP	80012212	1990-09-01
2	19290992743	Fernando Cavalcante	R. Dois de Fevereiro		Água Santa	Rio de Janeiro	RJ	22000000	2000-02-12
3	2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12
4	3623344710	Marcos Nougeira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ	22002012	1995-01-13
5	492472718	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19
6	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11
7	5576228758	Petra Oliveira	R. Benício de Abreu		Lapa	São Paulo	SP	80192029	1995-11-14
8	5648641702	Paulo César Mattos	Rua Hélio Beltrão		Tijuca	Rio de Janeiro	RJ	21002020	1991-08-30
9	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16

31) Observe o limite de crédito para cada estado, limitados a valores maiores ou iguais a 850000:

```
SELECT ESTADO, SUM([LIMITE DE CREDITO]) FROM [TABELA DE CLIENTES]
GROUP BY ESTADO
HAVING SUM([LIMITE DE CREDITO]) >= 850000
```

	ESTADO	(Nenhum nome de coluna)
1	RJ	995000,00

32) Junte condições no **HAVING**. Veja a consulta abaixo que ao mesmo tempo obtém a soma (**SUM**), o **MAX** e o **MIN** de preços:

```
SELECT EMBALAGEM, SUM([PREÇO DE LISTA]), MAX([PREÇO DE LISTA]),
MIN([PREÇO DE LISTA])
FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
```

	EMBALAGEM	(Nenhum nome de coluna)	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	Garrafa	76.9335	13.312	3.768
2	Lata	19.9935	4,56	2.808
3	PET	249.6425	38.012	7.004

33) E então aplique o **HAVING** sobre esta consulta:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]), MIN([PREÇO DE LISTA]) FROM
[TABELA DE PRODUTOS]
GROUP BY EMBALAGEM HAVING SUM([PREÇO DE LISTA]) <= 80
```

	EMBALAGEM	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	Garrafa	13.312	3.768
2	Lata	4,56	2.808

34) Você pode, inclusive, fazer condições compostas:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]), MIN([PREÇO DE LISTA]) FROM
[TABELA DE PRODUTOS]
GROUP BY EMBALAGEM HAVING SUM([PREÇO DE LISTA]) <= 80 AND MAX([PREÇO
DE LISTA]) >= 6
```

	EMBALAGEM	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	Garrafa	13.312	3.768

## Usando condições para classificar resultados

35) Abra o arquivo **Classificando Resultados.sql**.

36) Verifique a lista de produtos:

```
SELECT * FROM [TABELA DE PRODUTOS]
```

	CÓDIGO DO PRODUTO	NOME DO PRODUTO	EMBALAGEM	TAMANHO	SABOR	PREÇO DE LISTA
1	1000889	Sabor da Montanha - 700 ml - Uva	Garrafa	700 ml	Uva	6,309
2	1002334	Linha Citros - 1 Litro - Lima/Limão	PET	1 Litro	Lima/Limão	7,004
3	1002767	Videira do Campo - 700 ml - Cereja/Maçã	Garrafa	700 ml	Cereja/Maçã	8,41
4	1004327	Videira do Campo - 1,5 Litros - Melância	PET	1,5 Litros	Melância	19,51
5	1013793	Videira do Campo - 2 Litros - Cereja/Maçã	PET	2 Litros	Cereja/Maçã	24,01
6	1022450	Festival de Sabores - 2 Litros - Açaí	PET	2 Litros	Açaí	38,012
7	1037797	Clean - 2 Litros - Laranja	PET	2 Litros	Laranja	16,008
8	1040107	Light - 350 ml - Melância	Lata	350 ml	Melância	4,555
9	1041119	Linha Citros - 700 ml - Lima/Limão	Garrafa	700 ml	Lima/Limão	4,904
10	1078680	Frescor do Verão - 470 ml - Manga	Garrafa	470 ml	Manga	5,1795

37) Classifique os produtos em **CAROS** (preço maior que 12), **EM CONTA** (preço entre 7 e 12) ou **BARATOS** (preço menor que 7):

```
SELECT [NOME DO PRODUTO],
CASE WHEN [PREÇO DE LISTA] >= 12 THEN 'PRODUTO CARO'
WHEN [PREÇO DE LISTA] >= 7 AND [PREÇO DE LISTA] < 12 THEN 'PRODUTO EM CONTA'
ELSE 'PRODUTO BARATO' END
FROM
[TABELA DE PRODUTOS]
```

Você terá:

	NOME DO PRODUTO	(Nenhum nome de coluna)
1	Sabor da Montanha - 700 ml - Uva	PRODUTO BARATO
2	Linha Citros - 1 Litro - Lima/Limão	PRODUTO EM CONTA
3	Videira do Campo - 700 ml - Cereja/Maçã	PRODUTO EM CONTA
4	Videira do Campo - 1,5 Litros - Melância	PRODUTO CARO
5	Videira do Campo - 2 Litros - Cereja/Maçã	PRODUTO CARO
6	Festival de Sabores - 2 Litros - Açaí	PRODUTO CARO
7	Clean - 2 Litros - Laranja	PRODUTO CARO
8	Light - 350 ml - Melância	PRODUTO BARATO
9	Linha Citros - 700 ml - Lima/Limão	PRODUTO BARATO
10	Frescor do Verão - 470 ml - Manga	PRODUTO BARATO

38) Você pode juntar uma função de **GROUP BY** com a classificação. Como ela faz parte da agregação, a regra de construção da classificação deve estar presente na propriedades do **GROUP BY**. Veja abaixo:

```
SELECT [NOME DO PRODUTO],
CASE WHEN [PREÇO DE LISTA] >= 12 THEN 'PRODUTO CARO'
WHEN [PREÇO DE LISTA] >= 7 AND [PREÇO DE LISTA] < 12 THEN 'PRODUTO EM CONTA'
ELSE 'PRODUTO BARATO' END,
AVG([PREÇO DE LISTA])
FROM
```

```
[TABELA DE PRODUTOS]
GROUP BY [NOME DO PRODUTO],
CASE WHEN [PREÇO DE LISTA] >= 12 THEN 'PRODUTO CARO'
WHEN [PREÇO DE LISTA] >= 7 AND [PREÇO DE LISTA] < 12 THEN 'PRODUTO EM
CONTA'
ELSE 'PRODUTO BARATO' END
```

	NOME DO PRODUTO	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	Clean - 350 ml - Laranja	PRODUTO BARATO	2.808
2	Clean - 470 ml - Laranja	PRODUTO BARATO	3.768
3	Festival de Sabores - 700 ml - Maracujá	PRODUTO BARATO	4.912
4	Frescor do Verão - 350 ml - Manga	PRODUTO BARATO	3.8595
5	Frescor do Verão - 470 ml - Manga	PRODUTO BARATO	5.1795
6	Light - 350 ml - Melância	PRODUTO BARATO	4.555
7	Linha Cítricos - 700 ml - Líma/Limão	PRODUTO BARATO	4.904
8	Linha Refrescante - 700 ml - Morango/Limão	PRODUTO BARATO	6.3105
9	Pedaços de Frutas - 350 ml - Maça	PRODUTO BARATO	4.211
10	Sabor da Montanha - 700 ml - Uva	PRODUTO BARATO	6.309

39) Isso mostra que sempre que você usar uma fórmula ou função dentro da especificação do **SELECT**, e conter um agrupamento, esta função ou fórmula deve fazer parte do **GROUP BY**:

```
SELECT YEAR(DATA), COUNT(*) FROM [NOTAS FISCAIS] GROUP BY YEAR(DATA)
```

	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	2016	27284
2	2017	27001
3	2018	6500
4	2015	27092

## Usando o INNER JOIN

1) Abra o arquivo **Usando INNER JOIN.sql**.

2) Olhe as tabelas de vendedores e notas fiscais:

```
SELECT * FROM [TABELA DE VENDEDORES]
SELECT * FROM [NOTAS FISCAIS]
```

	MATRÍCULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca
2	00236	Cláudia Moraes	0.08	2013-09-17	1	Jardins
3	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana
4	00238	Pericles Alves	0.11	2016-08-21	0	Santo Amaro
	CPF	MATRÍCULA	DATA	NUMERO	IMPOSTO	
1	7771579779	00235	2015-01-01	100	0,1	
2	50534475787	00237	2015-01-01	101	0,12	
3	8502682733	00236	2015-01-01	102	0,12	
4	5840119709	00235	2015-01-01	103	0,12	
5	1471156710	00235	2015-01-01	104	0,12	

Há um campo em comum entre as duas tabelas, que é a **MATRICULA**. Por este campo é que você pode juntar as duas tabelas.

### 3) Execute:

```
SELECT * FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS
FISCAIS].MATRICULA
```

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

Neste resultado há todos os campos das duas tabelas.

### 4) Quando você faz a igualdade:

```
[TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
```

Tem que colocar o nome da tabela na frente do nome do campo. Isso porque se você usar:

```
MATRICULA = MATRICULA
```

O SQL Server não saberá de que tabela pertence os campos **MATRICULA**.

### 5) Crie um *alias*, que pode ser um nome ou uma letra, para colocar ao lado do nome da tabela. O comando ficará assim:

```
SELECT * FROM [TABELA DE VENDEDORES] A INNER JOIN
[NOTAS FISCAIS] B ON A.MATRICULA = B.MATRICULA
```

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

6) Selecione os campos a serem exibidos no **JOIN**. Você pode aplicar tudo o que já foi visto até agora nesta seleção (filtros, **TOP**, **DISTINCT**, **GROUP BY**, **ORDER BY**, etc). Por exemplo, conte o número de notas fiscais por cada vendedor:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], COUNT(*)
FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME]
```

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

7) Melhore a consulta e veja o número de notas fiscais, mas dentro de cada ano:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME],
YEAR(DATA),
COUNT(*) FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA)
```

	MATRICULA	NOME	(Nenhum nome de coluna)	(Nenhum nome de coluna)
8	00236	Cláudia Morais	2017	8968
9	00237	Roberta Martins	2015	9085
10	00235	Márcio Almeida Silva	2016	9152
11	00236	Cláudia Morais	2015	9032
12	00237	Roberta Martins	2018	2112

8) Nesta consulta final, ajuste a consulta acima, mostrando por ordem de data e vendedor:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME],
YEAR(DATA),
COUNT(*) FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA)
```

```
ORDER BY YEAR(DATA), [TABELA DE VENDEDORES].[NOME]
```

	MATRICULA	NOME	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	00236	Cláudia Morais	2015	9032
2	00235	Márcio Almeida Silva	2015	8975
3	00237	Roberta Martins	2015	9085
4	00236	Cláudia Morais	2016	9185
5	00235	Márcio Almeida Silva	2016	9152

9) Existe uma forma antiga de fazer **JOIN**, que é considerá-lo como se fosse um filtro e declarar as tabelas que vão ser unidas após o **FROM**, com as mesmas separadas por vírgula:

```
SELECT * FROM [TABELA DE VENDEDORES], [NOTAS FISCAIS]
WHERE [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
```

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0.08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01

## Usando LEFT e RIGHT JOIN

10) Abra o arquivo **Usando LEFT e RIGHT JOIN.sql**.

11) Junte a tabelas de clientes com a tabelas de notas fiscais pelo campo **CPF**, que é o campo em comum das duas:

```
SELECT * FROM [TABELA DE CLIENTES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCII
1	7771579779	Marcelo Mattos	R. Eduardo Luis Lopes		Brás	São Paulo	SP	88202912	1992-03-25
2	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11
3	8502682733	Valdeci da Silva	R. Srg. Edíson de Oliveira		Jardins	São Paulo	SP	82122020	1995-10-07
4	5840119709	Gabriel Araújo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16
5	1471156710	Érica Carvalho	R. Irajá		Jardins	São Paulo	SP	80012212	1990-09-01
6	94387575700	Walber Lontra	R. Cel. Almeida		Piedade	Rio de Janeiro	RJ	22000201	1989-06-20
7	3623344710	Marcos Nougeura	Av. Pastor Martin Luther King Junior		Iraráma	Rio de Janeiro	RJ	22002012	1995-01-13

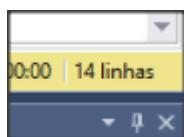
12) Aplicando o **COUNT(\*)** com **GROUP BY**, conte o número de notas fiscais por cada cliente:

```
SELECT [TABELA DE CLIENTES].[NOME], COUNT(*) FROM [TABELA DE CLIENTES]
INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
GROUP BY [TABELA DE CLIENTES].[NOME]
```

	NOME	(Nenhum nome de coluna)
1	Abel Silva	6365
2	Carlos Eduardo	6085
3	Cesar Teixeira	6226
4	Edson Meilleles	6308
5	Eduardo Jorge	6233
6	Erica Carvalho	6310
7	Fernando Cavalcante	6240
8	Gabriel Araujo	6273

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (51) | SUCOS\_VENDAS | 00:00:00 | 14 linhas

Note que esta consulta retornou 14 linhas:



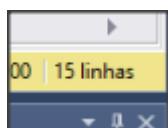
13) Verifique quantos clientes há na tabela de clientes:

```
SELECT * FROM [TABELA DE CLIENTES]
```

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASC
10	7771579779	Marcelo Mattos	R. Eduardo Luis Lopes		Brás	São Paulo	SP	88202912	1992-03-25
11	8502682733	Valdeci da Silva	R. Srg. Edíson de Oliveira		Jardins	São Paulo	SP	82122020	1995-10-07
12	8719655770	Carlos Eduardo	Av. Gen. Guedes da Fontoura		Jardins	São Paulo	SP	81192002	1983-12-20
13	9283760794	Edson Meilleles	R. Pinto de Azevedo		Cidade N...	Rio de Janeiro	RJ	22002002	1995-10-07
14	94387575700	Walber Lontra	R. Cel. Almeida		Piedade	Rio de Janeiro	RJ	22000201	1989-06-20
15	95939180787	Fábio Carvalho	R. dos Jacarandás da Península		Barra da ...	Rio de Janeiro	RJ	22002020	1992-01-05

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (51) | SUCOS\_VENDAS | 00:00:00 | 15 linhas

Esta consulta retorna 15 linhas:



Isto significa que, quando você vê a tabela de clientes, há 15 linhas, mas quando agrupa a consulta, fazendo **JOIN** com a tabela de notas fiscais, há 14 clientes. Logo, existe um cliente que faz parte do cadastro e que nunca fez uma compra. Pelo **LEFT JOIN**, você pode descobrir este cliente.

14) Para descobrir o cliente que nunca comprou, faça:

```
SELECT [TABELA DE CLIENTES].[NOME], COUNT(*) FROM [TABELA DE CLIENTES]
LEFT JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
```

```
GROUP BY [TABELA DE CLIENTES]. [NOME]
```

	NOME	(Nenhum nome de coluna)
5	Eduardo Jorge	6233
6	Érica Carvalho	6310
7	Fábio Carvalho	1
8	Fernando Cavalcante	6240
9	Gabriel Araujo	6273
10	Marcelo Mattos	6179
11	Marcos Nougueira	6352
12	Paulo César Mattos	6379

Consulta executada com êxito. DESKTOP-N51LB16 (14.0 RTM) | sa (51) | SUCOS\_VENDAS | 00:00:00 | 15 linhas

O cliente que tem valor de nota fiscal igual a 1 é o cliente que nunca comprou. O valor 1 aparece porque o COUNT(\*) faz uma contagem de registros e este cliente aparece na consulta, apesar de não ter nenhuma nota. Por causa do LEFT JOIN, ele aparece apenas uma vez e por isso o COUNT(\*) escreve para ele o valor 1.

15) Para ter certeza deste resultado, coloque o CPF na consulta:

```
SELECT [TABELA DE CLIENTES].[CPF], [TABELA DE CLIENTES].[NOME],
COUNT(*) FROM [TABELA DE CLIENTES] LEFT JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
GROUP BY [TABELA DE CLIENTES].[CPF], [TABELA DE CLIENTES].[NOME]
```

Você terá:

	CPF	NOME	(Nenhum nome de coluna)
9	5840119709	Gabriel Araujo	6273
10	7771579779	Marcelo Mattos	6179
11	8502682733	Valdeci da Silva	6251
12	8719655770	Carlos Eduardo	6085
13	9283760794	Edson Meilleles	6308
14	94397575700	Walber Lontra	6292
15	95939180787	Fábio Carvalho	1

Pegando o CPF que aparece na consulta acima, você pode ver se há alguma nota fiscal associada a ele na tabela de notas fiscais:

```
SELECT * FROM [NOTAS FISCAIS] WHERE CPF = '95939180787'
```

Como esperado, resultado vazio:

CPF	MATRICULA	DATA	NUMERO	IMPOSTO

## Usando FULL e CROSS JOIN

16) Abra o arquivo Usando FULL e CROSS JOIN.sql.

17) Veja a tabela de vendedores e de clientes:

```
SELECT * FROM [TABELA DE VENDEDORES]
SELECT * FROM [TABELA DE CLIENTES]
```

	MATRÍCULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO					
1	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca					
2	00236	Cláudia Moraes	0.08	2013-09-17	1	Jardins					
3	00237	Roberta Martins	0.11	2017-03-18	1	Copacabana					
	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO	IDADE	SEXO
1	1471156710	Érica Carvalho	R. Iniquitia		Jardins	São Paulo	SP	80012212	1990-09-01	27	F
2	19290992...	Fernando Ca...	R. Nois de F...		Água ...	Rio de J...	RJ	22000000	2000-02-12	18	M
3	2600586709	César Teixeira	Rua Conde ...		Tijuca	Rio de J...	RJ	22020001	2000-03-12	18	M

Resultado: 4 vendedores e 15 clientes. O campo em comum entre eles é  
O BAIRRO.

18) Faça o INNER JOIN entre a tabela de clientes e de vendedores:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE CLIENTES].BAIRRO FROM
[TABELA DE VENDEDORES] INNER JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```

	BAIRRO	BAIRRO
1	Jardins	Jardins
2	Tijuca	Tijuca
3	Tijuca	Tijuca
4	Tijuca	Tijuca
5	Santo Amaro	Santo Amaro
6	Jardins	Jardins
7	Jardins	Jardins

19) Como o resultado foi de 7 linhas, significa que existem vendedores com bairros que não estão na tabela de clientes e vice-versa. O LEFT JOIN pode fazer com que você possa descobrir estas não correspondências:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM
[TABELA DE VENDEDORES] LEFT JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```

	BAIRRO	NOME	BAIRRO	NOME
1	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
2	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
3	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos
4	Jardins	Cláudia Moraes	Jardins	Érica Carvalho
5	Jardins	Cláudia Moraes	Jardins	Valdeci da Silva
6	Jardins	Cláudia Moraes	Jardins	Carlos Eduardo
7	Copacabana	Roberta Martins	NULL	NULL
8	Santo Amaro	Pericles Alves	Santo Amaro	Gabriel Araujo

Resultado: 8 linhas

20) Como você ainda não obteve o resultado da maior tabela, significa que existem também vendedores com bairros que não existem na tabela de clientes. Para isso, use o **RIGHT JOIN** para descobrir isto:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],  
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM  
[TABELA DE VENDEDORES] RIGHT JOIN [TABELA DE CLIENTES] ON  
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```

	BAIRRO	NOME	BAIRRO	NOME
1	Jardins	Cláudia Moraes	Jardins	Érica Carvalho
2	NULL	NULL	Água Santa	Fernando Cavalcante
3	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
4	NULL	NULL	Inhauma	Marcos Nougeira
5	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
6	NULL	NULL	Humaitá	Abel Silva
7	NULL	NULL	Lapa	Petra Oliveira
8	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos

21) Para descobrir, ao mesmo tempo, o cliente que não tem bairro na tabela de vendedores e o vendedor que não tem bairro na tabela de clientes, use

**O FULL JOIN:**

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],  
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM  
[TABELA DE VENDEDORES] FULL JOIN [TABELA DE CLIENTES] ON  
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```

	BAIRRO	NOME	BAIRRO	NOME
1	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
2	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
3	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos
4	Jardins	Cláudia Moraes	Jardins	Érica Carvalho
5	Jardins	Cláudia Moraes	Jardins	Valdeci da Silva
6	Jardins	Cláudia Moraes	Jardins	Carlos Eduardo
7	Copacabana	Roberta Martins	NULL	NULL
8	Santo Amaro	Pericles Alves	Santo Amaro	Gabriel Araujo
9	NULL	NULL	Água Santa	Fernando Cavalcante
10	NULL	NULL	Inhauma	Marcos Nougeira
11	NULL	NULL	Humaitá	Abel Silva
12	NULL	NULL	Lapa	Petra Oliveira
13	NULL	NULL	Brás	Marcelo Mattos
14	NULL	NULL	Cidade Nova	Edson Meilleles
15	NULL	NULL	Piedade	Walber Lontra

22) O **CROSS JOIN** irá possibilitar a análise combinatória da tabela de vendedores e clientes:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE CLIENTES].BAIRRO FROM
[TABELA DE VENDEDORES] CROSS JOIN [TABELA DE CLIENTES]
```

	BAIRRO	BAIRRO
29	Jardins	Piedade
30	Jardins	Barra da ...
31	Copac...	Jardins
32	Copac...	Água Santa
33	Copac...	Tijuca
34	Copac...	Inhauma
35	Copac...	Tijuca

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS\_VENDAS | 00:00:00 | 60 linhas

O resultado foi justamente 60 linhas, que são os 15 clientes vezes os 4 vendedores.

## Juntando consultas

23) Abra o arquivo **Juntando consultas.sql**.

24) Veja a lista de bairros dos clientes e dos vendedores:

```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE
VENDEDORES]
```

	BAIRRO
6	Inhauma
7	Jardins
8	Lapa

	BAIRRO
1	Copacabana
2	Jardins
3	Santo Amaro
4	Tijuca

25) Use o **UNION**, juntando essas duas consultas:

```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
UNION
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE
VENDEDORES]
```

	BAIRRO
1	Água Santa
2	Brama da Tijuca
3	Brás
4	Cidade Nya
5	Copacabana
6	Humaitá
7	Inhauma
8	Jardins
9	Lapa
10	Piedade
11	Santo Amaro
12	Tijuca

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS\_VENDAS | 00:00:00 | 12 linhas

Com o **UNION**, você tem 12 bairros sem que nenhum se repita.

26) Execute agora o **UNION ALL**. Teremos:

```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE
VENDEDORES]
```

	BAIRRO
4	Cidade Nova
5	Humaitá
6	Inhauma
7	Jardins
8	Lapa
9	Piedade
10	Santo Amaro
11	Tijuca
12	Copacabana
13	Jardins
14	Santo Amaro
15	Tijuca

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS\_VENDAS | 00:00:00 | 15 linhas

Você terá 15 linhas como resultado, porque os bairros em comum aparecem duplicados.

27) Os campos de um **UNION** devem ter a mesma correspondência de tipo.

Veja a consulta abaixo:

```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE
CLIENTES].[DATA DE NASCIMENTO] FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE
VENDEDORES].[NOME] FROM [TABELA DE VENDEDORES]
```

	Mensagens
	Mensagem 241, Nível 16, Estado 1, Linha 14 Falha ao converter data e/ou hora da cadeia de caracteres.

Você terá um erro porque o segundo campo das duas consultas que estão sendo unidas pelo **UNION** são de tipos diferentes.

**28) Execute então:**

```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE
CLIENTES].[NOME] FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE
VENDEDORES].[NOME] FROM [TABELA DE VENDEDORES]
```

	BAIRRO	NOME
1	Aqua Santa	Fernando Cavalcante
2	Barra da Tijuca	Fábio Carvalho
3	Brás	Marcelo Mattos
4	Cidade Nova	Edson Meléllles
5	Humaitá	Abel Silva
6	Inhauma	Marcos Nogueira
7	Jardins	Carlos Eduardo
8	Jardins	Érica Carvalho
9	Jardins	Valdeci da Silva
10	Lapa	Petra Oliveira
11	Piedade	Walber Lontra
12	Santo Amaro	Gabriel Araújo

Consulta executada com êxito. DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS\_VENDAS | 00:00:00 | 19 linhas

**29) O que você pode fazer, em muitos dos casos, é criar uma coluna que identifique a origem do campo quando é efetuado o **UNION**:**

```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE
CLIENTES].[NOME], 'CLIENTE' FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE
VENDEDORES].[NOME], 'VENDEDOR' FROM [TABELA DE VENDEDORES]
```

	BAIRRO	NOME	(Nenhum nome de coluna)
8	Jardins	Érica Carvalho	CLIENTE
9	Jardins	Valdeci da Silva	CLIENTE
10	Lapa	Petra Oliveira	CLIENTE
11	Piedade	Walber Lontra	CLIENTE
12	Santo Amaro	Gabriel Araújo	CLIENTE
13	Tijuca	César Teixeira	CLIENTE
14	Tijuca	Eduardo Jorge	CLIENTE
15	Tijuca	Paulo César Mattos	CLIENTE
16	Copacabana	Roberta Martins	VENDEDOR
17	Jardins	Cláudia Moraes	VENDEDOR
18	Santo Amaro	Pericles Alves	VENDEDOR
19	Tijuca	Márcio Almeida Silva	VENDEDOR

Consulta executada com êxito. DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS\_VENDAS | 00:00:00 | 19 linhas

## Sub-consultas

**30) Abra o arquivo Sub consultas.sql.**

**31) Veja a consulta de clientes e seus respectivos bairros:**

```
SELECT NOME, BAIRRO FROM [TABELA DE CLIENTES]
```

NOME	BAIRRO
1 Érica Carvalho	Jardins
2 Fernando Cavalcante	Água Santa
3 César Teixeira	Tijuca
4 Marcos Nougeira	Inhauma
5 Eduardo Jorge	Tijuca
6 Abel Silva	Humaitá
7 Petra Oliveira	Lapa
8 Paulo César Mattos	Tijuca

Consulta executada com êxito.

32) Liste somente os clientes que possuam os mesmos bairros que os vendedores. Como você não sabe de antemão quais são os bairros dos vendedores, use a sub-consulta como condição de filtro:

```
SELECT NOME, BAIRRO FROM [TABELA DE CLIENTES]
WHERE BAIRRO IN (
    SELECT BAIRRO FROM [TABELA DE VENDEDORES] )
```

NOME	BAIRRO
1 Érica Carvalho	Jardins
2 César Teixeira	Tijuca
3 Eduardo Jorge	Tijuca
4 Paulo César Mattos	Tijuca
5 Gabriel Araujo	Santo Amaro
6 Valdeci da Silva	Jardins
7 Carlos Eduardo	Jardins

33) Veja o máximo dos preços por embalagem:

```
SELECT EMBALAGEM, MAX ([PREÇO DE LISTA]) FROM [TABELA DE PRODUTOS]
GROUP BY EMBALAGEM
```

EMBALAGEM	(Nenhum nome de coluna)
1 Garrafa	13,312
2 Lata	4,56
3 PET	38,012

34) Aplique um filtro sobre esta consulta, usando ela como uma sub-consulta no **FROM** de outra consulta:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM (SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM
[TABELA DE PRODUTOS] GROUP BY EMBALAGEM)
```

```
NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```

	EMBALAGEM	MAX_PRECO
1	Lata	4.56

O resultado acima acaba fazendo o mesmo que o **HAVING**.

## Visão

35) Abra o arquivo **Visões.sql**.

36) Note que na consulta abaixo:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM (SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM
[TABELA DE PRODUTOS] GROUP BY EMBALAGEM)
NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```

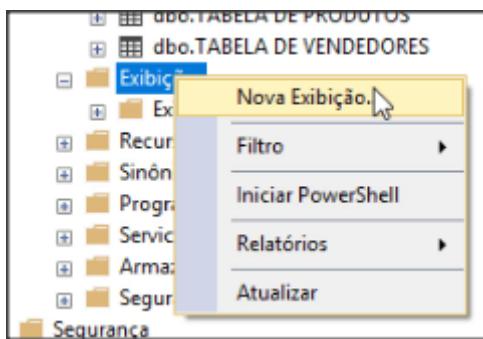
Há uma sub-consulta:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE
PRODUTOS] GROUP BY EMBALAGEM
```

Executando esta sub-consulta, você verá:

	EMBALAGEM	MAX_PRECO
1	Garrafa	13,312
2	Lata	4,56
3	PET	38,012

37) Expanda a pasta **Exibições** do banco de dados **SUCOS\_VENDAS**. Clique com o botão da direita do mouse sobre este item e escolha **Nova Exibição**:



38) Serão mostradas as tabelas. Aqui, você pode escolher as tabelas que farão parte das *views*. Mas, no caso, já há a consulta SQL pronta. Logo, clique em **OK** sem selecionar tabela alguma:

Coluna	Alias	Tabela	Saída	Tipo de Class...	Ordem de Cla...	Filtrar	Ou...	Ou...	Ou...

```
SELECT
FROM
```

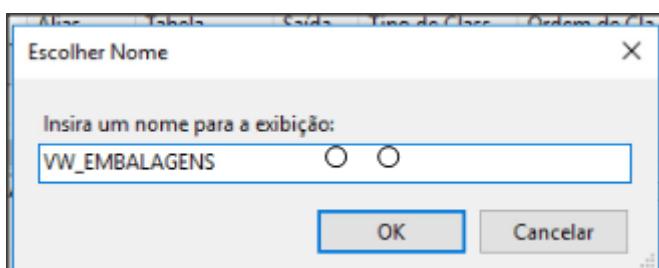
39) onde há escrito **SELECT FROM**, cole a sub-consulta que será usada para a criação da *view*:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
```

40) Clique no ícone **Salvar** na barra de ferramentas principal:



41) Crie um nome para esta visão (**VW\_EMBALAGENS**):



42) Clique em **OK** e você terá a representação gráfica da visão:

```

CREATE VIEW VW_EMBALAGENS AS
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO
FROM dbo.[TABELA DE PRODUTOS]
GROUP BY EMBALAGEM
  
```

43) Agora, use a visão na consulta como se fosse uma tabela. Veja abaixo a seleção original e a visão:

```

SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
SELECT * FROM [dbo].[VW_EMBALAGENS]
  
```

EMBALAGEM	MAX_PRECO
Garrafa	13,312
Lata	4,56
PET	38,012

EMBALAGEM	MAX_PRECO
Garrafa	13,312
Lata	4,56
PET	38,012

44) Você pode usar a visão como se fosse uma tabela na consulta que, antes, foi usada a sub-consulta:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM VW_EMBALAGENS NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```

	EMBALAGEM	MAX_PRECO
1	Lata	4.56

## Funções de string

1) Abra o arquivo **Consultas string.sql**

2) Execute cada linha e verifique o resultado de cada função:

```
SELECT LTRIM('      OLA')
SELECT RTRIM('OLA      ')
SELECT CONCAT('OLA ', 'TUDO BEM')
SELECT 'OLA ' + 'TUDO BEM'
SELECT LEFT('RUA AUGUSTA', 3)
SELECT RIGHT('RUA AUGUSTA', 7)
SELECT UPPER('rua augusta')
SELECT LOWER('RUA AUGUSTA')
SELECT REPLACE('R. AUGUSTA', 'R.', 'RUA')
SELECT SUBSTRING('RUA AUGUSTA', 1, 3)
SELECT SUBSTRING('RUA AUGUSTA', 2, 4)
SELECT LEN('RUA AUGUSTA')
SELECT * FROM [TABELA DE CLIENTES]
SELECT CONCAT(NOME, ' (' , CPF, ') ') FROM [TABELA DE CLIENTES]
```

## Funções de datas

3) Abra o arquivo **Funcoes de datas.sql**.

4) Execute cada linha e verifique o resultado de cada função:

```
SELECT SYSDATETIME()
SELECT SYSDATETIMEOFFSET()
SELECT SYSUTCDATETIME()
SELECT CURRENT_TIMESTAMP
SELECT GETDATE()
SELECT GETUTCDATE()
SELECT DATENAME(YEAR, GETDATE())
SELECT DATENAME(MICROSECOND, GETDATE())
SELECT DATENAME(MINUTE, GETDATE())
SELECT DATENAME(MONTH, GETDATE())
SELECT DATEPART(MONTH, GETDATE())
SELECT DAY(GETDATE())
```

```

SELECT YEAR(GETDATE())
SELECT DATEFROMPARTS(2015, 9, 1)
SELECT DATENAME(MONTH, DATEFROMPARTS(2015, 9, 1))
SELECT DATETIME2FROMPARTS(2015, 9, 1, 13, 12, 11, 120, 4)
SELECT DATEDIFF(MONTH, DATEFROMPARTS(2015, 9, 1), GETDATE())
SELECT DATEADD(MONTH, 5, GETDATE())
SELECT ISDATE('2018-01-01')
SELECT ISDATE('2018-25-28')
SELECT * FROM [NOTAS FISCAIS]

SELECT [DATA], CONCAT(DATENAME(DAY, [DATA]), ' ', DATENAME(MONTH, [DATA]), ' ', DATENAME(YEAR, [DATA]))
FROM [NOTAS FISCAIS]

```

## Funções matemáticas

5) Abra o arquivo **Funcoes matemáticas.sql**

6) Execute cada linha e verifique o resultado de cada função:

```

SELECT CEILING(12.333223)
SELECT FLOOR(12.333223)
SELECT RAND()
SELECT ROUND(12.33323323, 2)
SELECT * FROM [ITENS NOTAS FISCAIS]
SELECT [QUANTIDADE], [PREÇO], ROUND(([QUANTIDADE] * [PREÇO]), 1) FROM
[ITENS NOTAS FISCAIS]

```

## Conversão de dados

7) Abra o arquivo **Conversao de dados.sql**.

8) Execute cada linha e verifique o resultado de cada função:

```

SELECT CONVERT(VARCHAR, GETDATE(), 101)
SELECT CONVERT(VARCHAR, GETDATE(), 113)
SELECT CONVERT(VARCHAR, GETDATE(), 130)
SELECT CONVERT(decimal(10,5), 193.57)
SELECT * FROM [TABELA DE PRODUTOS]

SELECT 'O preço do produto ' + [NOME DO PRODUTO] + ' é ' + [PREÇO DE LISTA]
FROM [TABELA DE PRODUTOS]

SELECT 'O preço do produto ' + [NOME DO PRODUTO] + ' é ' +
CONVERT(VARCHAR, [PREÇO DE LISTA])
FROM [TABELA DE PRODUTOS]

```

## Relatório de vendas válidas

1) Objetivo desta prática é a de montar o relatório de vendas por cliente e comparar se o total de venda (quantidade) em um determinado mês superou ou não o limite máximo de venda por cliente (quantidade limite contida no cadastro):

	NOME	MES_ANO	QUANTIDADE	(Nenhum nome de coluna)
1	Abel Silva	2015-01	23176	VENDA VÁLIDA
2	Abel Silva	2015-02	21325	VENDA VÁLIDA
3	Abel Silva	2015-03	23925	VENDA VÁLIDA
4	Abel Silva	2015-04	22390	VENDA VÁLIDA
5	Abel Silva	2015-05	22567	VENDA VÁLIDA
6	Abel Silva	2015-06	22851	VENDA VÁLIDA
7	Abel Silva	2015-07	19573	VENDA VÁLIDA
8	Abel Silva	2015-08	20160	VENDA VÁLIDA
9	Abel Silva	2015-09	22153	VENDA VÁLIDA
10	Abel Silva	2015-10	22114	VENDA VÁLIDA
11	Abel Silva	2015-11	19077	VENDA VÁLIDA
12	Abel Silva	2015-12	21968	VENDA VÁLIDA
13	Abel Silva	2016-01	26541	VENDA INVÁLIDA
14	Abel Silva	2016-02	21912	VENDA VÁLIDA
15	Abel Silva	2016-03	22377	VENDA VÁLIDA
16	Abel Silva	2016-04	20908	VENDA VÁLIDA
17	Abel Silva	2016-05	24125	VENDA VÁLIDA
18	Abel Silva	2016-06	19353	VENDA VÁLIDA

2) Abra o arquivo **Vendas válidas EXEMPLO.sql**.

3) Há 3 tabelas que você irá usar para montar este relatório. A tabela de clientes, para buscar os dados do clientes (nome, por exemplo) e também o valor total de crédito contido no seu cadastro. A outra tabela é a de notas fiscais, onde tem a data da venda. A ligação entre estas duas tabelas é o CPF do cliente. Finalmente, você irá acessar a tabela de itens de nota fiscais, onde você obtém o valor faturado. Esta última tabela irá se juntar com a tabela de notas fiscais pelo número da nota:

```
SELECT * FROM [TABELA DE CLIENTES]
SELECT * FROM [ITENS NOTAS FISCAIS]
SELECT * FROM [NOTAS FISCAIS]
```

4) Primeiramente, junte a tabela de notas fiscais com a tabela de itens:

```
SELECT * FROM [NOTAS FISCAIS] NF
INNER JOIN [ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
```

	CPF	MATRICULA	DATA	NUMERO	IMPOSTO	NUMERO	CÓDIGO DO PRODUTO	QUANTIDADE	PREÇO
1	7771579779	00235	2015-01-01	100	0,1	100	1013793	63	24,01
2	7771579779	00235	2015-01-01	100	0,1	100	1101035	26	9,0105
3	7771579779	00235	2015-01-01	100	0,1	100	520380	67	12,011
4	7771579779	00235	2015-01-01	100	0,1	100	773912	66	8,008
5	50534475787	00237	2015-01-01	101	0,12	101	1000889	35	6,309
6	50534475787	00237	2015-01-01	101	0,12	101	479745	65	3,768
7	8502682733	00236	2015-01-01	102	0,12	102	1101035	84	9,0105
8	8502682733	00236	2015-01-01	102	0,12	102	235653	37	3,8595
9	8502682733	00236	2015-01-01	102	0,12	102	394479	74	8,409
10	5840119709	00235	2015-01-01	103	0,12	103	520380	29	12,011

5) A seguir, segregue os campos importantes deste **JOIN**, aplicando algumas funções. Você irá obter apenas o CPF, a data convertida para mês e ano (para ver somente os dados dentro do mês) e a quantidade vendida:

```
SELECT NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7) AS ANO_MES,
       INF.QUANTIDADE FROM [NOTAS FISCAIS] NF
INNER JOIN [ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
```

	CPF	DATA_CONVERTIDA	QUANTIDADE
361	50534475787	2015-01-03	65
362	5648641702	2015-01-03	91
363	5576228758	2015-01-03	79
364	5576228758	2015-01-03	19
365	5576228758	2015-01-03	21
366	8719655770	2015-01-03	17
367	8719655770	2015-01-03	91
368	1471156710	2015-01-03	40
369	1471156710	2015-01-03	78
370	1471156710	2015-01-03	21

6) Agrupe esta consulta para obter o valor total, por cliente e por mês/ano:

```
SELECT NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7) AS ANO_MES,
       SUM(INF.QUANTIDADE) AS QUANTIDADE_MES FROM [NOTAS FISCAIS] NF
INNER JOIN [ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
GROUP BY NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7)
```

Resultados Mensagens

	CPF	ANO_MES	QUANTIDADE_MES
1	9283760794	2016-04	23352
2	9283760794	2017-07	21705
3	8719659070	2016-09	22665
4	9283760794	2015-02	19614
5	8502682733	2016-06	23751
6	1471156710	2017-08	20377
7	2600586709	2015-07	21672
8	8502682733	2017-04	19405
9	5648641702	2016-10	21866
10	19290992743	2017-06	20034

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 546 linhas

7) Guarde esta consulta e verifique, agora, os dados do limite de vendas em quantidades contidas na tabela de clientes. Selecione apenas os campos necessários a serem usados no relatório:

```
SELECT TC.NOME, TC.[VOLUME DE COMPRA] FROM [TABELA DE CLIENTES] TC
```

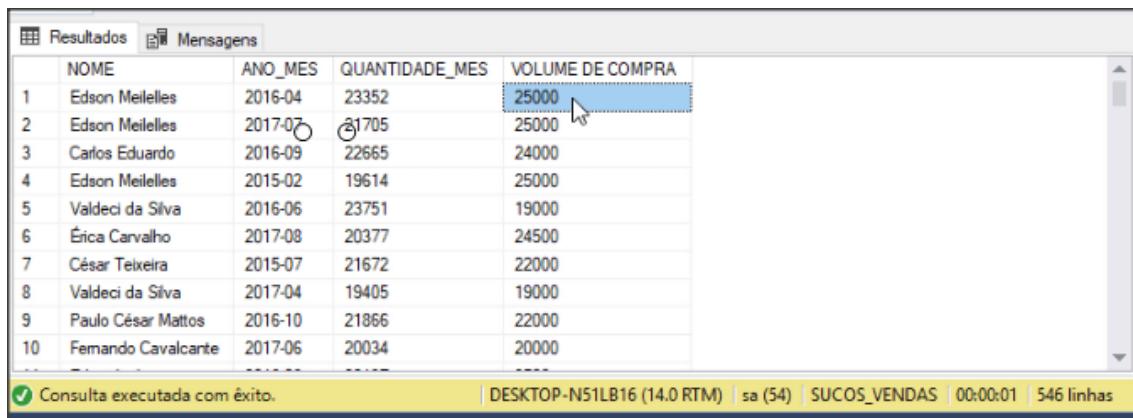
Resultados Mensagens

	NOME	VOLUME DE COMPRA
1	Érica Carvalho	24500
2	Fernando Cavalcante	20000
3	César Teixeira	22000
4	Marcos Nougeira	22000
5	Eduardo Jorge	9500
6	Abel Silva	26000
7	Petra Oliveira	16000
8	Paulo César Mattos	22000
9	Gabriel Araujo	21000
10	Marcelo Mattos	20000

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 15 linhas

8) Junte as duas consultas acima. A que obtém dados de vendas e a que obtém dados do limite máximo:

```
SELECT TC.NOME, CQ.ANO_MES, CQ.QUANTIDADE_MES, TC.[VOLUME DE COMPRA]
FROM
(SELECT NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7) AS
ANO_MES,
SUM(INF.QUANTIDADE) AS QUANTIDADE_MES FROM [NOTAS FISCAIS] NF
INNER JOIN [ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
GROUP BY NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7)) CQ
INNER JOIN [TABELA DE CLIENTES] TC ON TC.CPF = CQ.CPF
```

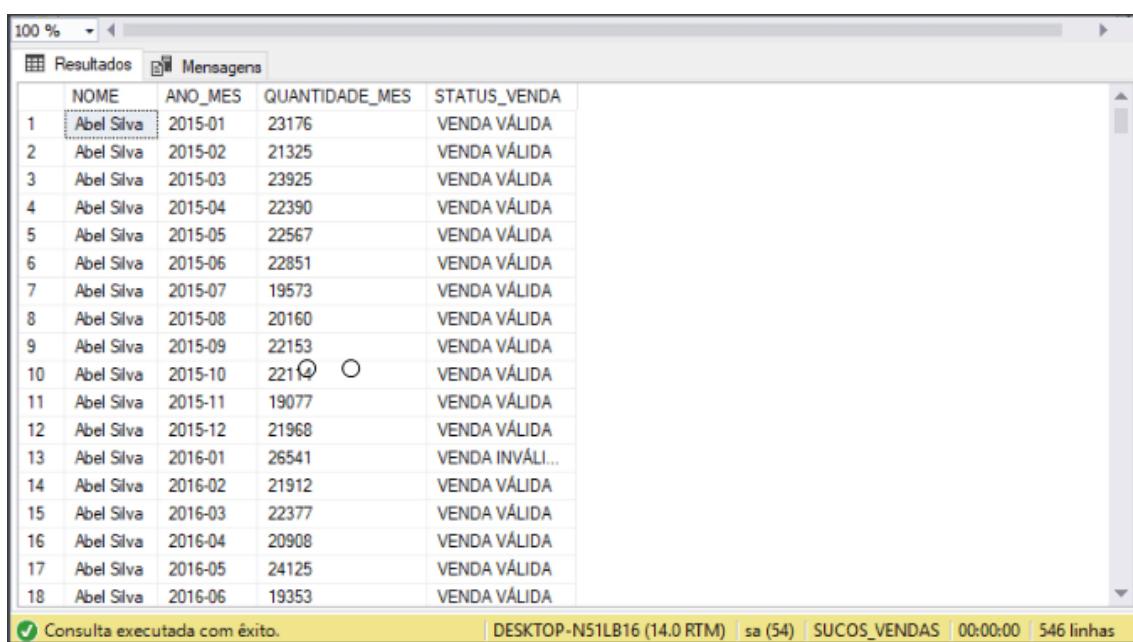


	NOME	ANO_MES	QUANTIDADE_MES	VOLUME DE COMPRA
1	Edson Meilleles	2016-04	23352	25000
2	Edson Meilleles	2017-07	1705	25000
3	Carlos Eduardo	2016-09	22665	24000
4	Edson Meilleles	2015-02	19614	25000
5	Valdeci da Silva	2016-06	23751	19000
6	Érica Carvalho	2017-08	20377	24500
7	César Teixeira	2015-07	21672	22000
8	Valdeci da Silva	2017-04	19405	19000
9	Paulo César Mattos	2016-10	21866	22000
10	Fernando Cavalcante	2017-06	20034	20000

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:01 | 546 linhas

9) Com duas colunas comparando as vendas com os limites, crie um **CASE** que irá testar se a venda do mês superou ou não a venda limite e escrever um *label* para isso. Apresente esta consulta ordenada e você terá o relatório final:

```
SELECT AUX1.NOME, AUX1.ANO_MES, AUX1.QUANTIDADE_MES,
CASE WHEN AUX1.QUANTIDADE_MES <= AUX1.[VOLUME DE COMPRA/10] THEN
'VENDA VÁLIDA'
WHEN AUX1.QUANTIDADE_MES > AUX1.[VOLUME DE COMPRA/10] THEN 'VENDA INVÁLIDA'
END AS STATUS_VENDA
FROM
(SELECT TC.NOME, CQ.ANO_MES, CQ.QUANTIDADE_MES, TC.[VOLUME DE COMPRA]
FROM
(SELECT NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7) AS
ANO_MES,
SUM(INF.QUANTIDADE) AS QUANTIDADE_MES FROM [NOTAS FISCAIS] NF
INNER JOIN [ITENS NOTAS FISCAIS] INF
ON NF.NUMERO = INF.NUMERO
GROUP BY NF.CPF, SUBSTRING(CONVERT(VARCHAR, NF.[DATA], 120), 1, 7)) CQ
INNER JOIN [TABELA DE CLIENTES] TC ON TC.CPF = CQ.CPF) AUX1
ORDER BY AUX1.NOME, AUX1.ANO_MES
```



	NOME	ANO_MES	QUANTIDADE_MES	STATUS_VENDA
1	Abel Silva	2015-01	23176	VENDA VÁLIDA
2	Abel Silva	2015-02	21325	VENDA VÁLIDA
3	Abel Silva	2015-03	23925	VENDA VÁLIDA
4	Abel Silva	2015-04	22390	VENDA VÁLIDA
5	Abel Silva	2015-05	22567	VENDA VÁLIDA
6	Abel Silva	2015-06	22851	VENDA VÁLIDA
7	Abel Silva	2015-07	19573	VENDA VÁLIDA
8	Abel Silva	2015-08	20160	VENDA VÁLIDA
9	Abel Silva	2015-09	22153	VENDA VÁLIDA
10	Abel Silva	2015-10	22100	VENDA VÁLIDA
11	Abel Silva	2015-11	19077	VENDA VÁLIDA
12	Abel Silva	2015-12	21968	VENDA VÁLIDA
13	Abel Silva	2016-01	26541	VENDA INVÁLIDA
14	Abel Silva	2016-02	21912	VENDA VÁLIDA
15	Abel Silva	2016-03	22377	VENDA VÁLIDA
16	Abel Silva	2016-04	20908	VENDA VÁLIDA
17	Abel Silva	2016-05	24125	VENDA VÁLIDA
18	Abel Silva	2016-06	19353	VENDA VÁLIDA

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 546 linhas

## Relatório de vendas por sabor

10) O objetivo é montar um relatório de vendas (faturamento) por sabor, dentro de um período de tempo (no caso, no ano de 2016) e criar uma coluna com o percentual de participação de cada venda. É importante exibir os dados ordenados por valor do faturamento, do maior para o menor:

	SABOR	ANO	FATURAMENTO	PARTICIPACAO
1	Açaí	2016	10024014.10	23.6627%
2	Melância	2016	6128144.29	14.4661%
3	Manga	2016	5725142.63	13.5148%
4	Maçã	2016	4321369.74	10.201%
5	Cereja/...	2016	4071635.98	9.6115%
6	Laranja	2016	3844934.55	9.07635%
7	Maracujá	2016	1987886.75	4.6926%
8	Morang...	2016	1913604.25	4.51725%
9	Lima/Li...	2016	1508300.36	3.56049%
10	Cereja	2016	1063754.48	2.5111%
11	Morango	2016	974442.27	2.30027%
12	Uva	2016	798888.80	1.88586%

11) Abra o arquivo **Vendas por sabor EXEMPLO.sql**.

12) Esta consulta envolverá a tabela de produtos, para buscar os sabores, a tabela de itens de notas fiscais, para obter o faturamento (quantidade X preço), e a tabela de notas fiscais, onde tem a data da venda. Você irá juntar a tabela de produtos e a tabela de itens de notas fiscais pelo código de produto, e em seguida juntará com a tabela de notas fiscais:

```
SELECT * FROM [TABELA DE PRODUTOS] TP
SELECT * FROM [NOTAS FISCAIS] NF
SELECT * FROM [ITENS NOTAS FISCAIS] INF
```

13) Junte as três tabelas, selecionando os campos que são importantes para o relatório:

```
SELECT TP.SABOR, NF.DATA, (INF.QUANTIDADE * INF.PREÇO) AS FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CÓDIGO DO PRODUTO] = INF.[CÓDIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
```

	SABOR	DATA	FATURAMENTO
1	Cereja/Maçã	2015-01-01	1512,63
2	Morango/Limão	2015-01-01	234,273
3	Maçã	2015-01-01	804,737
4	Laranja	2015-01-01	528,528
5	Uva	2015-01-01	220,815
6	Laranja	2015-01-01	244,92
7	Morango/Limão	2015-01-01	756,882
8	Manga	2015-01-01	142,8015

Consulta executada com êxito. DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 213364 linhas

14) Agrupe esta consulta por sabor e ano:

```
SELECT TP.SABOR, YEAR(NF.DATA) AS ANO, SUM (INF.QUANTIDADE * INF.PREÇO) AS FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CÓDIGO DO PRODUTO] = INF.[CÓDIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
GROUP BY TP.SABOR, YEAR(NF.DATA)
```

	SABOR	ANO	FATURAMENTO
1	Açaí	2015	9345514,908
2	Cereja	2015	1028420,7
3	Cereja/Maçã	2015	3953306,06
4	Laranja	2015	3653928,152
5	Lima/Limão	2015	1386509,452
6	Maçã	2015	4135919,30700001
7	Manga	2015	5321977,0015
8	Maracujá	2015	1794187,536

Consulta executada com êxito. DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 48 linhas

15) Aplique o filtro para ter somente as vendas de 2016:

```
SELECT TP.SABOR, YEAR(NF.DATA) AS ANO, SUM (INF.QUANTIDADE * INF.PREÇO) AS FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CÓDIGO DO PRODUTO] = INF.[CÓDIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY TP.SABOR, YEAR(NF.DATA)
```

	SABOR	ANO	FATURAMENTO
6	Maçã	2016	4321369,73729999
7	Manga	2016	5725142,62582499
8	Maracujá	2016	1987886,74560001
9	Melância	2016	6128144,29499999
10	Morango	2016	974442,268800002
11	Morango/Li...	2016	1913604,25095
12	Uva	2016	798888,796650002

16) Guarde esta consulta e faça outra para obter a soma total de vendas no mesmo período. Para isso, aproveite a consulta anterior, mas tire o sabor do grupo:

```
SELECT YEAR(NF.DATA) AS ANO, SUM (INF.QUANTIDADE * INF.PREÇO) AS
FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CODIGO DO PRODUTO] = INF.[CODIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY YEAR(NF.DATA)
```

	Resultados	Mensagens
	ANO	FATURAMENTO
1	2016	42362118.1877254

17) Junte as duas consultas pelo campo que representa o ano:

```
SELECT AUX1.SABOR, AUX1.ANO, AUX1.FATURAMENTO, AUX2.TOTAL FROM
(SELECT TP.SABOR, YEAR(NF.DATA) AS ANO, SUM (INF.QUANTIDADE *
INF.PREÇO) AS FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CODIGO DO PRODUTO] = INF.[CODIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY TP.SABOR, YEAR(NF.DATA)) AUX1
INNER JOIN (SELECT YEAR(NF.DATA) AS ANO, SUM (INF.QUANTIDADE *
INF.PREÇO) AS TOTAL
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CODIGO DO PRODUTO] = INF.[CODIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY YEAR(NF.DATA)) AUX2
ON AUX1.ANO = AUX2.ANO
```

	SABOR	ANO	FATURAMENTO	TOTAL
1	Açai	2016	10024014.105	42362118.1877254
2	Cereja	2016	1063754.47709999	42362118.1877254
3	Cereja/Maçã	2016	4071635.9775	42362118.1877254
4	Laranja	2016	3844934.5452	42362118.1877254
5	Lima/Limão	2016	1508300.3628	42362118.1877254
6	Maçã	2016	4321369.73729999	42362118.1877254
7	Manga	2016	5725142.62582499	42362118.1877254
8	Maracujá	2016	1987886.74560001	42362118.1877254

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (54) | SUCOS\_VENDAS | 00:00:00 | 12 linhas

18) Para encerrar o relatório, calcule a coluna percentual e ordene a saída do maior para o menor faturamento. Aproveite e converta os dados de faturamento e percentual para ser representado com duas casas decimais:

```

SELECT AUX1.SABOR, AUX1.ANO, CONVERT(DECIMAL(15,2), AUX1.FATURAMENTO)
AS FATURAMENTO,
CONVERT(VARCHAR,
CONVERT(DECIMAL(15,2), (AUX1.FATURAMENTO/AUX2.TOTAL) * 100)) + ' %'
AS PERCENTUAL FROM
(SELECT TP.SABOR, YEAR(NF.DATA) AS ANO, SUM(INF.QUANTIDADE *
INF.PREÇO) AS FATURAMENTO
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CÓDIGO DO PRODUTO] = INF.[CÓDIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY TP.SABOR, YEAR(NF.DATA)) AUX1
INNER JOIN (SELECT YEAR(NF.DATA) AS ANO, SUM(INF.QUANTIDADE *
INF.PREÇO) AS TOTAL
FROM [ITENS NOTAS FISCAIS] INF INNER JOIN [TABELA DE PRODUTOS] TP
ON TP.[CÓDIGO DO PRODUTO] = INF.[CÓDIGO DO PRODUTO]
INNER JOIN [NOTAS FISCAIS] NF
ON NF.NUMERO = INF.NUMERO
WHERE YEAR(NF.DATA) = 2016
GROUP BY YEAR(NF.DATA)) AUX2
ON AUX1.ANO = AUX2.ANO
ORDER BY AUX1.FATURAMENTO DESC

```

	SABOR	ANO	FATURAMENTO	PERCENTUAL
1	Açaí	2016	10024014.10	23.66 %
2	Melancia	2016	6128144.29	14.47 %
3	Manga	2016	5725142.63	13.51 %
4	Maça	2016	4321369.74	10.20 %
5	Cereja/Maça	2016	4071635.98	9.61 %
6	Laranja	2016	3844934.55	9.08 %
7	Maracujá	2016	1987886.75	4.69 %
8	Morango/Limão	2016	1913604.25	4.52 %
9	Lima/Limão	2016	1508300.36	3.56 %
10	Cereja	2016	1062554.40	2.51 %
11	Morango	2016	974442.27	2.30 %
12	Uva	2016	798888.80	1.89 %