

Download das bases do projeto

Criando um banco de dados

Importando arquivos para tabelas de carga

Verificando dados das tabelas

Criando a tabela de produção

Tipos de campos

Comando Update para ajustar campos

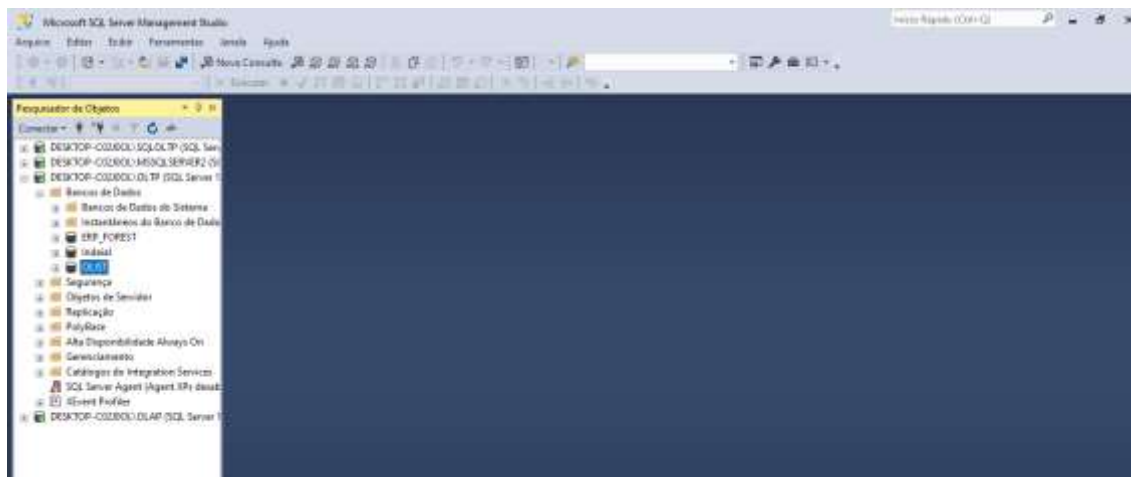
Comando Join para juntar tabelas

Conectando dados no Excel

PROJETO USANDO SQL

E-Commerce – Olist

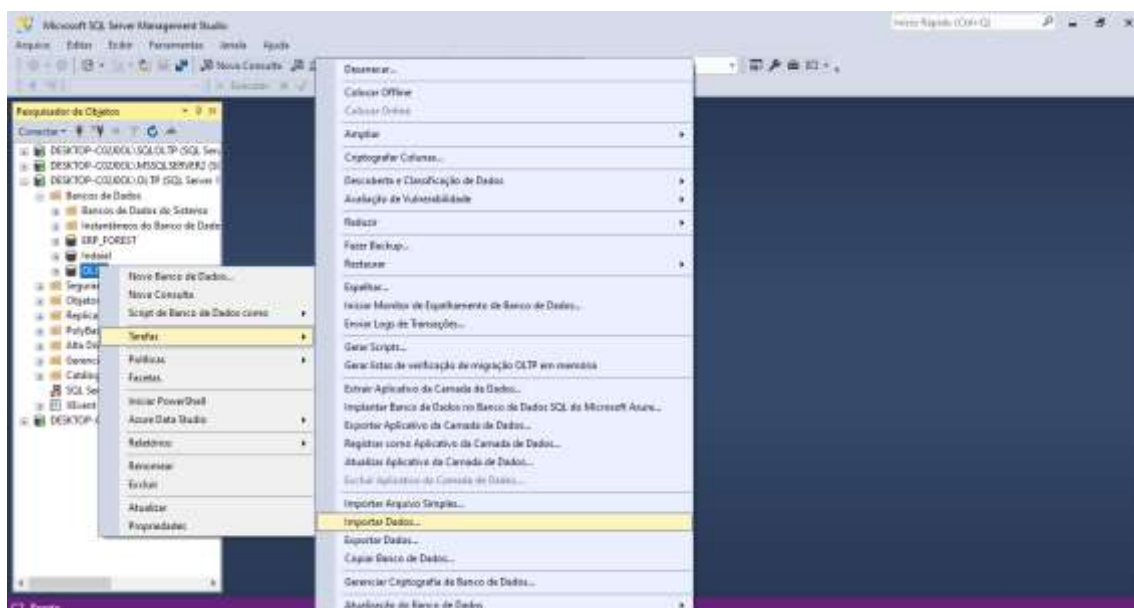
Criar o banco de dados OLIST no SQL SERVER



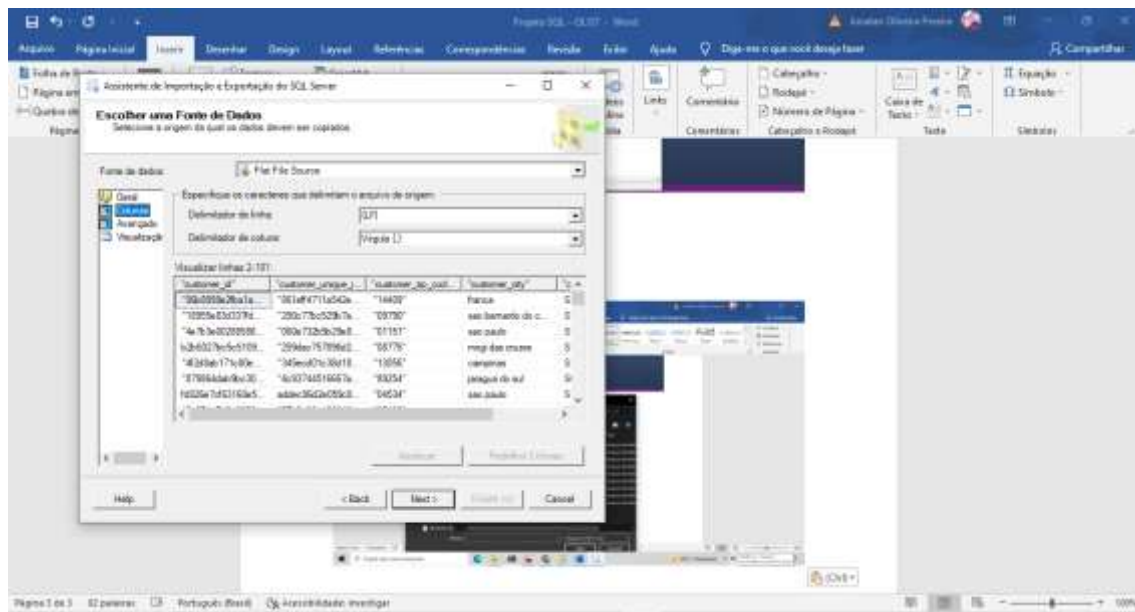
Tabelas para serem importadas para o SQL

Nome	Data de modificação	Tipo	Tamanho
tbl_customer_detail	01/10/2017 18:00	Arquivo de Tabela...	6.821 KB
tbl_geolocation_detail	01/10/2017 18:00	Arquivo de Tabela...	19.000 KB
tbl_order_item_detail	01/10/2017 18:00	Arquivo de Tabela...	13.077 KB
tbl_order_payment_detail	01/10/2017 18:00	Arquivo de Tabela...	3.642 KB
tbl_order_return_detail	01/10/2017 18:00	Arquivo de Tabela...	14.110 KB
tbl_order_detail	01/10/2017 18:00	Arquivo de Tabela...	17.240 KB
tbl_product_detail	01/10/2017 18:00	Arquivo de Tabela...	3.334 KB
tbl_sales_detail	01/10/2017 18:00	Arquivo de Tabela...	177 KB
product_category_name_translation	01/10/2017 18:00	Arquivo de Tabela...	1 KB

Importar os dados para o SQL SERVER



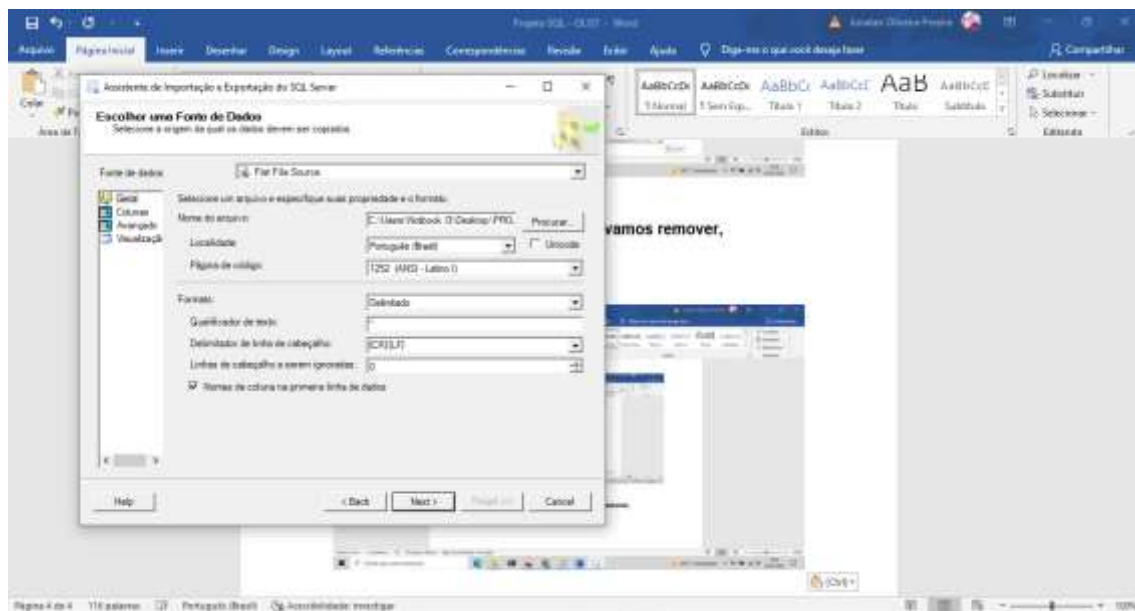
ARQUIVOS CSV e TXT utilizar o FLAT FILE SOURCE



Não importar com os delimitadores

Identificado o delimitador (") agora vamos remover,

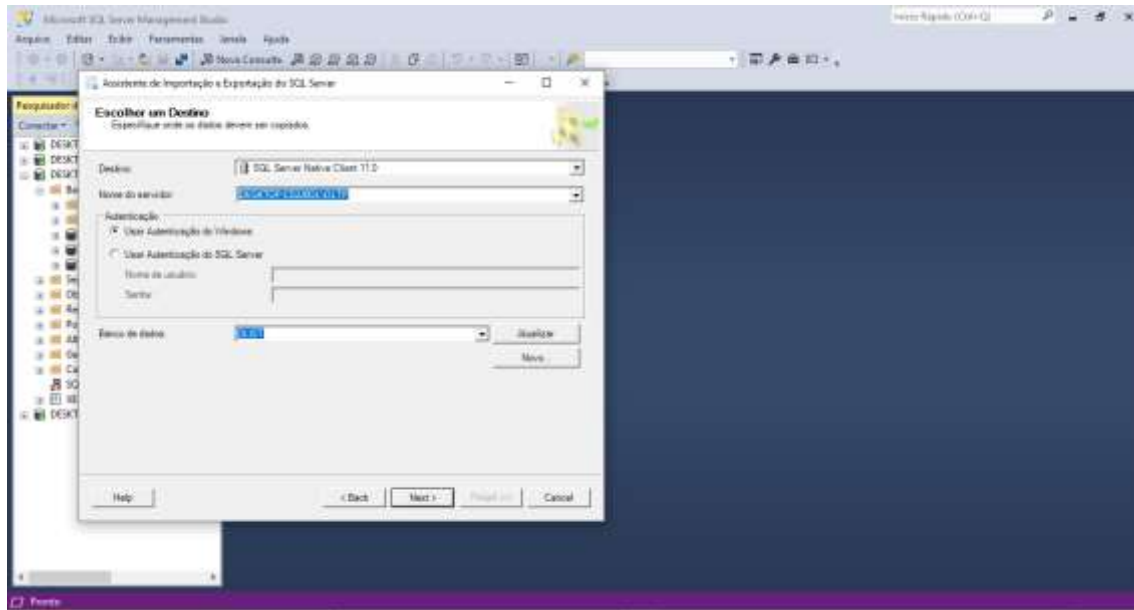
clicar em geral, qualificador de texto



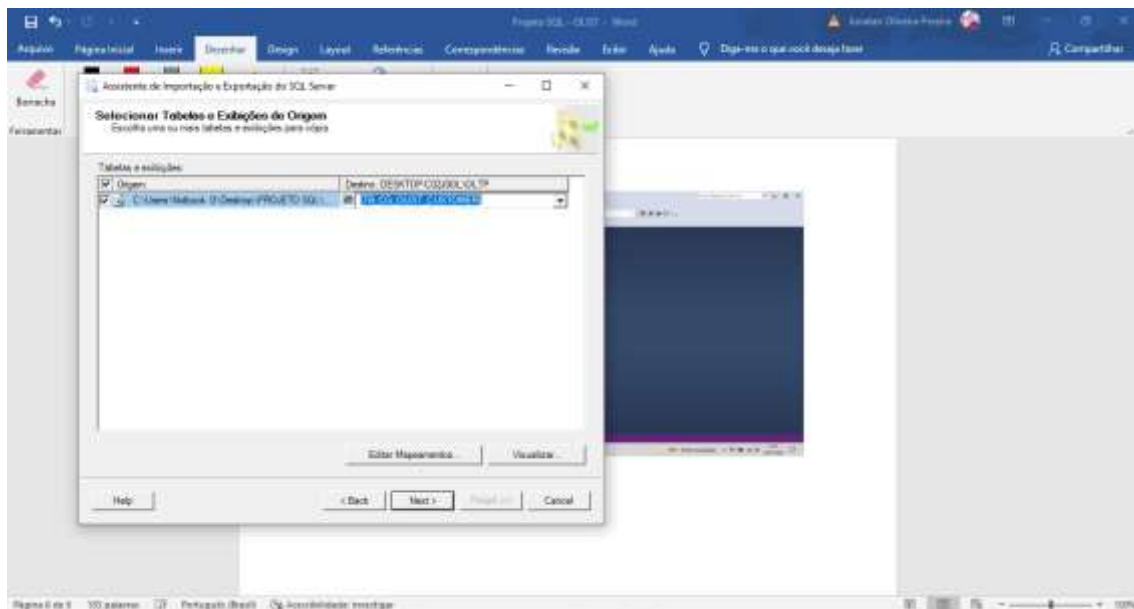
[illegible]

Destino seleccionar SQL SERVER NATIVE CLIENTE

Nome do Servidor: pode ser utilizado o IP, Hostname ou nome do servidor e aperta next



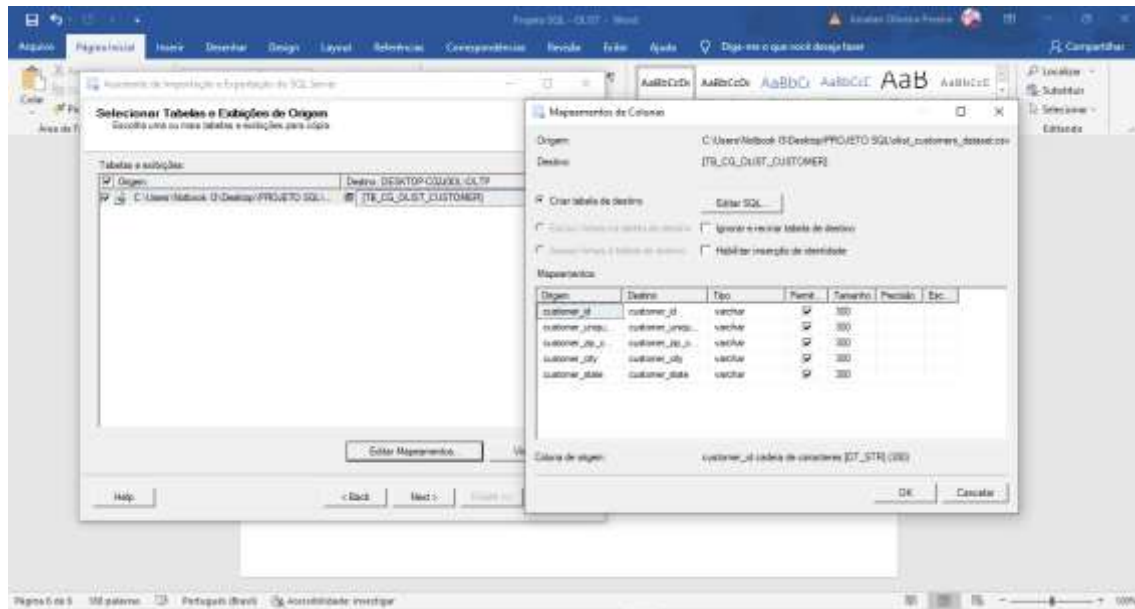
Caso queira alterar o nome



Por ser uma tabela de Carga não iremos alterar o tipo de campo

por que no SQL precisamos ter o campos

campo texto criado como texto,
campo numérico como numérico



Clicar em OK, NEXT, NEXT e FINISH

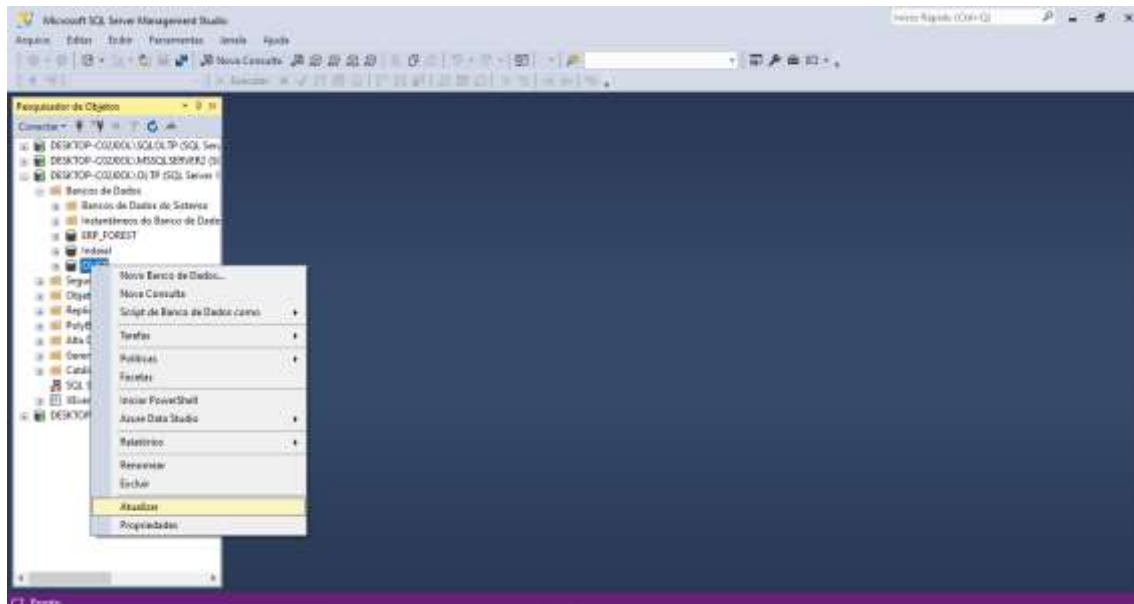
Começar a rodar o processo, informação de sucesso

Clicar em Close

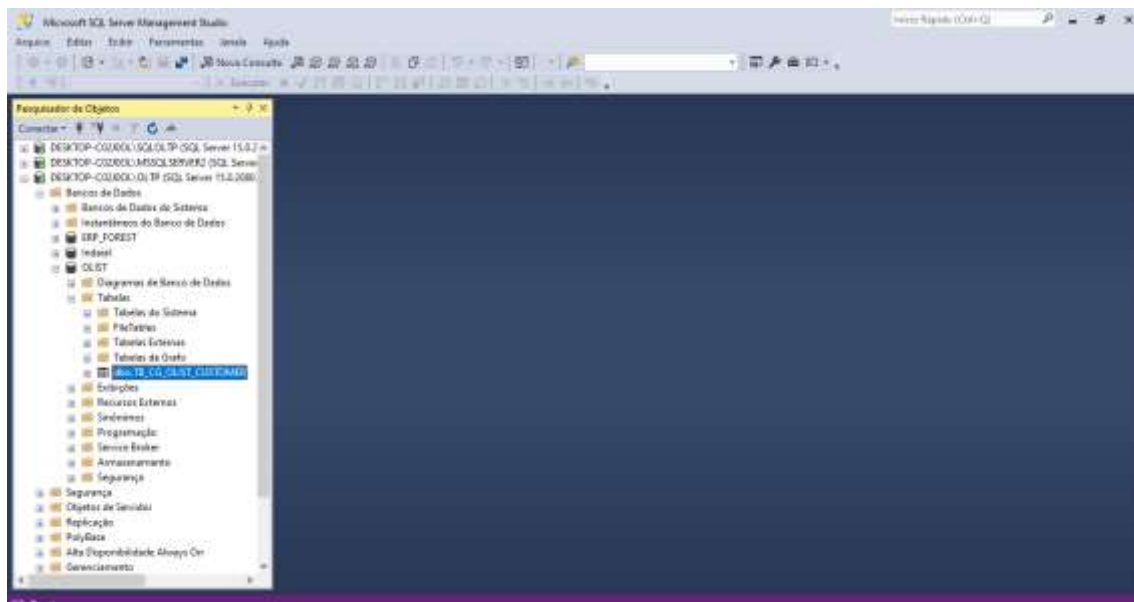


Ir até o bando de dados no SQL SERVER

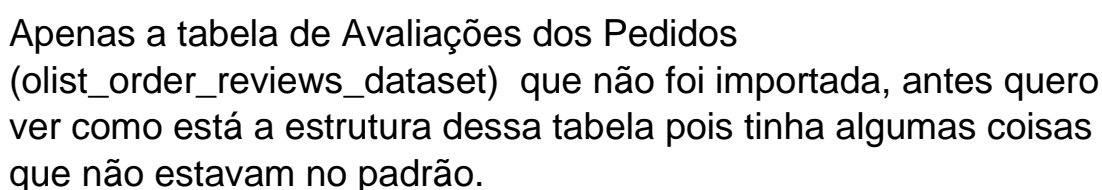
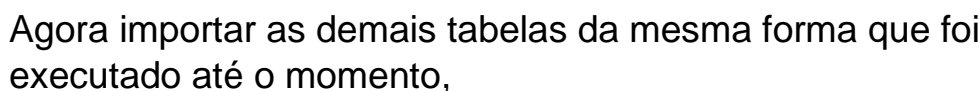
Atualizar as tabelas



Verificando se a tabela já está disponível




```
SELECT * FROM TB CG OLIST CUSTOMER
```



Agora precisamos identificar o relacionamento entre as tabelas

```
SELECT * FROM TP_C4_OLIST_CUSTOMER
SELECT * FROM [dbo].[product_category_name_translation]
SELECT * FROM [dbo].[olist_sellers_dataset]
SELECT * FROM [dbo].[olist_products_dataset]
SELECT * FROM [dbo].[olist_orders_dataset]
SELECT * FROM [dbo].[olist_order_payments_dataset]
SELECT * FROM [dbo].[olist_order_items_dataset]
SELECT * FROM [dbo].[olist_geolocation_dataset]
```

customer_id	customer_unique_id	customer_jid_code_prefix	customer_city	customer_state
c4835040a230a52f5d5a2a78a0	2ca0b240a70ca0f00a12776103770	00750	fortaleza	CE
a25e40534cc329a71a030a3a3a070	20103032a7e74124c0230a0a530a	90410	porto alegre	RS
c07070a0e154a7a03a5a0a7a07	b4b4050a303070a0400a2a145a0a	10070	caracuba	SP
800a40013a701a0a0a0a0a0a0a	800a400a0a0a777070a012a00a7	80007	chapaco	SC
a070a0a0e154a7a03a5a0a7a07	13171a0a0a0a0a0a0a0a0a0a0a	01301	seo paulo	SP
201a0a0a0a0a0a0a0a0a0a0a0a	00a0700a0a0a0a0a0a0a0a0a0a	57030	parqueira	AL
907070a0e154a7a03a5a0a7a07	a70a13a7a0a0a0a0a0a0a0a0a0	71020	brasilia	DF
3a0a0a0a0a0a0a0a0a0a0a0a0a	0a0a1a0a0a0a0a0a0a0a0a0a0a	80000	estrela	RS
a0a0a0a0e154a7a03a5a0a7a07	400a0a0a0a0a0a0a0a0a0a0a0a	70770	condragua	MT
0a0a0a0a0a0a0a0a0a0a0a0a0a	0a0a0a0a0a0a0a0a0a0a0a0a0a	41000	salvador	BA
20a0a0a0e154a7a03a5a0a7a07	20a0a0a0a0a0a0a0a0a0a0a0a0	30000	canela	RS
c24a700a0a0a0a0a0a0a0a0a0a	070a0a0a0a0a0a0a0a0a0a0a0a	32400	seo paulo	SP
c30a0a0a0a0a0a0a0a0a0a0a0a	80c70a0a0a0a0a0a0a0a0a0a0a	21510	rio de janeiro	RJ
70a0a0a0e154a7a03a5a0a7a07	b4b4050a303070a0400a2a145a0a	83040	seo paulo	PR
2a1720a0a0e154a7a03a5a0a7a07	640a0a0a0a0a0a0a0a0a0a0a0a	80520	lagos	SC
a0a0a0a0e154a7a03a5a0a7a07	0a0a0a0a0a0a0a0a0a0a0a0a0a	14320	dourado	SP

Vamos verificar os TOP 10 de cada tabela

```
SELECT TOP 10 * FROM TP_C4_OLIST_CUSTOMER
SELECT TOP 10 * FROM [dbo].[product_category_name_translation]
SELECT TOP 10 * FROM [dbo].[olist_sellers_dataset]
SELECT TOP 10 * FROM [dbo].[olist_products_dataset]
SELECT TOP 10 * FROM [dbo].[olist_orders_dataset]
SELECT TOP 10 * FROM [dbo].[olist_order_payments_dataset]
SELECT TOP 10 * FROM [dbo].[olist_order_items_dataset]
SELECT TOP 10 * FROM [dbo].[olist_geolocation_dataset]
```

customer_id	customer_unique_id	customer_jid_code_prefix	customer_city	customer_state
c07070a0e154a7a03a5a0a7a07	b4b4050a303070a0400a2a145a0a	10070	caracuba	SP
800a40013a701a0a0a0a0a0a0a	800a400a0a0a777070a012a00a7	80007	chapaco	SC
a070a0a0e154a7a03a5a0a7a07	13171a0a0a0a0a0a0a0a0a0a0a	01301	seo paulo	SP
201a0a0a0a0a0a0a0a0a0a0a0a	00a0700a0a0a0a0a0a0a0a0a0a	57030	parqueira	AL
907070a0e154a7a03a5a0a7a07	a70a13a7a0a0a0a0a0a0a0a0a0	71020	brasilia	DF

product_id	product_category_name	product_name_length	product_description_length	product_photos_qty	product_weight_g	product_length_cm	product_height_cm
3442000a0a0a0a0a0a0a0a0a0a	automotivo	auto					
c07070a0e154a7a03a5a0a7a07	cama_mesa_banho	bed_bath_table					
a070a0a0e154a7a03a5a0a7a07	movel_movelado	furniture_decor					
201a0a0a0a0a0a0a0a0a0a0a0a	roupas_femin	girls_fashion					

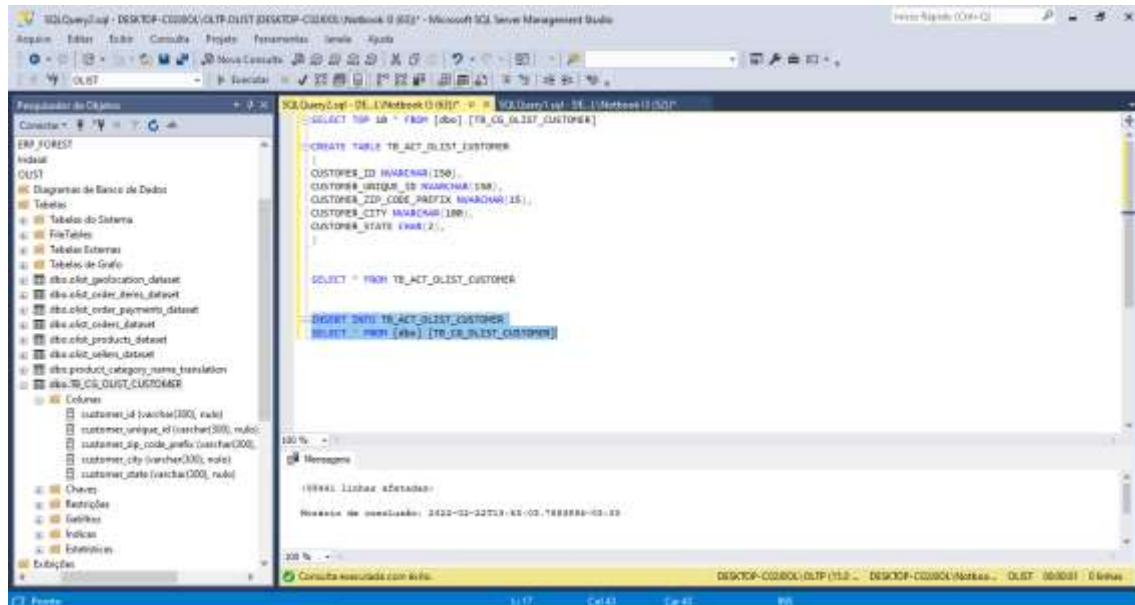
```
[dbo].[TB_CG_OLIST_CUSTOMER] em TB_ACT_OLIST_CUSTOMER
```

NVARCHAR utilizar para texto se for número usar INT

```
INSERT INTO TB_ACT_OLIST_CUSTOMER
```

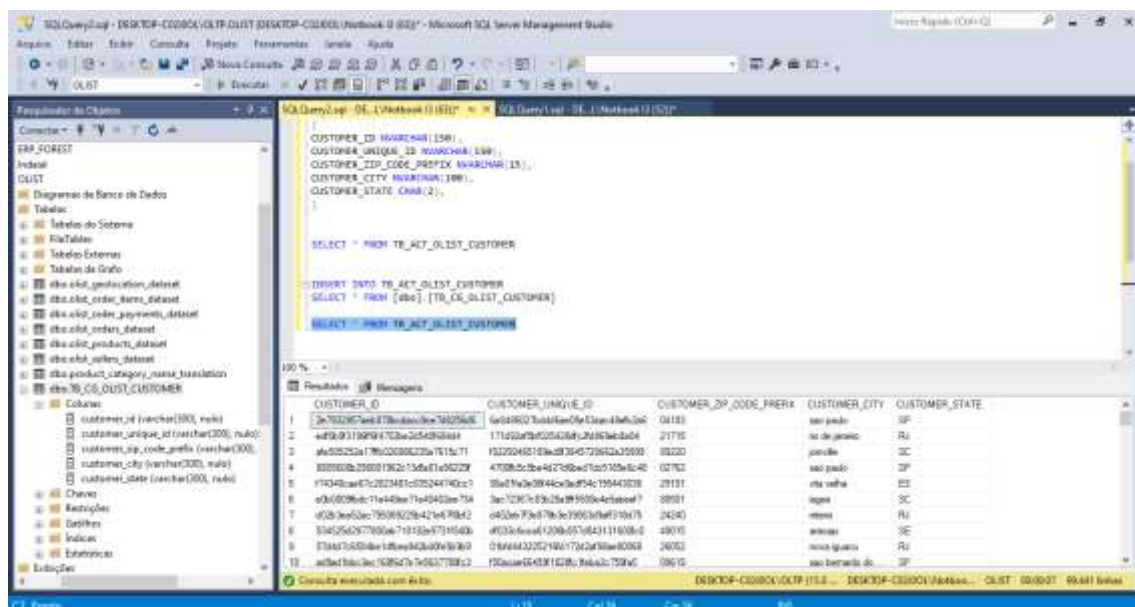
```
SELECT * FROM [dbo].[TB_CG_OLIST_CUSTOMER]
```

Não executar mais de uma vez para não Duplicar

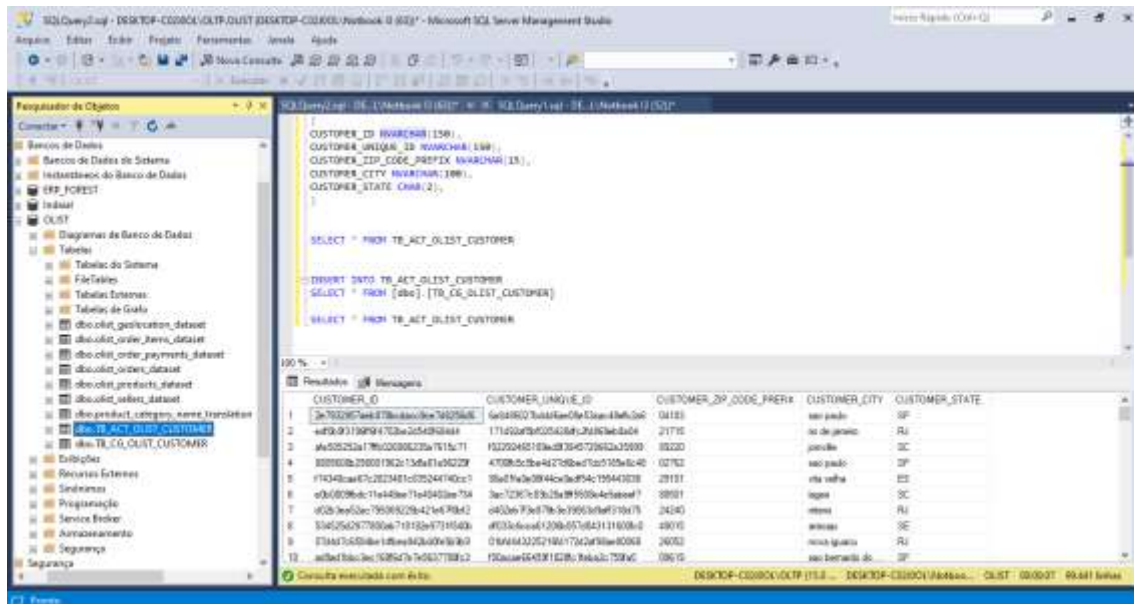


Vamos verificar a transformação para TB_ACT_OLIST_CUSTOMER

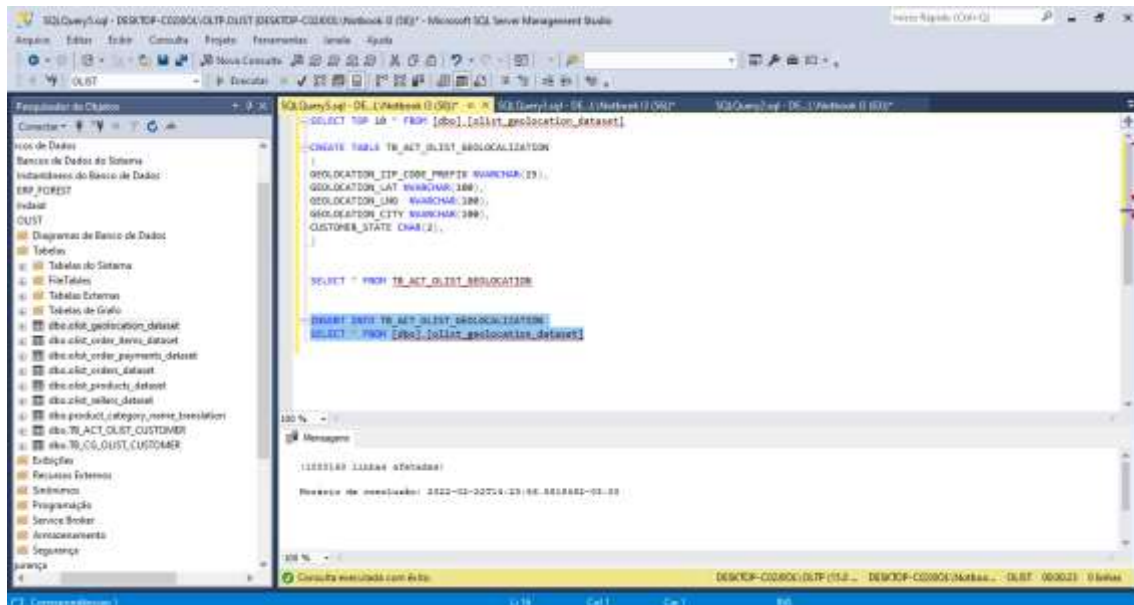
```
SELECT * FROM TB_ACT_OLIST_CUSTOMER
```



Atualizar tabelas para carregar a nova tabela TB_ACT_OLIST_CUSTOMER

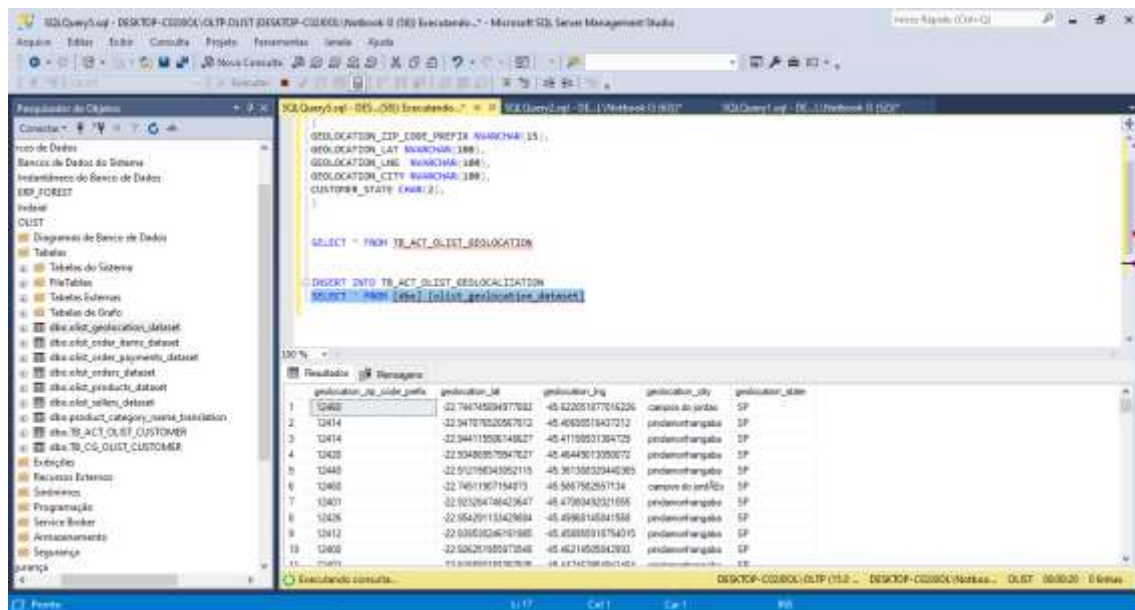


Transformando as demais tabelas Cargas em Produção

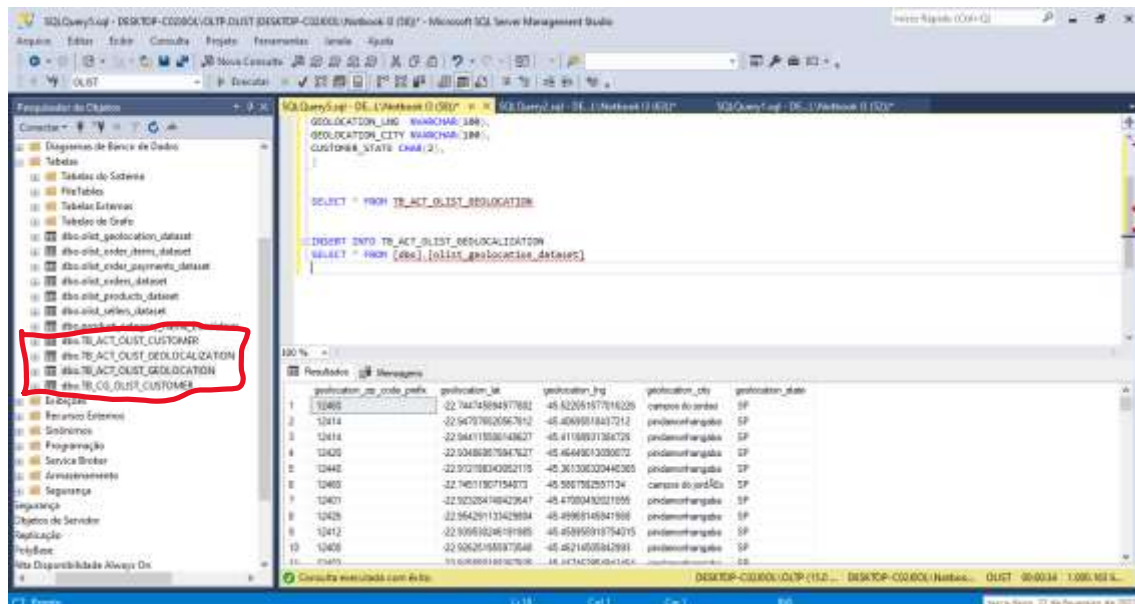


Agora verificando as informações fazendo um

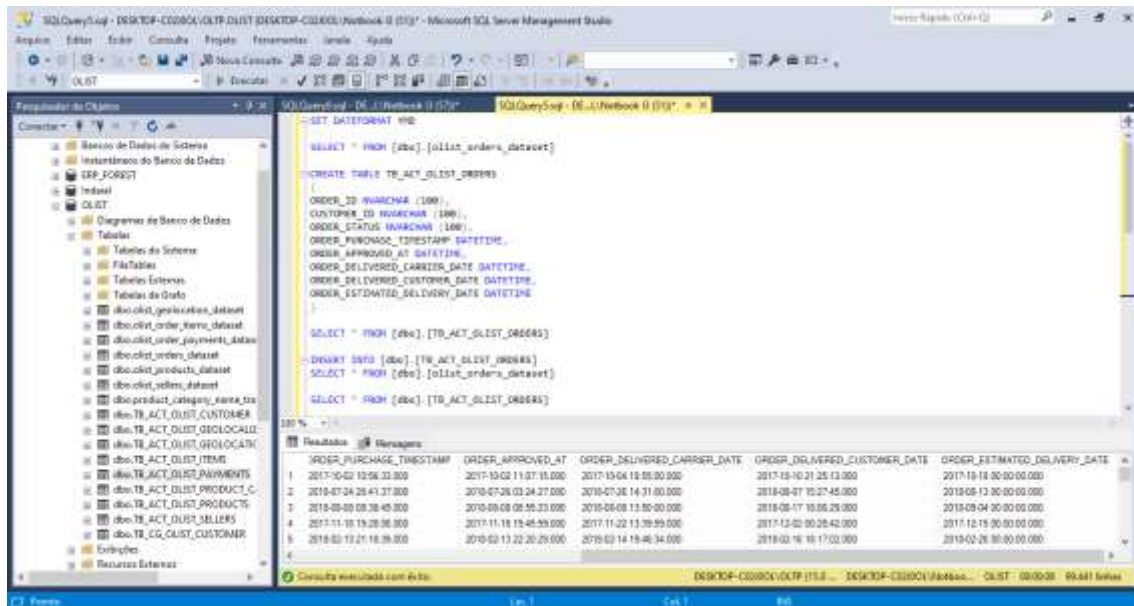
`SELECT * FROM [dbo].[olist_geolocation_dataset]`



Atualizando as tabelas

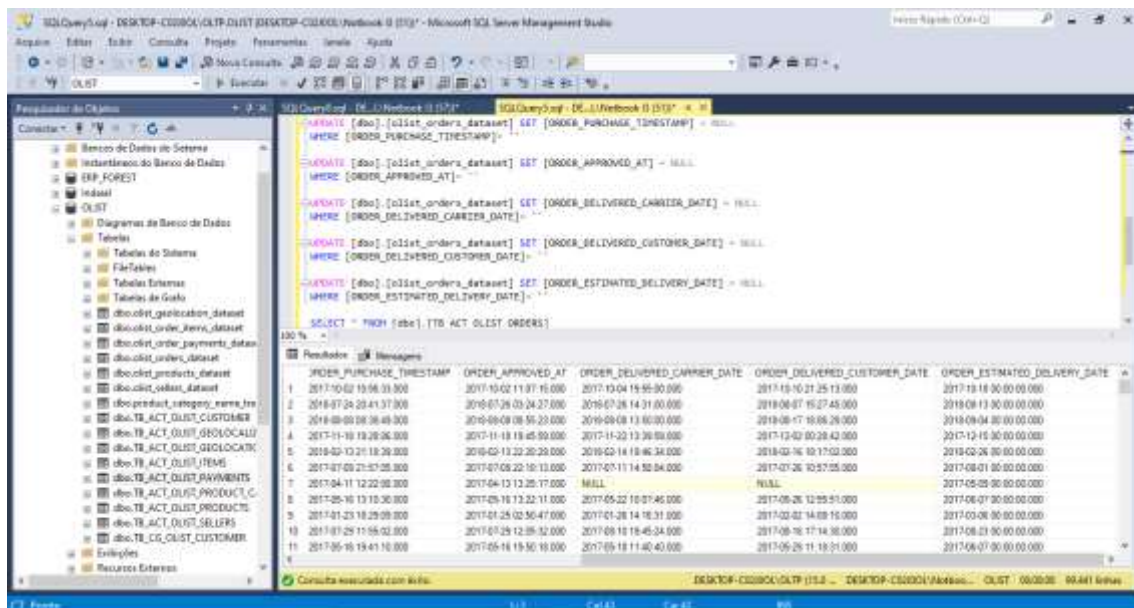


SELECT TOP 10 * FROM [dbo].[olist_sellers_dataset]

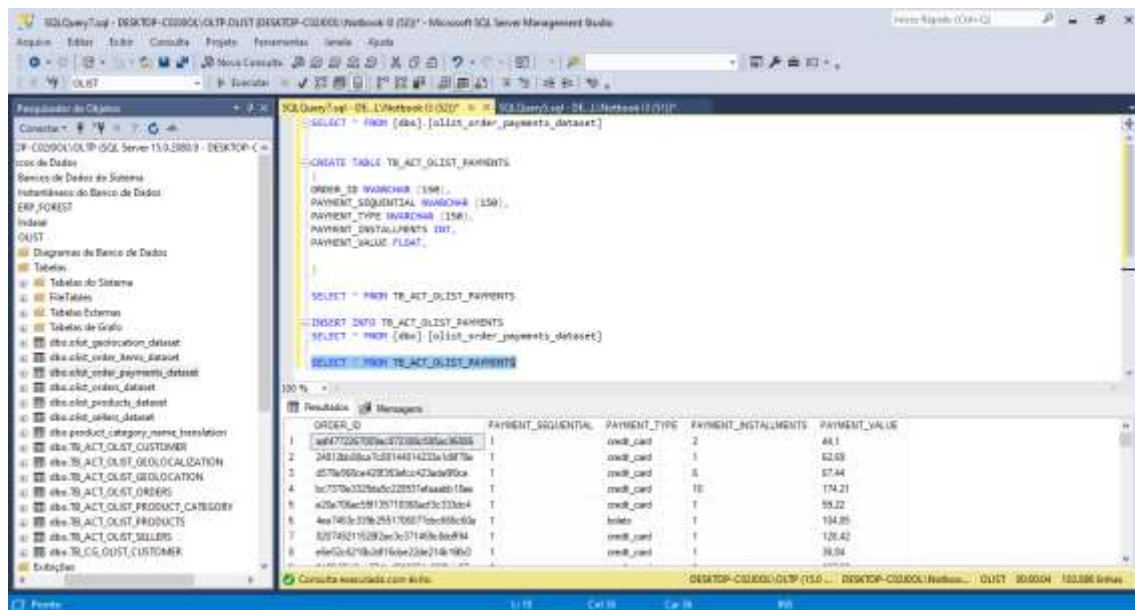


Preenchendo os valores em branco com NULL nas tabelas de Carga e verificando o preenchimento

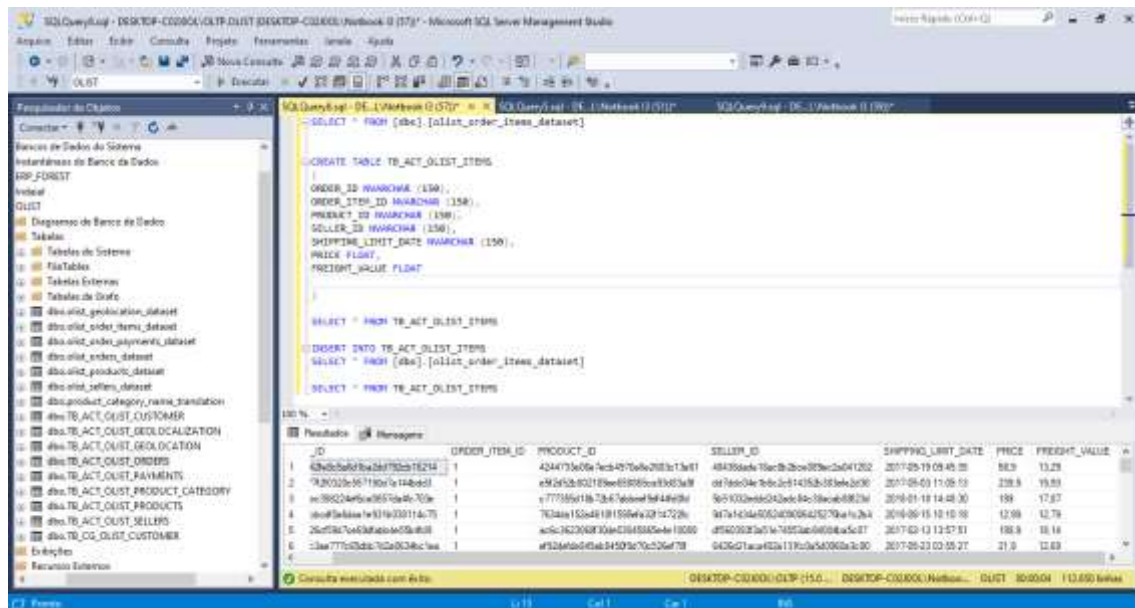
`SELECT * FROM [dbo].[TB_ACT_OLIST_ORDERS]`



`SELECT * FROM [dbo].[olist_order_payments_dataset]`



`SELECT * FROM [dbo].[olist_order_items_dataset]`

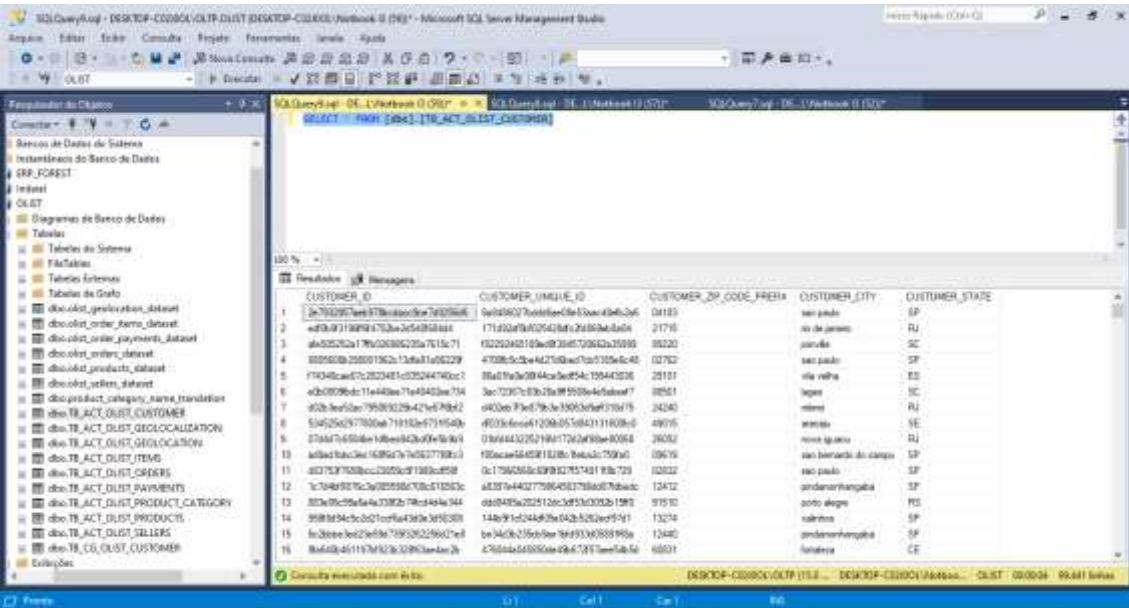


PERGUNTAS DE NEGOCIO

QUAIS CLIENTES QUE FIZERAM UM PEDIDO?

Consultando os clientes que foram cadastrados

```
SELECT * FROM [dbo].[TB_ACT_OLIST_CUSTOMER]
```

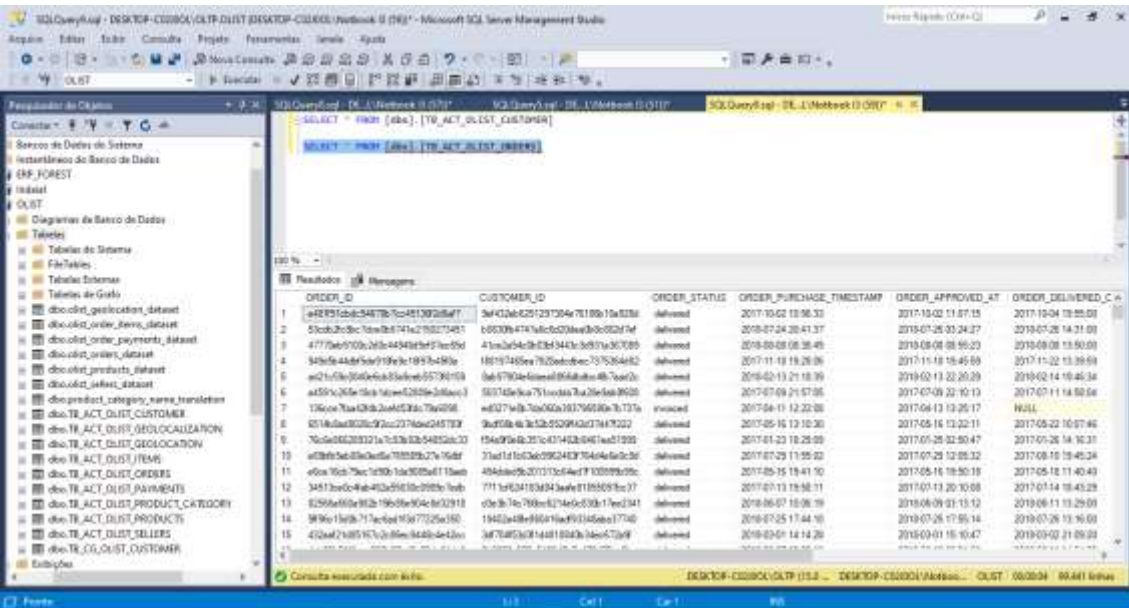


The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure, including the 'TB_ACT_OLIST_CUSTOMER' table. The right pane shows the query results for the query 'SELECT * FROM [dbo].[TB_ACT_OLIST_CUSTOMER]'. The results are displayed in a grid with 16 columns: CUSTOMER_ID, CUSTOMER_UNIQUE_ID, CUSTOMER_ZIP_CODE_FREE, CUSTOMER_CITY, CUSTOMER_STATE, and several other columns. The data is sorted by CUSTOMER_ID.

CUSTOMER_ID	CUSTOMER_UNIQUE_ID	CUSTOMER_ZIP_CODE_FREE	CUSTOMER_CITY	CUSTOMER_STATE
1	2e702957ee978e0a0e7e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
2	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
3	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
4	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
5	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
6	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
7	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
8	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
9	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
10	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
11	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
12	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
13	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
14	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
15	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
16	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e

Consultando os Pedidos

```
SELECT * FROM [dbo].[TB_ACT_OLIST_ORDERS]
```



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure, including the 'TB_ACT_OLIST_ORDERS' table. The right pane shows the query results for the query 'SELECT * FROM [dbo].[TB_ACT_OLIST_ORDERS]'. The results are displayed in a grid with 7 columns: ORDER_ID, CUSTOMER_ID, ORDER_STATUS, ORDER_PURCHASE_TIMESTAMP, ORDER_APPROVED_AT, and ORDER_DELIVERED_C_A. The data is sorted by ORDER_ID.

ORDER_ID	CUSTOMER_ID	ORDER_STATUS	ORDER_PURCHASE_TIMESTAMP	ORDER_APPROVED_AT	ORDER_DELIVERED_C_A
1	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
2	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
3	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
4	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
5	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
6	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
7	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
8	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
9	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
10	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
11	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
12	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
13	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
14	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e
15	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e	0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e0e

Agora vamos fazer o cruzamento entre duas tabelas (JOIN)

`SELECT * FROM [dbo].[TB_ACT_OLIST_CUSTOMER] AS O => Apelido O`

`SELECT * FROM [dbo].[TB_ACT_OLIST_ORDERS] AS C => Apelido C`

Relacionamento entre as tabelas

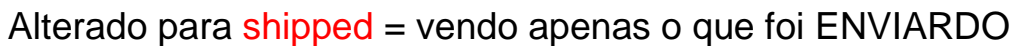
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database schema for 'TB_ACT_OLIST'. The 'CUSTOMER' table is highlighted, showing its columns: 'CUSTOMER_ID' (nvarchar(100), null), 'CUSTOMER_UNIQUE_ID' (nvarchar(100), null), 'CUSTOMER_ZIP_CODE_PREFIX' (nvarchar(10), null), 'CUSTOMER_CITY' (nvarchar(100), null), and 'CUSTOMER_STATE' (char(2), null). The 'ORDERS' table is also highlighted, showing its columns: 'ORDER_ID' (nvarchar(100), null), 'CUSTOMER_ID' (nvarchar(100), null), 'ORDER_STATUS' (nvarchar(10), null), 'ORDER_PURCHASE_TIMESTAMP' (datetime), 'ORDER_APPROVED_AT' (datetime), and 'ORDER_DELIVERED_C' (datetime). On the right, the 'Query Editor' pane shows a query that joins the 'CUSTOMER' and 'ORDERS' tables. The query is: `SELECT * FROM [dbo].[TB_ACT_OLIST_CUSTOMER] AS O JOIN [dbo].[TB_ACT_OLIST_ORDERS] AS C ON O.CUSTOMER_ID = C.CUSTOMER_ID`. The 'Results' pane shows the output of the query, displaying columns: 'ORDER_ID', 'CUSTOMER_ID', 'ORDER_STATUS', 'ORDER_PURCHASE_TIMESTAMP', 'ORDER_APPROVED_AT', and 'ORDER_DELIVERED_C'. The results show a list of orders with their corresponding customer information.

CRIANDO FUNÇÃO DE VISUALIZAÇÃO

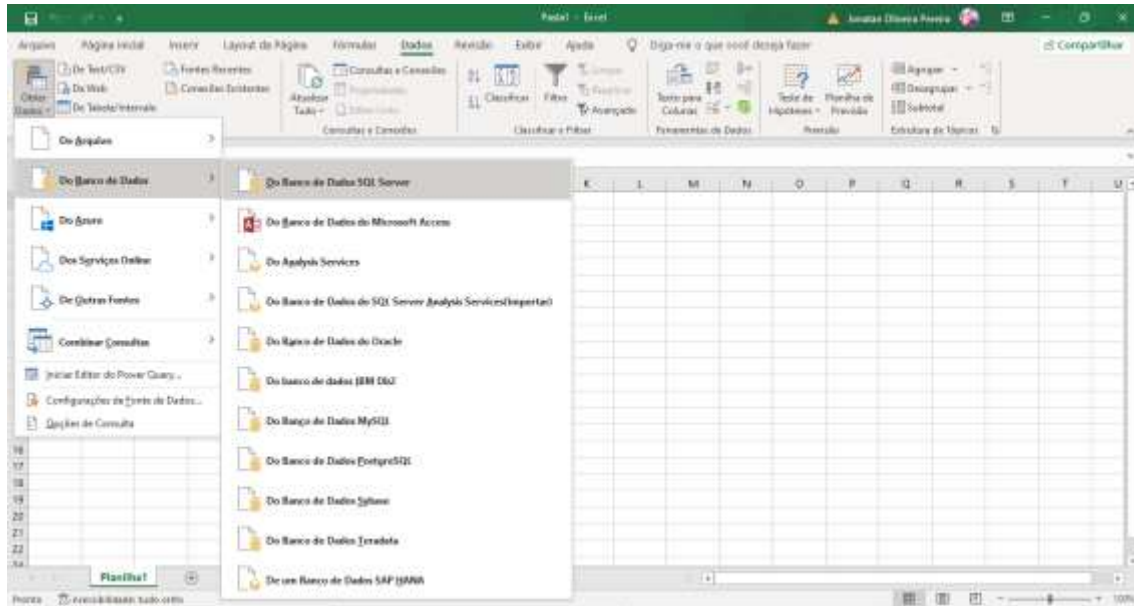
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database schema for 'TB_ACT_OLIST'. The 'CUSTOMER' table is highlighted, showing its columns: 'CUSTOMER_ID' (nvarchar(100), null), 'CUSTOMER_UNIQUE_ID' (nvarchar(100), null), 'CUSTOMER_ZIP_CODE_PREFIX' (nvarchar(10), null), 'CUSTOMER_CITY' (nvarchar(100), null), and 'CUSTOMER_STATE' (char(2), null). The 'ORDERS' table is also highlighted, showing its columns: 'ORDER_ID' (nvarchar(100), null), 'CUSTOMER_ID' (nvarchar(100), null), 'ORDER_STATUS' (nvarchar(10), null), 'ORDER_PURCHASE_TIMESTAMP' (datetime), 'ORDER_APPROVED_AT' (datetime), and 'ORDER_DELIVERED_C' (datetime). On the right, the 'Query Editor' pane shows a query that joins the 'CUSTOMER' and 'ORDERS' tables. The query is: `SELECT * FROM [dbo].[TB_ACT_OLIST_CUSTOMER] AS O JOIN [dbo].[TB_ACT_OLIST_ORDERS] AS C ON O.CUSTOMER_ID = C.CUSTOMER_ID`. The 'Results' pane shows the output of the query, displaying columns: 'ORDER_ID', 'CUSTOMER_ID', 'ORDER_STATUS', 'ORDER_PURCHASE_TIMESTAMP', 'ORDER_APPROVED_AT', and 'ORDER_DELIVERED_C'. The results show a list of orders with their corresponding customer information.

Alterado para **delivered** = vendo apenas o que foi ENTREGUE

Alterado para **delivered** = vendo apenas o que foi ENTREGUE



ABRIR EXCEL, EM OBTER DADOS> DO BANCO DE DADOS> DO BANCO DE DADOS SQL SERVER

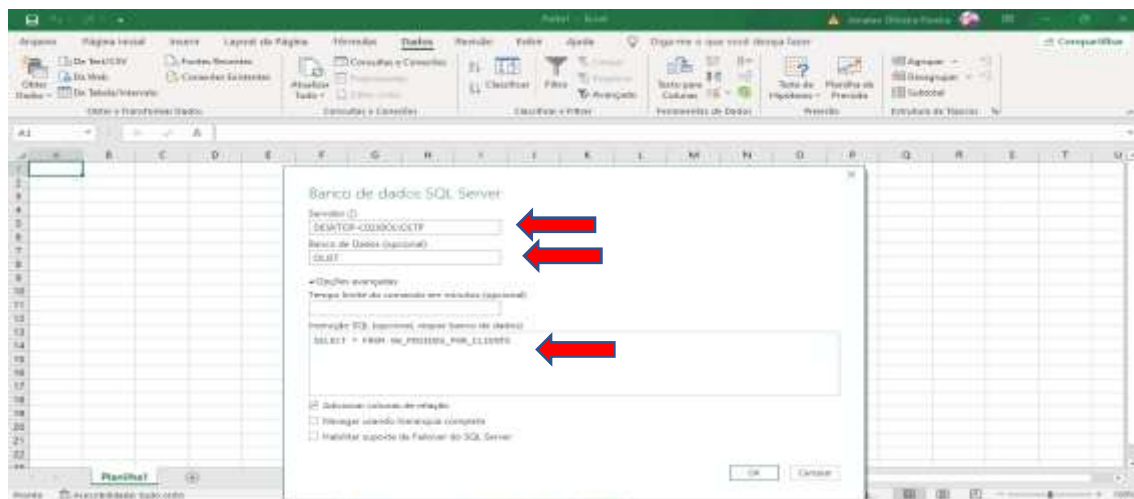


INSERIR O NOME DO SERVIDOR E O BANCO QUE DESEJA SE
CONECTAR,

CASO QUEIRA MOSTRAR APENAS UMA VISUALIZAÇÃO
INFORMAR NO CAMPO A FUNÇÃO NESSE CASO AQUI A
FUNÇÃO É

SELECT * FROM VW_PEDIDOS_POR_CLIENTE

CLICAR EM OK



CARREGANDO OS DADOS NO EXCEL

The screenshot shows an Excel spreadsheet with the following columns: ORDER_APPROVED_AT, ORDER_DELIVERED_CARRIER_DATE, ORDER_DELIVERED_CUSTOMER_DATE, ORDER_ESTIMATED_DELIVERY_DATE, CUSTOMER_CITY, and CUSTOMER_STATE. The data is organized into rows, with the first row being a header and subsequent rows containing specific order and customer details.

ORDER_APPROVED_AT	ORDER_DELIVERED_CARRIER_DATE	ORDER_DELIVERED_CUSTOMER_DATE	ORDER_ESTIMATED_DELIVERY_DATE	CUSTOMER_CITY	CUSTOMER_STATE
10/01/2018 10:31	11/01/2018 14:21	15/01/2018 18:06	05/02/2018 00:00	Itaquera	SP
08/04/2018 21:20	13/04/2018 10:22	30/04/2018 16:04	16/04/2018 00:00	rio de janeiro	RJ
11/04/2017 16:32	12/04/2017 10:21	19/04/2017 14:44	03/05/2017 00:00	sao jose do turvo	SP
11/11/2017 22:33	26/11/2017 19:28	24/11/2017 19:02	06/12/2017 00:00	rio de janeiro	RJ
26/06/2018 05:37	28/06/2018 14:52	03/07/2018 18:28	25/07/2018 00:00	pianopolis	SC
05/06/2018 04:52	07/06/2018 14:25	08/06/2018 13:06	28/06/2018 00:00	poa	SP
08/07/2018 21:09	10/07/2018 14:39	11/07/2018 19:03	20/07/2018 00:00	sorocaba	SP
24/02/2018 09:07	26/02/2018 13:49	14/03/2018 13:22	21/03/2018 00:00	almeida	MG
26/06/2018 15:37	28/06/2018 14:12	04/07/2018 16:27	27/07/2018 00:00	brasil	DF
11/06/2018 21:23	17/06/2018 14:36	22/06/2018 13:52	11/06/2018 00:00	boa esperanca do sul	SC
10/06/2017 02:43	14/06/2017 13:52	10/06/2017 13:14	23/06/2017 00:00	taboao da santa	SP
08/07/2017 20:35	10/07/2017 12:47	11/07/2017 19:21	21/07/2017 00:00	santos	SP
02/02/2018 12:38	05/02/2018 22:28	02/02/2018 19:55	26/02/2018 00:00	belo horizonte	MG
26/01/2018 20:58	24/01/2018 20:36	30/01/2018 22:11	02/02/2018 00:00	sao paulo	SP
05/03/2018 16:36	04/03/2018 13:41	11/03/2018 23:08	01/04/2018 00:00	gorema	GO
16/06/2018 05:30	16/06/2018 11:47	23/06/2018 18:43	10/06/2018 00:00	paulista	PE
09/07/2017 17:08	11/07/2017 18:49	20/07/2017 19:22	01/08/2017 00:00	belo horizonte	MG
11/04/2018 21:51	17/04/2018 01:33	21/04/2018 17:10	07/05/2018 00:00	barretos	SP
08/05/2018 04:54	08/05/2018 07:18	21/05/2018 19:45	11/06/2018 00:00	gromado	RS
25/07/2017 16:30	20/07/2017 19:13	31/07/2017 17:52	10/08/2017 00:00	londrina	PR
01/06/2017 16:50	04/06/2017 17:43	08/06/2017 17:28	24/06/2017 00:00	sao paulo	SP
01/11/2017 11:11	04/11/2017 14:31	14/11/2017 13:53	30/11/2017 00:00	londrina	PR

CIDADES QUE MAIS GERARAM PEDIDOS

The screenshot shows a pivot table in Excel with 'Cidades' as the row labels and 'Contagem de ORDER_ID' as the column labels. The data is sorted by the count of orders in descending order. A red arrow points to the 'Contagem de ORDER_ID' column header.

Cidades	Contagem de ORDER_ID
sao paulo	15045
rio de janeiro	6603
belo horizonte	2657
brasilia	2073
sorocaba	1489
campinas	1406
porto alegre	1342
salvador	1188
gustafsvik	1144
sao bernardo do campo	913
ribeirao	825
santo andre	778
maraca	724
santos	700
sao jose dos campos	687
goiania	660
fortaleza	618
sorocaba	609
recife	583
joazeiro	556
londrina	548

Top 20

