



PROGRAMA DE CAPACITAÇÃO

DATA SCIENCE + DATA INTELLIGENCE

NC 03 – Introdução ao uso de SQL para processos de extração e transformação de dados.

Prof. Dr.: José Carlos Conti



Ementa



- SQL (Linguagem Estruturada de Consulta) tem sido, há muito tempo, bastante útil para extração e consulta a bancos de dados relacionais, para capturar matéria prima para análises, relatórios e indicadores.
- É fundamental que os profissionais dominem SQL para acelerar o processo de extração, tratamento e análise de dados.



Conteúdos a serem trabalhados

1

- Conceituação e importância da linguagem SQL

3

- Uso da cláusula SELECT para consulta ao banco de dados

5

- Trabalhando com operações lógicas.

7

- Cruzamento de dados de diferentes tabelas (JOIN).

2

- Instalação do SQL Server Express.

4

- Uso de filtros (cláusula WHERE) nas consultas.

6

- Agrupando dados em uma consulta por meio de GROUP BY.

8

- Práticas usando SQL Server Express e Python

Programação das aulas (15h – 5 sessões de 3h cada)



Softwares a serem trabalhados

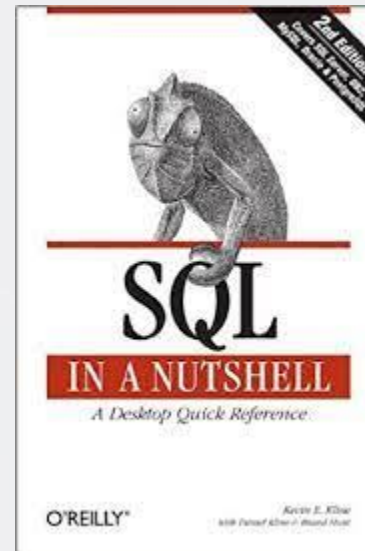
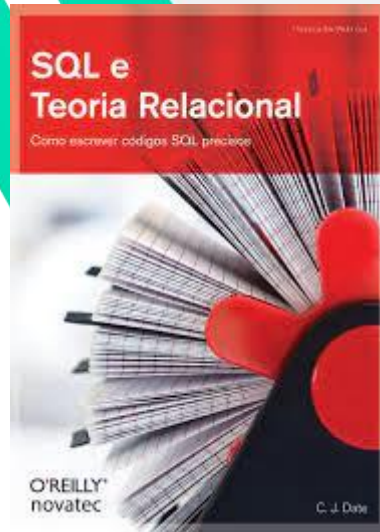
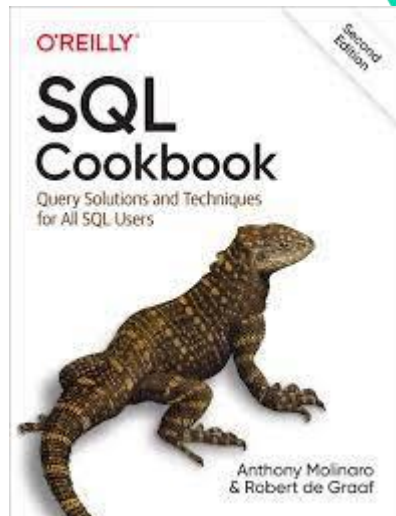
- Jupyter Notebooks (Python 3)
- SQL Server Express



Bibliografia básica



- SQL Cookbook – Anthony Molinaro, Robert de Graaf
- SQL e Teoria Relacional – C.J. Date
- SQL in a Nutshell – Kevin Kline, Daniel Kline, Brand Hunt
- Getting Started with SQL – Thomas Nield



Aula 2



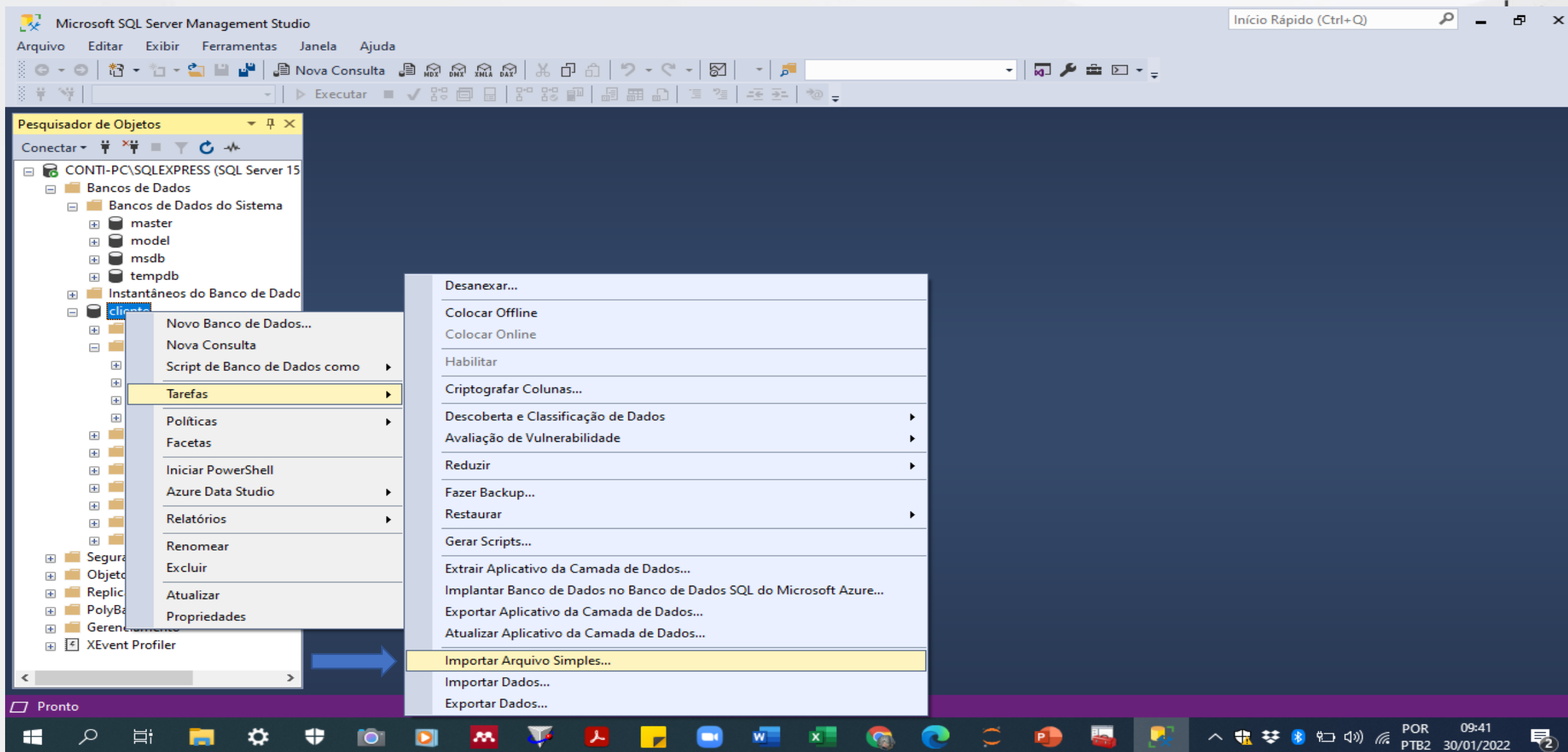
- Uso da cláusula SELECT para consulta ao banco de dados
- Uso de filtros (cláusula WHERE) nas consultas

Aula 2

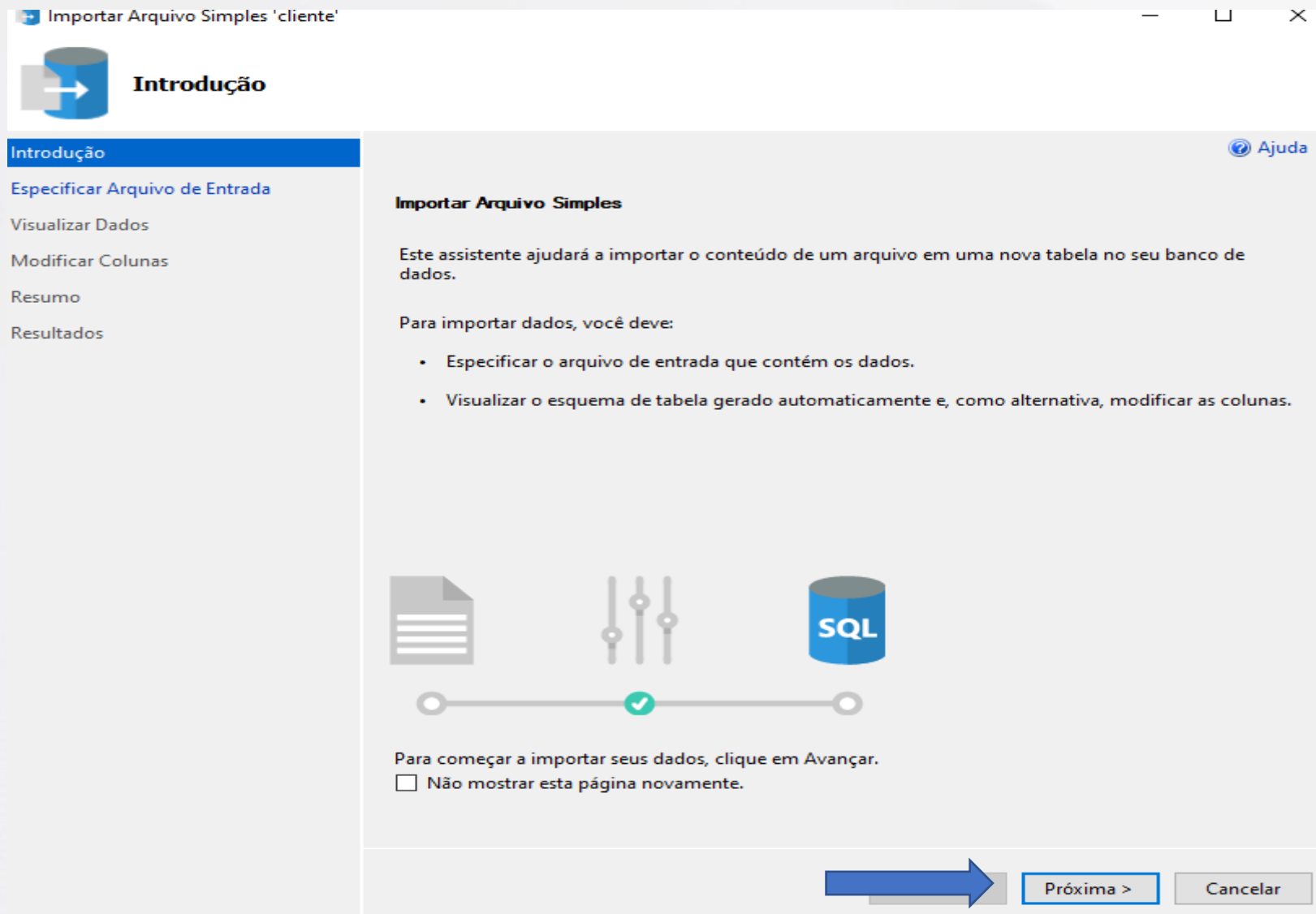


- Apesar das dificuldades, foi possível instalar e configurar o SQL Server Express ?

Revisão: Criando tabela importando arquivo



Revisão: Criando tabela importando arquivo



Revisão: Criando tabela importando arquivo

Microsoft SQL Server Management Studio

Arquivo Editar Exibir Ferramentas Jan

Pesquisador de Objetos

Conectar

CONTI-PC\SQLEXPRESS (SQL Server 15)

- Bancos de Dados
 - Bancos de Dados do Sistema
 - master
 - model
 - msdb
 - tempdb
 - Instantâneos do Banco de Dado
 - cliente
 - Diagramas de Banco de Dado
 - Tabelas
 - Tabelas do Sistema
 - FileTables
 - Tabelas Externas
 - Tabelas de Grafo
 - Exibições
 - Recursos Externos
 - Sinônimos
 - Programação
 - Service Broker
 - Armazenamento
 - Segurança
 - Segurança
 - Objetos de Servidor
 - Replicação
 - PolyBase
 - Gerenciamento
 - XEvent Profiler

Importar Arquivo Simples 'cliente'

Especificar Arquivo de Entrada

Introdução

Especificar Arquivo de Entrada

Visualizar Dados

Modificar Colunas

Resumo

Resultados

Especificar Arquivo de Entrada

Esta operação criará uma tabela do seu arquivo de entrada.

Local do arquivo a ser importado

C:\Aulas_BI\clientes.txt

Procurar...

Novo nome da tabela:

t_clientes

Esquema de tabela:

dbo

Próxima >


Cancelar

Pronto

POR PTB2 10:39 30/01/2022

Revisão: Criando tabela importando arquivo

Importar Arquivo Simples 'cliente'

 **Visualizar Dados**

Introdução

Especificar Arquivo de Entrada

Visualizar Dados

Modificar Colunas

Resumo

Resultados


Ajuda

Visualizar Dados

Esta operação analisou a estrutura do arquivo de entrada para gerar a visualização abaixo para as primeiras 50 linhas.

	CODIGO_CLIENTE	NOME_CLIENTE	DATA_NASCIMEN	EMAIL	TELEFONE_CELUL	UF
1		Jose A1	30/09/1950	jose a1@email....	11911111111	SP
2		Jose A2	01/10/1950	jose a2@email....	11911111112	SP
3		Jose A3	02/10/1950	jose a3@email....	11911111113	SP
4		Jose A4	03/10/1950	jose a4@email....	11911111114	SP
5		Jose A5	04/10/1950	jose a5@email....	11911111115	SP
6		Jose A6	05/10/1950	jose a6@email....	11911111116	SP
7		Jose A7	06/10/1950	jose a7@email....	11911111117	SP
8		Jose A8	07/10/1950	jose a8@email....	11911111118	SP
9		Jose A9	08/10/1950	jose a9@email....	11911111119	SP
10		Jose A10	09/10/1950	jose a10@email...	11911111120	SP
11		Jose A11	10/10/1950	jose a11@email...	11911111121	SP
12		Jose A12	11/10/1950	jose a12@email...	11911111122	SP
13		Jose A13	12/10/1950	jose a13@email...	11911111123	SP
14		Jose A14	13/10/1950	jose a14@email...	11911111124	SP
15		Jose A15	14/10/1950	jose a15@email...	11911111125	SP
16		Jose A16	15/10/1950	jose a16@email...	11911111126	SP
17		Jose A17	16/10/1950	jose a17@email...	11911111127	SP
18		Jose A18	17/10/1950	jose a18@email...	11911111128	SP

☒ Usar a Detecção de Tipos de Dados Avançados – pode oferecer um melhor ajuste de tipo. No entanto, as células com valores anormais podem ser removidas.



Próxima >

Cancelar




POR
PTB2

10:41
30/01/2022



Revisão: Criando tabela importando arquivo

Importar Arquivo Simples 'cliente'

**Modificar Colunas**

Introdução

Especificar Arquivo de Entrada

Visualizar Dados

Modificar Colunas

Resumo

Resultados

Ajuda

Modificar Colunas

Esta operação gerou o seguinte esquema de tabela. Verifique se o esquema é preciso e, se não for, faça as alterações.

Nome da Coluna	Tipo de Dados	Chave Primária	<input type="checkbox"/> Permitir Nulos
CODIGO_CLIENTE	smallint	<input type="checkbox"/>	<input type="checkbox"/>
NOME_CLIENTE	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
DATA_NASCIMENTO	date	<input type="checkbox"/>	<input type="checkbox"/>
EMAIL	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
TELEFONE_CELULAR	bigint	<input type="checkbox"/>	<input type="checkbox"/>
UF	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>

Granularidade de linha do relatório de erros (impacto no desempenho com intervalos menores)

Sem Intervalo

Próxima >


Cancelar

Windows taskbar icons

POR PTB2 10:44 30/01/2022

Revisão: Criando tabela importando arquivo

Importar Arquivo Simples 'cliente'

**Resumo**


[Introdução](#)
[Especificar Arquivo de Entrada](#)
[Visualizar Dados](#)
[Modificar Colunas](#)
[Resumo](#)
[Resultados](#)

Ajuda

Resumo
Para concluir a operação usando as entradas especificadas, clique em Concluir.


Importar Informações

Nome: CONTI-PC\SQLEXPRESS
Nome do Banco de Dados: cliente
Nome da Tabela: dbo.t_clientes
Arquivo a ser importado: C:\Aulas_BI\clientes.txt




Concluir

Cancelar





POR
PTB2

10:45
30/01/2022



Revisão: Criando tabela importando arquivo

 Importar Arquivo Simples 'cliente'

 **Resultados**

Introdução


Especificar Arquivo de Entrada


Visualizar Dados

Modificar Colunas


Resumo

Resultados


 Ajuda

 **Operação Concluída**

Resumo:

	Nome	Resultado
	Inserir Dados	Êxito

< Voltar



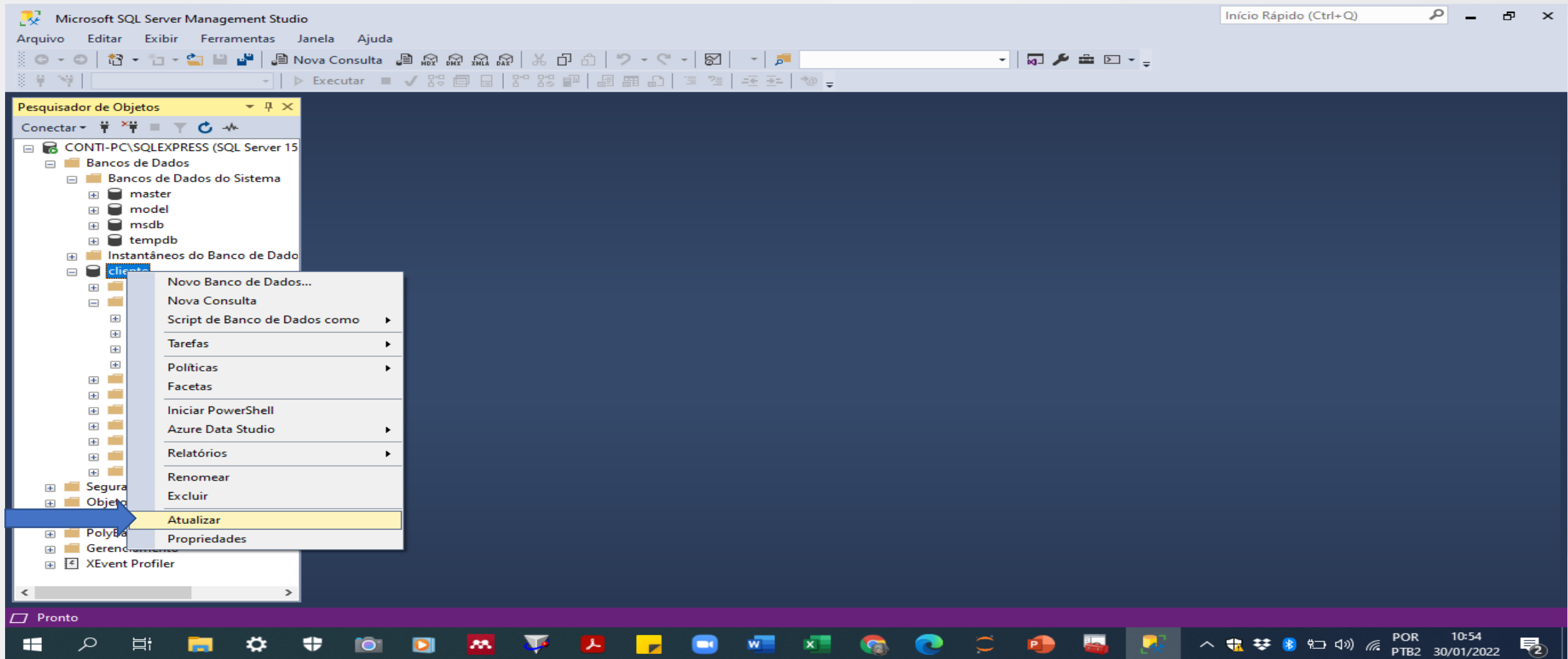
Fechar



POR PTB2 10:46 30/01/2022 2

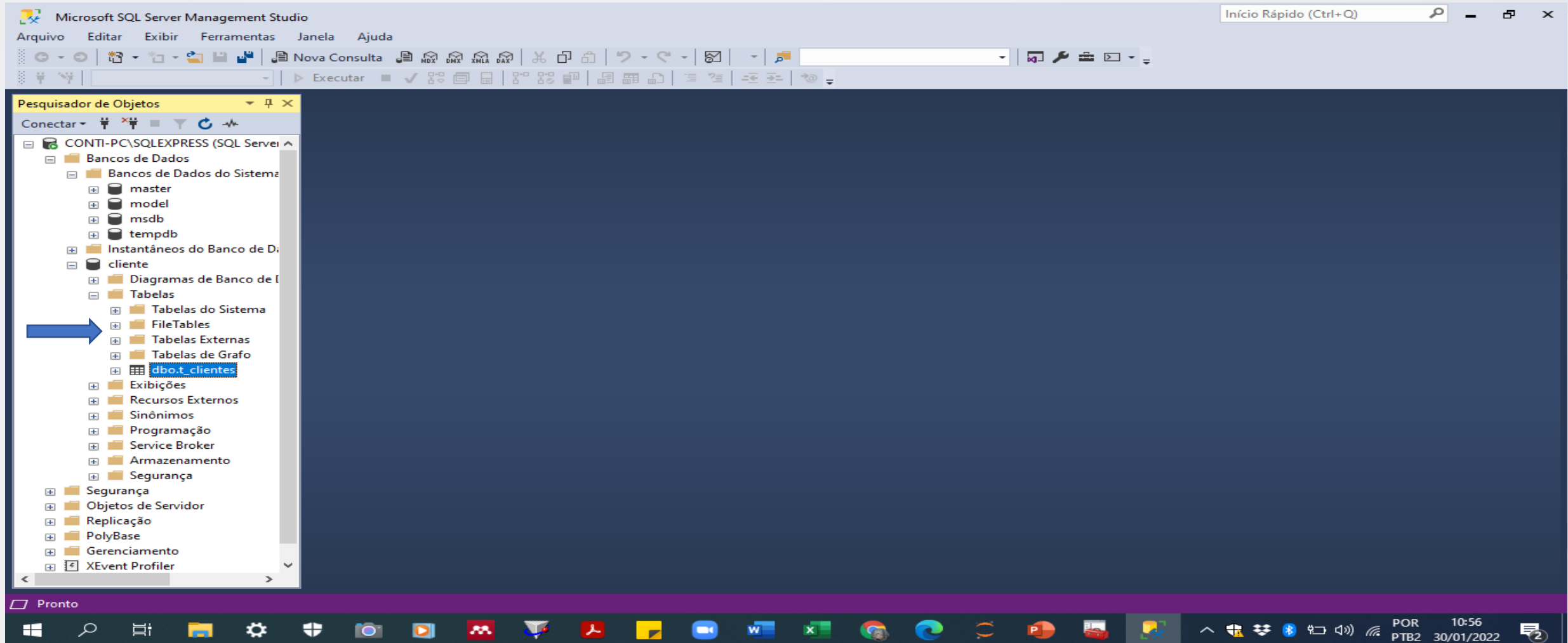
Revisão: Criando tabela importando arquivo

- A seguir atualizar o banco de dados cliente para visualizar a tabela t_clientes criada.



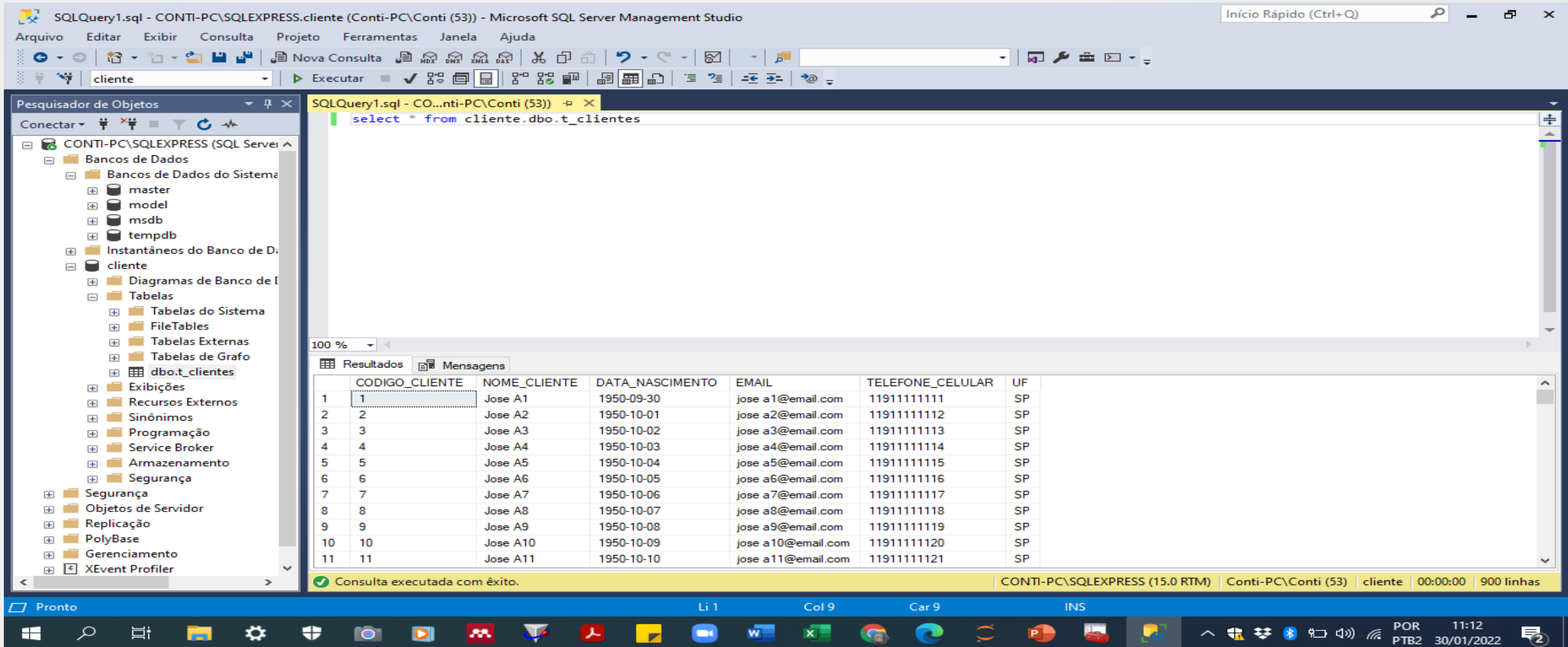
Revisão: Criando tabela importando arquivo

- Note que a tabela t_clientes é apresentada no banco de dados cliente.



Revisão: Criando tabela importando arquivo

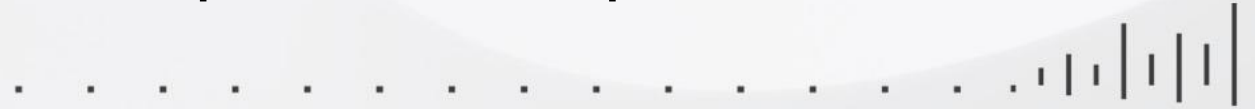
- Efetue uma consulta simples na tabela t_clientes para visualizar os dados.



The screenshot displays the Microsoft SQL Server Management Studio interface. The title bar indicates the connection to 'CONTI-PC\SQLEXPRESS.cliente (Conti-PC\Conti (53))'. The 'Pesquisador de Objetos' (Object Explorer) on the left shows the database structure, with 'cliente' expanded to show 'Tabelas' (Tables), and 'dbo.t_clientes' selected. The 'SQLQuery1.sql' editor in the center contains the query: `select * from cliente.dbo.t_clientes`. The 'Resultados' (Results) pane at the bottom shows the query output as a table with 11 rows and 7 columns. The status bar at the bottom indicates 'Consulta executada com êxito.' (Query executed successfully).

	CODIGO_CLIENTE	NOME_CLIENTE	DATA_NASCIMENTO	EMAIL	TELEFONE_CELULAR	UF
1	1	Jose A1	1950-09-30	jose a1@email.com	11911111111	SP
2	2	Jose A2	1950-10-01	jose a2@email.com	11911111112	SP
3	3	Jose A3	1950-10-02	jose a3@email.com	11911111113	SP
4	4	Jose A4	1950-10-03	jose a4@email.com	11911111114	SP
5	5	Jose A5	1950-10-04	jose a5@email.com	11911111115	SP
6	6	Jose A6	1950-10-05	jose a6@email.com	11911111116	SP
7	7	Jose A7	1950-10-06	jose a7@email.com	11911111117	SP
8	8	Jose A8	1950-10-07	jose a8@email.com	11911111118	SP
9	9	Jose A9	1950-10-08	jose a9@email.com	11911111119	SP
10	10	Jose A10	1950-10-09	jose a10@email.com	11911111120	SP
11	11	Jose A11	1950-10-10	jose a11@email.com	11911111121	SP

Revisão: Criando tabela importando arquivo



- Atividade
 - Considerando o exemplo usado para importar o arquivo clientes.txt para criar a tabela t_clientes, aplicar o mesmo procedimento para importar o arquivo vendas.txt para criar a tabela t_vendas.

Revisão: Criando tabela importando arquivo

- Scripts para criação das tabelas

```
USE [cliente]  
GO
```

```
/****** Object: Table [dbo].[t_clientes]   Script Date: 18/04/2022 22:13:30  
*****/
```

```
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[t_clientes](  
[CODIGO_CLIENTE] [smallint] NOT NULL,  
[NOME_CLIENTE] [nvarchar](50) NOT NULL,  
[DATA_NASCIMENTO] [date] NOT NULL,  
[EMAIL] [nvarchar](50) NOT NULL,  
[TELEFONE_CELULAR] [bigint] NOT NULL,  
[UF] [nvarchar](50) NOT NULL  
) ON [PRIMARY]  
GO
```

Revisão: Criando tabela importando arquivo

- Scripts para criação das tabelas

```
USE [cliente]  
GO
```

```
/***** Object: Table [dbo].[t_vendas]    Script Date: 18/04/2022  
22:13:54 *****/
```

```
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[t_vendas](  
[SKU] [nvarchar](50) NOT NULL,  
[TAMANHO_PEDIDO] [tinyint] NOT NULL,  
[LOJA] [nvarchar](50) NOT NULL,  
[DATA_VENDA] [date] NOT NULL,  
[MARCA] [nvarchar](50) NOT NULL,  
[CODIGO_CLIENTE] [smallint] NOT NULL,  
[PRECO_UNITARIO] [smallint] NOT NULL,  
[CUSTO_UNITARIO] [smallint] NOT NULL  
) ON [PRIMARY]  
GO
```

Uso da cláusula SELECT para consulta ao banco de dados

Go to www.menti.com and use the code 73 37 33 0

P03 - Em SQL, para selecionar na tabela vendas todas as vendas das lojas São Paulo e Rio de Janeiro, a sintaxe correta para o comando é ?

Mentimeter

0	0	0
SELECT * FROM VENDAS WHERE LOJA in ('São Paulo', 'Rio de Janeiro')	SELECT * IN VENDAS WITH LOJA = 'Rio' or LOJA = 'S Paulo'	SELECT LOJA FROM VENDAS

Uso da cláusula SELECT para consulta ao banco de dados:

- Recupera linhas do banco de dados e permite a seleção de uma ou várias linhas ou colunas de uma ou várias tabelas no SQL Server.
- A sintaxe completa da instrução SELECT é complexa, mas as cláusulas principais podem ser assim resumidas:

```
SELECT select_list [ INTO new_table ]  
  
[ FROM table_source ] [ WHERE search_condition ]  
  
[ GROUP BY group_by_expression ]  
  
[ HAVING search_condition ]  
  
[ ORDER BY order_expression [ ASC | DESC ]]
```

Uso da cláusula SELECT para consulta ao banco de dados

- Considere a estrutura das tabelas que trabalharemos nas próximas aulas:
- tabela t_vendas.

Nome do campo	Descrição do campo
SKU	Stock Keeping Unit (Unidade de Manutenção de Estoque). Código único de identificação do produto.
TAMANHO_PEDIDO	Tamanho do pedido, identifica a quantidade de pedidos efetuados.
LOJA	Nome da Loja, identificado por Filial / Cidade.
DATA_VENDA	Data da venda do produto.
MARCA	Marca do produto.
CODIGO_CLIENTE	Código único do cliente.
PRECO_UNITARIO	Preço unitário do produto.
CUSTO_UNITARIO	Custo de produção unitário do produto.

Uso da cláusula SELECT para consulta ao banco de dados

- Considere a estrutura das tabelas que trabalharemos nas próximas aulas:
- tabela t_clientes.

Nome do campo	Descrição do campo
CODIGO_CLIENTE	Código único do cliente.
NOME_CLIENTE	Nome do cliente.
DATA_NASCIMENTO	Data de nascimento do cliente.
EMAIL	Endereço do email do cliente.
TELEFONE_CELULAR	Número do telefone celular do cliente.
UF	Unidade Federativa do Endereço do cliente.

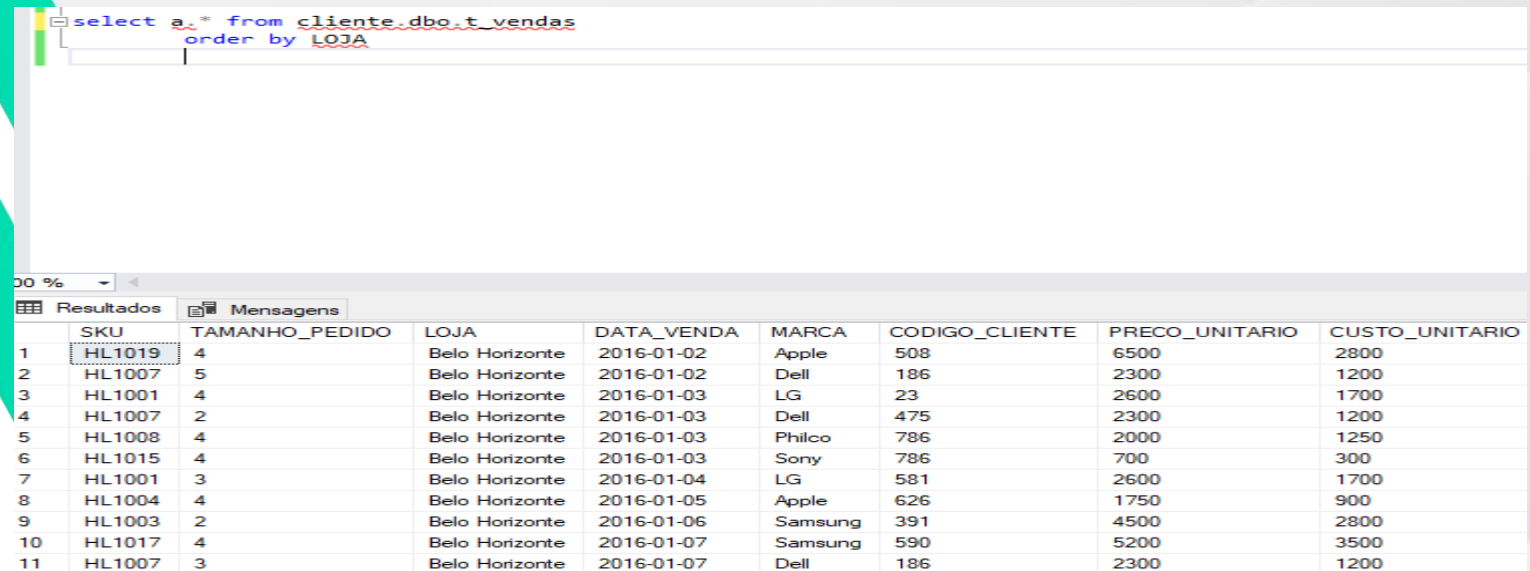
Uso da cláusula SELECT para consulta ao banco de dados

- Usando Select para recuperar linhas e colunas
 - Esta seção mostra três exemplos de código. Este primeiro exemplo de código retorna todas as linhas (nenhuma cláusula WHERE foi especificada) e todas as colunas (usando o *) da tabela t_vendas.

```
select * from cliente.dbo.t_vendas  
order by LOJA
```

Em que:

- cliente = nome do banco de dados
- dbo = nome do schema (padrão do SQL Server)
- t_vendas = nome da tabela

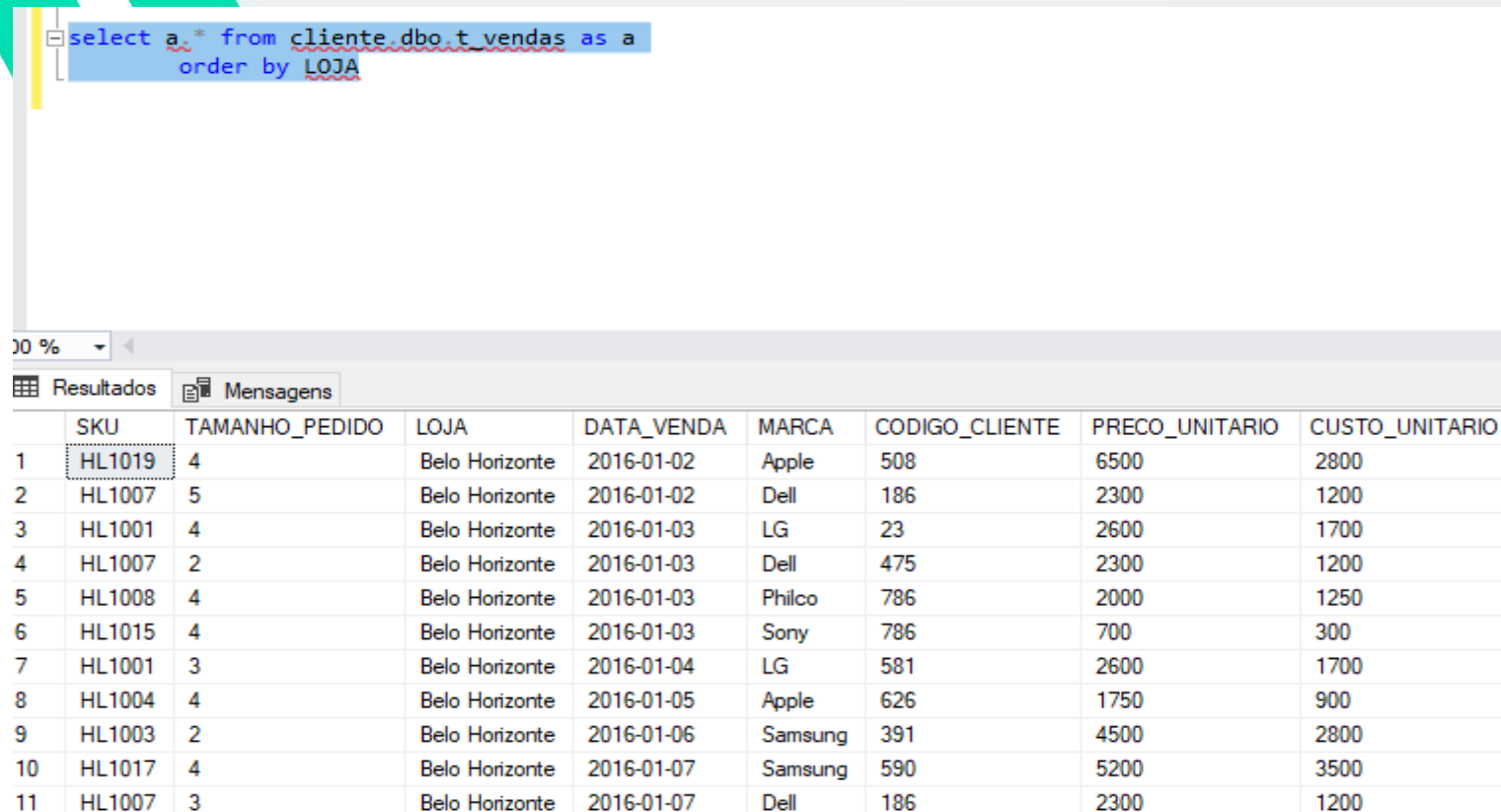


	SKU	TAMANHO_PEDIDO	LOJA	DATA_VENDA	MARCA	CODIGO_CLIENTE	PRECO_UNITARIO	CUSTO_UNITARIO
1	HL1019	4	Belo Horizonte	2016-01-02	Apple	508	6500	2800
2	HL1007	5	Belo Horizonte	2016-01-02	Dell	186	2300	1200
3	HL1001	4	Belo Horizonte	2016-01-03	LG	23	2600	1700
4	HL1007	2	Belo Horizonte	2016-01-03	Dell	475	2300	1200
5	HL1008	4	Belo Horizonte	2016-01-03	Philco	786	2000	1250
6	HL1015	4	Belo Horizonte	2016-01-03	Sony	786	700	300
7	HL1001	3	Belo Horizonte	2016-01-04	LG	581	2600	1700
8	HL1004	4	Belo Horizonte	2016-01-05	Apple	626	1750	900
9	HL1003	2	Belo Horizonte	2016-01-06	Samsung	391	4500	2800
10	HL1017	4	Belo Horizonte	2016-01-07	Samsung	590	5200	3500
11	HL1007	3	Belo Horizonte	2016-01-07	Dell	186	2300	1200

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Este próximo exemplo aplica-se a definição de alias da tabela t_vendas para obter o mesmo resultado.

```
select a.* from cliente.dbo.t_vendas as a  
order by LOJA
```



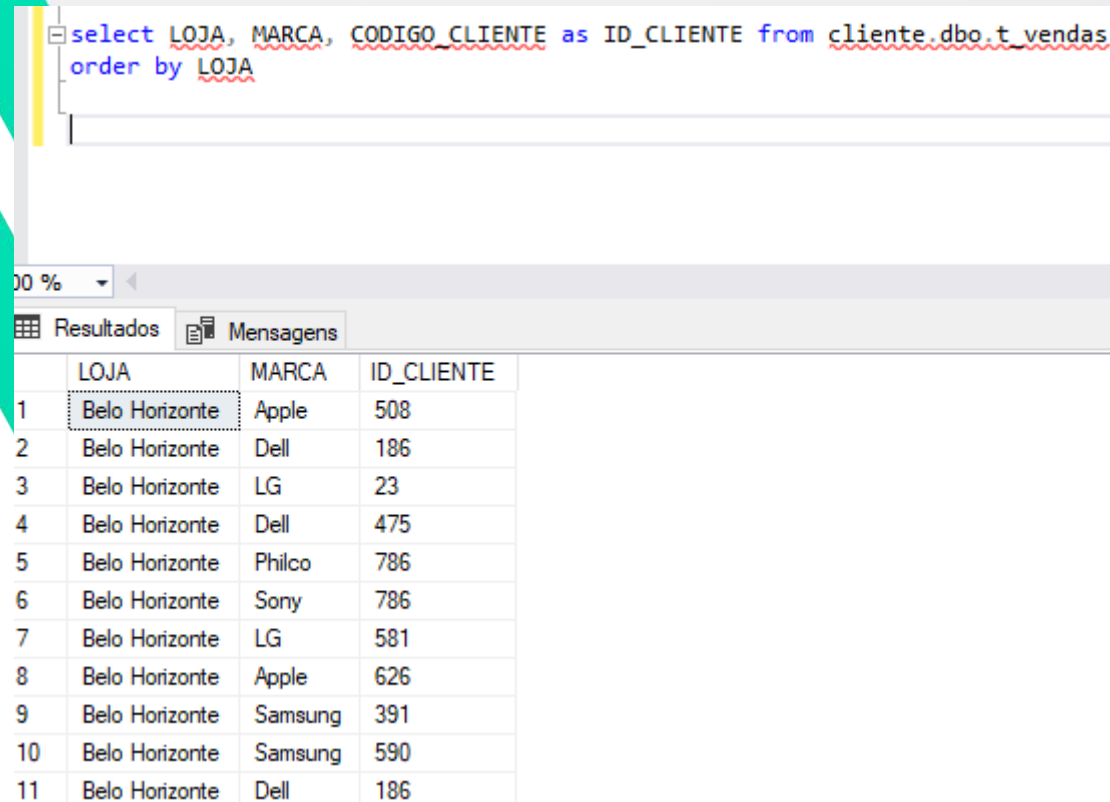
```
select a.* from cliente.dbo.t_vendas as a  
order by LOJA
```

	SKU	TAMANHO_PEDIDO	LOJA	DATA_VENDA	MARCA	CODIGO_CLIENTE	PRECO_UNITARIO	CUSTO_UNITARIO
1	HL1019	4	Belo Horizonte	2016-01-02	Apple	508	6500	2800
2	HL1007	5	Belo Horizonte	2016-01-02	Dell	186	2300	1200
3	HL1001	4	Belo Horizonte	2016-01-03	LG	23	2600	1700
4	HL1007	2	Belo Horizonte	2016-01-03	Dell	475	2300	1200
5	HL1008	4	Belo Horizonte	2016-01-03	Philco	786	2000	1250
6	HL1015	4	Belo Horizonte	2016-01-03	Sony	786	700	300
7	HL1001	3	Belo Horizonte	2016-01-04	LG	581	2600	1700
8	HL1004	4	Belo Horizonte	2016-01-05	Apple	626	1750	900
9	HL1003	2	Belo Horizonte	2016-01-06	Samsung	391	4500	2800
10	HL1017	4	Belo Horizonte	2016-01-07	Samsung	590	5200	3500
11	HL1007	3	Belo Horizonte	2016-01-07	Dell	186	2300	1200

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
- O próximo exemplo retorna todas as linhas (nenhuma cláusula WHERE foi especificada) e um subconjunto das colunas (LOJA, MARCA, CODIGO_CLIENTE) da tabela t_vendas no banco de dados cliente. O terceiro título de coluna é renomeado como ID_CLIENTE

```
select LOJA, MARCA, CODIGO_CLIENTE as ID_CLIENTE from cliente.dbo.t_vendas  
order by LOJA
```



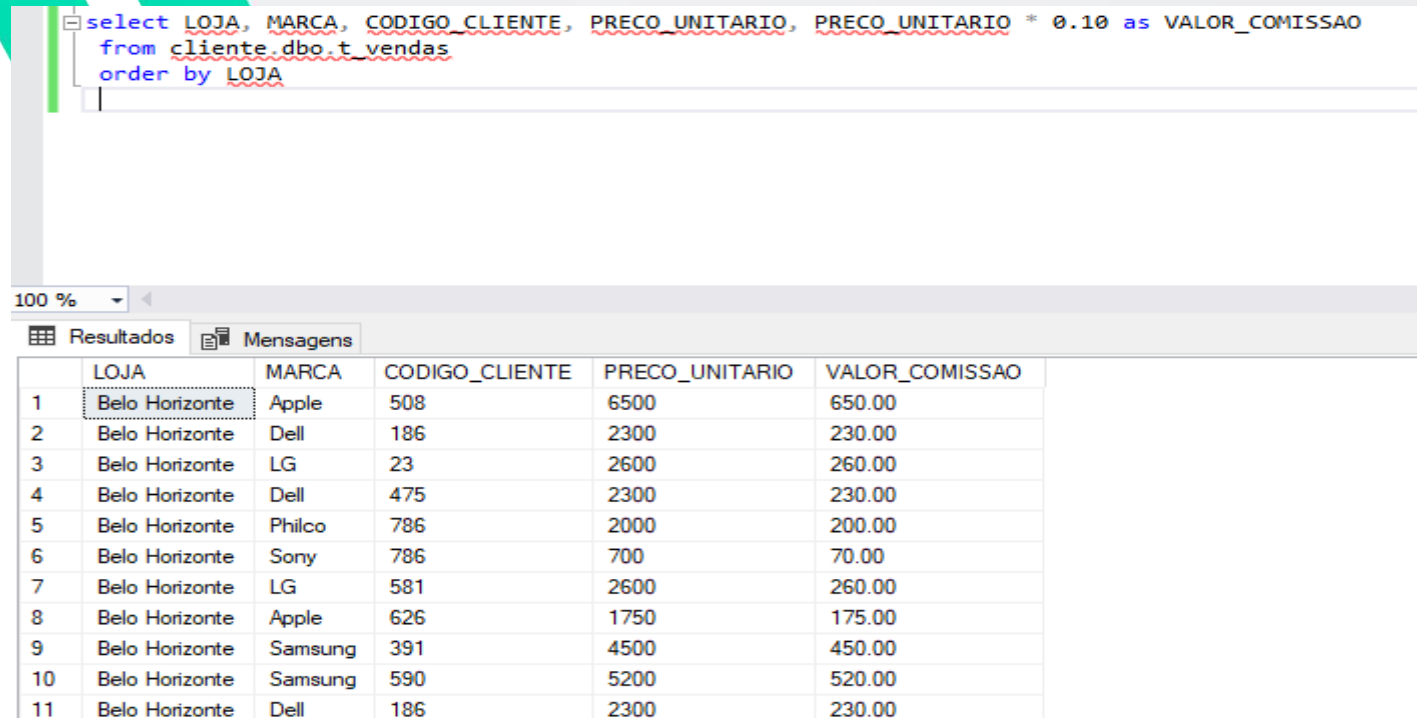
```
select LOJA, MARCA, CODIGO_CLIENTE as ID_CLIENTE from cliente.dbo.t_vendas  
order by LOJA
```

	LOJA	MARCA	ID_CLIENTE
1	Belo Horizonte	Apple	508
2	Belo Horizonte	Dell	186
3	Belo Horizonte	LG	23
4	Belo Horizonte	Dell	475
5	Belo Horizonte	Philco	786
6	Belo Horizonte	Sony	786
7	Belo Horizonte	LG	581
8	Belo Horizonte	Apple	626
9	Belo Horizonte	Samsung	391
10	Belo Horizonte	Samsung	590
11	Belo Horizonte	Dell	186

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando SELECT com títulos de coluna e cálculos
 - O exemplo a seguir retorna todas as linhas da tabela t_vendas e calcula o valor_comissão para cada preço unitário da venda.

```
select LOJA, MARCA, CODIGO_CLIENTE, PRECO_UNITARIO,  
       PRECO_UNITARIO * 0.10 as VALOR_COMISSAO  
from cliente.dbo.t_vendas  
order by LOJA
```



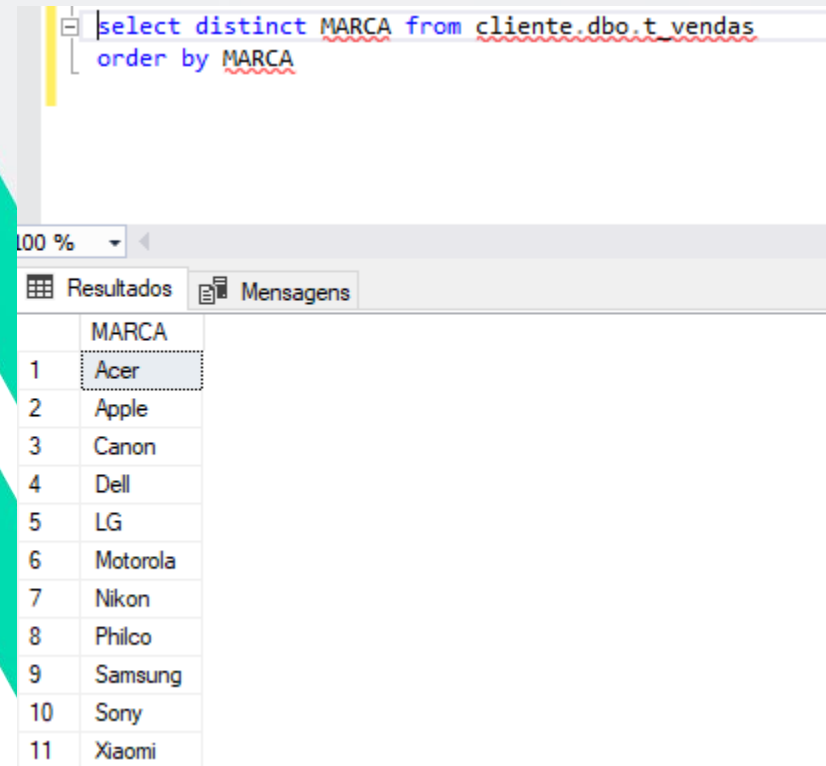
```
select LOJA, MARCA, CODIGO_CLIENTE, PRECO_UNITARIO, PRECO_UNITARIO * 0.10 as VALOR_COMISSAO  
from cliente.dbo.t_vendas  
order by LOJA
```

	LOJA	MARCA	CODIGO_CLIENTE	PRECO_UNITARIO	VALOR_COMISSAO
1	Belo Horizonte	Apple	508	6500	650.00
2	Belo Horizonte	Dell	186	2300	230.00
3	Belo Horizonte	LG	23	2600	260.00
4	Belo Horizonte	Dell	475	2300	230.00
5	Belo Horizonte	Philco	786	2000	200.00
6	Belo Horizonte	Sony	786	700	70.00
7	Belo Horizonte	LG	581	2600	260.00
8	Belo Horizonte	Apple	626	1750	175.00
9	Belo Horizonte	Samsung	391	4500	450.00
10	Belo Horizonte	Samsung	590	5200	520.00
11	Belo Horizonte	Dell	186	2300	230.00

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando DISTINCT com SELECT
 - O exemplo a seguir usa DISTINCT para gerar uma lista de todas as MARCAS de produtos na tabela t_vendas

```
select distinct MARCA from cliente.dbo.t_vendas  
order by MARCA
```

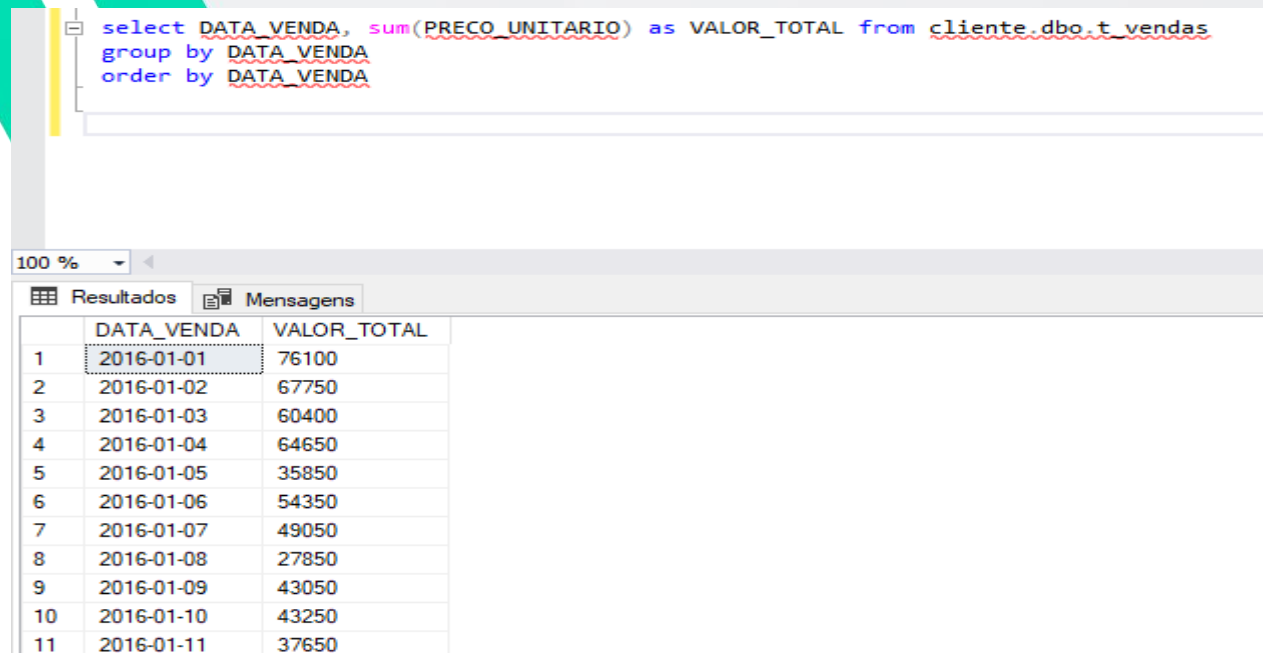


	MARCA
1	Acer
2	Apple
3	Canon
4	Dell
5	LG
6	Motorola
7	Nikon
8	Philco
9	Samsung
10	Sony
11	Xiaomi

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando o GROUP BY
 - O exemplo a seguir localiza o valor total da venda por data.

```
select DATA_VENDA, sum(PRECO_UNITARIO) as VALOR_TOTAL from  
cliente.dbo.t_vendas  
group by DATA_VENDA  
order by DATA_VENDA
```



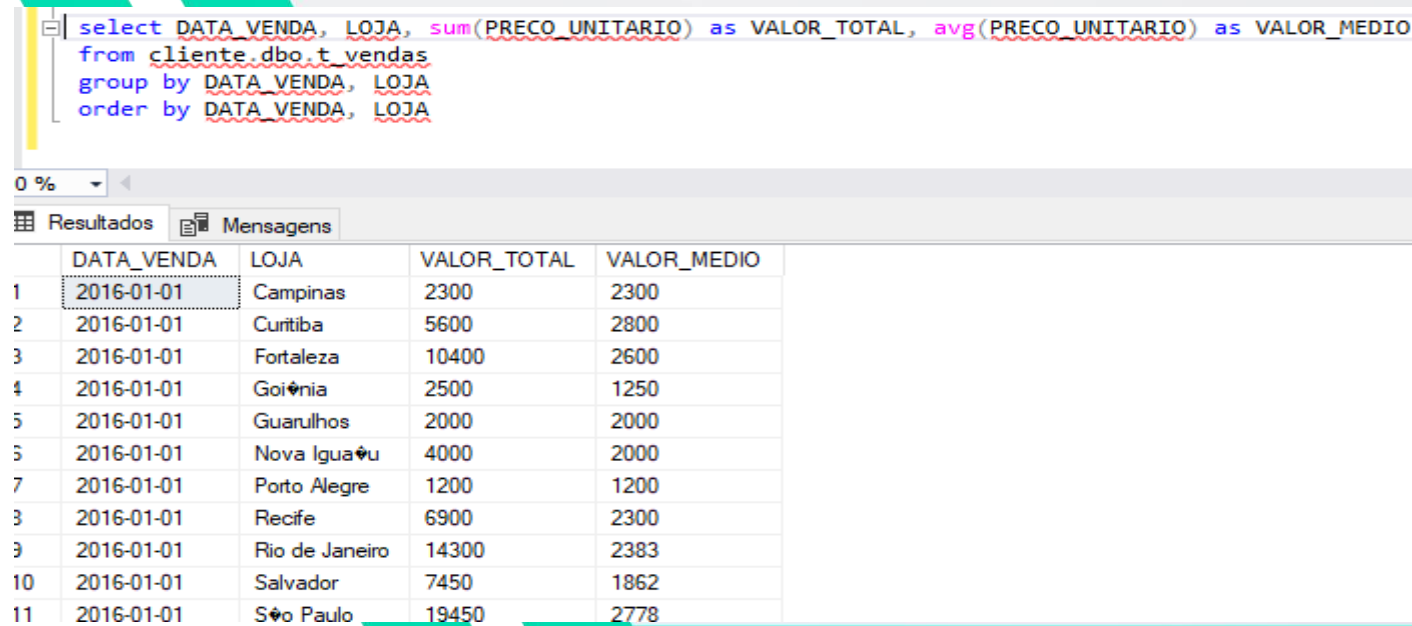
```
select DATA_VENDA, sum(PRECO_UNITARIO) as VALOR_TOTAL from cliente.dbo.t_vendas  
group by DATA_VENDA  
order by DATA_VENDA
```

	DATA_VENDA	VALOR_TOTAL
1	2016-01-01	76100
2	2016-01-02	67750
3	2016-01-03	60400
4	2016-01-04	64650
5	2016-01-05	35850
6	2016-01-06	54350
7	2016-01-07	49050
8	2016-01-08	27850
9	2016-01-09	43050
10	2016-01-10	43250
11	2016-01-11	37650

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando GROUP BY com vários grupos
 - O exemplo a seguir apresenta a soma e o valor médio das vendas para cada data, agrupados por data da venda e loja.

```
select DATA_VENDA, LOJA, sum(PRECO_UNITARIO) as  
VALOR_TOTAL, avg(PRECO_UNITARIO) as VALOR_MEDIO  
from cliente.dbo.t_vendas  
group by DATA_VENDA, LOJA  
order by DATA_VENDA, LOJA
```



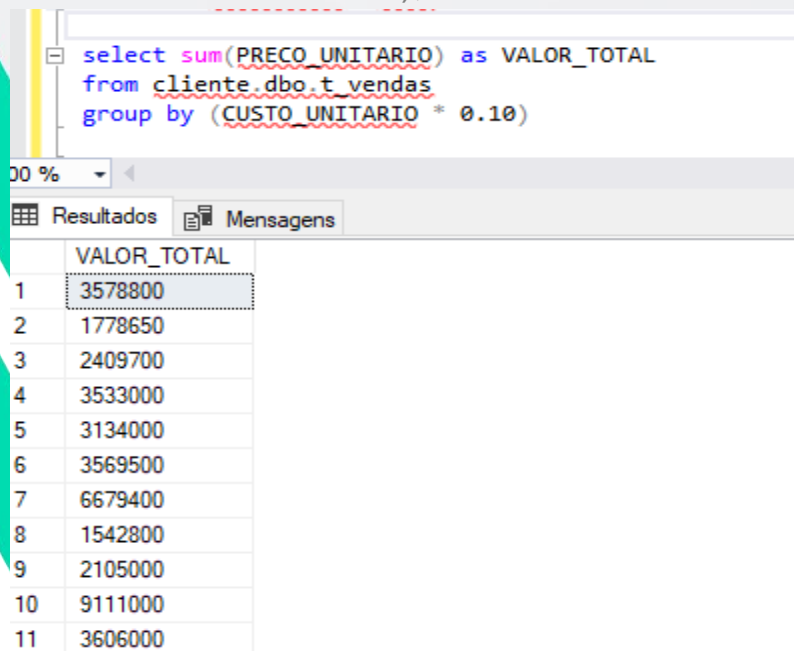
The screenshot shows a SQL query in a text editor with syntax highlighting. Below the editor, a table displays the results of the query. The table has five columns: DATA_VENDA, LOJA, VALOR_TOTAL, and VALOR_MEDIO. The results are grouped by date and store.

	DATA_VENDA	LOJA	VALOR_TOTAL	VALOR_MEDIO
1	2016-01-01	Campinas	2300	2300
2	2016-01-01	Curitiba	5600	2800
3	2016-01-01	Fortaleza	10400	2600
4	2016-01-01	Goiânia	2500	1250
5	2016-01-01	Guarulhos	2000	2000
5	2016-01-01	Nova Iguaçu	4000	2000
7	2016-01-01	Porto Alegre	1200	1200
3	2016-01-01	Recife	6900	2300
3	2016-01-01	Rio de Janeiro	14300	2383
10	2016-01-01	Salvador	7450	1862
11	2016-01-01	São Paulo	19450	2778

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando GROUP BY com uma expressão
 - O exemplo a seguir agrupa por uma expressão.

```
select sum(PRECO_UNITARIO) as VALOR_TOTAL  
from cliente.dbo.t_vendas  
group by (CUSTO_UNITARIO * 0.10);
```



	VALOR_TOTAL
1	3578800
2	1778650
3	2409700
4	3533000
5	3134000
6	3569500
7	6679400
8	1542800
9	2105000
10	9111000
11	3606000

Uso da cláusula SELECT para consulta ao banco de dados

- Uso da cláusula SELECT para consulta ao banco de dados
 - Usando a cláusula HAVING
 - Essa consulta usa a cláusula HAVING para restringir os resultados.

```
select LOJA, DATA_VENDA, sum(PRECO_UNITARIO) as VALOR_TOTAL
from cliente.dbo.t_vendas
group by LOJA, DATA_VENDA
HAVING LOJA <> 'Rio de Janeiro'
order by DATA_VENDA
```

133 %

Resultados Mensagens

	LOJA	DATA_VENDA	VALOR_TOTAL
1	Recife	2016-01-01	6900
2	Porto Alegre	2016-01-01	1200
3	São Paulo	2016-01-01	19450
4	Guarulhos	2016-01-01	2000
5	Curitiba	2016-01-01	5600
6	Campinas	2016-01-01	2300
7	Nova Iguaçu	2016-01-01	4000
8	Goiânia	2016-01-01	2500
9	Salvador	2016-01-01	7450
10	Fortaleza	2016-01-01	10400
11	Curitiba	2016-01-02	5650

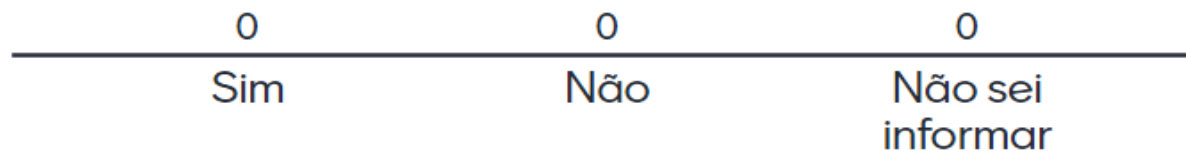
Uso de filtros (cláusula WHERE) nas consultas



Go to www.menti.com and use the code 6860 1603

P04: Em uma consulta SQL, é possível definir na cláusula WHERE um campo que não esteja no Select ?

Mentimeter



Uso de filtros (cláusula WHERE) nas consultas



- Uso de filtros (cláusula WHERE) nas consultas
 - **Argumentos**
 - `< search_condition >` define a condição que deverá ser atendida para as linhas que retornarão na consulta.
 - Não há nenhum limite para o número de predicados que podem ser incluídos em um critério de pesquisa.
 - **Exemplos**
 - Os exemplos a seguir mostram como usar algumas condições de pesquisa comuns na cláusula WHERE.

Uso de filtros (cláusula WHERE) nas consultas



- Uso de filtros (cláusula WHERE) nas consultas
 - Localizando uma linha com o uso de uma igualdade simples

```
select * from cliente.dbo.t_vendas  
where LOJA = 'Porto Alegre'
```

- Localizando linhas que contêm um valor como parte de uma cadeia de caracteres

```
select * from cliente.dbo.t_vendas  
where LOJA LIKE ('%Porto%')
```

133 %

Resultados Mensagens

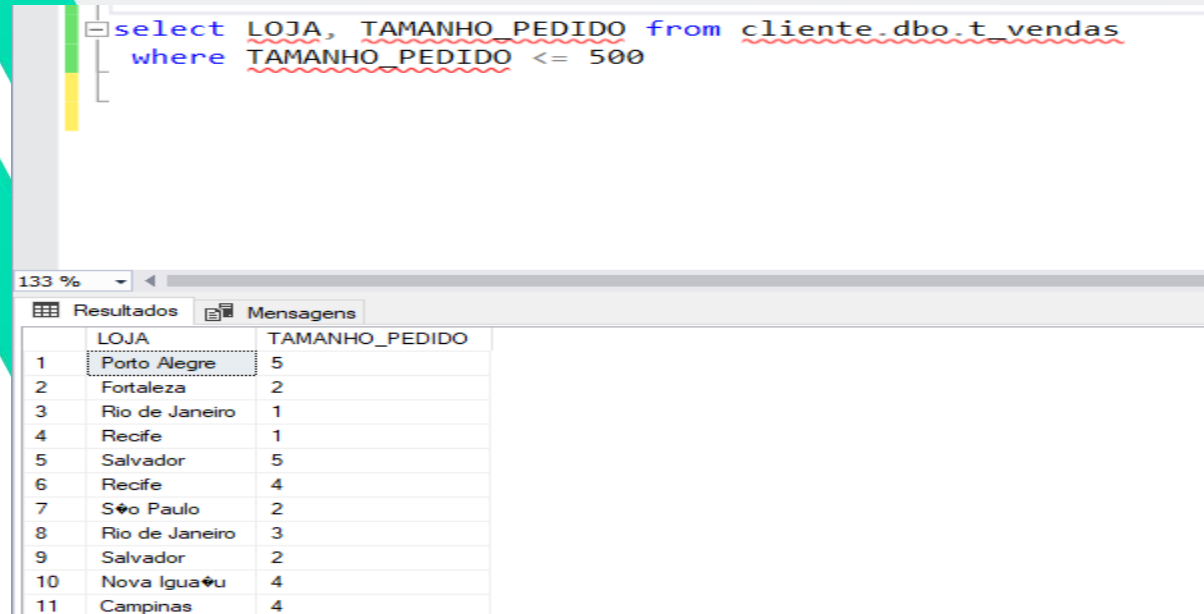
	SKU	TAMANHO_PEDIDO	LOJA	DATA_VENDA	MARCA	CODIGO_CLIENTE	PRECO_UNITARIO	CUSTO_UNITARIO
1	HL1018	5	Porto Alegre	2016-01-01	Xiaomi	64	1200	650
2	HL1023	3	Porto Alegre	2016-01-04	Samsung	554	1500	550
3	HL1007	4	Porto Alegre	2016-01-05	Dell	266	2300	1200
4	HL1013	2	Porto Alegre	2016-01-05	Nikon	525	1550	700
5	HL1008	1	Porto Alegre	2016-01-05	Philco	253	2000	1250
6	HL1024	2	Porto Alegre	2016-01-05	Acer	230	2400	1150
7	HL1008	4	Porto Alegre	2016-01-06	Philco	189	2000	1250
8	HL1009	3	Porto Alegre	2016-01-07	Motorola	68	1400	750
9	HL1016	3	Porto Alegre	2016-01-08	Samsung	115	1450	700
10	HL1001	5	Porto Alegre	2016-01-09	LG	794	2600	1700
11	HL1001	3	Porto Alegre	2016-01-11	LG	780	2600	1700

Uso de filtros (cláusula WHERE) nas consultas

- Uso de filtros (cláusula WHERE) nas consultas

- Localizando linhas com o uso de um operador de comparação

```
select LOJA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO <= 500
```



The screenshot shows a SQL query editor with the following query:

```
select LOJA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO <= 500
```

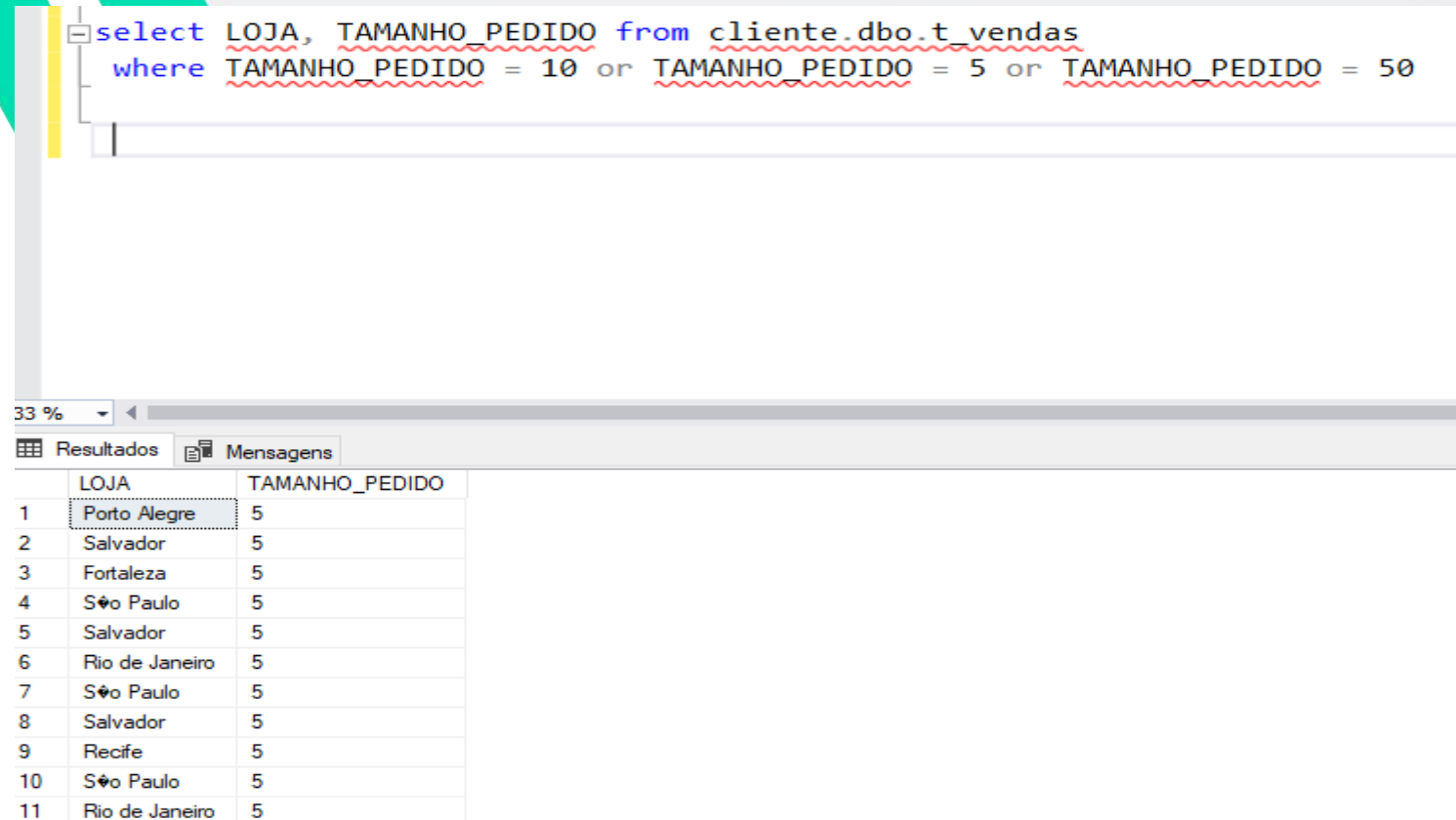
Below the query editor, the results are displayed in a table with two columns: LOJA and TAMANHO_PEDIDO. The table contains 11 rows of data.

	LOJA	TAMANHO_PEDIDO
1	Porto Alegre	5
2	Fortaleza	2
3	Rio de Janeiro	1
4	Recife	1
5	Salvador	5
6	Recife	4
7	São Paulo	2
8	Rio de Janeiro	3
9	Salvador	2
10	Nova Iguaçu	4
11	Campinas	4

Uso de filtros (cláusula WHERE) nas consultas

- Uso de filtros (cláusula WHERE) nas consultas
 - Localizando linhas que atendem a qualquer uma de três condições

```
select LOJA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO = 10 or TAMANHO_PEDIDO = 5 or  
TAMANHO_PEDIDO = 50
```



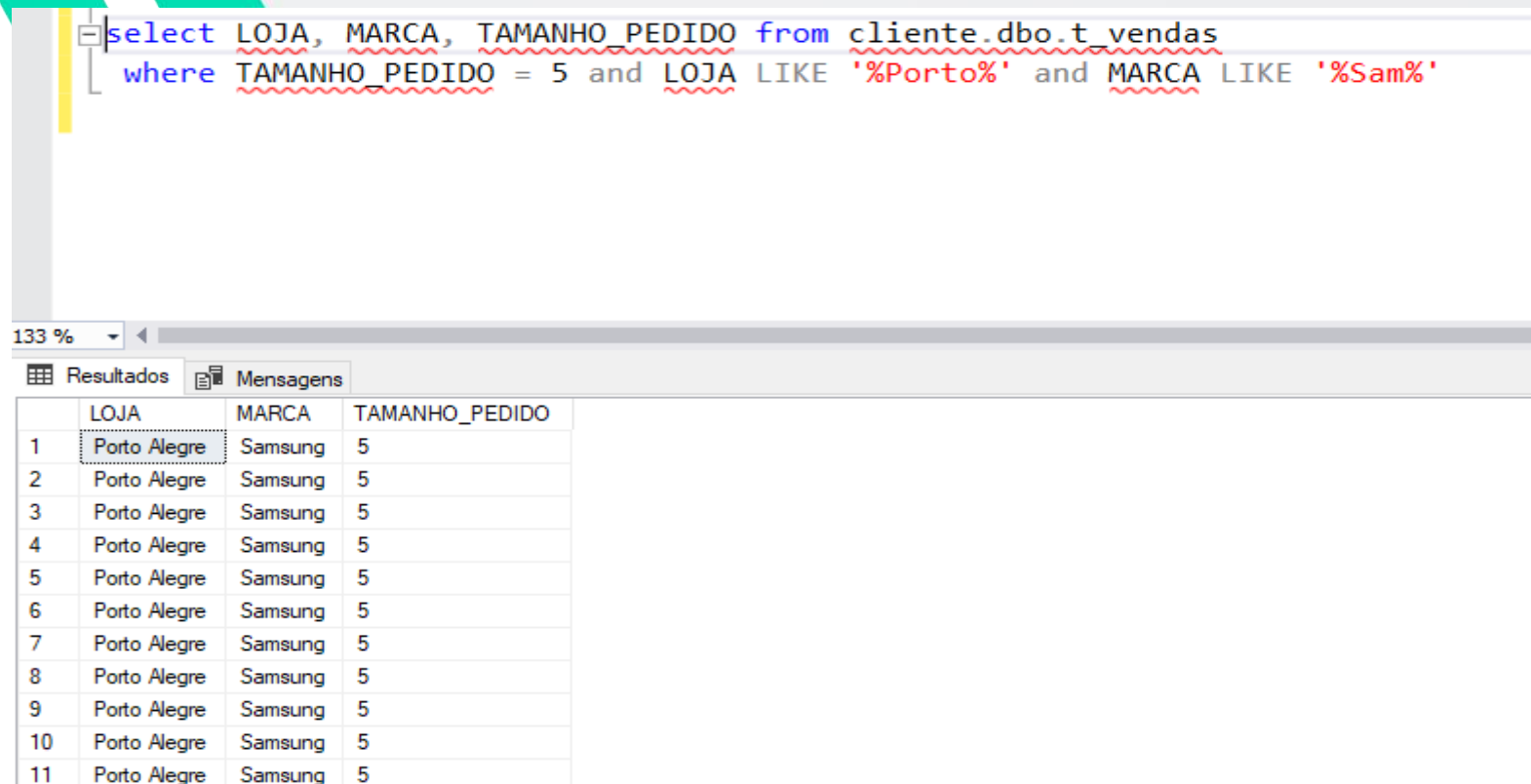
```
select LOJA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO = 10 or TAMANHO_PEDIDO = 5 or TAMANHO_PEDIDO = 50
```

	LOJA	TAMANHO_PEDIDO
1	Porto Alegre	5
2	Salvador	5
3	Fortaleza	5
4	São Paulo	5
5	Salvador	5
6	Rio de Janeiro	5
7	São Paulo	5
8	Salvador	5
9	Recife	5
10	São Paulo	5
11	Rio de Janeiro	5

Uso de filtros (cláusula WHERE) nas consultas

- Uso de filtros (cláusula WHERE) nas consultas
 - Localizando linhas que devem atender a várias condições

```
select LOJA, MARCA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO = 5 and LOJA LIKE '%Porto%' and MARCA  
LIKE '%Sam%'
```



133 %

Resultados Mensagens

	LOJA	MARCA	TAMANHO_PEDIDO
1	Porto Alegre	Samsung	5
2	Porto Alegre	Samsung	5
3	Porto Alegre	Samsung	5
4	Porto Alegre	Samsung	5
5	Porto Alegre	Samsung	5
6	Porto Alegre	Samsung	5
7	Porto Alegre	Samsung	5
8	Porto Alegre	Samsung	5
9	Porto Alegre	Samsung	5
10	Porto Alegre	Samsung	5
11	Porto Alegre	Samsung	5

Uso de filtros (cláusula WHERE) nas consultas



- Uso de filtros (cláusula WHERE) nas consultas
 - Localizando linhas que estão em uma lista de valores

```
select LOJA, MARCA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where LOJA in ('Rio de Janeiro', 'Porto Alegre', 'Fortaleza')
```

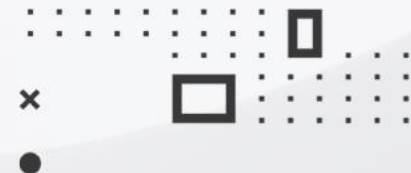
```
select LOJA, MARCA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where LOJA in ('Rio de Janeiro', 'Porto Alegre', 'Fortaleza')
```

33 %

Resultados Mensagens

	LOJA	MARCA	TAMANHO_PEDIDO
1	Porto Alegre	Xiaomi	5
2	Fortaleza	LG	2
3	Rio de Janeiro	Philco	1
4	Rio de Janeiro	Samsung	3
5	Fortaleza	Samsung	5
6	Rio de Janeiro	Motorola	4
7	Fortaleza	Apple	4
8	Rio de Janeiro	Canon	5
9	Rio de Janeiro	Apple	2
10	Rio de Janeiro	Samsung	4
11	Fortaleza	Dell	2

Uso de filtros (cláusula WHERE) nas consultas



- Uso de filtros (cláusula WHERE) nas consultas
 - Localizando linhas que têm um valor entre dois valores

```
select LOJA, MARCA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO between 5 and 50
```

The screenshot shows a SQL query editor with the following query:

```
select LOJA, MARCA, TAMANHO_PEDIDO from cliente.dbo.t_vendas  
where TAMANHO_PEDIDO between 5 and 50
```

Below the query editor, there is a tab labeled 'Resultados' (Results) and a tab labeled 'Mensagens' (Messages). The 'Resultados' tab is active, displaying a table with 11 rows and 4 columns: LOJA, MARCA, and TAMANHO_PEDIDO. The first row is highlighted with a dashed border.

	LOJA	MARCA	TAMANHO_PEDIDO
1	Porto Alegre	Xiaomi	5
2	Salvador	Philco	5
3	Fortaleza	Samsung	5
4	São Paulo	Philco	5
5	Salvador	Acer	5
6	Rio de Janeiro	Canon	5
7	São Paulo	LG	5
8	Salvador	Samsung	5
9	Recife	Canon	5
10	São Paulo	Canon	5
11	Rio de Janeiro	Samsung	5

- Atividades:

1. Consulte os dados da tabela t_vendas, selecione todos os campos.
2. Consulte os dados da tabela t_vendas, selecione os campos: codigo_cliente, loja, tamanho_pedido.
3. Consulte os dados da tabela t_vendas, selecione os campos: codigo_cliente, loja, tamanho_pedido, acrescente na condição WHERE loja = 'Rio de Janeiro'.
4. Consulte os dados da tabela t_vendas, selecione todos os campos , acrescente na condição WHERE loja in ('Rio de Janeiro', 'Salvador')
5. Consulte os dados da tabela t_vendas, selecione todos os campos, acrescente na condição WHERE tamanho_pedido > 5 e codigo_cliente entre 100 e 200.
6. Consulte os dados da tabela t_vendas, crie o campo UF consultando o campo LOJA em que: LOJA = Rio de Janeiro a UF será RJ; LOJA = São Paulo, a UF será SP; LOJA = Salvador a UF será BA; LOJA = Belo Horizonte a UF será MG.
7. Consulte os dados da tabela t_clientes, selecione todos os campos.
8. Consulte os dados da tabela t_clientes, selecione todos os cliente em que a UF = SP.
9. Consulte os dados da tabela t_clientes, selecione todos os clientes em que o código do cliente esteja entre 10 e 20.
10. Consulte os dados da tabela t_clientes, selecione todos os clientes em que a UF = RJ e o código do cliente seja maior que 50.

Dúvidas ?
Obrigado !!

conti30@gmail.com