# Limpeza de Dados com SQL

O principal objetivo deste projeto é mostrar meus conhecimentos na limpeza, manipulação e tratamento de dados utilizando a linguagem SQL. Abordarei os principais conceitos de limpeza de dados como: Remover Duplicatas, Valor Ausentes, Valores Nulos, Dados Inconsistentes, Formato dos Dados etc.

## Entendendo o conjunto de dados

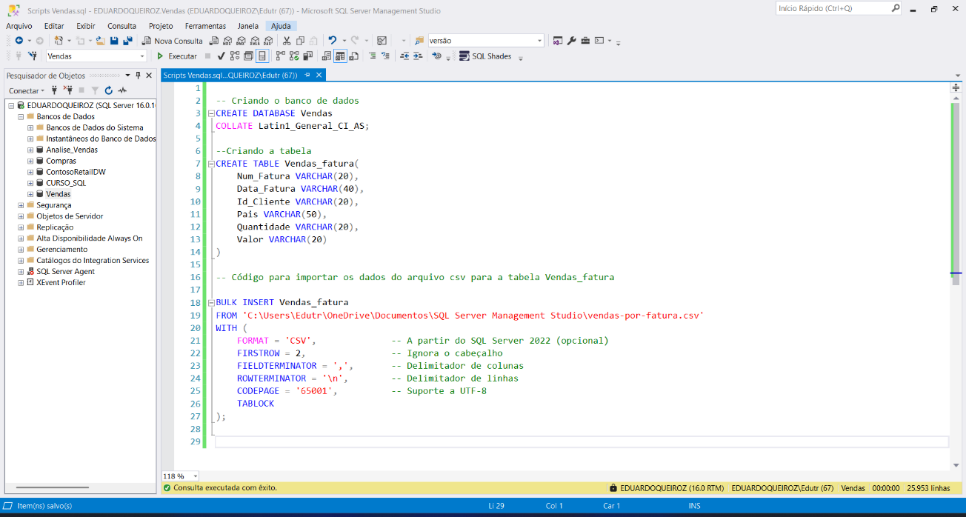
A base de dados que trataremos foi retirada do site Kaggle link(vendas-por-fatura.csv). Os dados estão armazenados originalmente em uma única tabela no formato *csv,* importei os dados para o SGBD Microsoft SQL Server onde iremos fazer o tratamento e limpeza dos dados com o SQL.

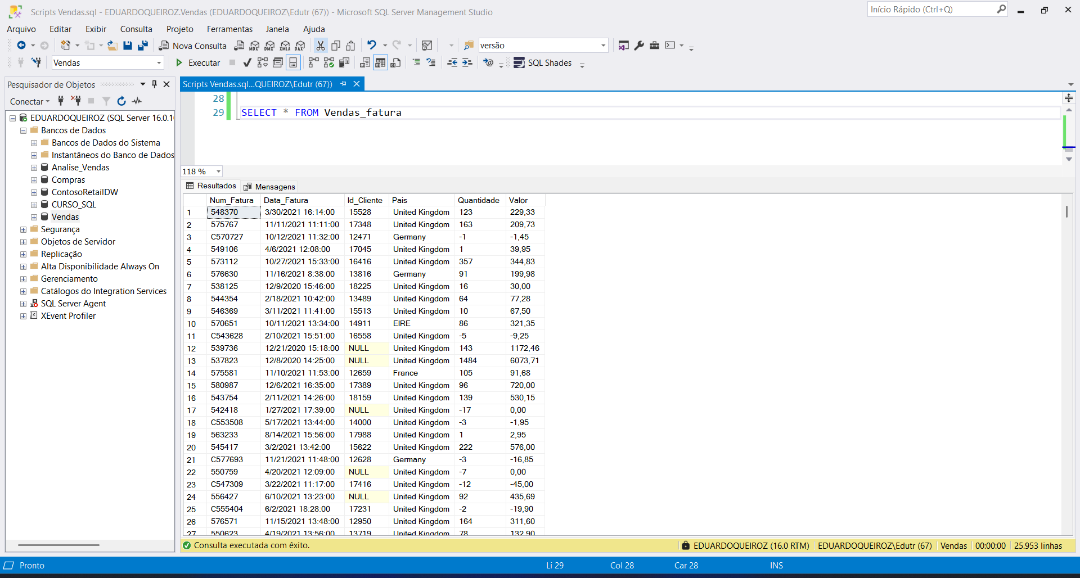
Essa base contém registros de vendas de uma empresa de ecommerce sediada no Reino Unido, com informações detalhadas por fatura individual. Cada linha representa uma transação ou item vendido dentro de uma fatura.

Descrição das colunas da tabela vendas-por-fatura:

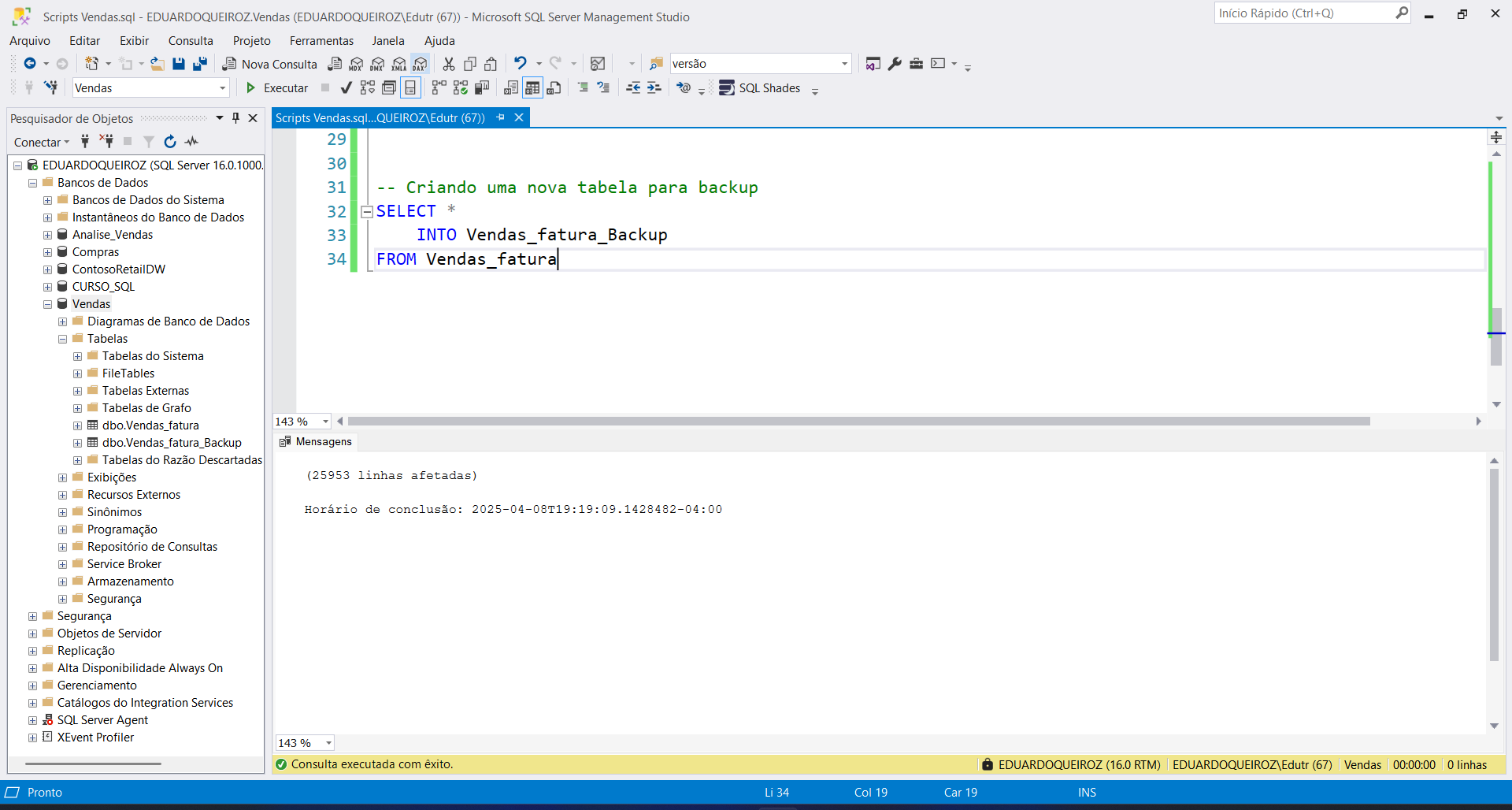
* **Num\_fatura:** Um número atribuído exclusivamente a cada transação. Se começar com "c", indica uma substituição.
* **Data\_fatura:** Data e hora da fatura, dia e hora em que a transação foi gerada.
* **ID\_Cliente:** Número de clientes. Um número integral de 5 dígitos atribuído exclusivamente a cada cliente.
* **País:** O nome do país onde cada cliente reside.
* **Quantidade:** As quantidades de cada produto (item) por transação.
* **Valor:** Valor da fatura em libras esterlinas.

## Importação dos Dados

Como os dados estão no formato CSV, vamos importá-los para o SQL Server para trabalharmos com eles. Primeiro, criaremos um banco de dados chamado **Vendas**. Em seguida, criaremos a tabela **Vendas\_fatura** para armazenar os dados provenientes do arquivo CSV. Durante a criação do script SQL, os tipos de dados foram definidos como **VARCHAR**, ou seja, do tipo texto, para evitar erros durante a importação.

Visualizando os dados inseridos:

## Backup dos dados

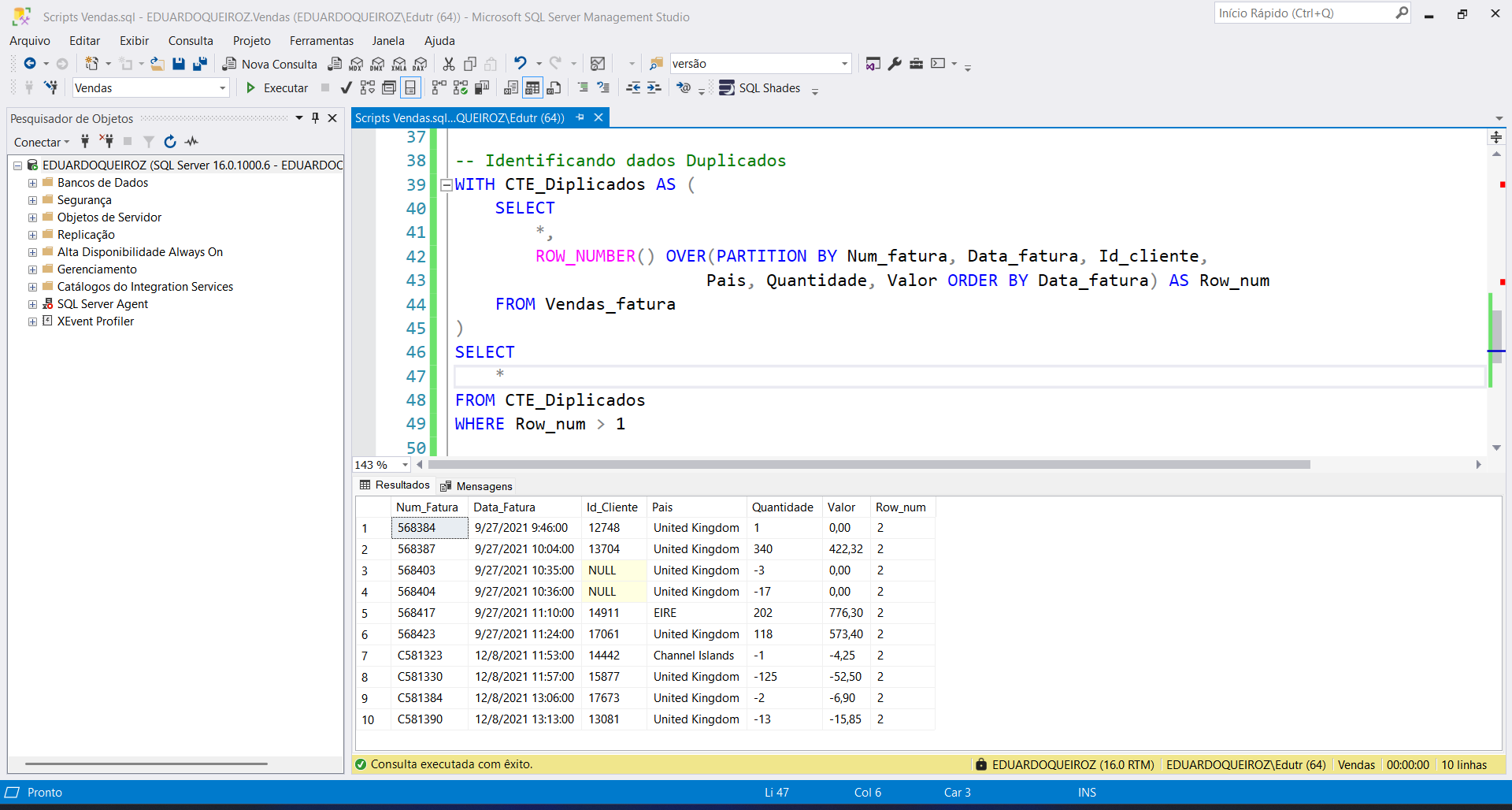
Para garantir que possamos trabalhar com segurança e evitar erros como deletar dados ou inserir informações equivocadas, clonaremos a tabela **Vendas\_fatura**. Em seguida, criaremos uma tabela adicional que servirá como backup, caso precisemos restaurar algum dado no futuro.

Agora que concluímos a etapa de entendimento, preparação e importação dos dados, é hora de colocar a mão na massa e iniciar a limpeza e tratamento dos dados propriamente ditos. Mas antes criarei um passo a passo de como será realizada a transformação da base de dados

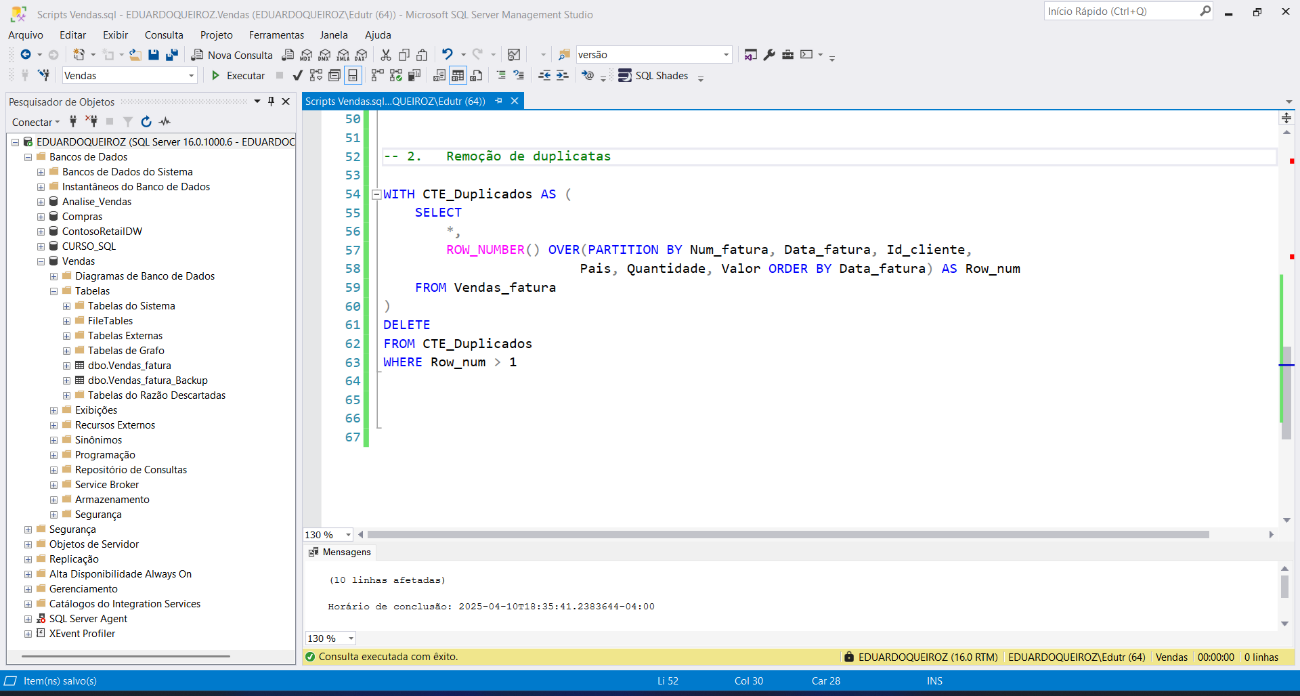
1. Remoção de dados duplicados
2. Remoção linhas com dados ausentes
3. Remover valores negativos
4. Limpeza coluna *num\_fatura*
5. Limpeza coluna *data\_fatura*
6. Padronizando os tipos de dados corretos das colunas

### Remoção de dados duplicados

1. **Analisando dados duplicados:** Primeiro passo, analisar as linhas duplicadas na tabela. Vamos usar uma consulta SQL com CTE e a função de janela Row\_Number para encontrar registros que tenham valores duplicados em uma ou mais colunas.

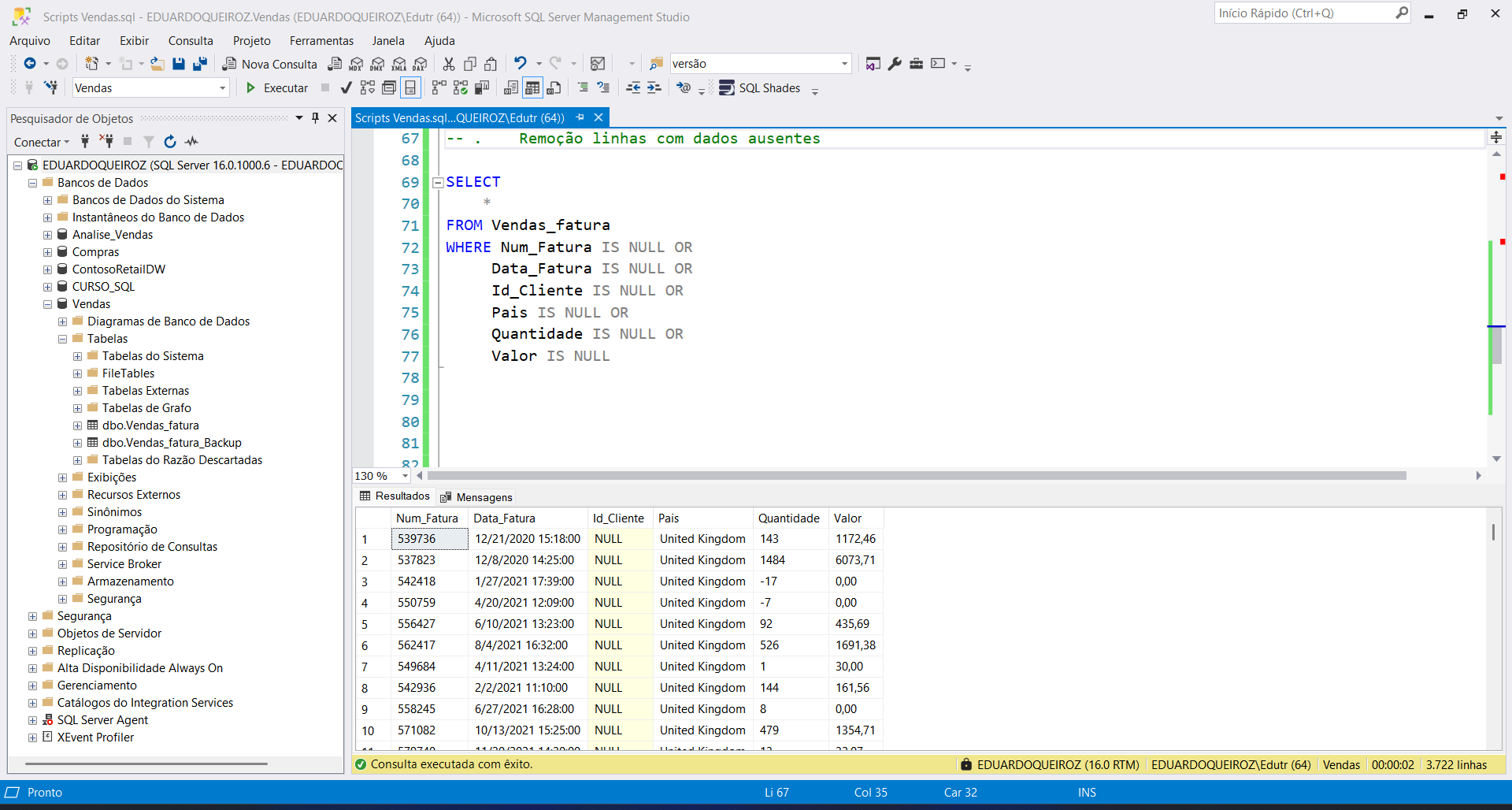


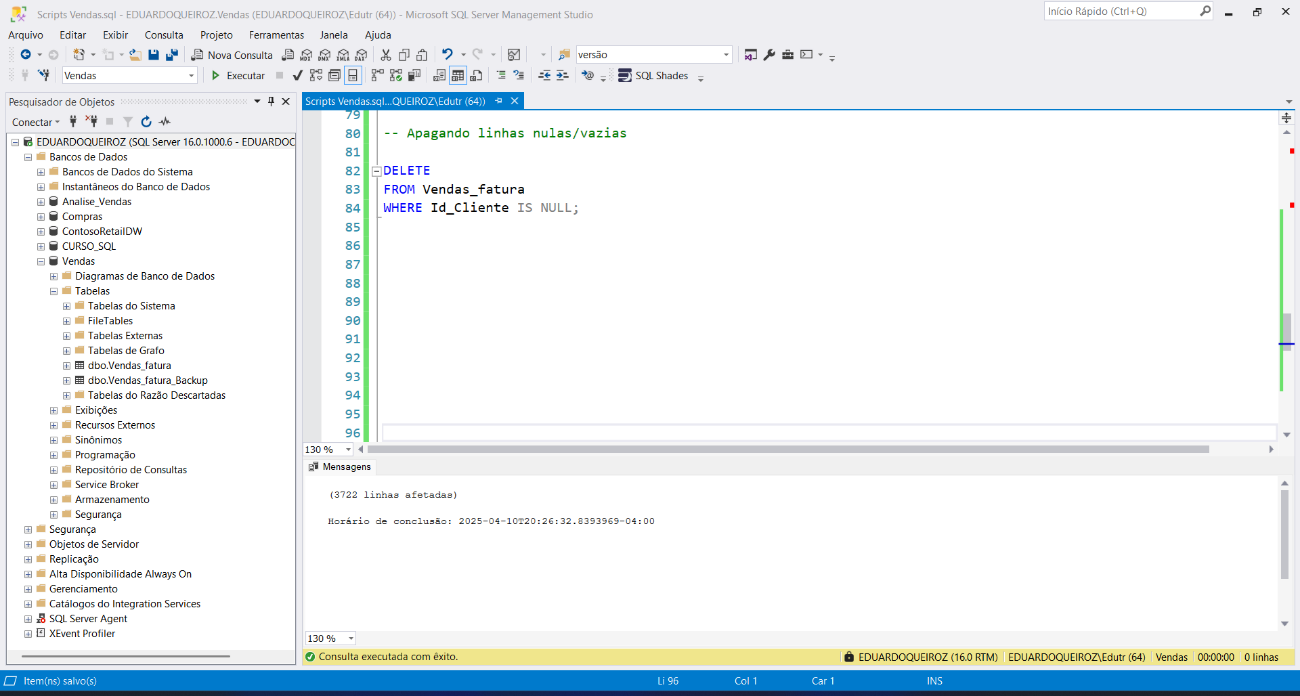
1. **Remoção de duplicatas:** Depois de identificar as duplicatas, agora removeremos essas linhas da tabela.



### Remoção linhas com dados ausentes

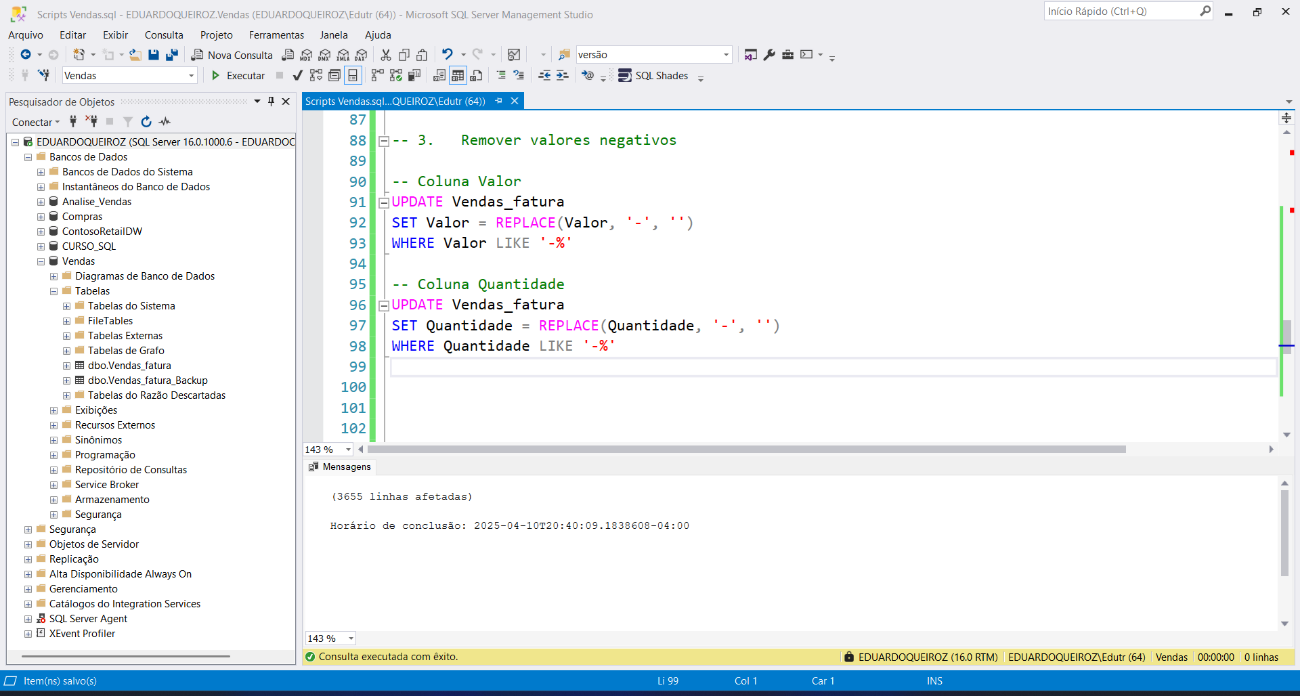
Para identificar todas as linhas com registros nulos nas colunas da tabela Vendas\_fatura, usaremos a seguinte consulta SQL:



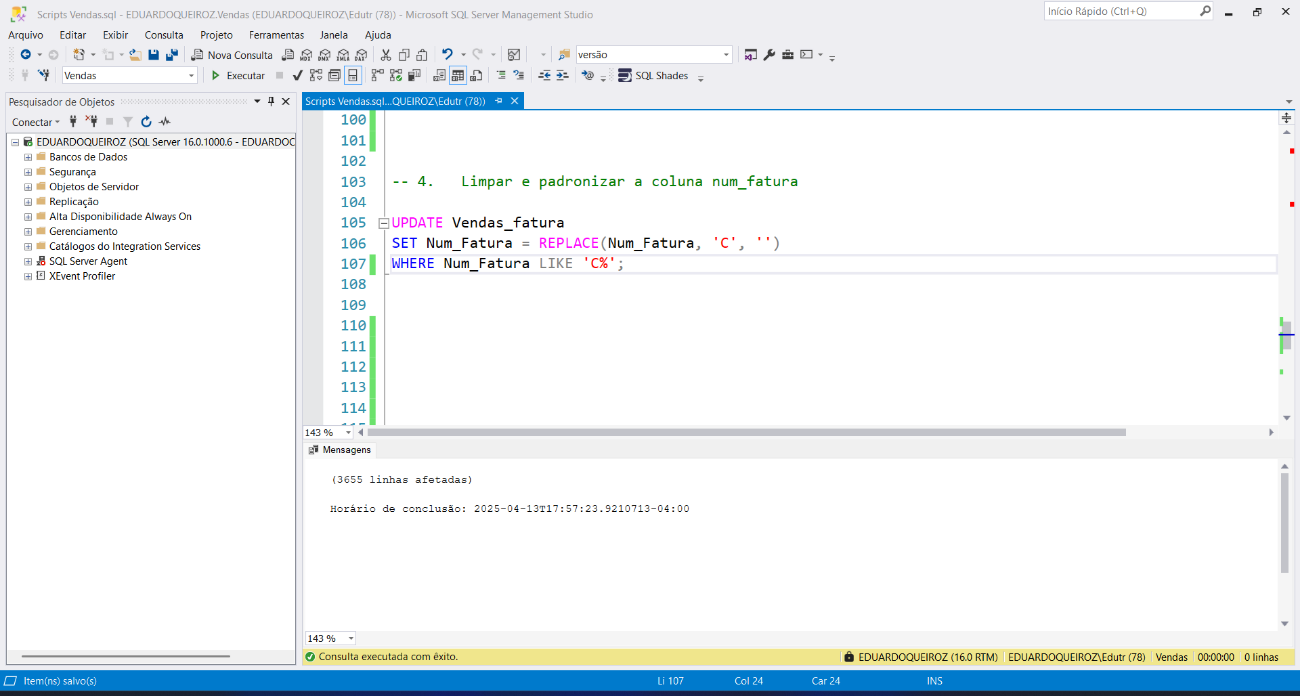
Ao analisar o resultado da consulta, observamos que os valores nulos pertencem à coluna id\_cliente. Isso impede a identificação dos clientes. Portanto, decidi excluir todas as 3722 linhas que contêm valores nulos.

### Remover valores negativos

As colunas que possuem valores negativos são Valor e Quantidade. Como o tipo de dados dessas colunas é texto, podemos utilizar a função Replace para substituir o sinal de menos.

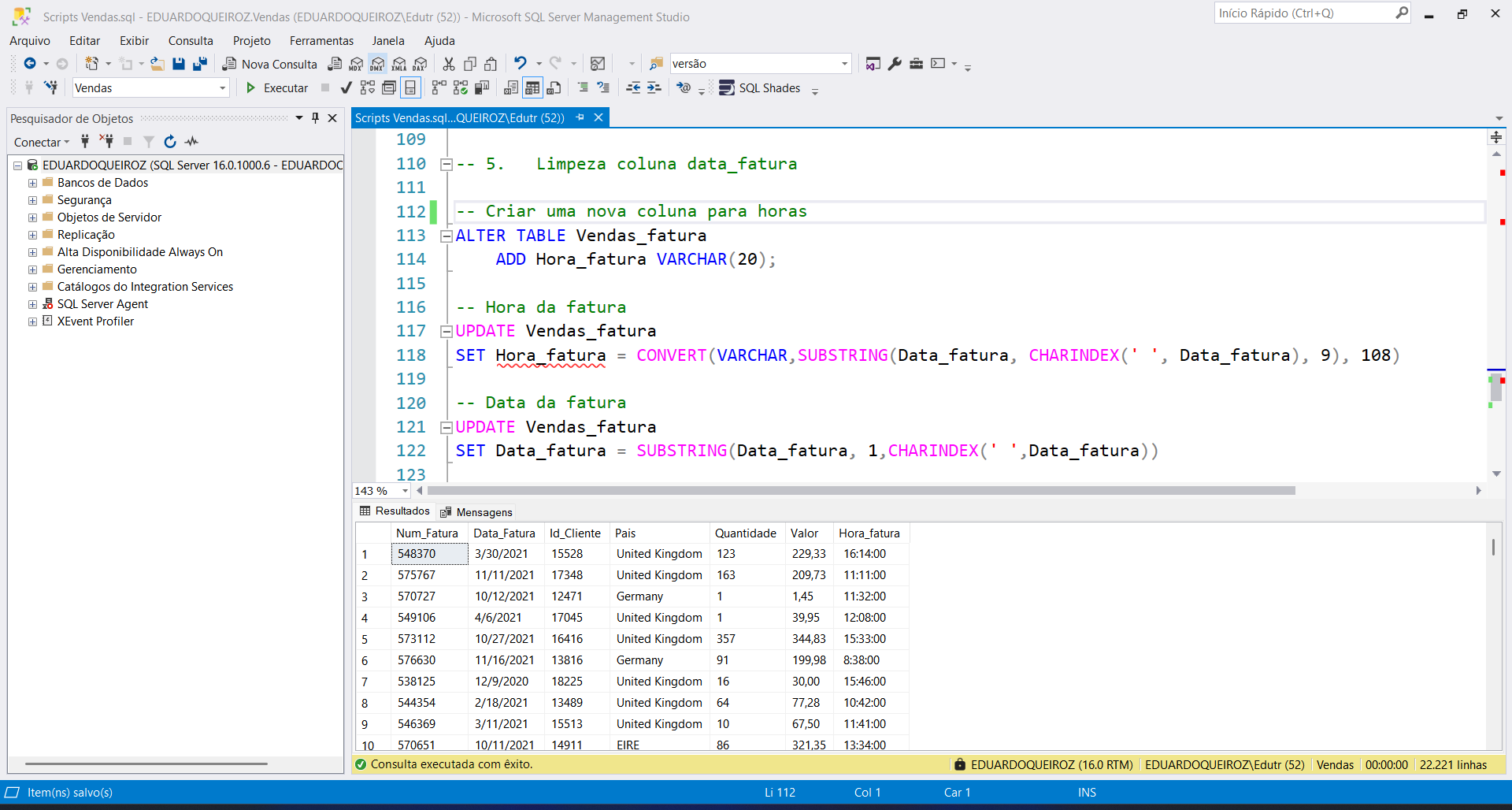


### Limpar e padronizar a coluna *num\_fatura*

Analisando a coluna *num\_fatura*, observamos a presença de caracteres não numéricos. É necessário remover esses caracteres para garantir que o tipo de dado correto para essa coluna seja numérico.

### Limpeza coluna *data\_fatura*

A coluna data\_fatura contém dois tipos de dados: data e hora. Embora eu pudesse manter apenas o tipo data, decidi criar uma coluna para armazenar exclusivamente a hora.



### Padronizando os tipos de dados corretos das colunas

Depois de realizar a limpeza dos dados, padronizei o tipo correto de dados de cada coluna para garantir a integridade e a consistência das informações. Esse processo envolveu a conversão de tipos de dados para formatos apropriados, como transformar colunas de texto em valores numéricos ou datas, conforme necessário. Além disso, foi essencial remover caracteres não numéricos e ajustar os formatos de data e hora para facilitar a análise e manipulação dos dados.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.