# Data Warehousing report for Q100 Waste data

By Amit

This report is to to provide the analysis, evaluation of the design, creation, robustness of the model and functionality of the data warehouse made, the associated queries and the visualisation dashboard.

Two similar datasets were given from the *Q100 Waste data for: WasterDataFlow - Local Authority waste management published by Departemnt for Enviroment, Food and Rural Affairs* for 2015 - 2016 and 2016 - 2017.

Resources used: [Link for data from 2015 to 2016](http://data.defra.gov.uk/Waste/Q100_data_England_2015_2016.csv)

[Link for data from 2016 to 2017](http://data.defra.gov.uk/Waste/Q100_data_England_2015_2016.csv)

**Table of Contents:**

1. *Designing and Reasoning*
2. *Setting up the Facts and dimention tables in MS SQL through T-SQL*
3. *Cleaning the data*
4. *Setting up the staging tables*
5. *Performing ETL operations*
6. *Further required manipulations*
7. *Asked queries executions*
8. *BI Dashboard*

*Designing and Reasoning*

*------------------------------------*

Analysing the raw data from the website, the data from 2015 - 16 had 31 attributes and the data from 2016 - 17 had 35 attributes. On further look it was found that majority of the attributes were functionally dependent on only few attributes. This makes the a point to go for normalising the data I.e. opting for Snowflake-Schema but going for snowflake schema also comes with increase the transaction processing time as we have to many joins to get the result and also makes a query complex.

In the end, the choice to go Star-Schema was made as the the many different attributes had to be accounted for simultaneosly in the asked queries but were simple operations. So Star schema here provides simplicity and a performance in these circumstances.

The specifications of the data warehouse is it would have 2 dimension tables namely “main” and “secondary” and a fact table name “facts”.

The division of attributes are:

Looking at the queries to process, they deal with periodic properties of the attributes and the total tonnes of waste produced for particular Aurthorities, Materiaals and other categorical attributes.

The facts table handle the Total tonnes part, secondary table world store the periodic attributes of the dataset, And the various other named informations like Authorities, Factory names, material group and id is stored in the main table.

For the main table the composite primary key chosen is (Material Id, Waste Processor Id, Waste Processor Output Id). It would store the major portion of the data.

For the secondary table th composite primary key I s(Waste Stream id, Sender Waste Processor Id).

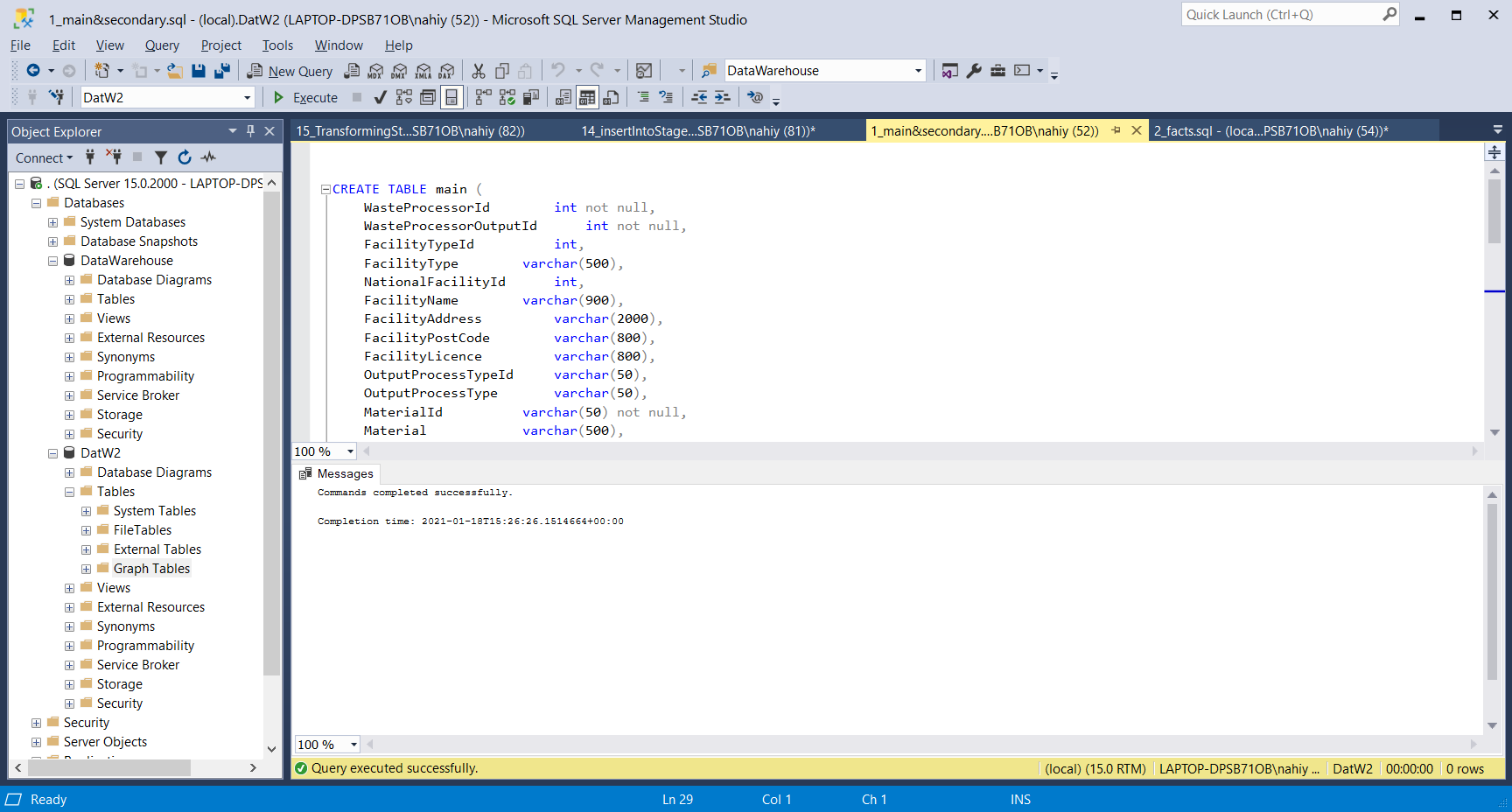
The facts table has references to both the table by the primary key and the TotalTonnes attribute is the measure field required.

*Setting up the Facts and dimention tables in MS SQL through T-SQL*

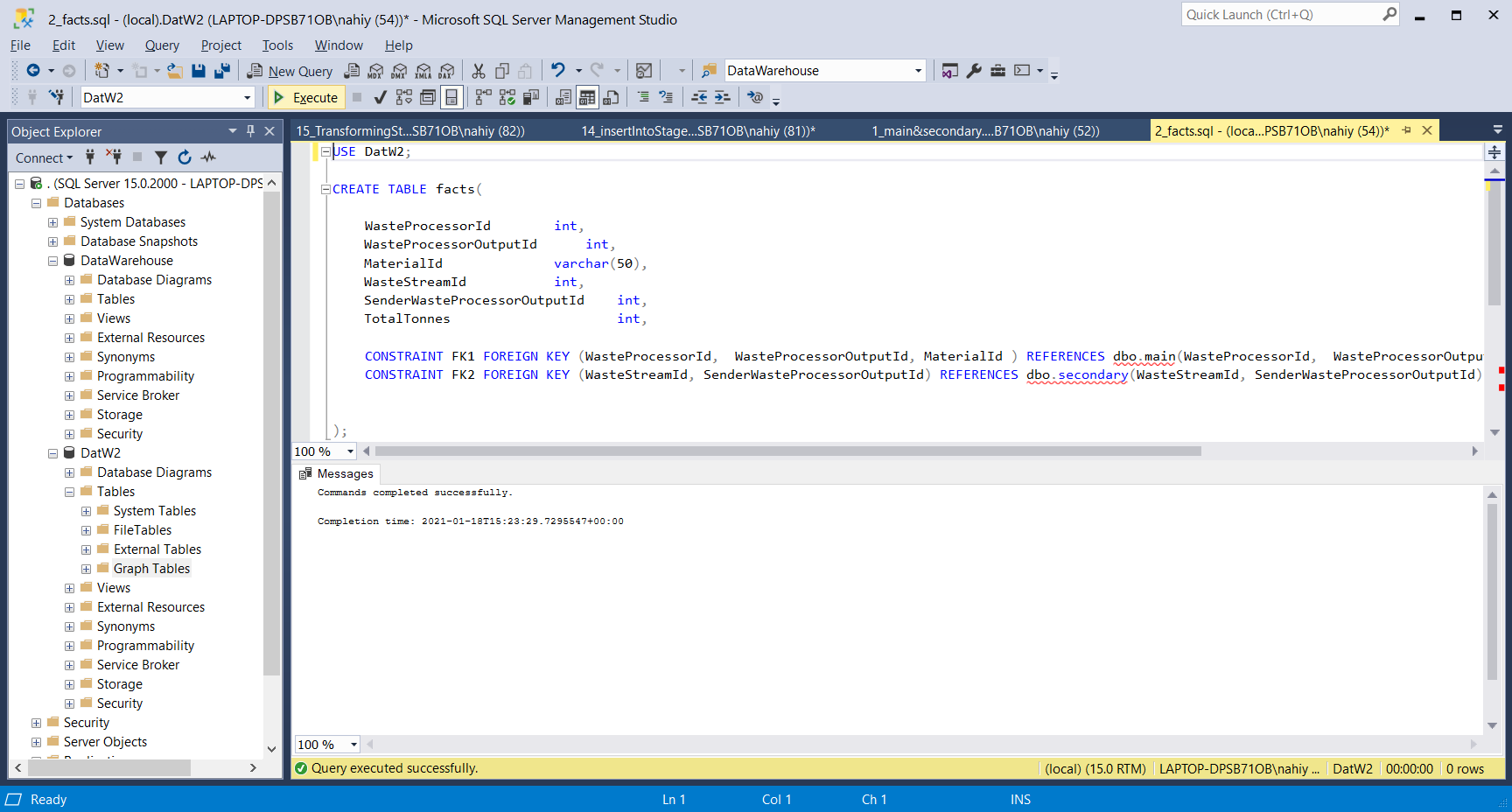
*-----------------------------------------------------------------------------------------*

1. SQL queries executed ->

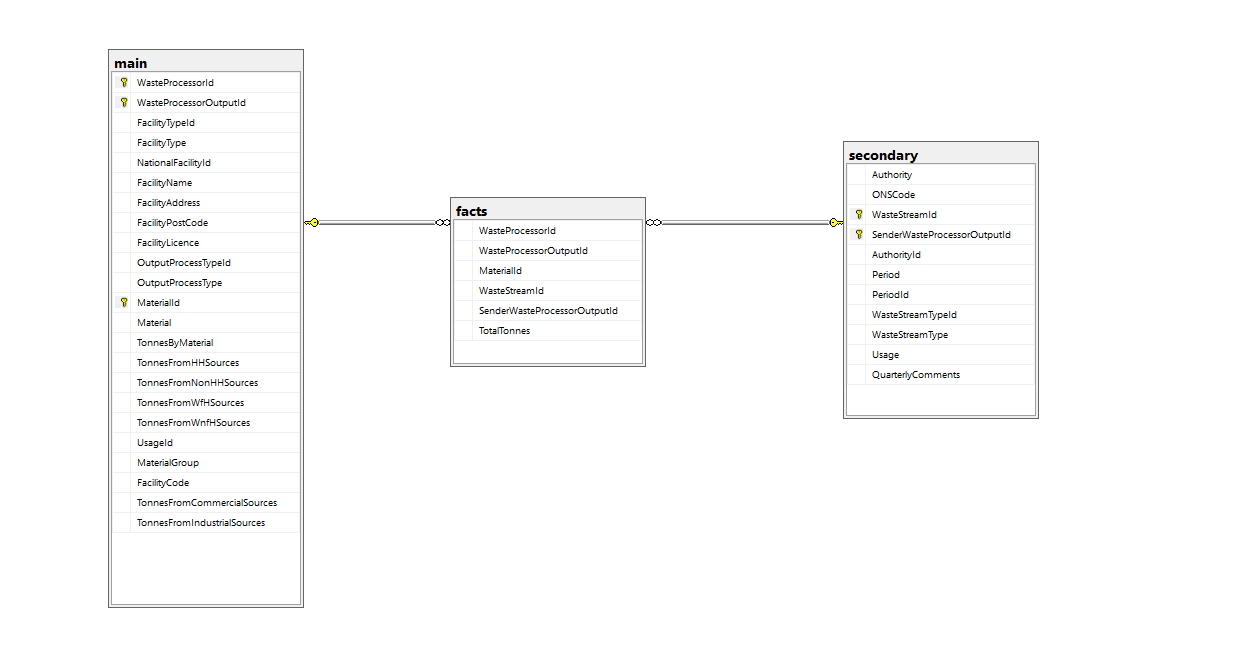
For main & secondary table:



For facts table:



Schema Diagram:

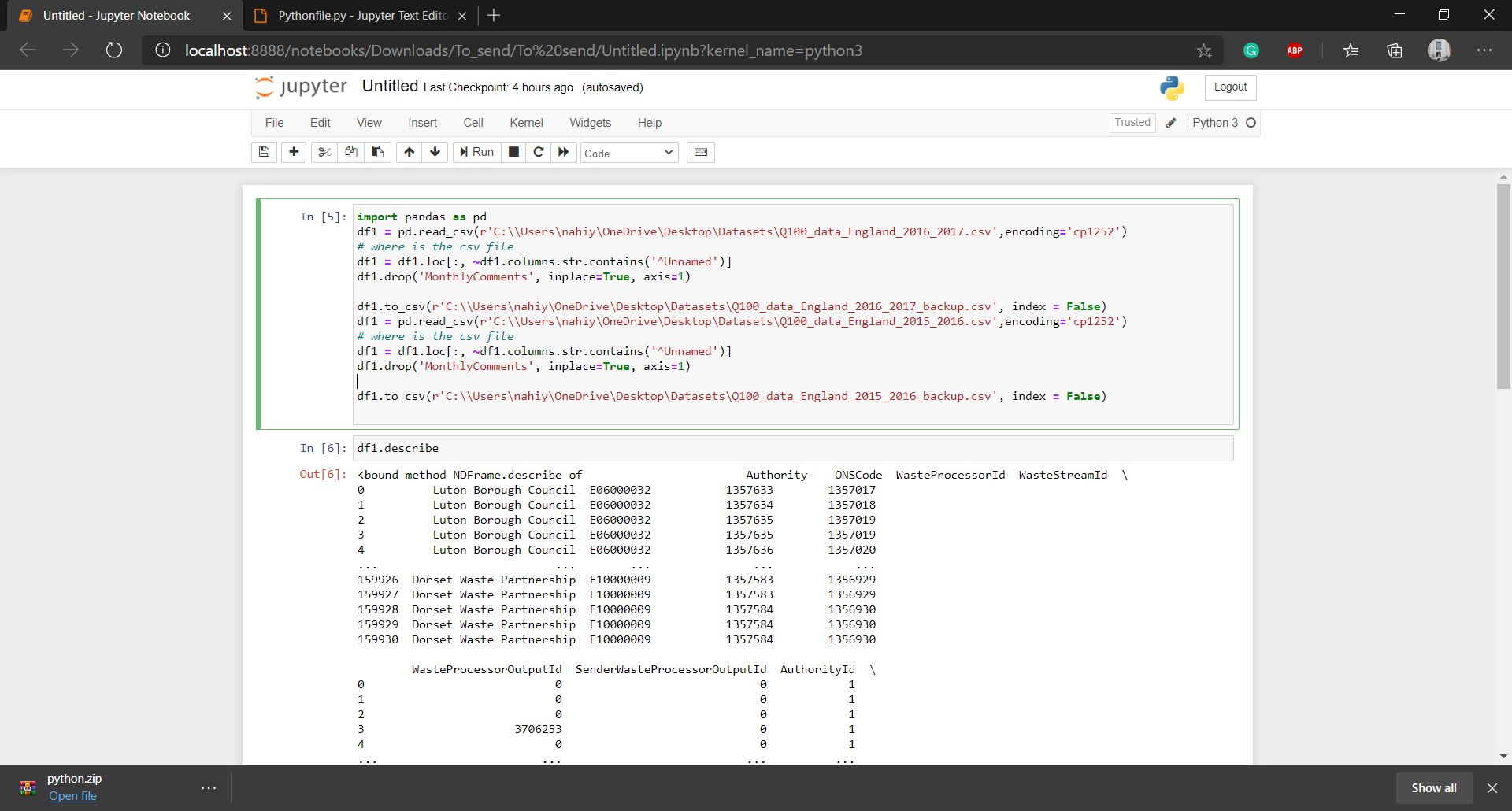


*Cleaning the data*

*------------------------*

As the file is .CSV file it is read by SQL having comma as an Field Terminator, So any commas are un desriable inside an attribute. Also there is new line characters present, i.e. used as row terminators so that is also not wanted. The monthly comments attribute has both of thoise things and also more some of its value has above 3000 characters. And there is an unnamed attribute in the data that needs cleaning.

Used Python to get rid of the MonthlyComments and the Unnamed attribute. As followed…



To get rid all of the commas in the attribute, used MS Excel to Find ‘ , ‘ and replaced it with ‘-’. This is all the data cleaning we do outside of MS SQL.

*Setting up the staging tables*

*---------------------------------------*

For staging, 6 tables are used…

1. **data2015\_2016**:

To BULK INSERT or Import from the Q100 Waste Data 2015 to 2016 dataset

1. **data2016\_2017:**

To BULK INSERT or Import from the Q100 Waste Data 2015 to 2016 dataset

1. **MERGE2015\_2017:**

To insert all the data from the data2015\_2016 and data2016\_2017 table.

1. **Stage\_main:**

To stage the data for the main table.

1. **Stage\_secondary:**

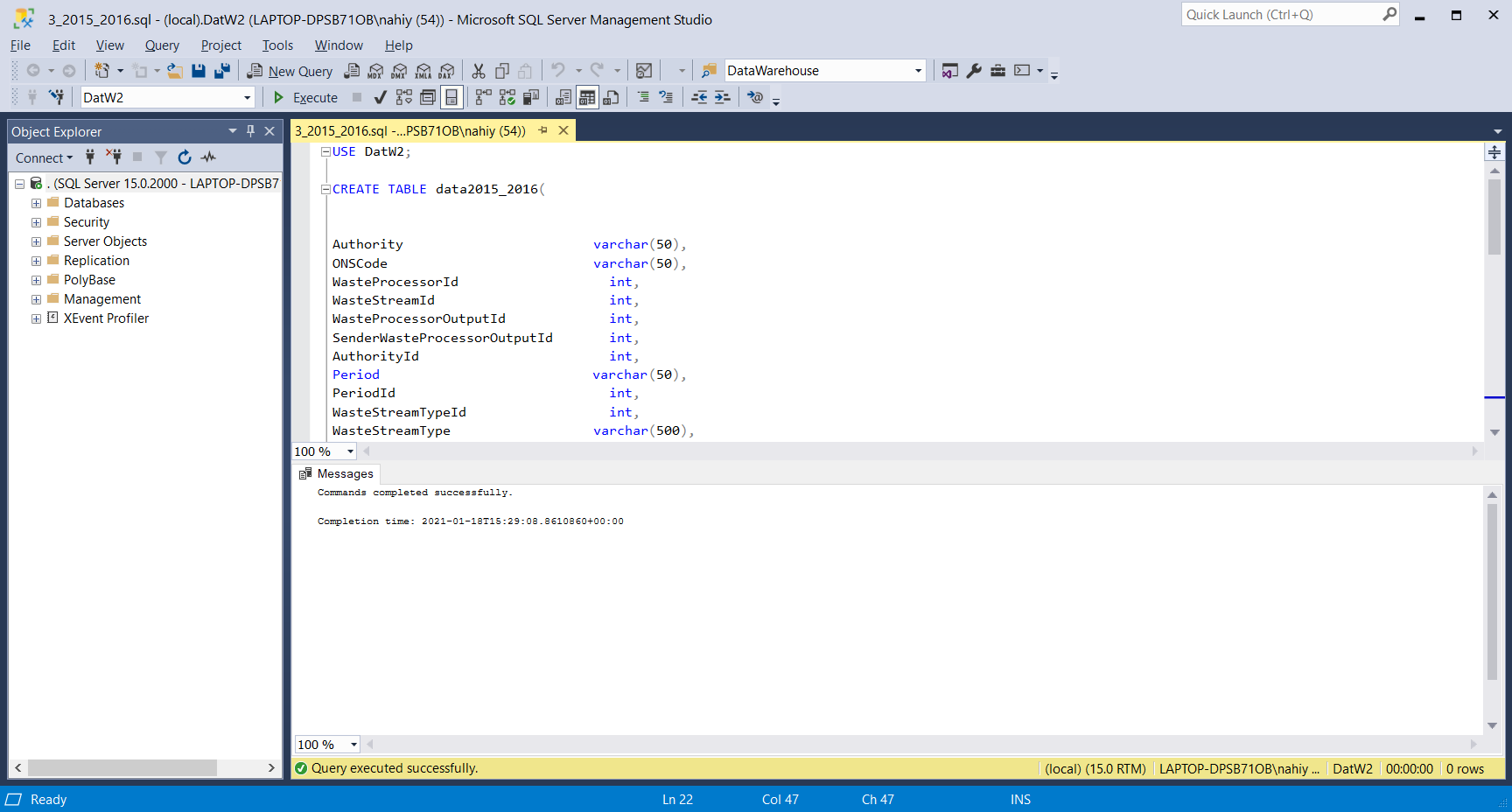
To stage the data for the secondary table

1. **Stage\_facts:**

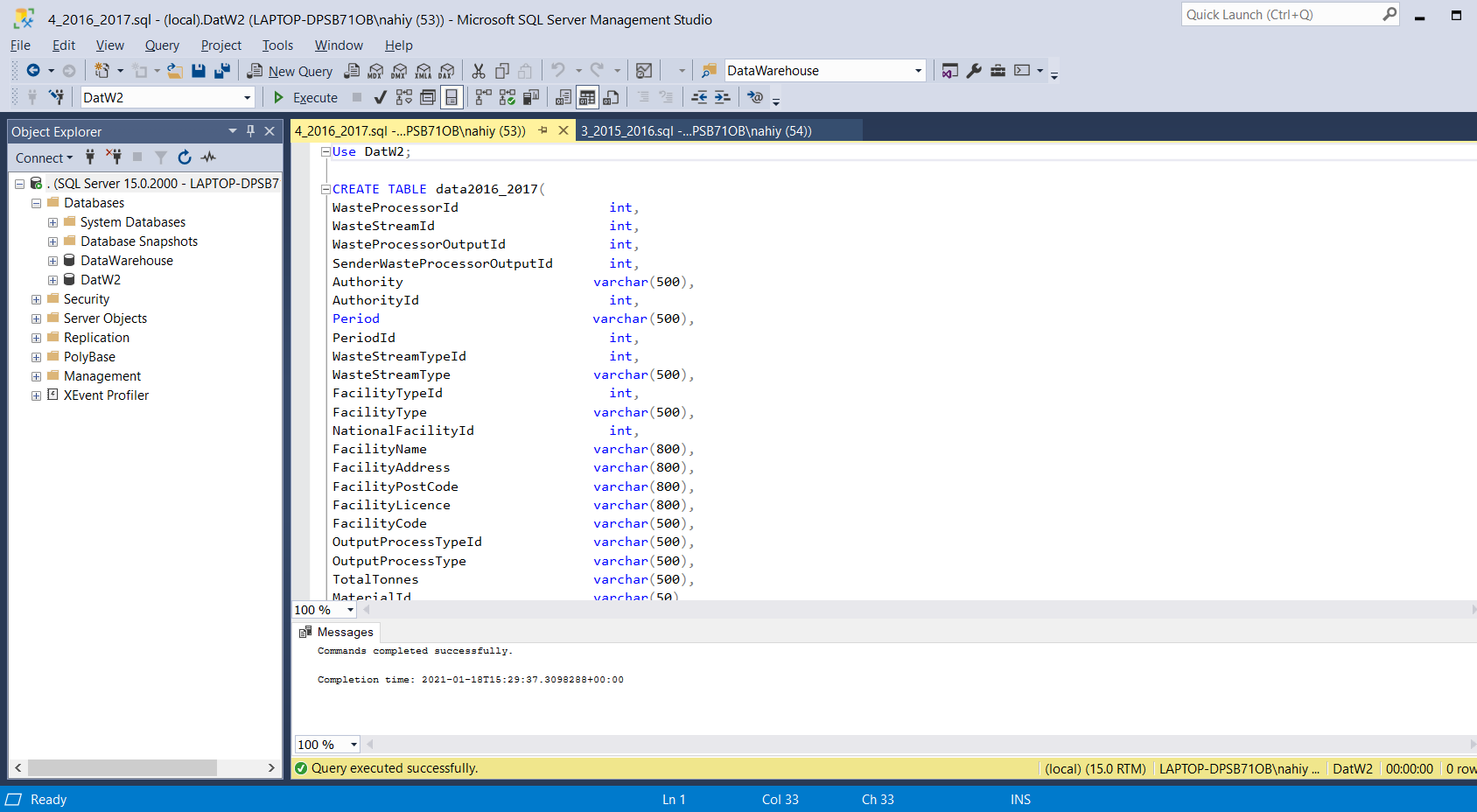
To stage the data for the facts table

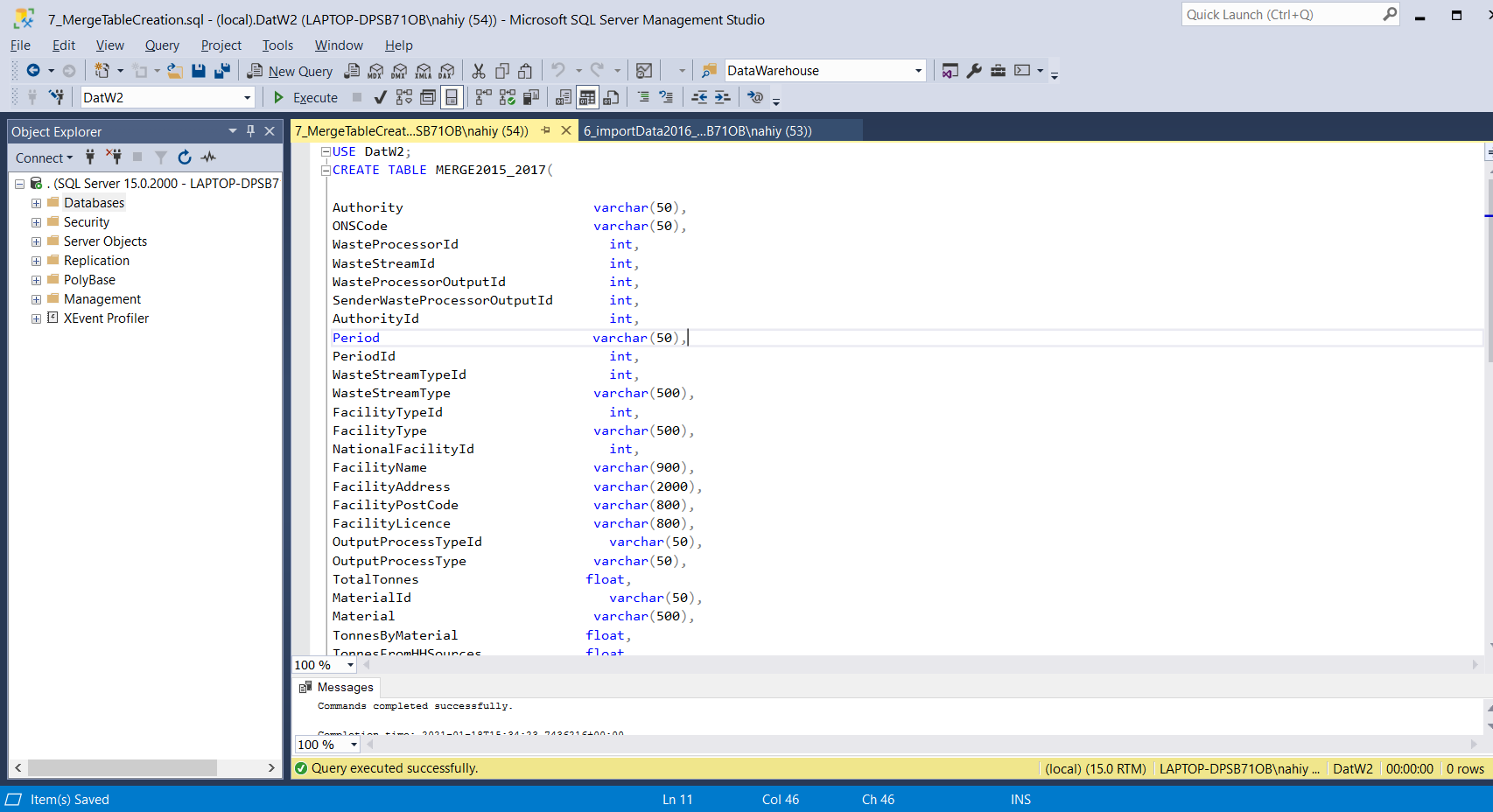
**SQL queries screenshots of creations->**

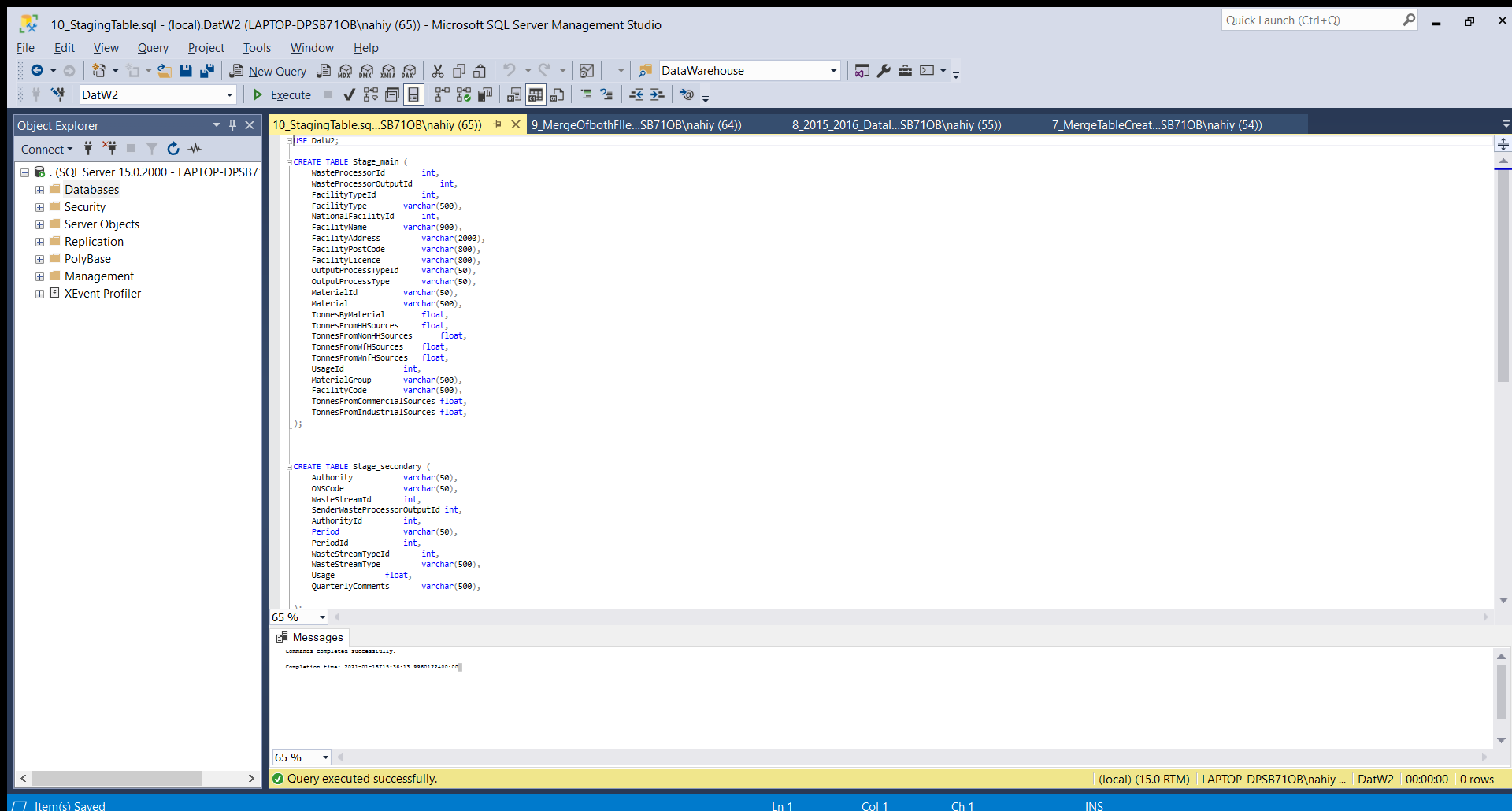
1. **data2015\_2016**:



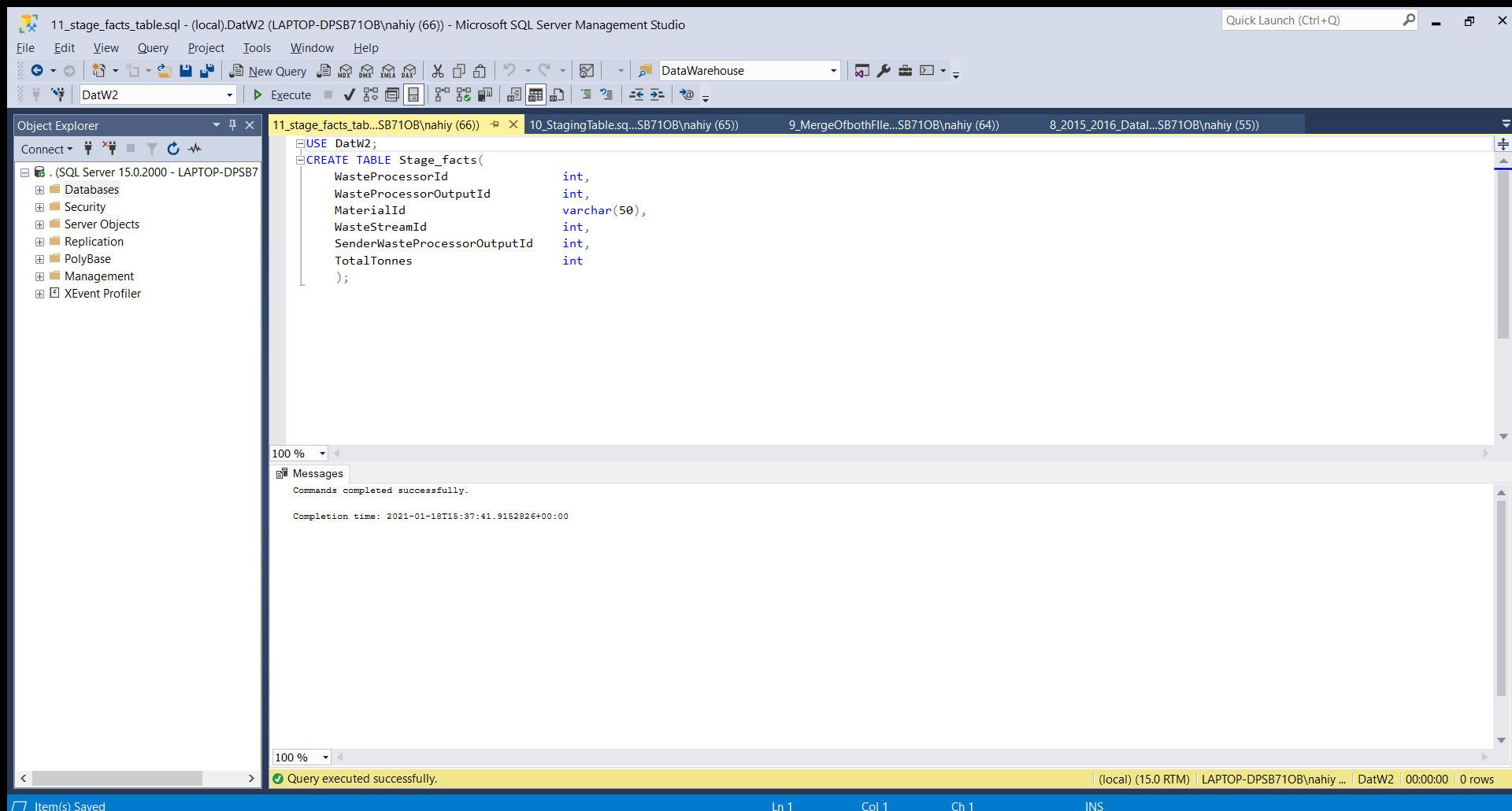
1. **data2016\_2017:**

****

1. **MERGE2015\_2017:**
2. **Stage\_main &**
3. **Stage\_secondary:**

****

1. **Stage\_facts:**



*Performing ETL operations*

*------------------------------------*

For Extraction: using BULK INSERT, the data is inserted to.

Deciding the parameters…

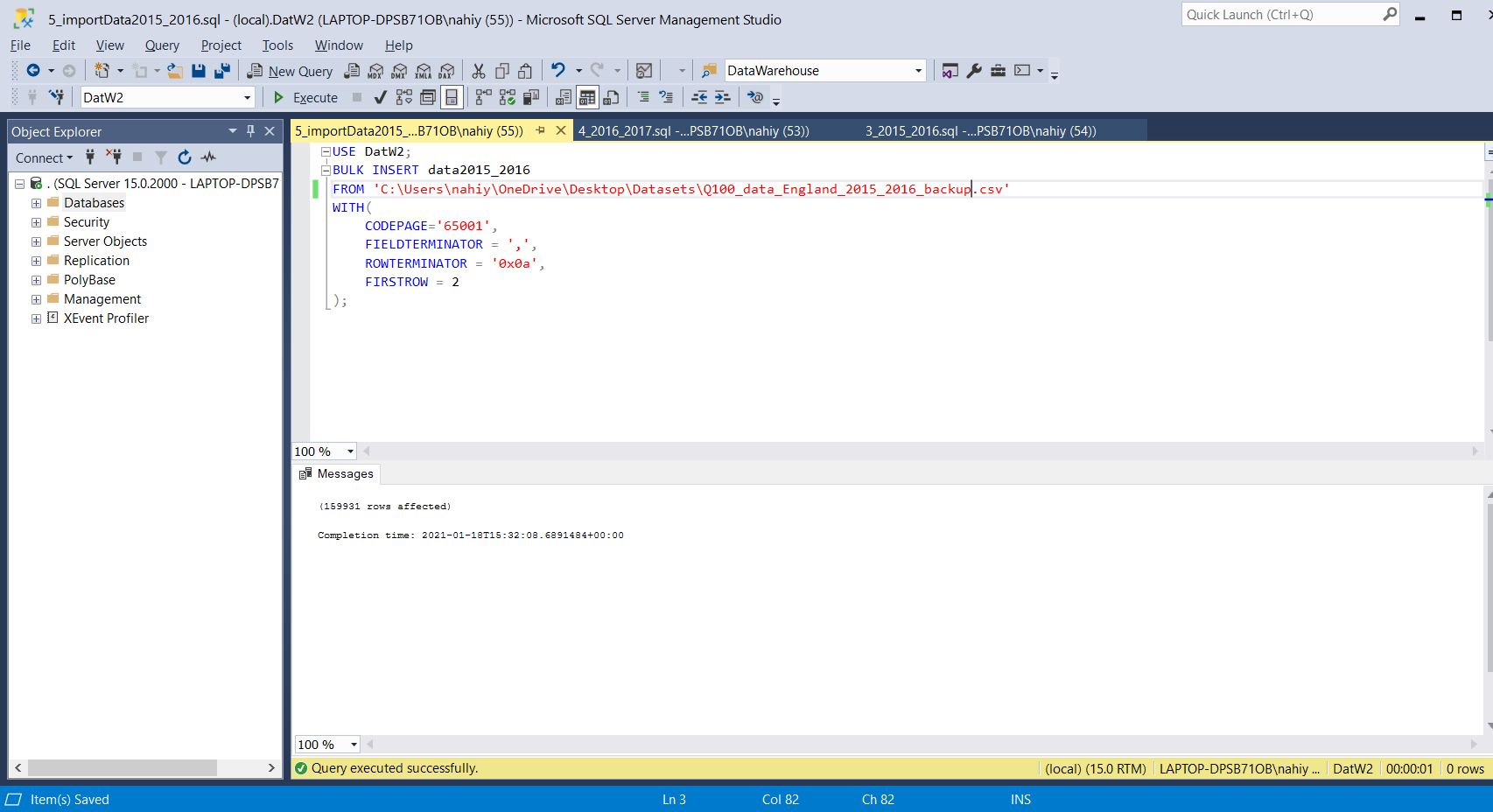
As it is csv file, FieldTerminator is set to ‘ , ’ to get the attributes in a row.

The rowterminator show be set to a new line character.

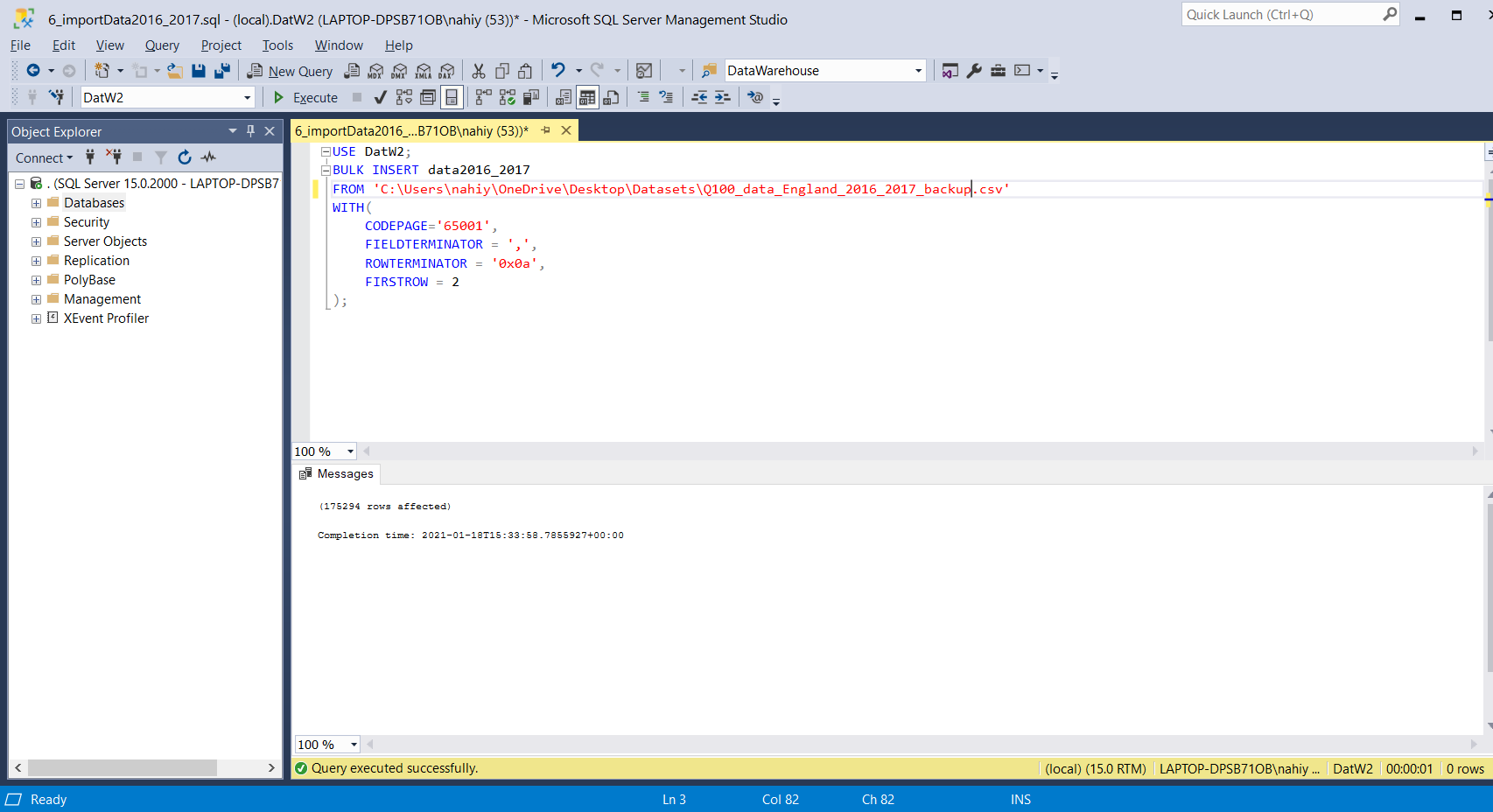
Since the actural data starts with starts with second row(First row is the columns) FirstRow is set to 2.

EXTRACTION…

EXTRACTING THE Q100 Waste data for 2015 to 2016 to the staginf table **data2015\_2016**:

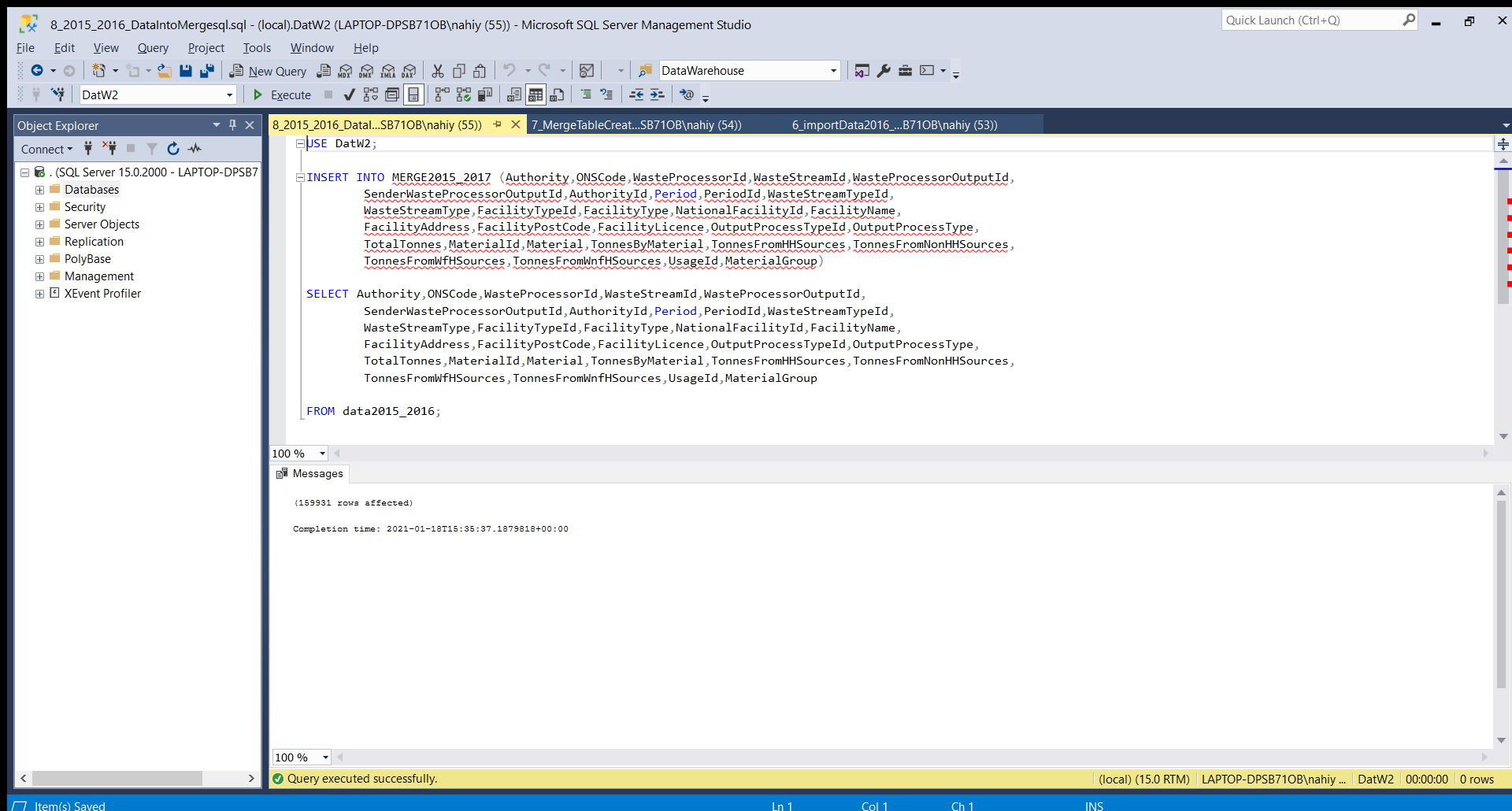


EXTRACTING THE Q100 Waste data for 2016 to 2017 to the staginf table **data2016\_2017**:

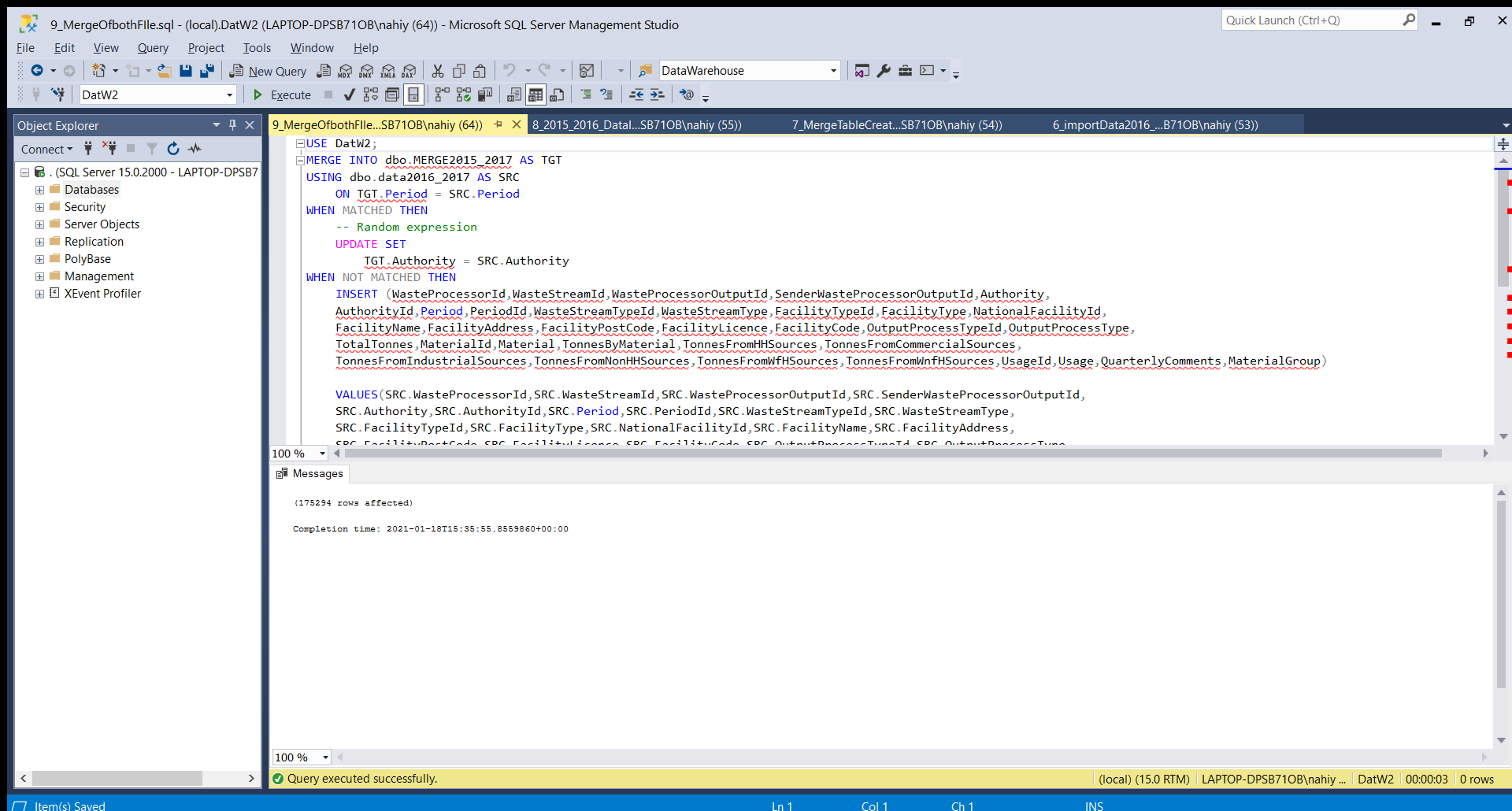


FURTHER STAGING BY STORING DATA TO FACILITATE FURTHER OPERATION…

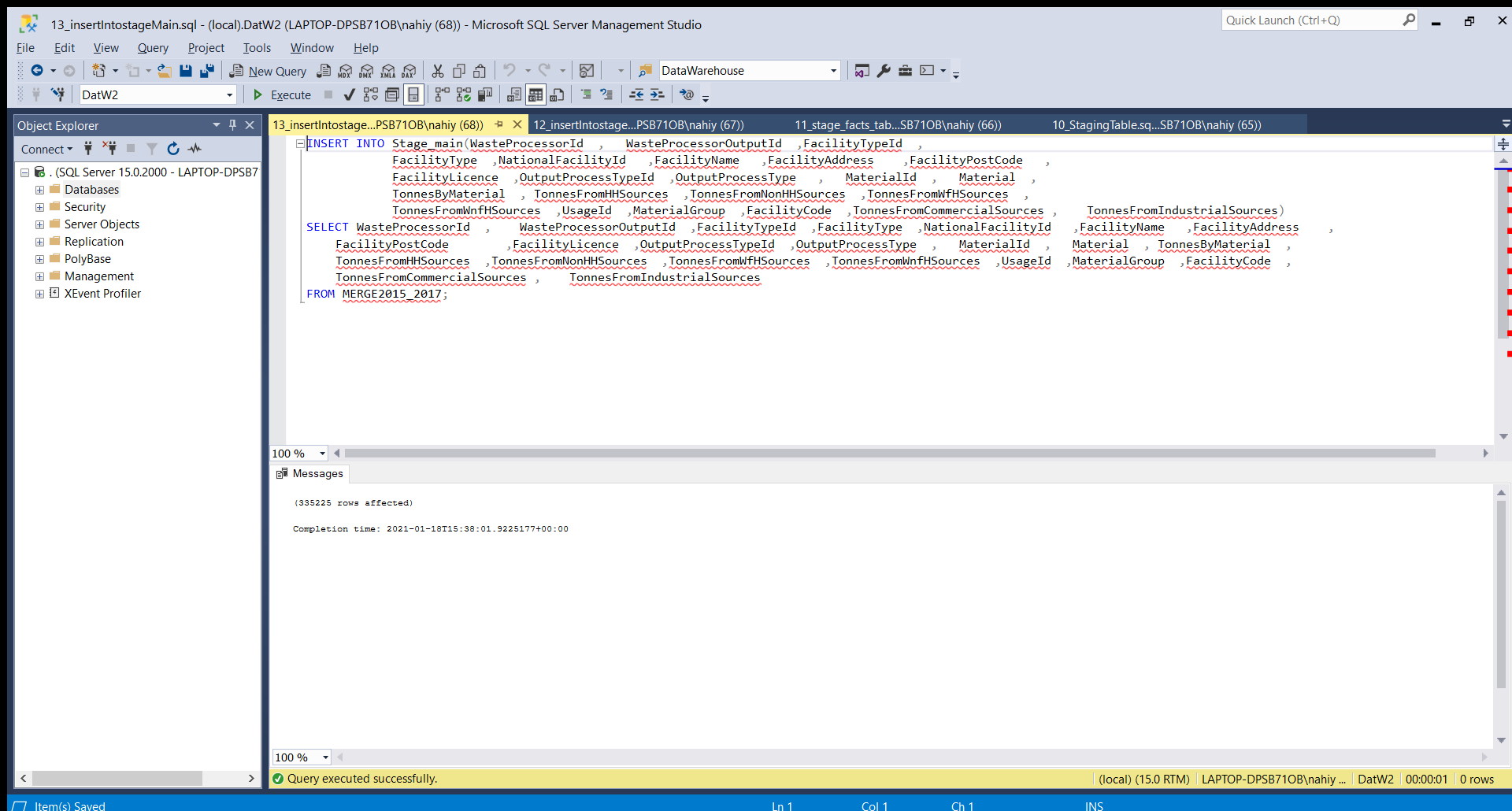
INSERTING all the data from the data2015\_2016 table to the **MERGE2015\_2017:**

****

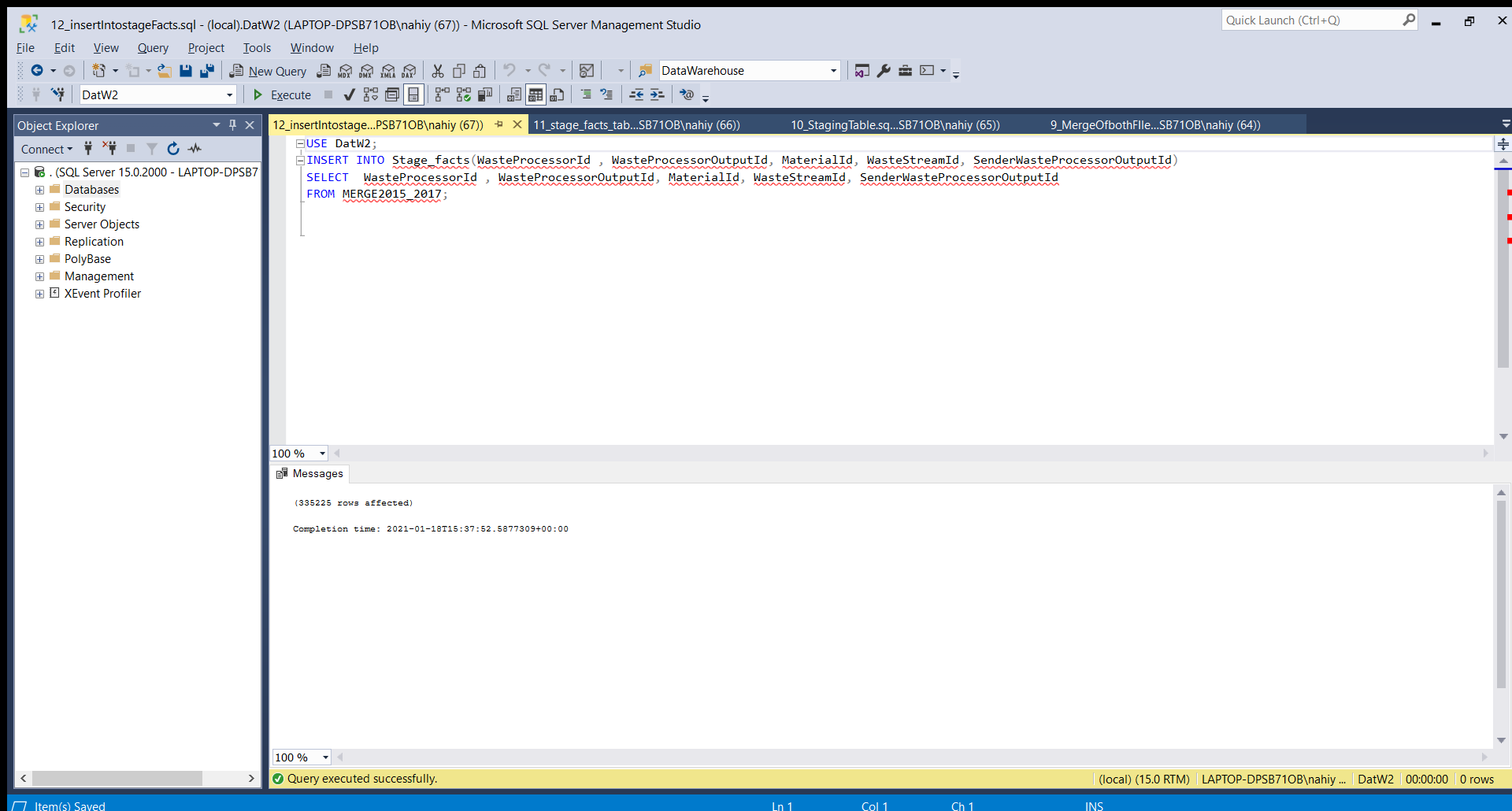
**MERGING** all the data from the data2016\_2017 table to the **MERGE2015\_2017:**

****

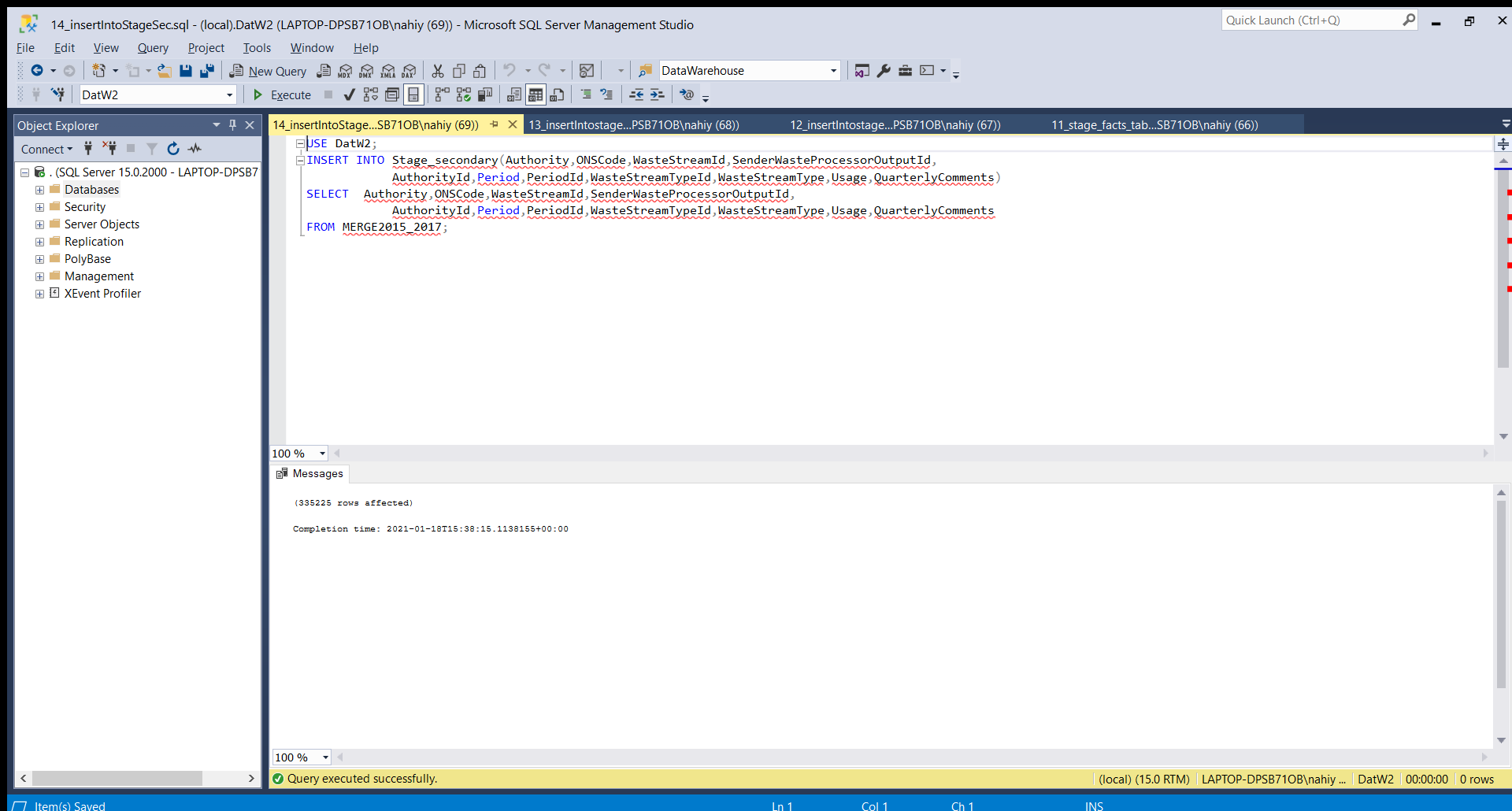
INSERTING the required data in appropriate attributes from MERGE2015\_2016 to **Stage\_main**:



INSERTING the required data in appropriate attributes from MERGE2015\_2016 to **Stage\_facts**:

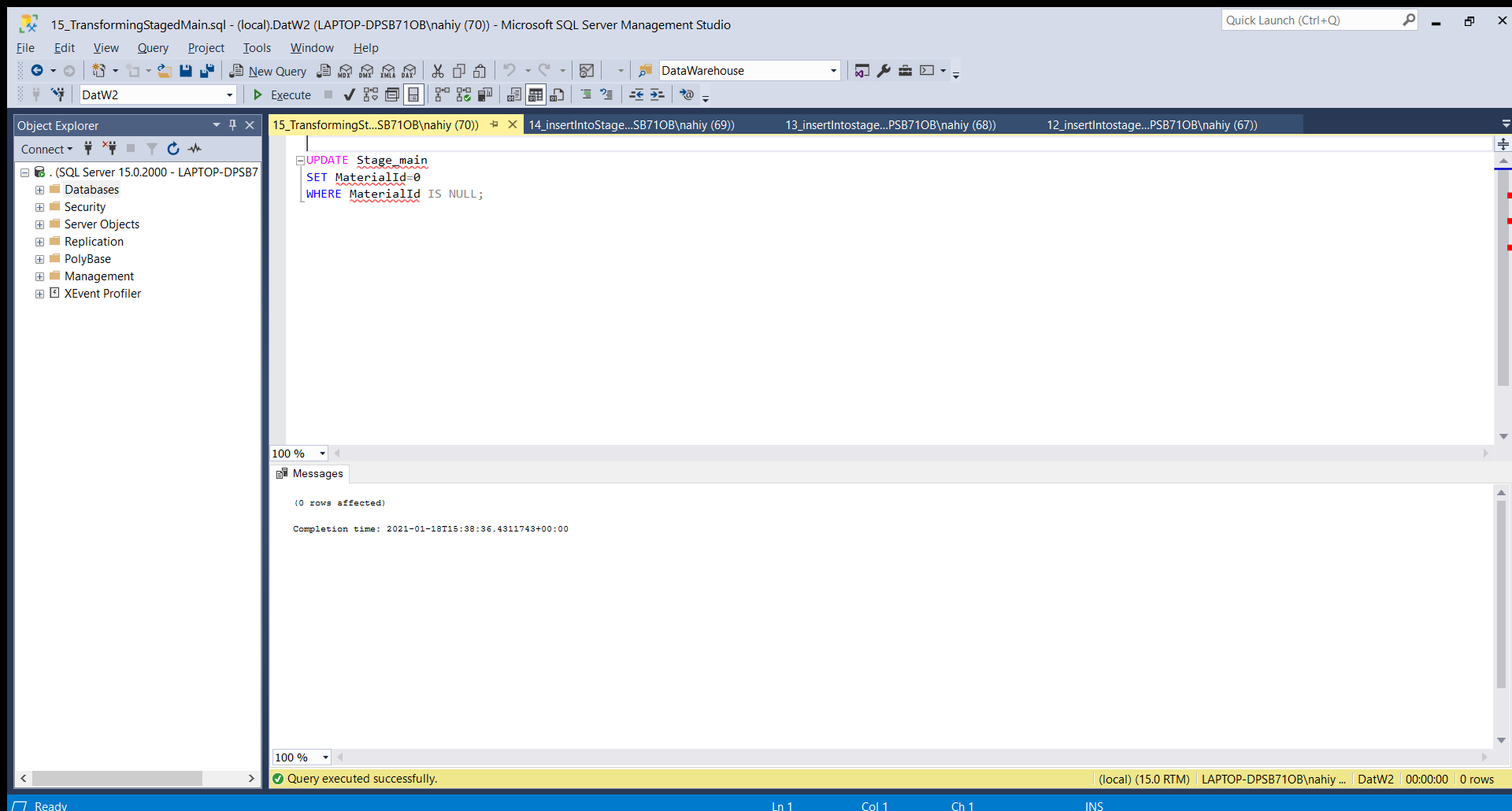


INSERTING the required data in appropriate attributes from MERGE2015\_2016 to **Stage\_secondary**:

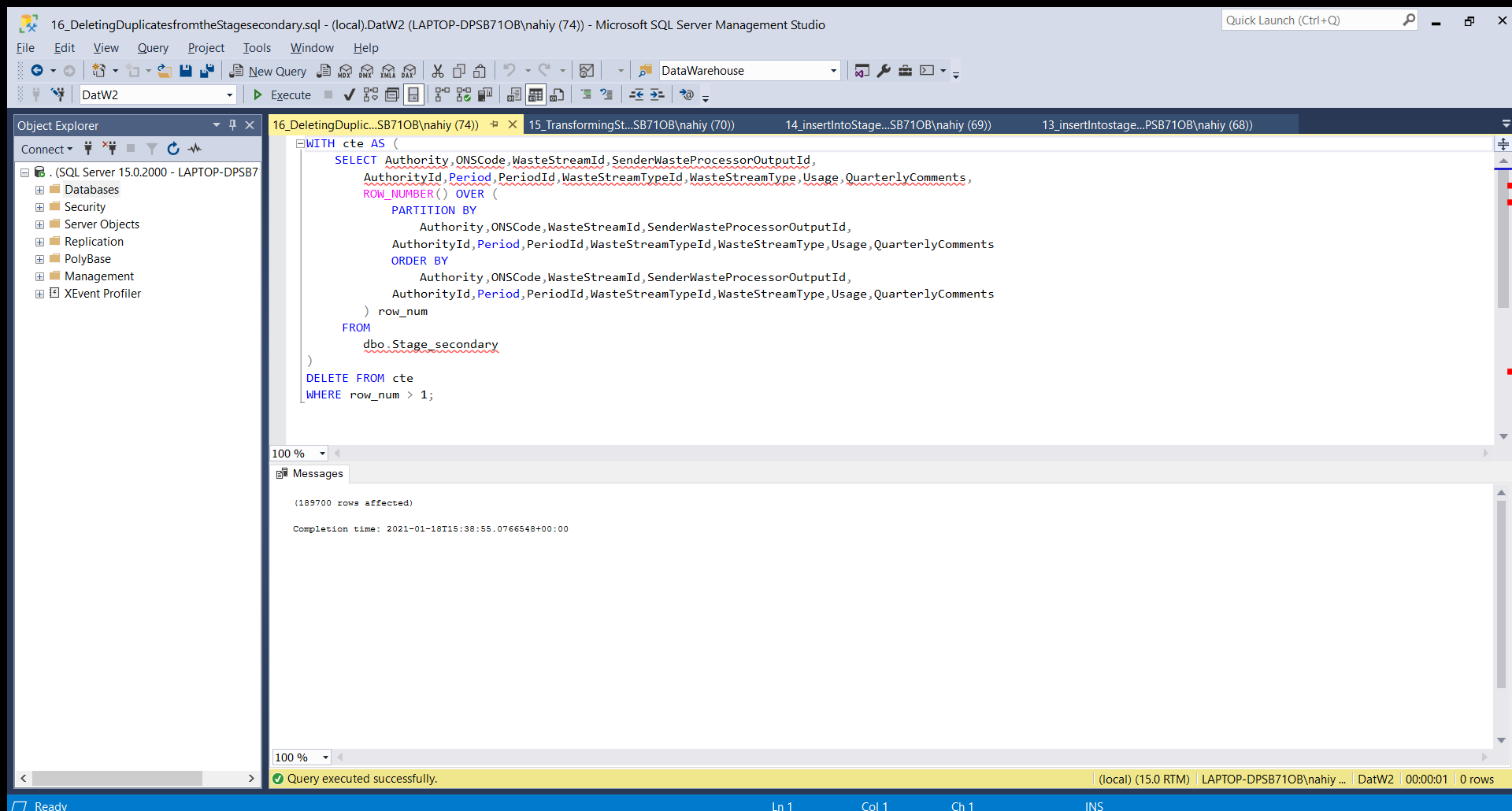


TRANSFORMING AND CLEANING THE DATA

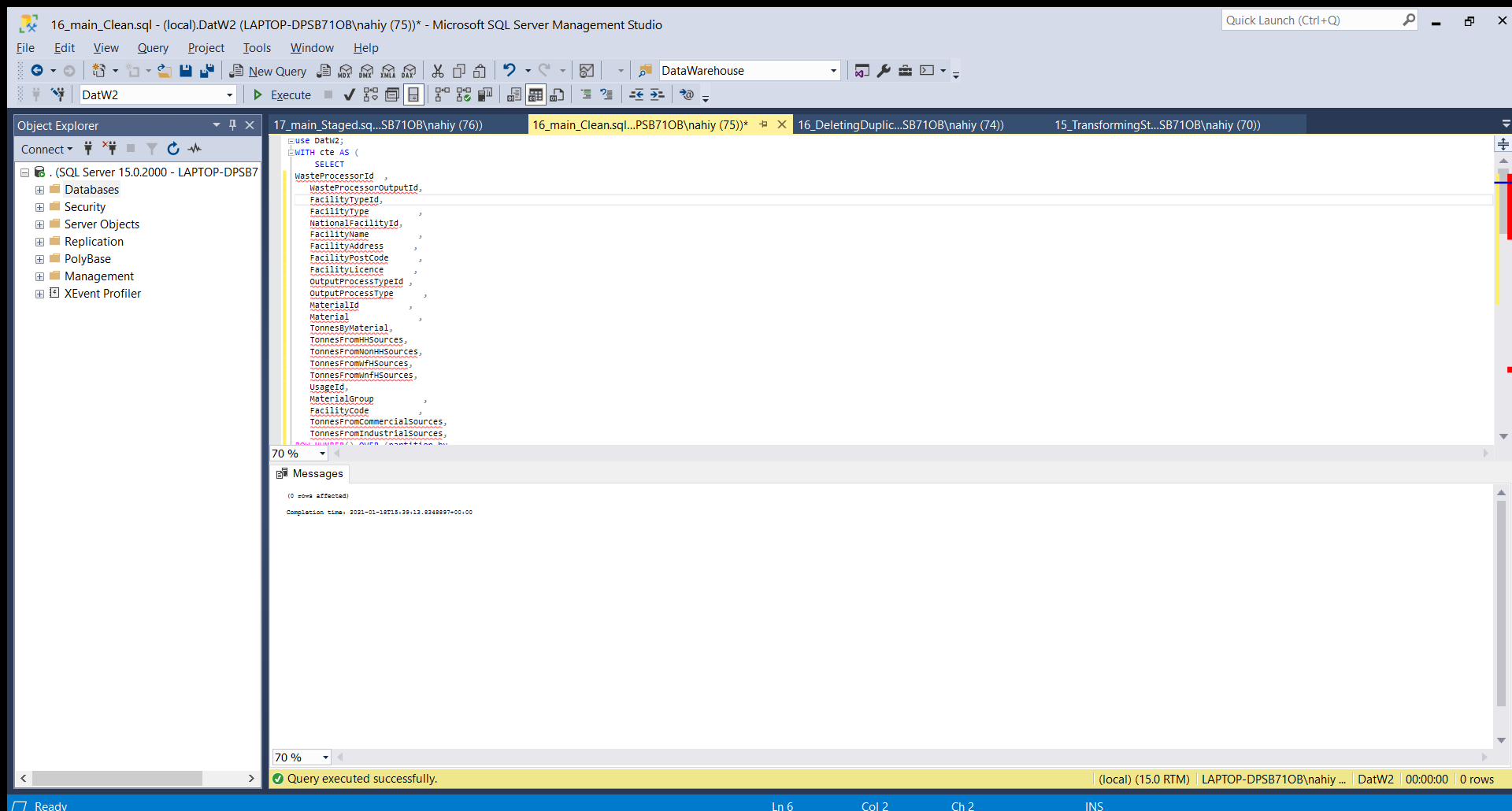
UPDATING Stage\_main (Material Id is a primary key in main table, It can’t be NULL)



DELETING Duplicating rows from the Stage\_secondary table

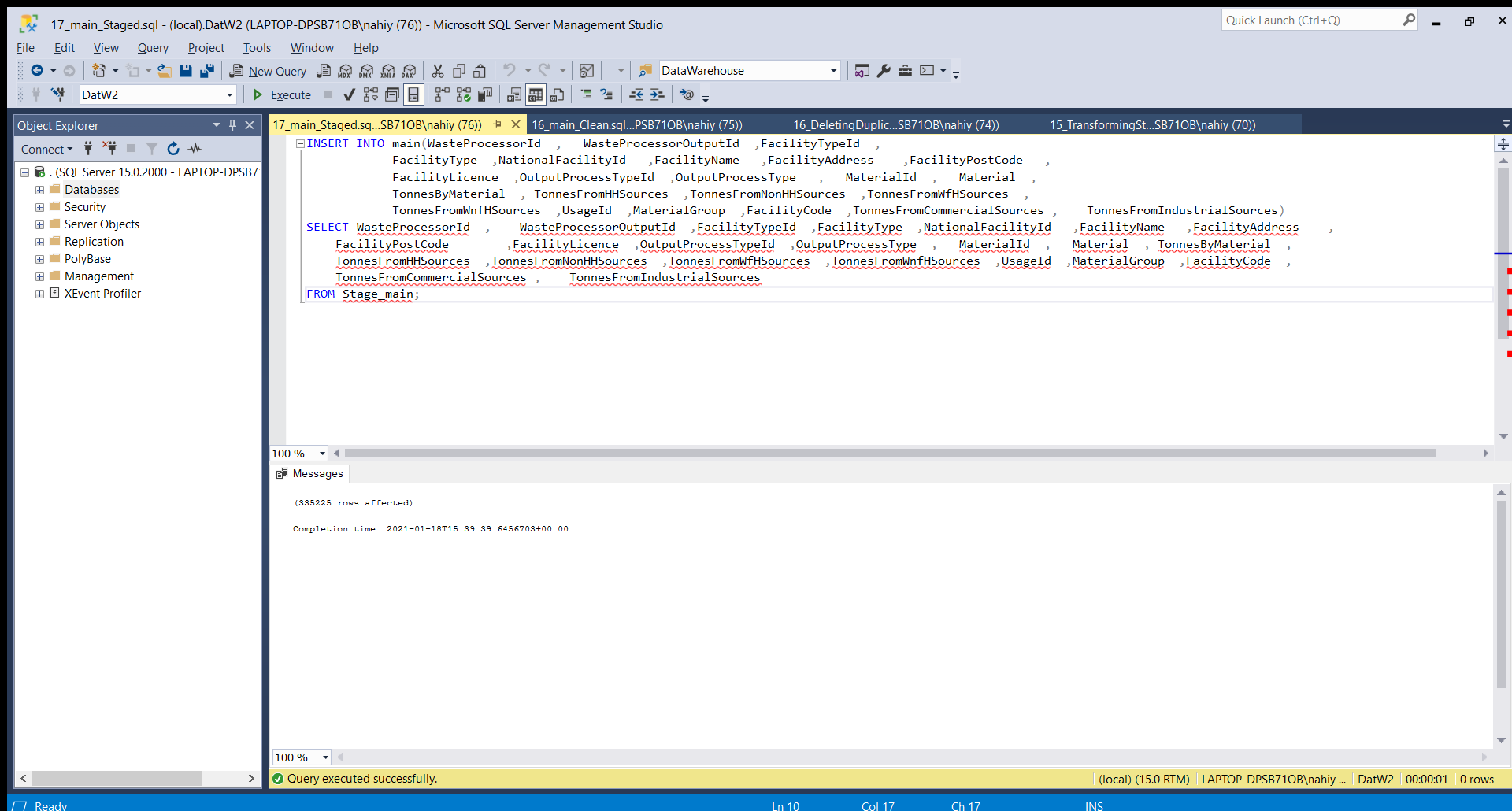


DELETING Duplicating rows from the Stage\_main table

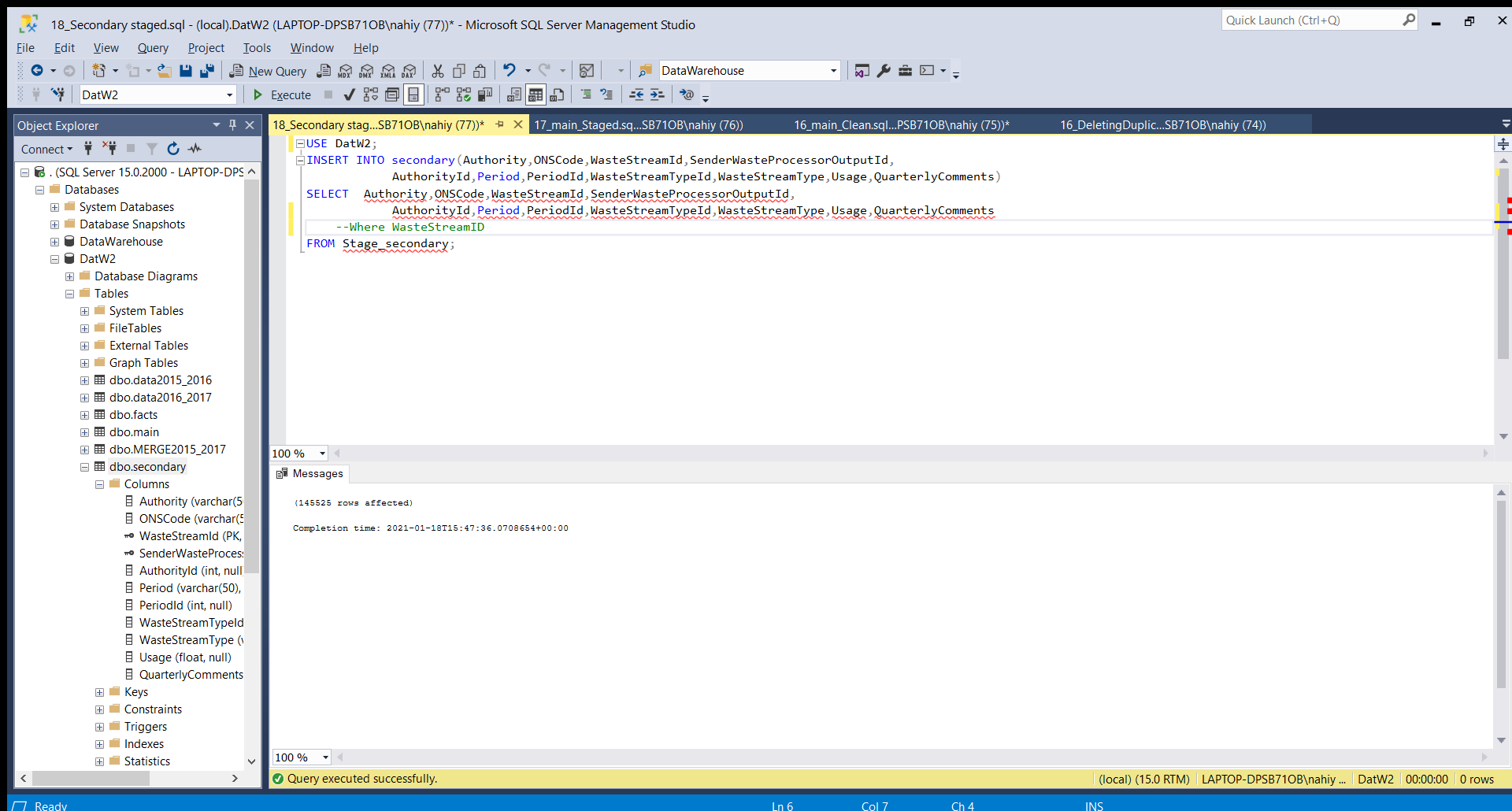


LOADING…

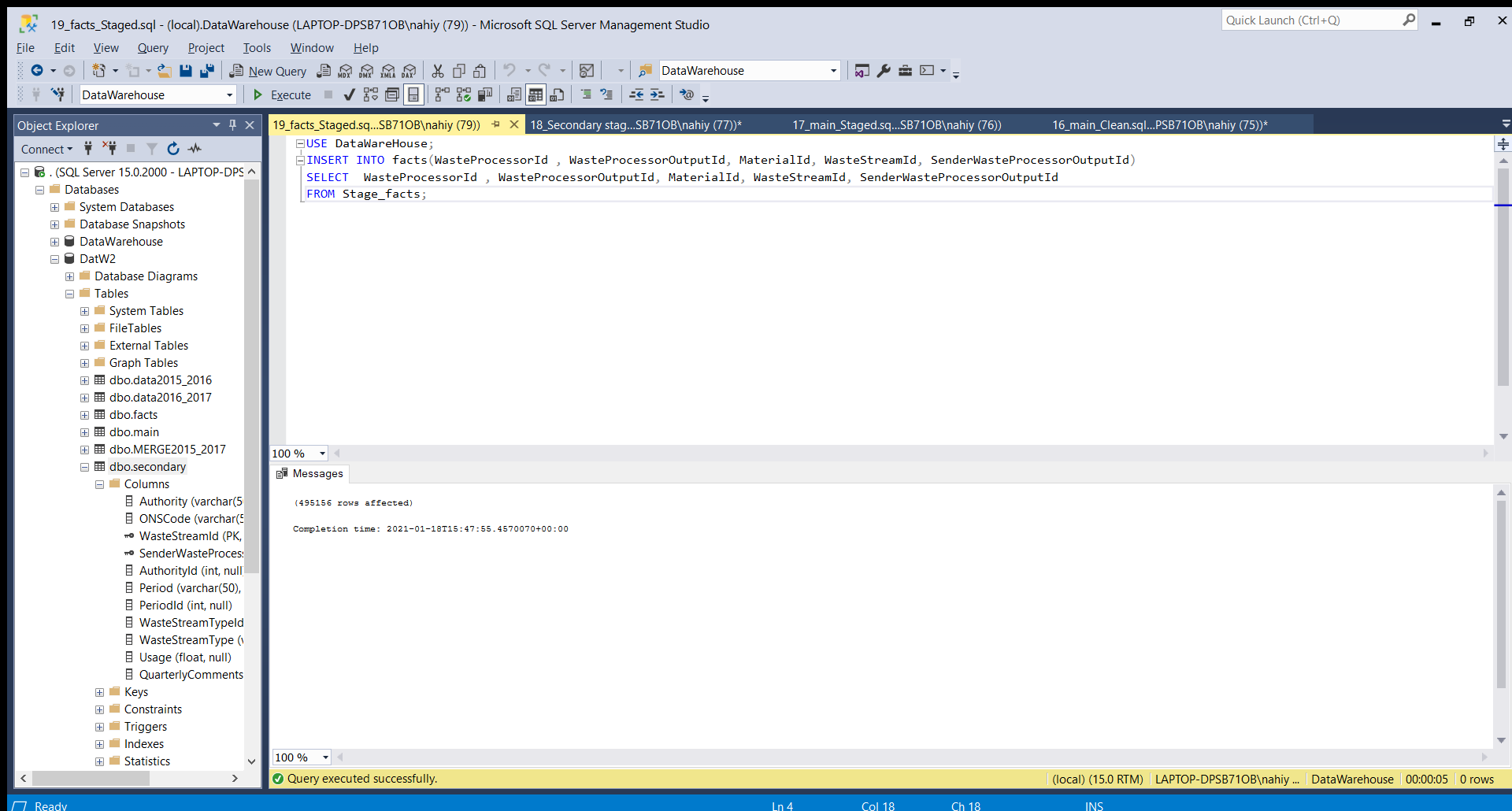
INSERTING all the data from the Stage\_main table to the main table:



INSERTING all the data from the Stage\_secondary table to the secondary table:



INSERTING all the data from the Stage\_facts table to the facts table:

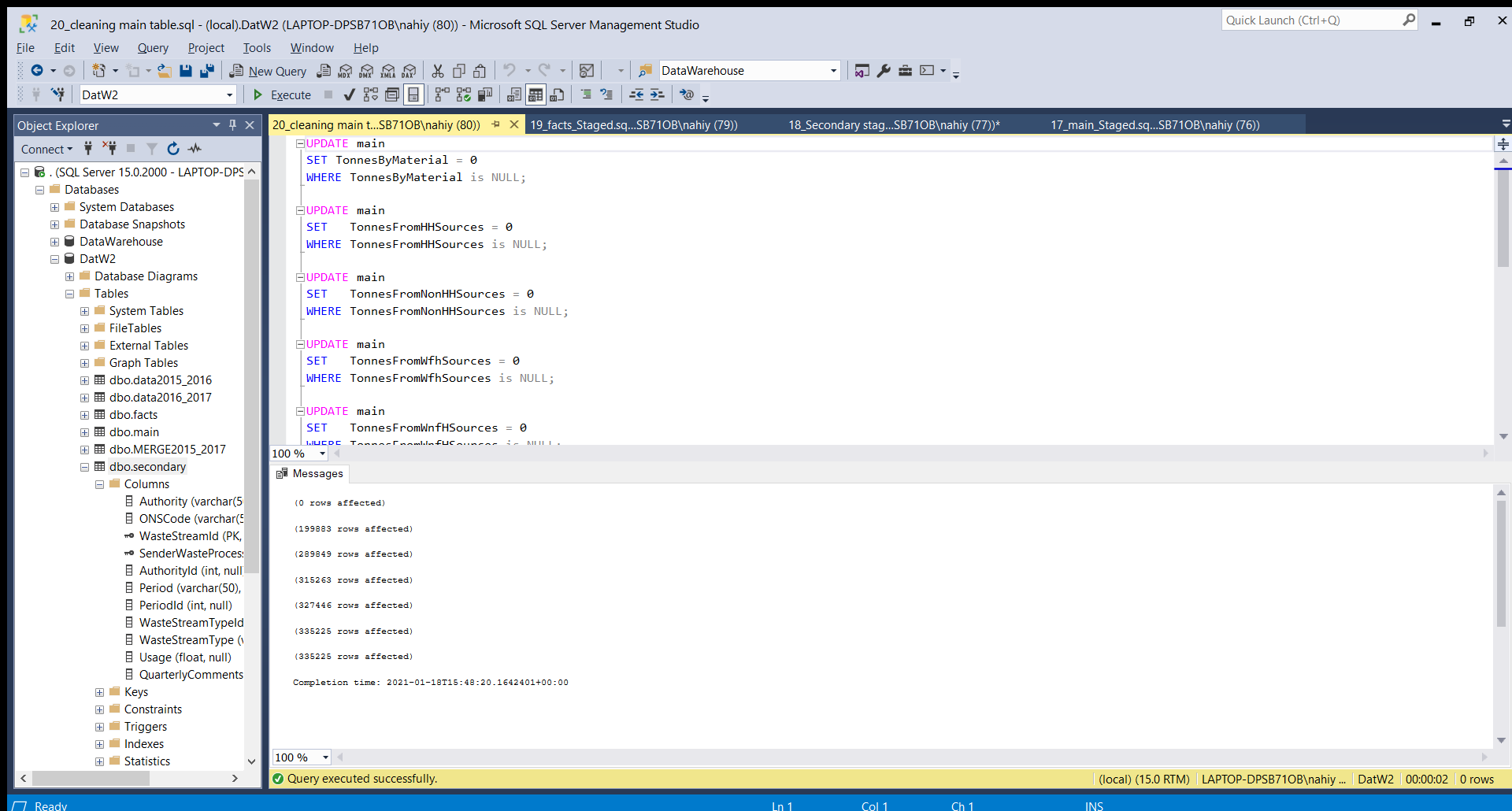


*Further required manipulations*

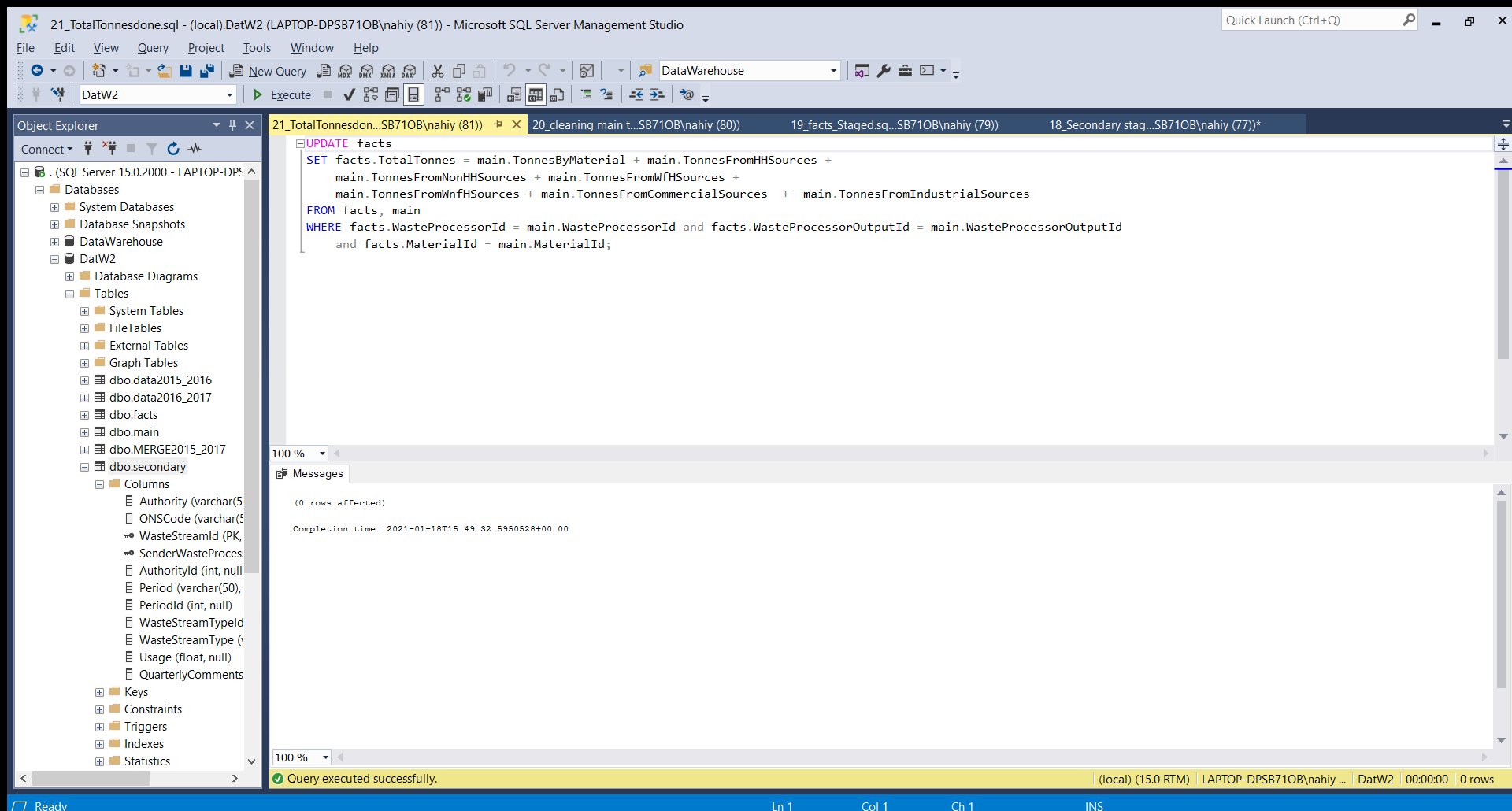
The data still needs to be optimized for better use and also needs to put accurate information of TotalTonnes attribute of the facts table

FURTHER CLEANING

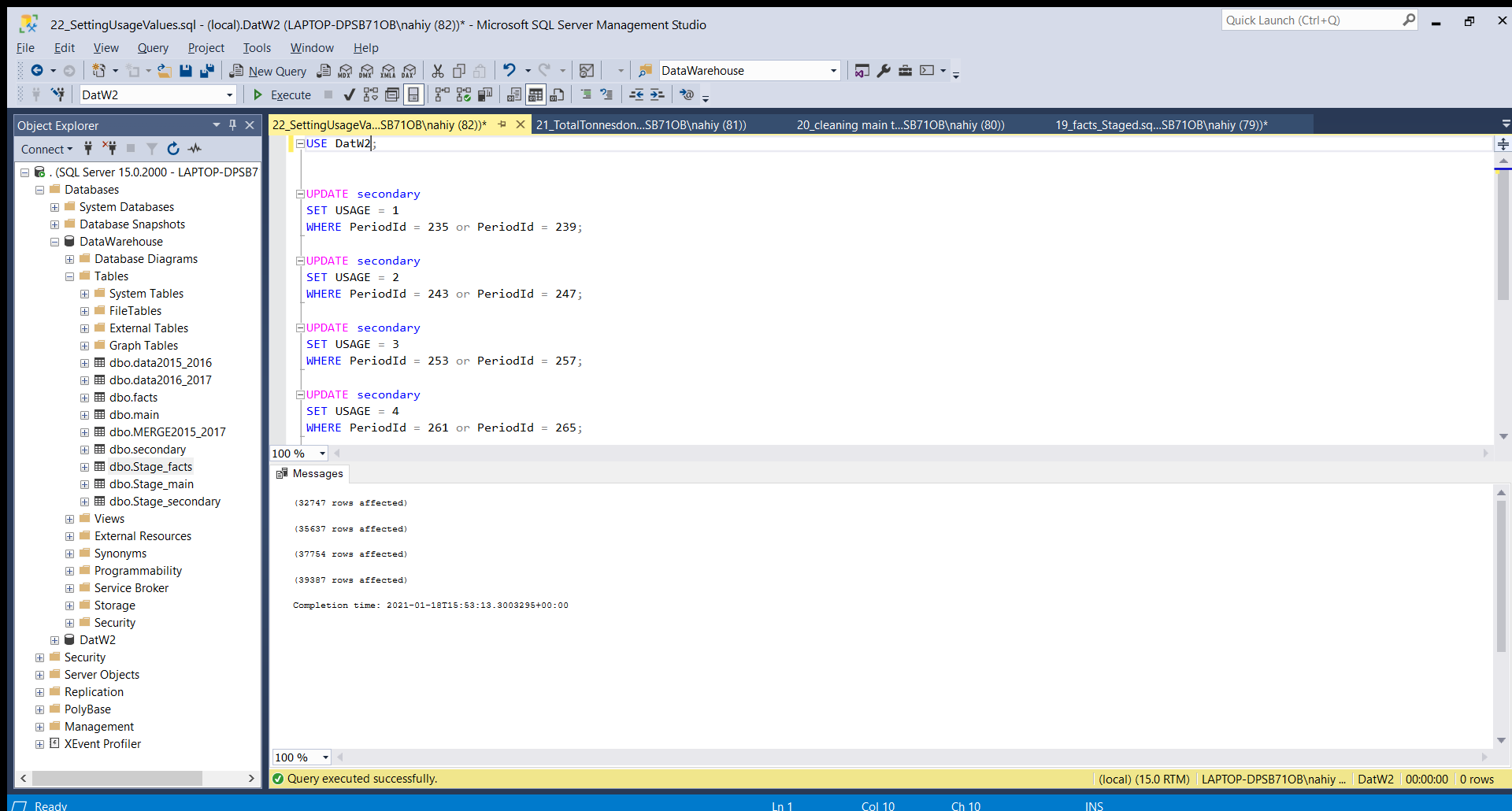
UPDATING main table by putting 0 in the places where a measurable quantity is left NULL



UPDATING facts table by correcting the Total Tonnes parameter so it becomes the sum of the weight measures(in Tonees) of all sources of the waste production:



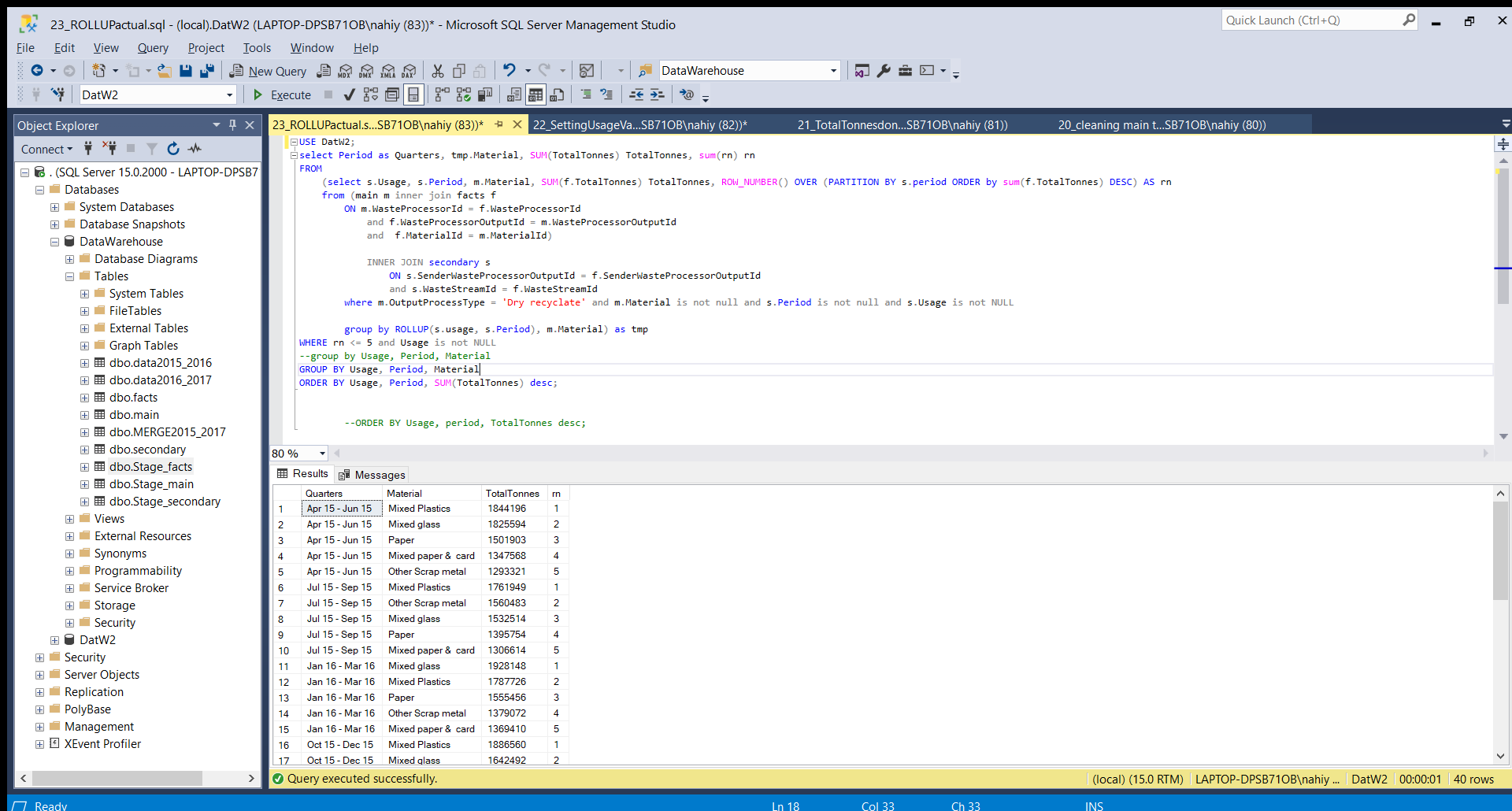
UPDATING the USAGE parameter of the secondary table by making it represent 6 months period



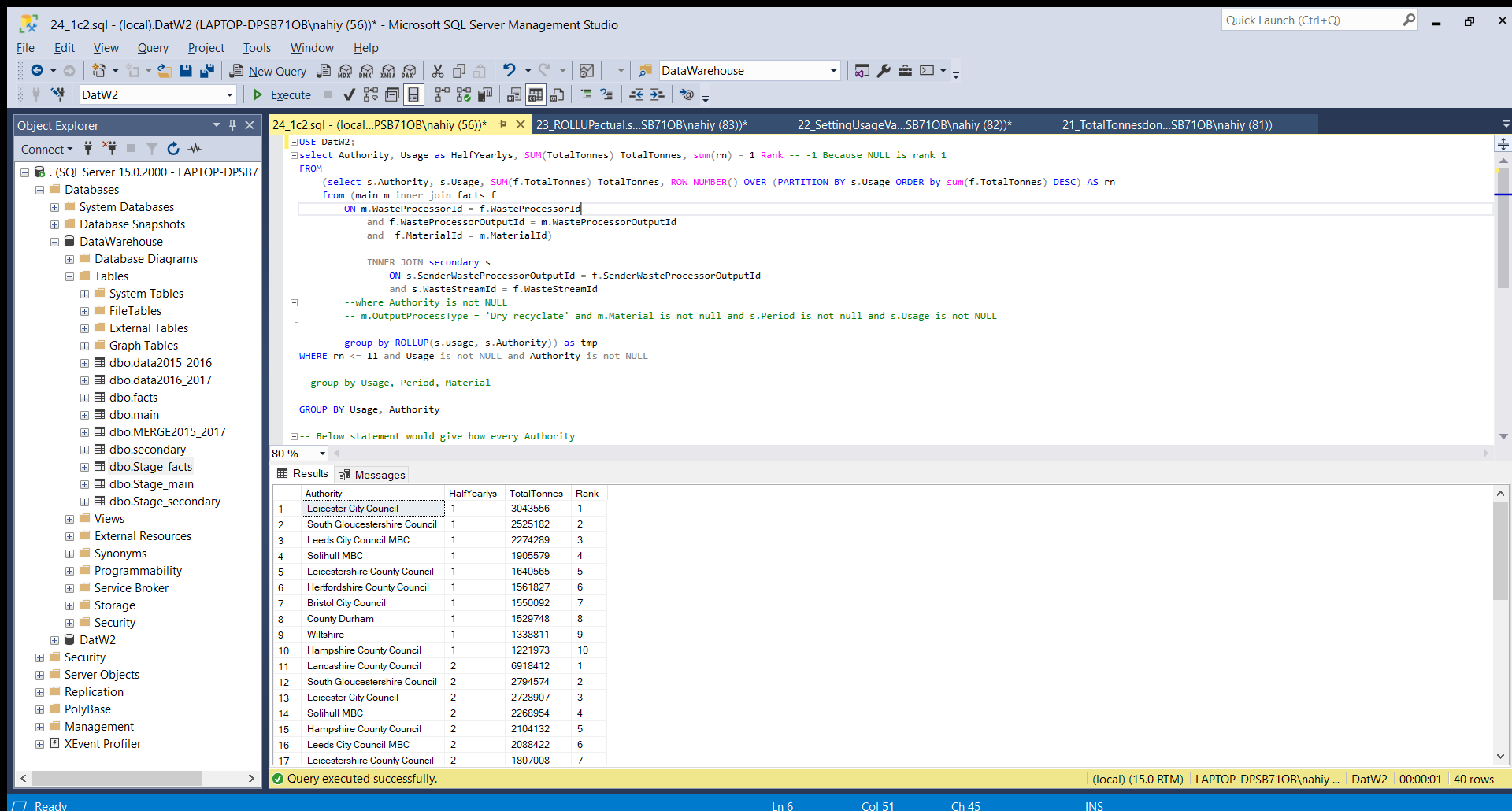
*Asked queries executions*

*---------------------------------*

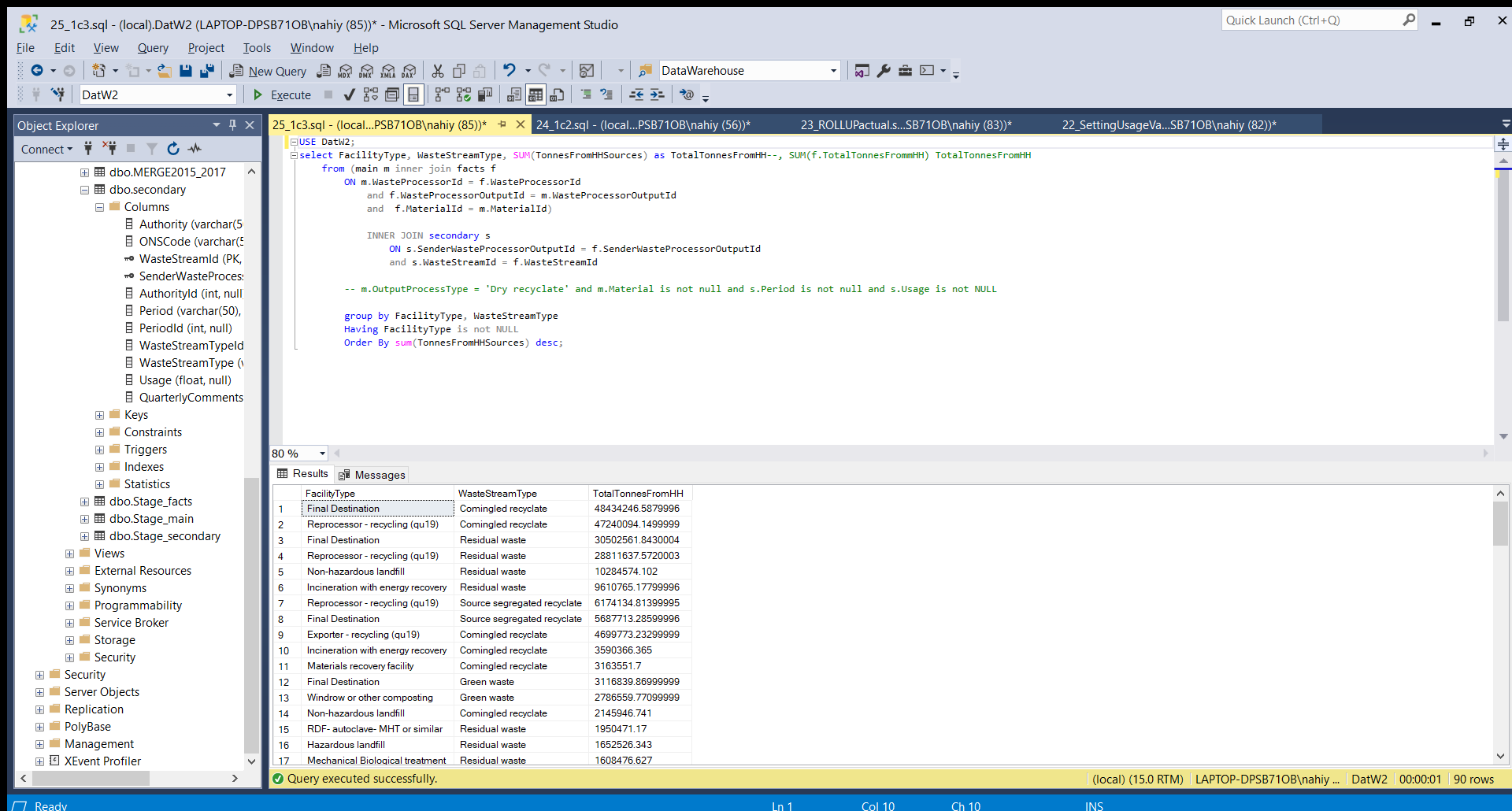
In the 1c2 we need top 5 “Dry recyclate” waste Materials(belongs in the main) TotalTonnes(belongs in the facts) for each period(belongs in the secondary). We have to perform join on the foregin keys, and do the required operations



In 1c2 we have rank top 10 Authorities on the basis of the TotalTonnes of waste for eacy six-month period.



In 1c3 we need to print TotalTonnesFromHH and WasteStreamType for each facilty type.



*BI Dashboard*

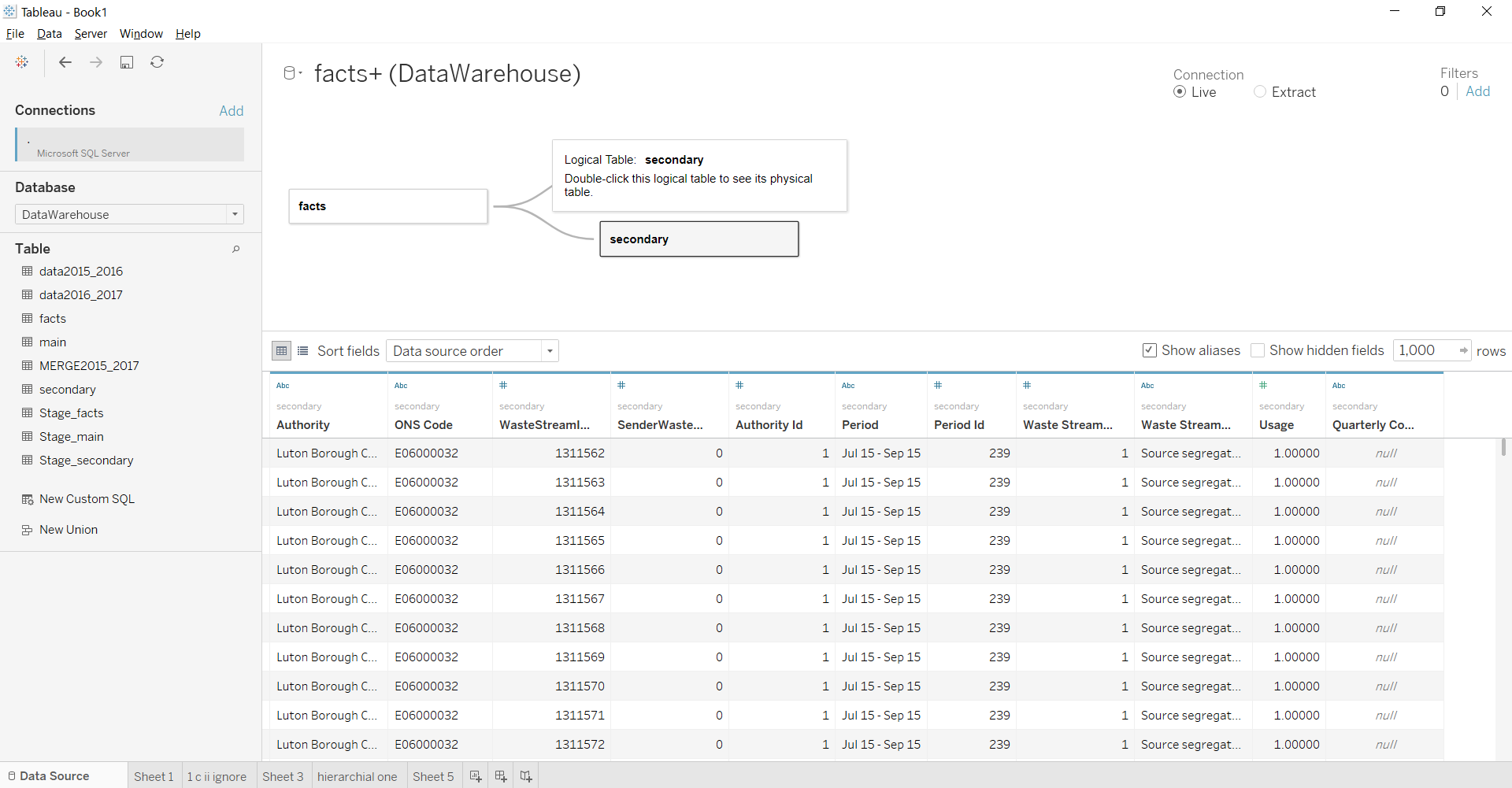
*------------------*

Connecting with the server (named ‘ , ‘)

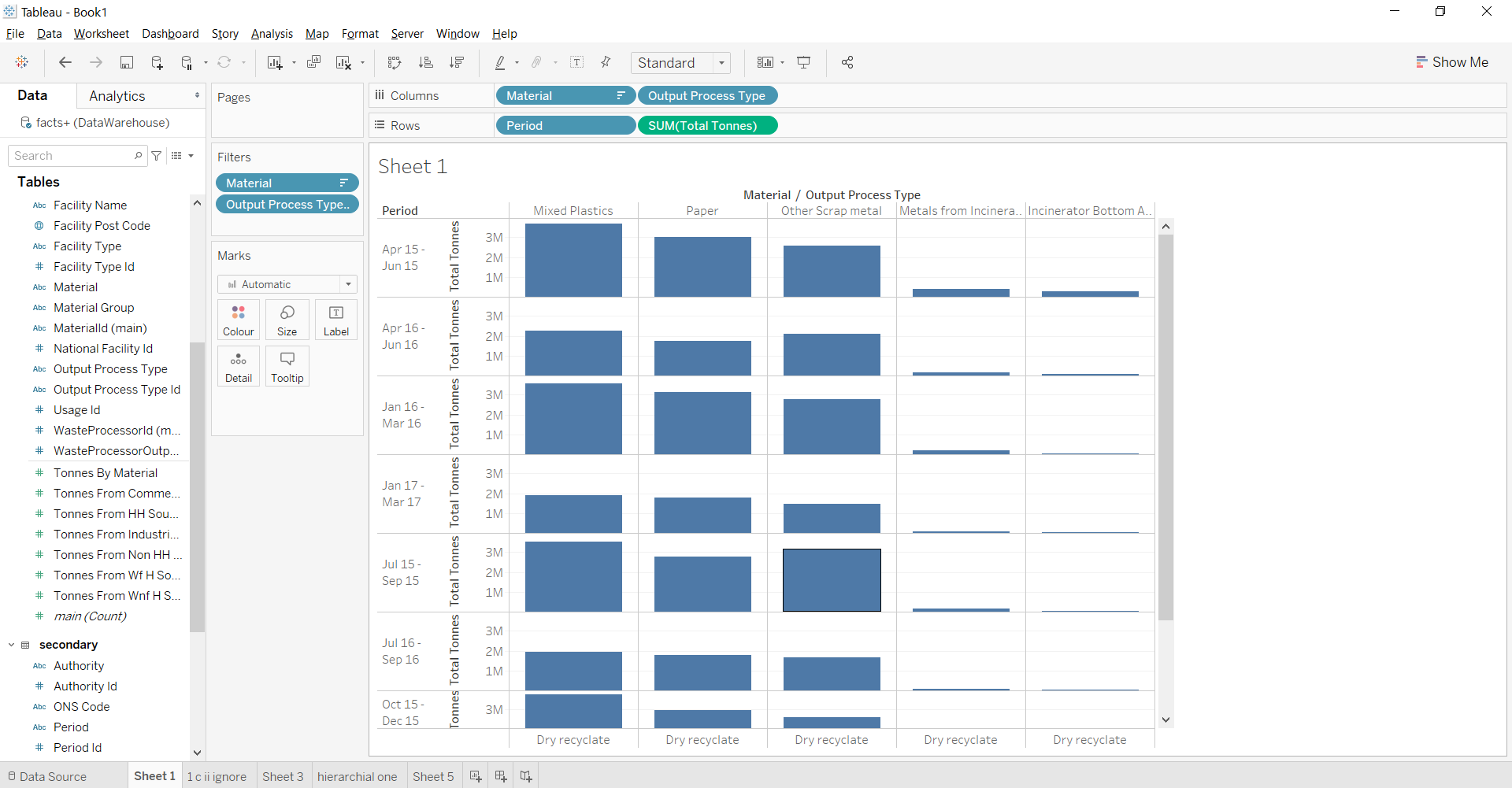
Selecting the Database

Drag and Drop the required tables (facts, main and secondary)

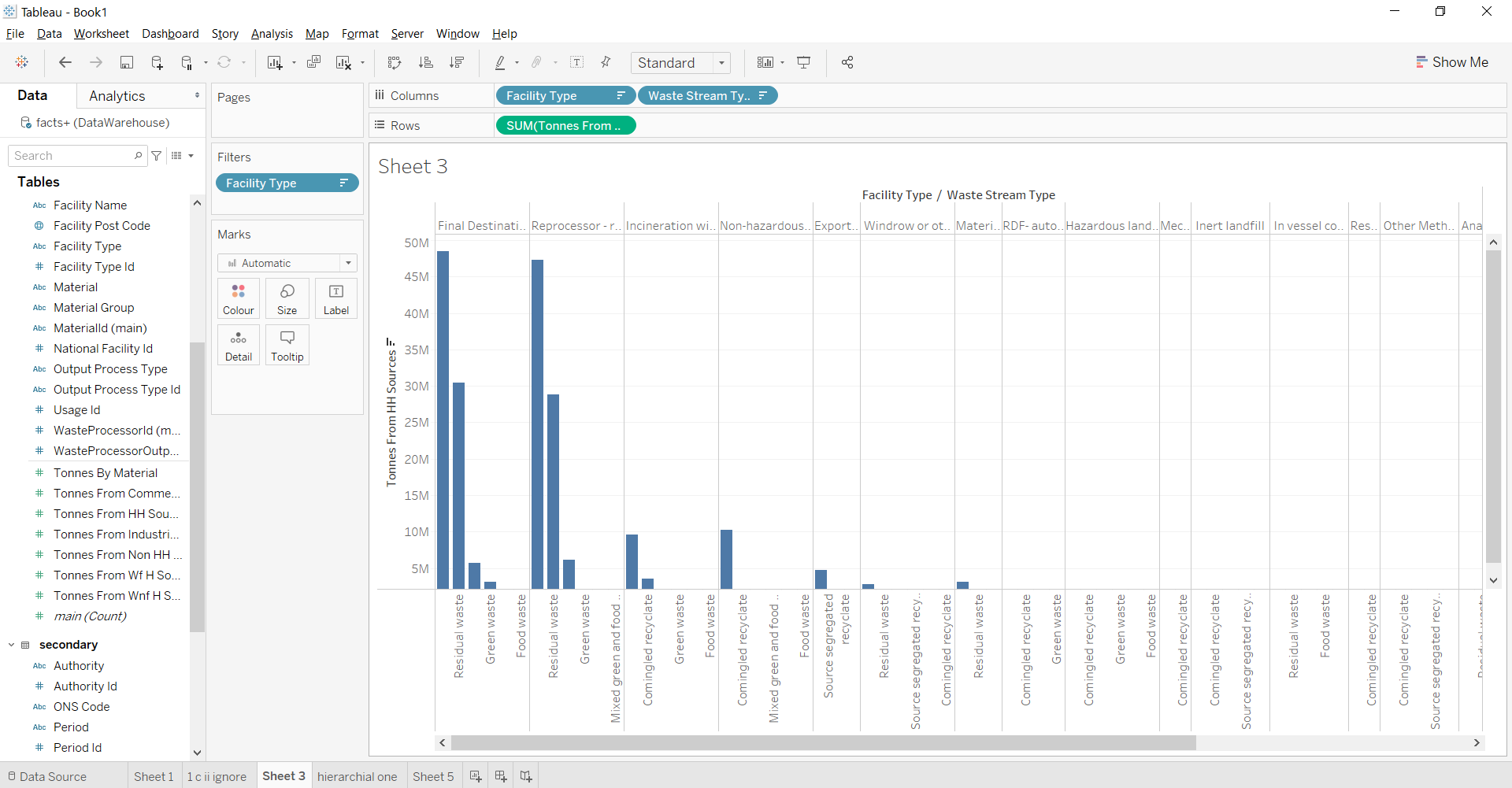
Match the relations among them (Primary keys from main & secondary to the Foriegn keys in the facts table)



Query one visualisation:



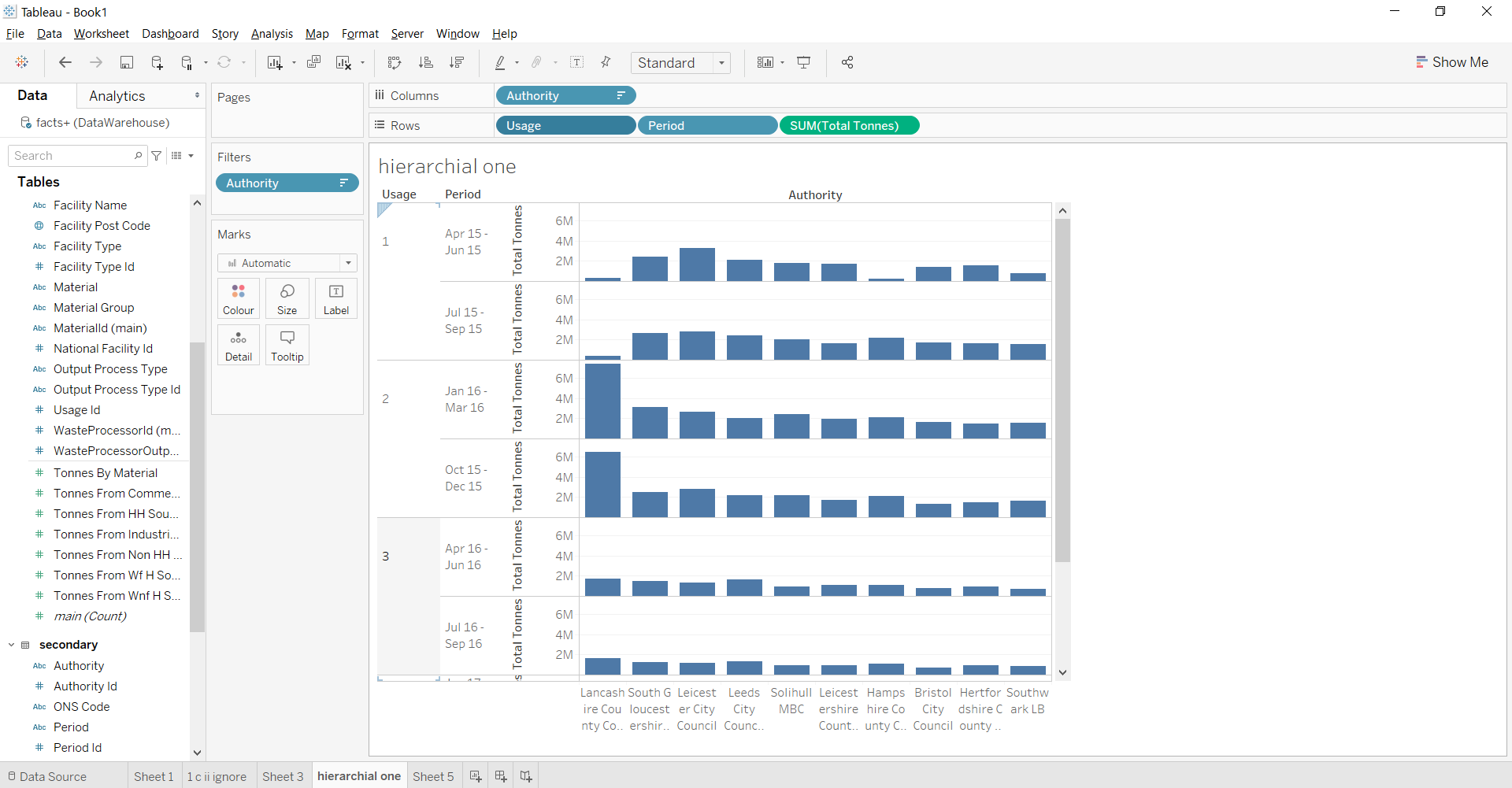
Query 3(1c.3) visulisation:



Query:

Show the top 10 Authorities producing TotalTonnes of Waste for each quarter and six month period.

Visualisation:



Query: Show TotalTonnes produced by each material and also show the material group it belongs to.

Visualisation:

