Logo

Description automatically generated

Data Warehousing & Business Intelligence

Y3 S2

**Assignment 1**

Submitted to

Sri Lanka Institute of Information Technology

By

Fernando WTH – IT22313652

Weekend Batch

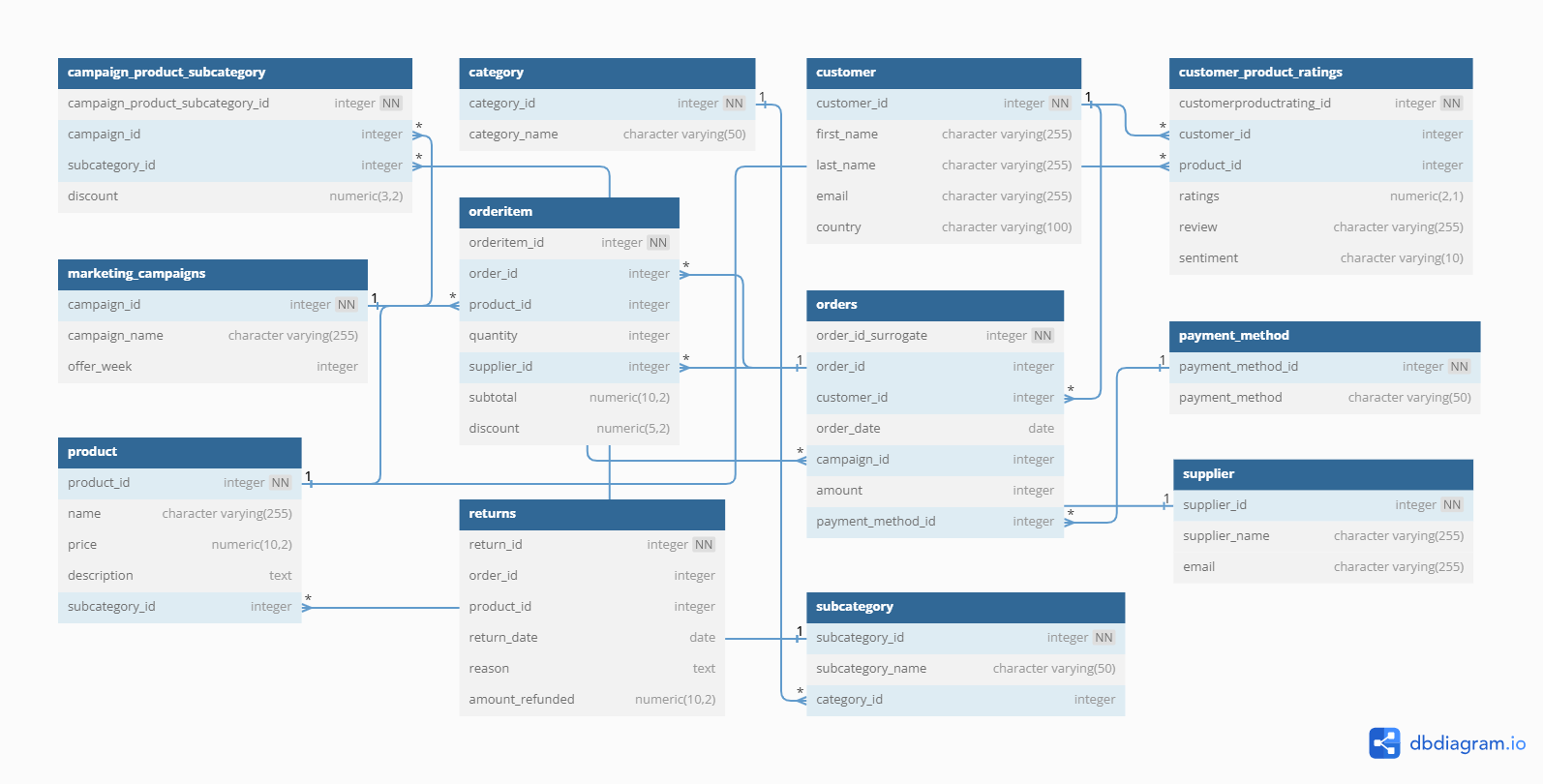
**Step 1: Data Set Selection**

The selected dataset is a publicly available dataset that simulates a real-world e-commerce online retail platform. It can be used to create a data warehouse solution for order lifecycle tracking and advanced customer behavior analytics. Each table represents either a business entity or a transaction which aligns with the OLTP characteristics.

Following the previously set guidelines for this assignment, the dataset consisted of around one year’s worth of data and records and attributes. The contents of this dataset was only csv files but using them I created 3 different data sources (CSV, text file and database). As the data was sufficient to create a data warehouse, I was also able to perform ETL functions with this dataset. I am also able to identify the various hierarchies, dimensions and aggregates within this dataset and as this is a data collection I am also able to generate reports.

Do to this I selected this Ecommerce Dataset as my chosen dataset for this assignment.  
Link: <https://www.kaggle.com/datasets/sharangkulkarni/oltp-ecommerce-data?select=eCommerce_schema.sql>

ER Diagram Schema



**Step 02: Preparation of Data Sources**

There were 12 CSV files available in the dataset. They were category.csv, customer.csv, customer\_product\_ratings.csv, marketing\_campaigns.csv, orderitem.csv, orders.csv, payment\_method.csv, product.csv, returns.csv, subcategory.csv, supplier.csv and campaign\_product\_subcategory.csv.

Using 9 of those CSV files I created a database an used that as one of my sources. For the remaining 3 files, category.csv, customer.csv and payment\_method.csv, I decided to convert the customer file to a text file and use the remaining two as CSV files bringing my data source count to 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Source** | **Data Source Type** | **Description** | **Final Type** |
| category.csv | csv | This file contains the basic information about the various product categories. | CSV |
| customer.csv | csv | This file contains the basic information about the various customers. | Text File(.txt) |
| customer\_product\_ratings.csv | csv | This file contains information about the customer product ratings. |  |
| marketing\_campaigns.csv | csv | This file contains the basic information about the marketing campaigns. | Ecommerce\_OLTP datablase |
| orderitem.csv | csv | This file contains the basic information about the various ordered items. |  |
| orders.csv | csv | This file contains information about the customer orders. |  |
| payment\_method.csv | csv | This file contains the payment methods. | CSV |
| product.csv | csv | This file contains the basic information about the various products. |  |
| returns.csv | csv | This file contains the information regarding order returns. | Ecommerce\_OLTP datablase |
| subcategory.csv | csv | This file contains the basic information about the subcategories of products. |  |
| supplier.csv | csv | This file contains the basic information about the suppliers. |  |
| campaign\_product\_subcategory.csv | csv | This file contains the information about the ampaign product subcategories. . | Ecommerce\_OLTP datablase |

Shown below are the files I imported to the database

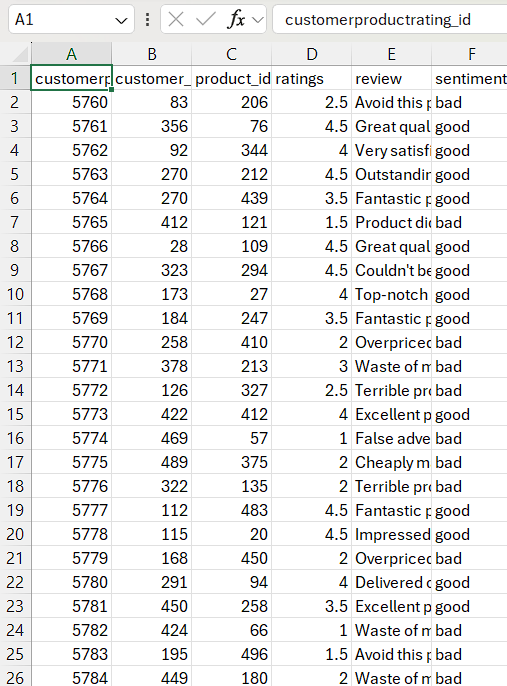
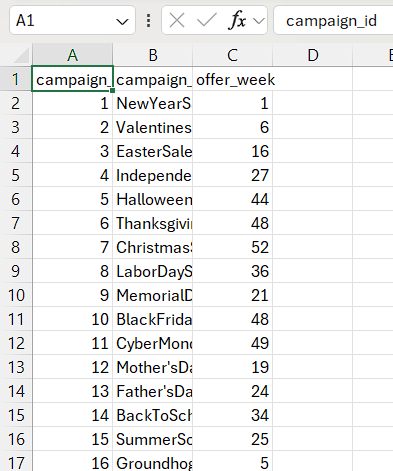


Figure 2 marketing\_campaigns.csv

Figure1 customer\_product\_ratings.csv

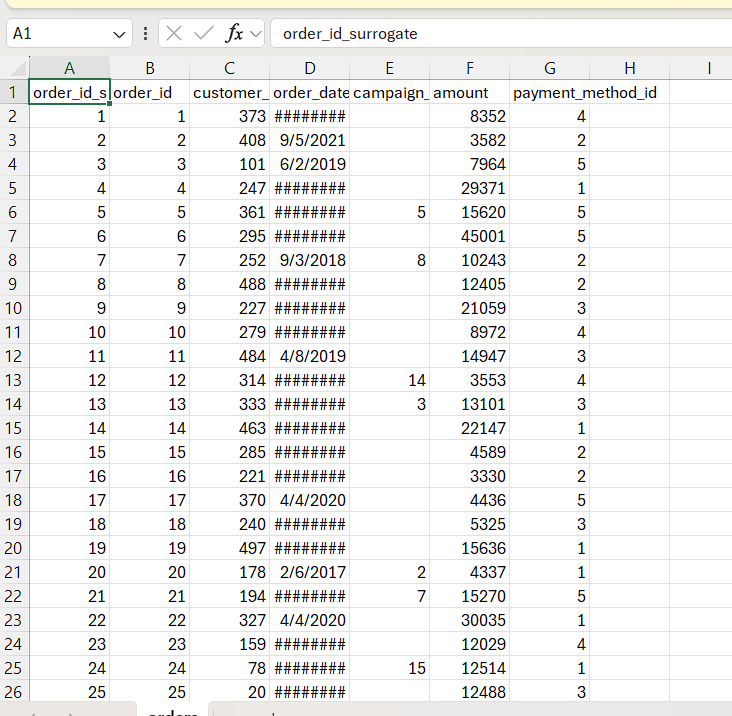


Figure 4 orders.csv

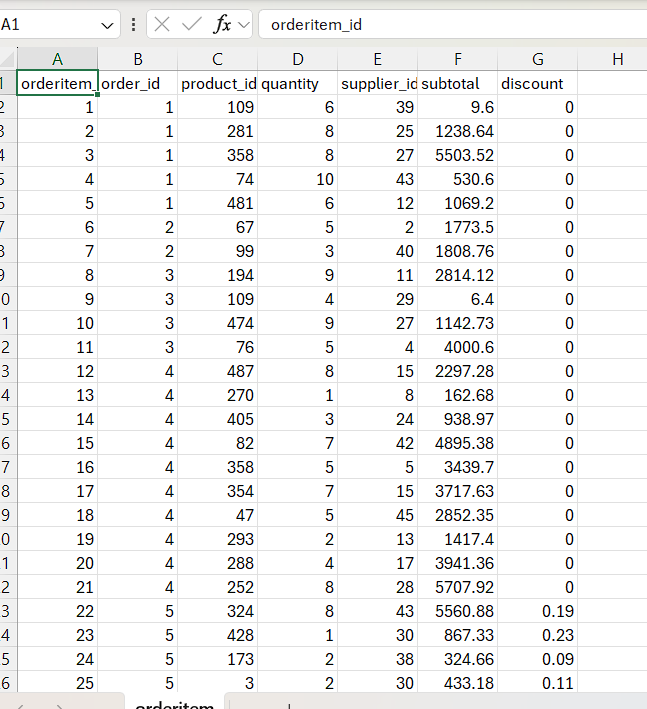


Figure 3 orderitem.csv

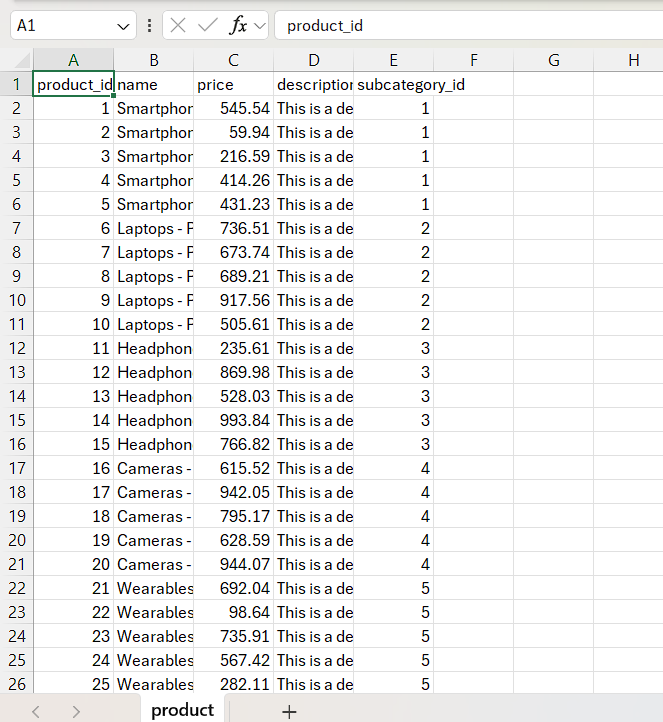
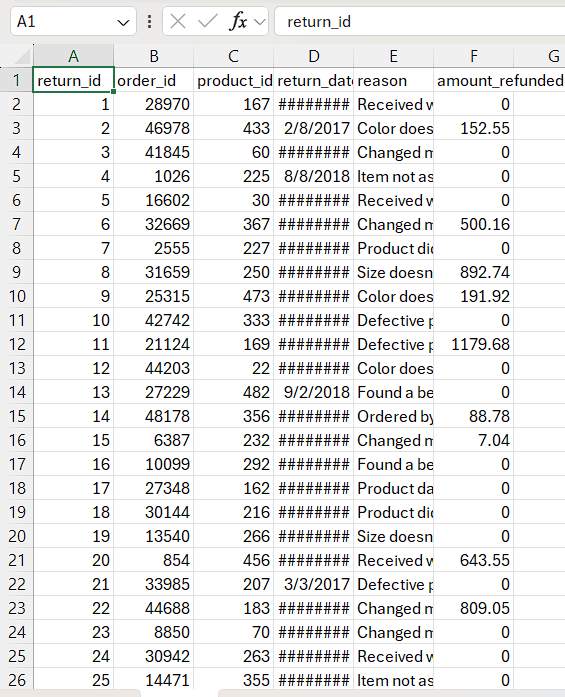


Figure 6 returns.csv

Figure 5 product.csv

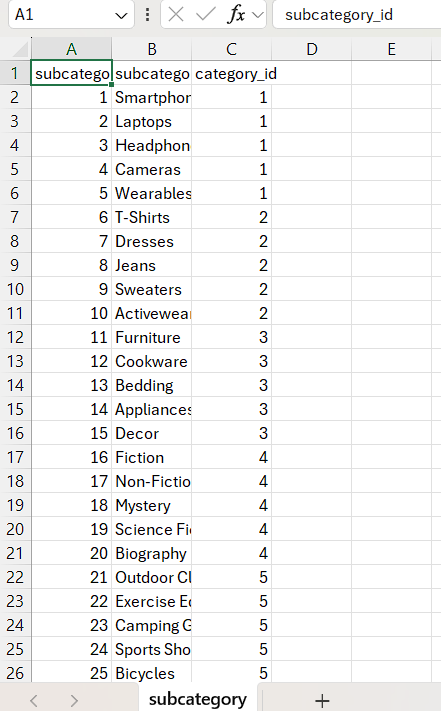


Figure 7 subcategory.csv

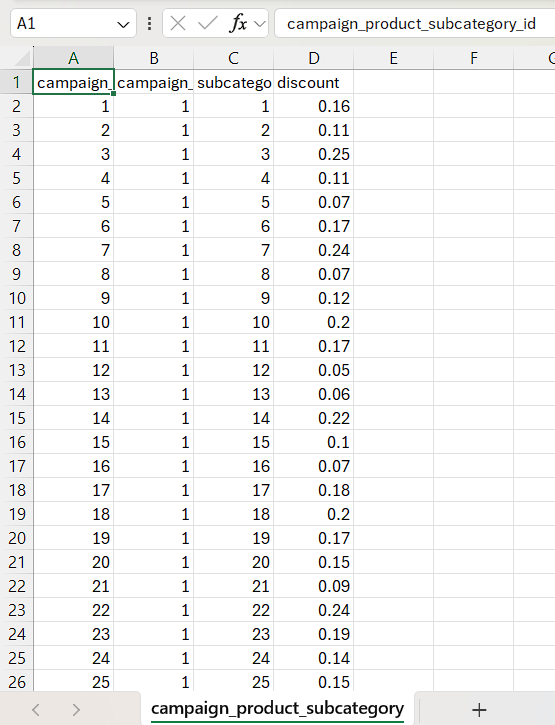
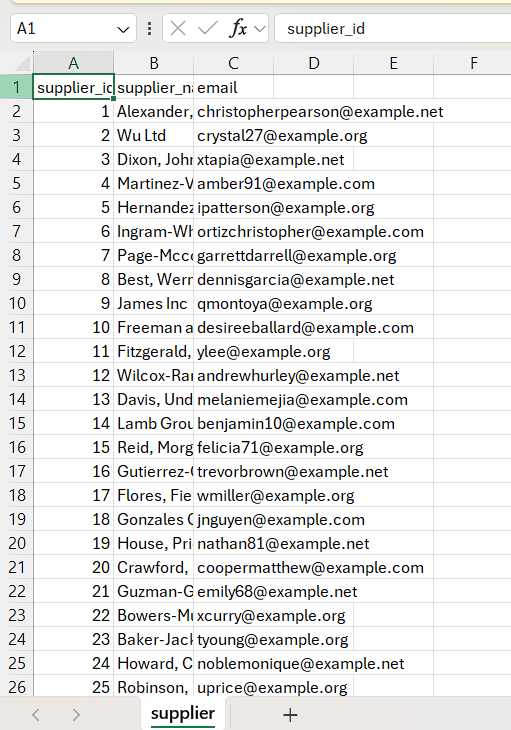
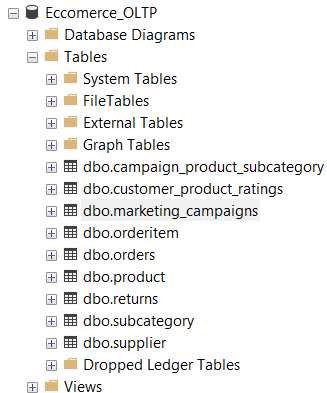
 

Figure 9 campaign\_product\_subcategory.csv

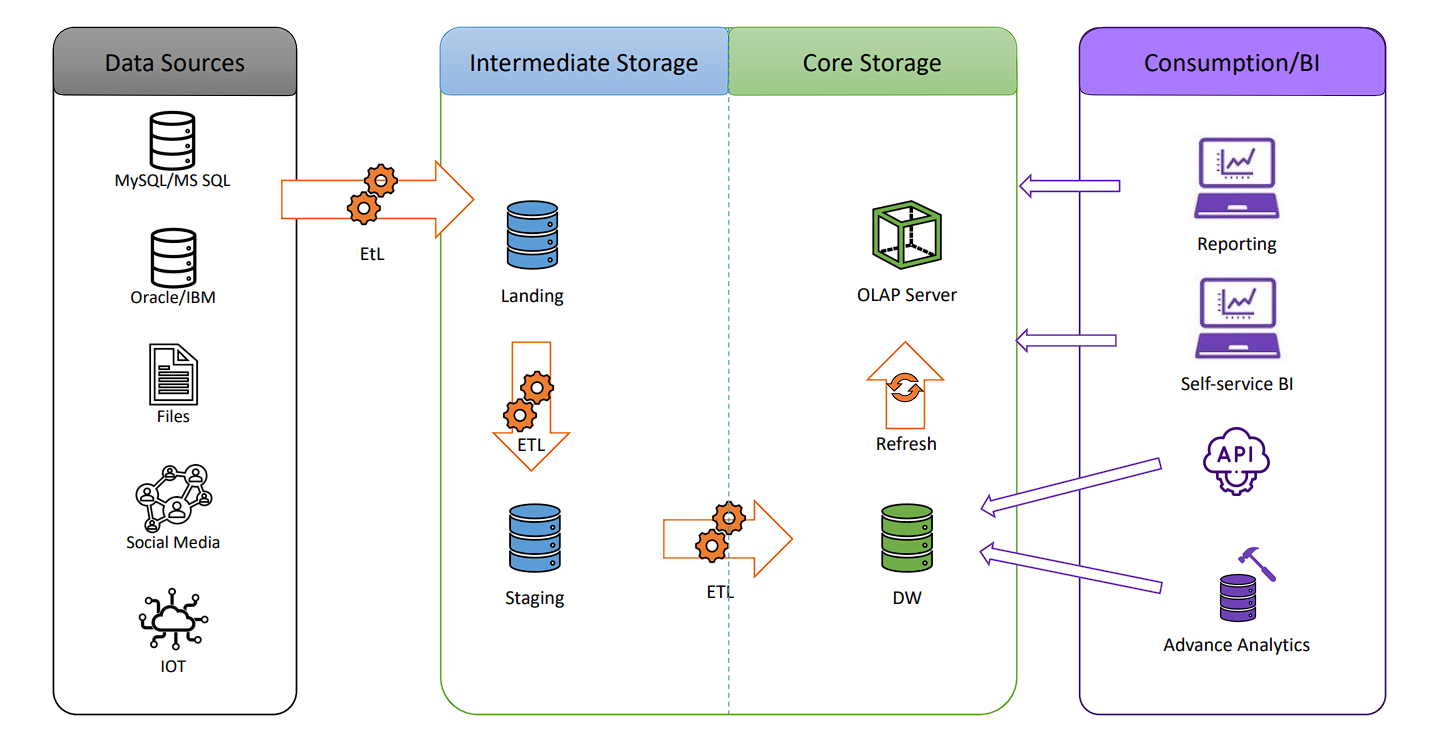
Figure 8 supplier.csv

I loaded them all into a database as shown below.



I also created a data warehouse name Ecommerce\_OLTP\_DW where I created all my dimension tables and fact tables.

**Step 3: Solution Architecture**



1. **Source Systems**: Text, CSV, SQL tables
2. **ETL Layer (SSIS)**: Extract from sources, transform data (cleansing, surrogate keys, lookups), load into DW
3. **Data Warehouse (SQL Server)**: Snowflake schema containing dimensions and fact tables
4. **Reporting Tools**: Power BI

**Step 4: Data warehouse design & development**

For my Data Set the schema that I chose was the snowflake schema. In my data there are 8-dimension tables and 3 fact tables. The slowly changing dimension table is the Customer table as the country the customer is currently residing in can change from time to time as well as their email, so we have to maintain historical data.

**Dimension Tables Created:**

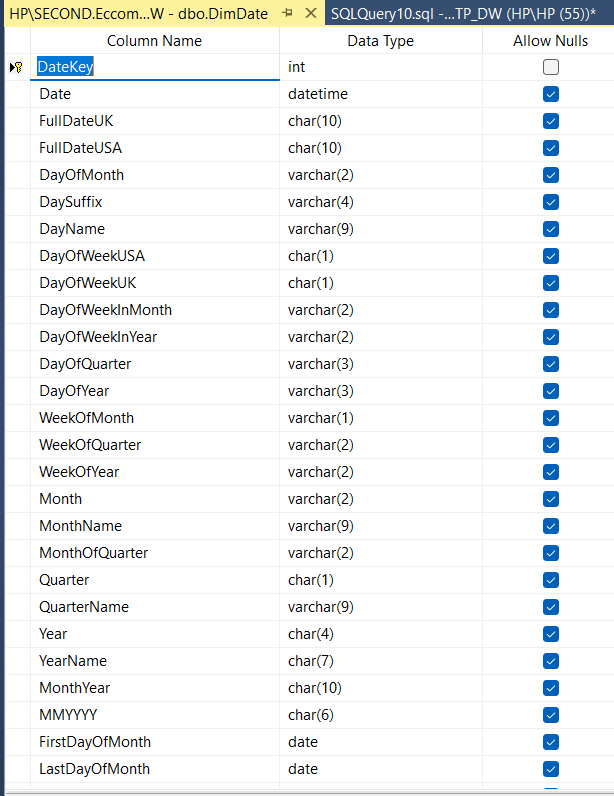
* DimCustomer (SCD)
* DimProduct
* DimSupplier
* DimCategory
* DimSubCategory
* DimCampaign
* DimCampaignSubcategory
* DimDate

Each dimension includes insert\_date, modified\_date, and in the case of DimCustomer, also includes start\_date, end\_date.

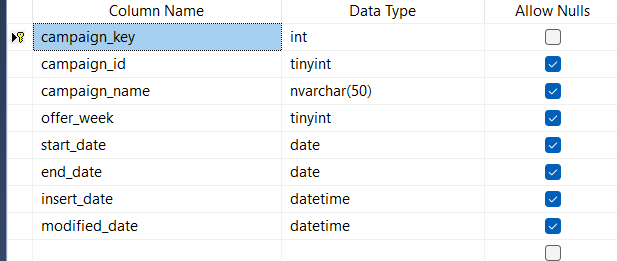
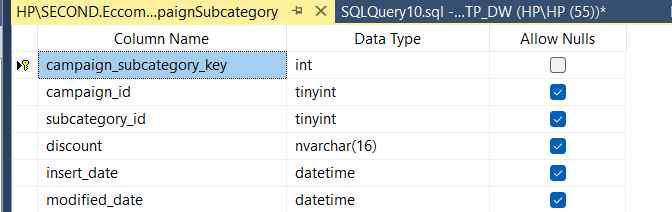
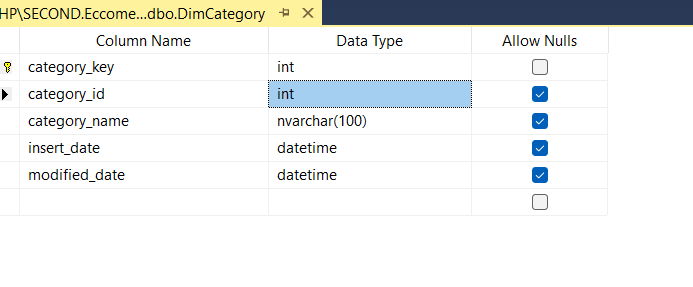
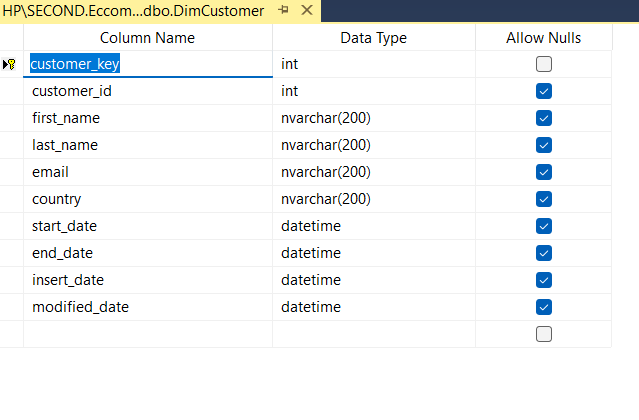
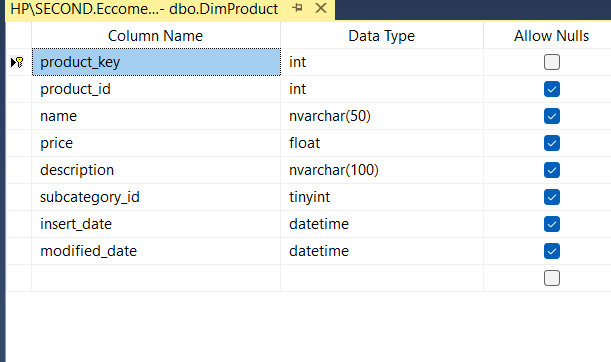
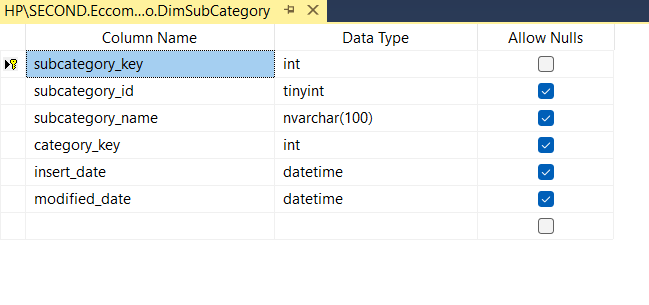
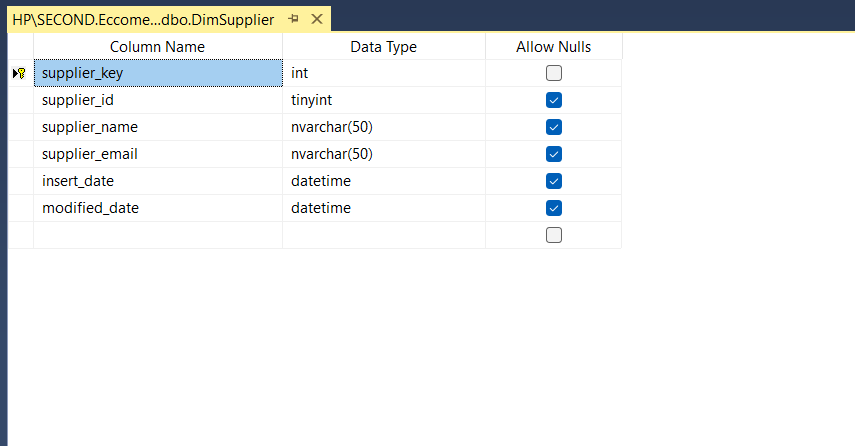
Logical foreign key relationships are maintained through surrogate keys but not enforced physically for ETL performance.

**Dimension Table Creation**

Before going forward with the other dimension tables I first created a date dimension table.



Then following that I created a total of 7 other dimension tables.



DimSupplier

DimCustomer

DimSubCategory

DimProduct

DimCategory

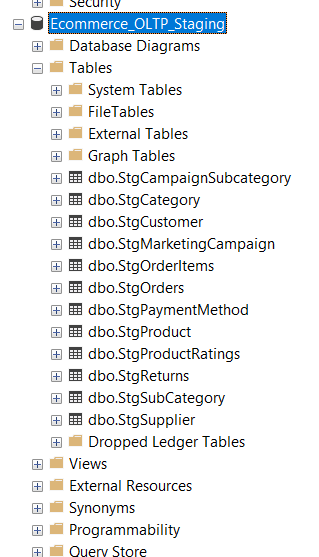
DimCampaign

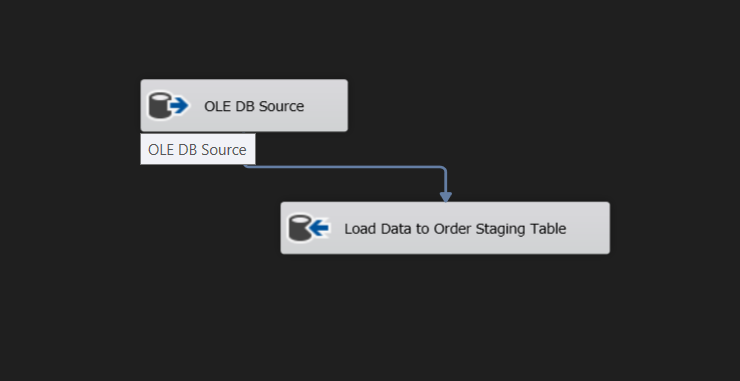
DimCampaignSubCategory

**Step 5: ETL Development**

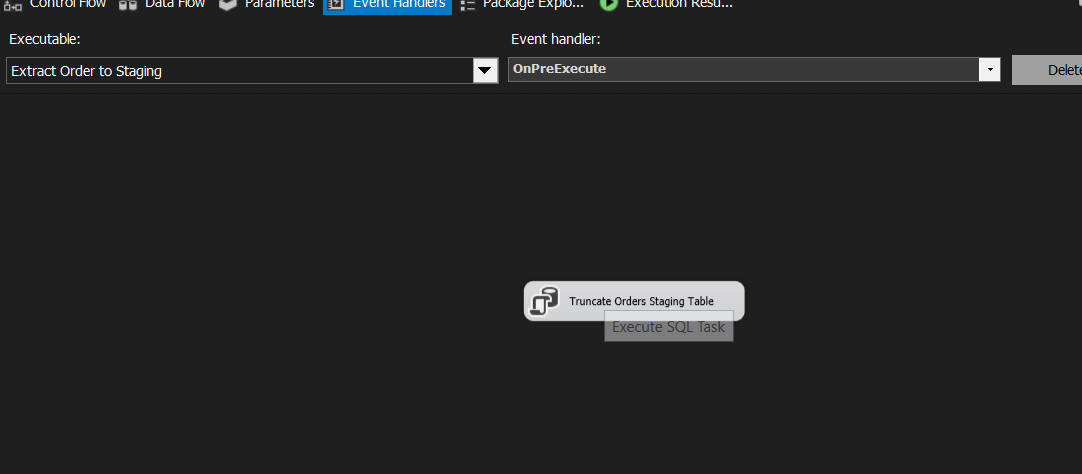


Using the SQL Server Integration Service available in Visual Studio, I extracted all the data from the tables that were in the source database and the 2 separate CSV files and the text file to a separate staging DB called Ecommerce\_OLTP\_Staging.



**Extract Order Data to Staging**

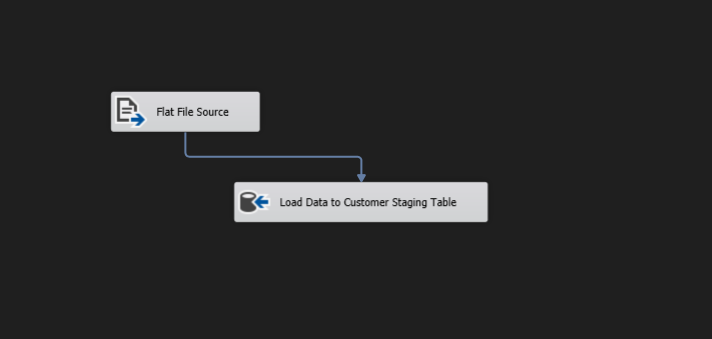
Used the OLE DB Source to connect to the source table dbo.orders table in the Ecommerce\_OLT P source database and loaded that data to a staging table called dbo.StgOrders in the Ecommerce\_OLTP\_Staging database using a OLE DB Destination.

****

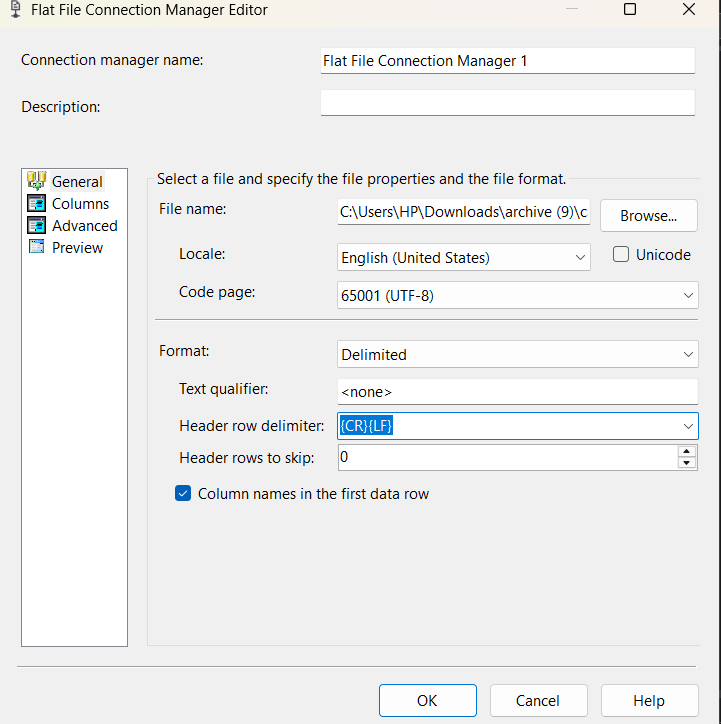
Used an Execute SQL task tool in SSIS to truncate the orders table for each time it’s loaded to avoid duplicates.

This process was repeated for all the tables in the source database.

**Extract Customer Data to Staging**



Used a flat file source to extract data from the text file, which is considered as a flat file along with csv files.

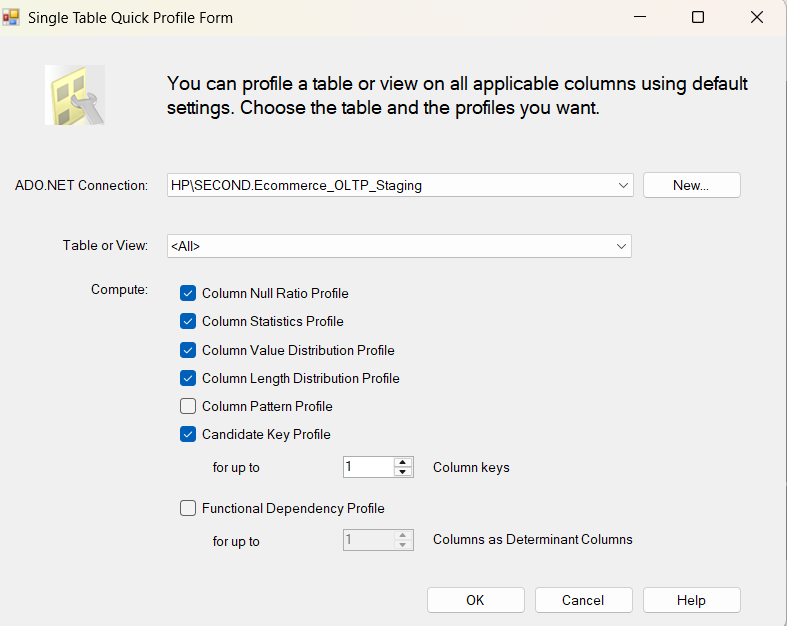


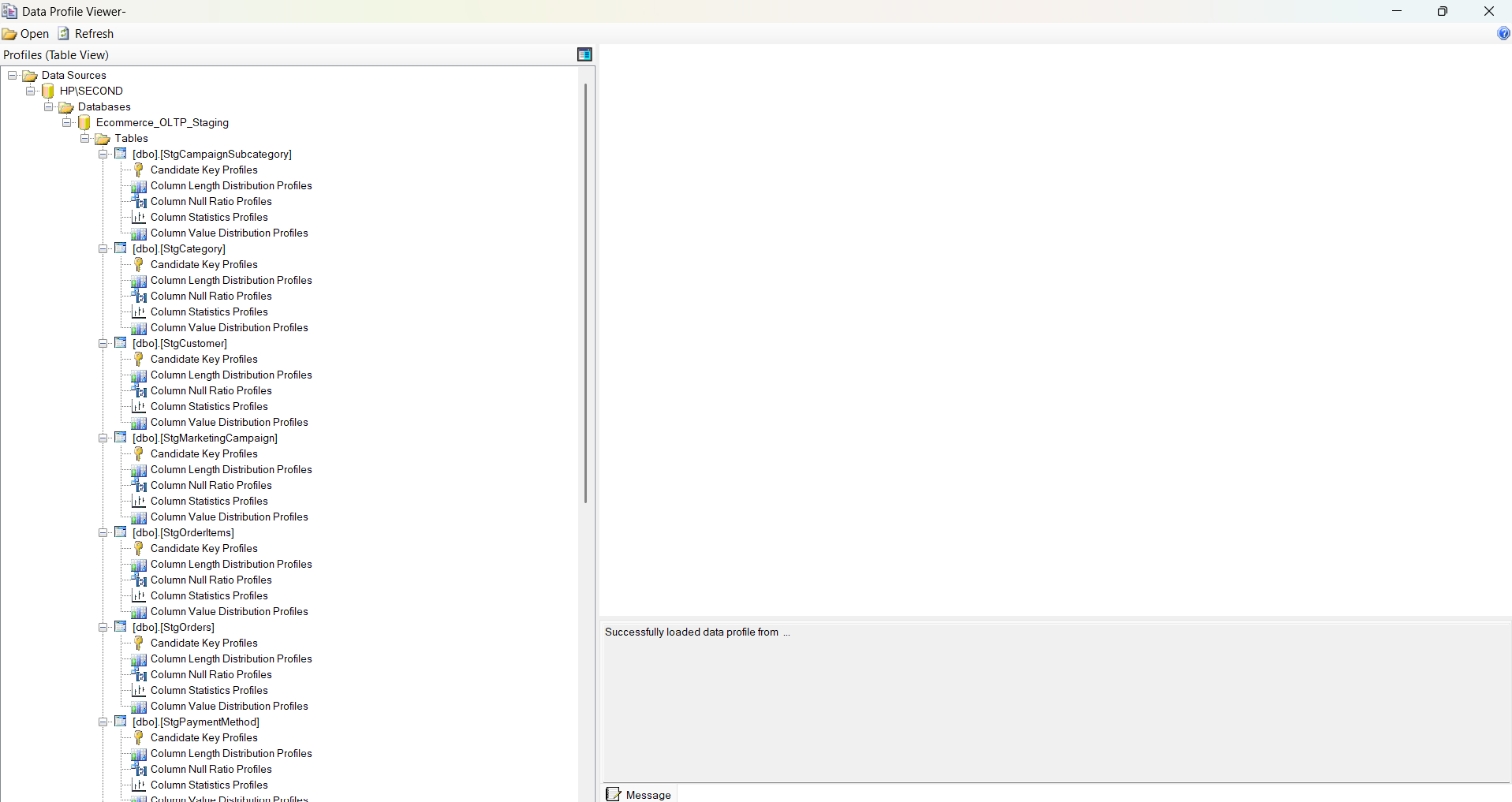
Used the flat file connection manager to properly identify the columns and data types of the flat file data.

Then used an execute sql task to truncate the data from the file before loading to the table.

This method was then applied to the remaining two flat files as well until all the data was loaded into separate staging tables.

**Data Profiling**

I used the staging table data to do some data profiling to analyze the data and to determine and understand what transformations need to be done.

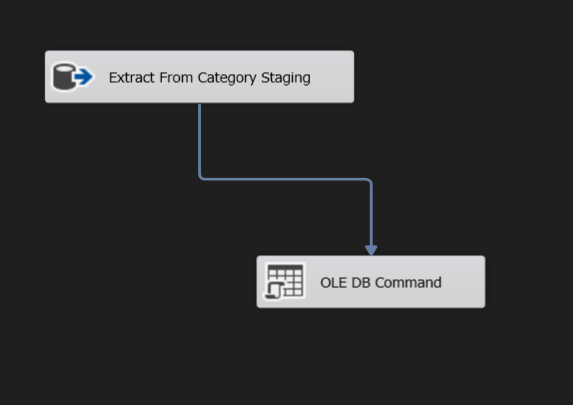


**Data Transformation**

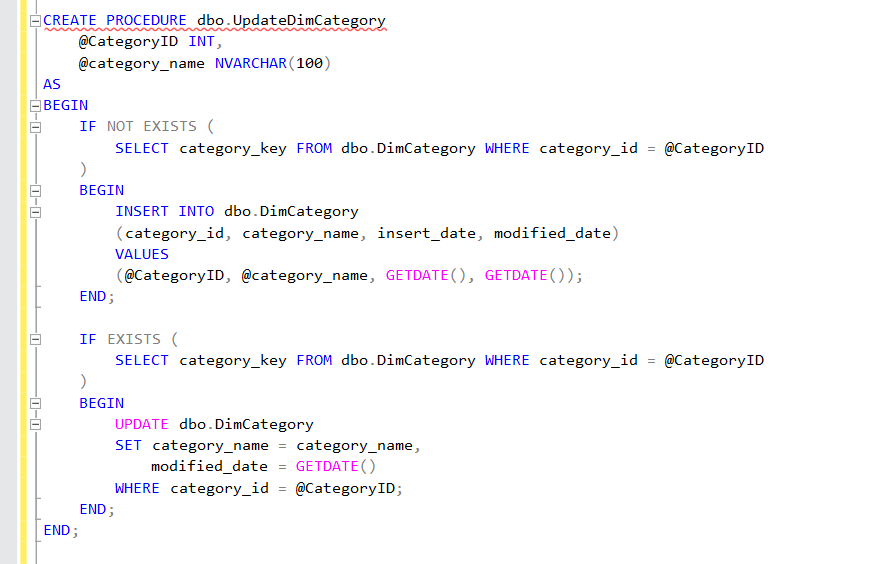
To begin the Data Transformation section I first created a new package called Ecommerce\_Load\_DW. Considering the orders and hierarchy I first loaded the category data followed by the subcategory and product tables.

**Transform and Load Category Data**

I created a new Data Flow Task within the newly created package. Dragged and dropped a OLE DB Source to extract data from the Category Staging into the Dimension Table using an OLE DB Command that contained the sql command for executing the created procedure.

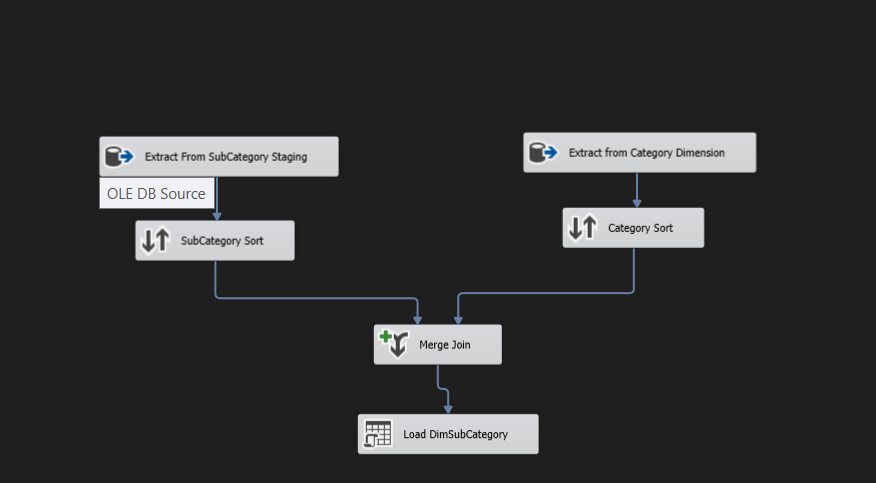


The procedure was created in the SSMS in the data warehouse database.



**Transform and Load SubCategory Data**

I created the DimSubCategory Table by sorting and merging the subcategory staging with the category dimension table as it uses the category id as well.

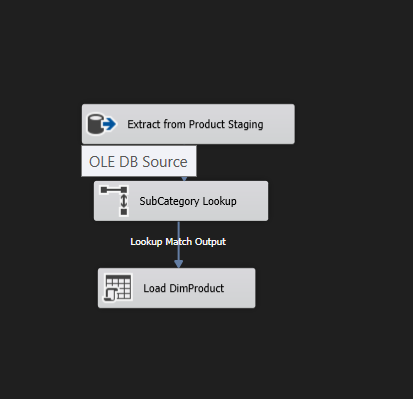


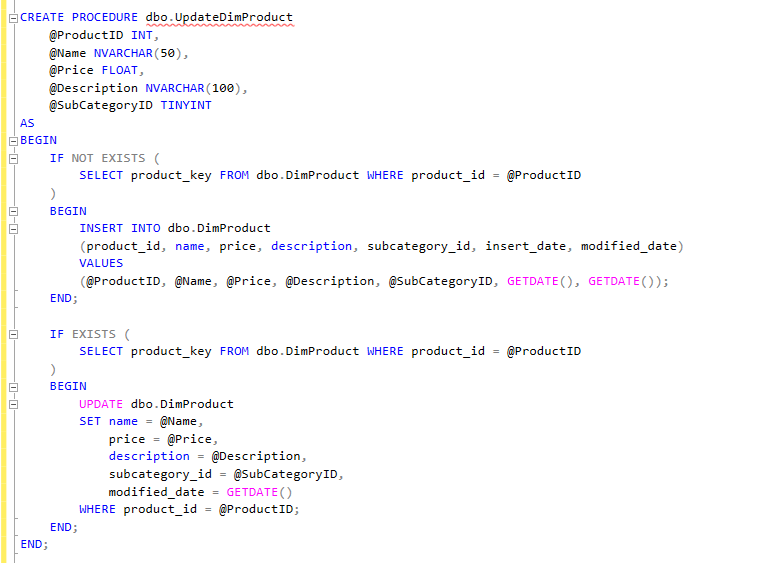
The procedure was written in the SSMS and the command to execute it was entered into the OLE DB Command.



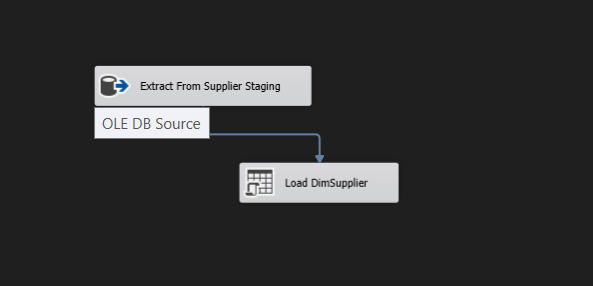
**Transform and Load ProductData**

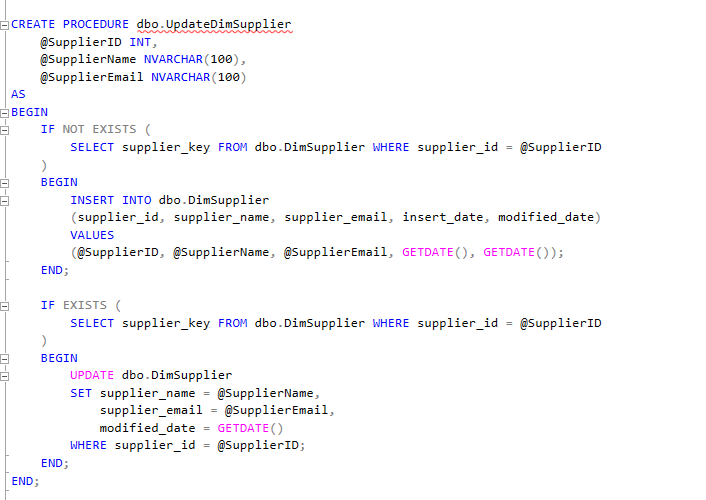
For the Product table we used a lookup as the product table includes the subcategory ID and it is a much easier and simpler method rather than using the merge and sort components. We repeated this process for the other dimension tables.



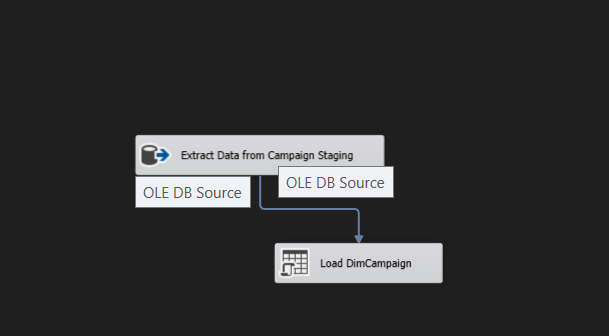


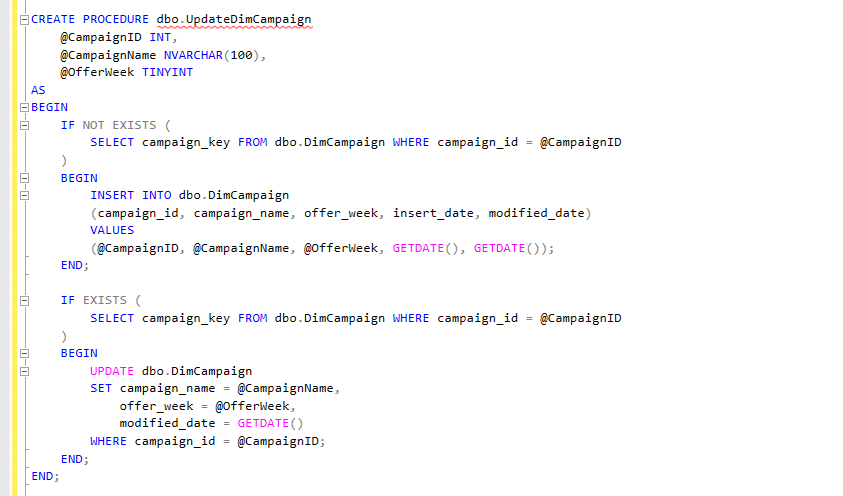
**Transform and Load Supplier Data**



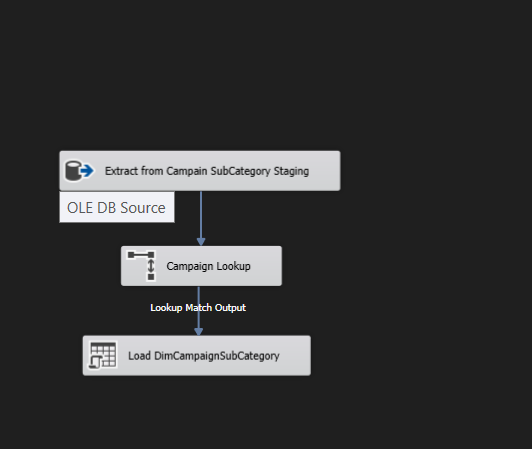


**Transform and Load Campaign Data**





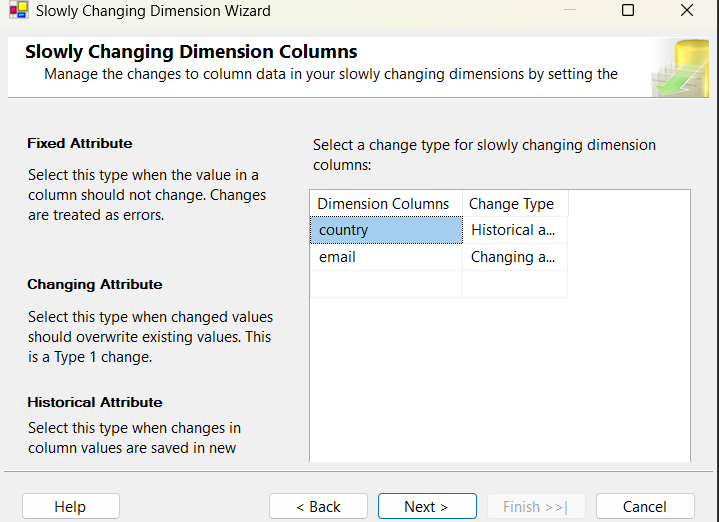
**Transform and Load Campaign SubCategory Data**

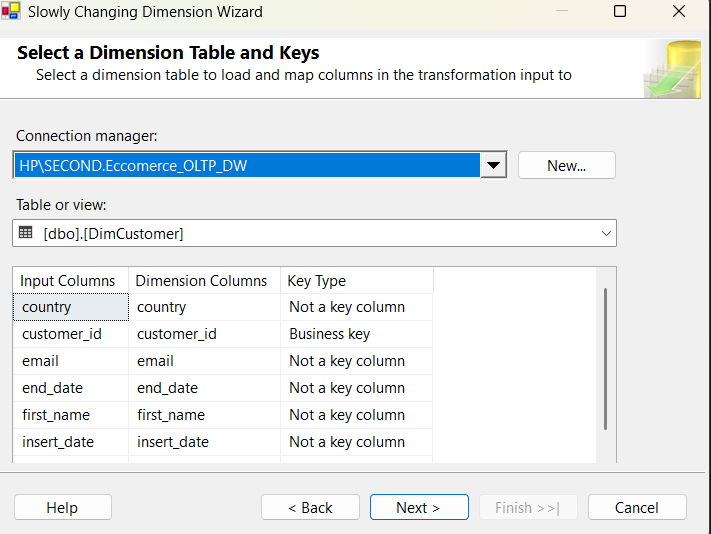


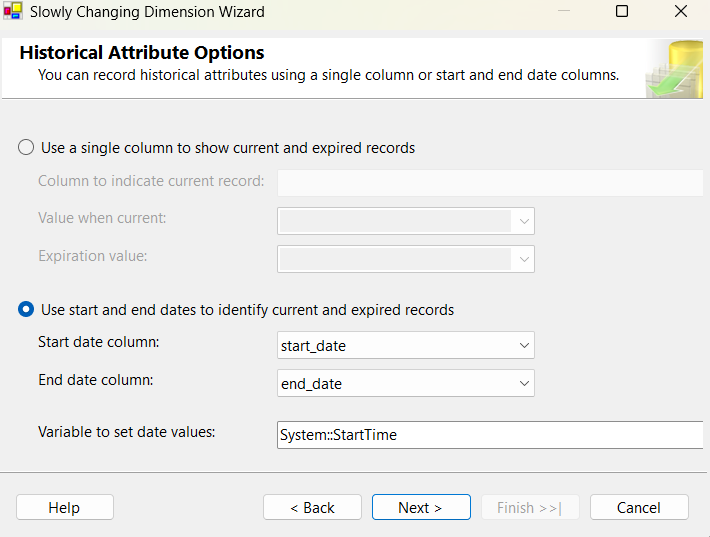


**Transform and Load Customer Data(SCD)**

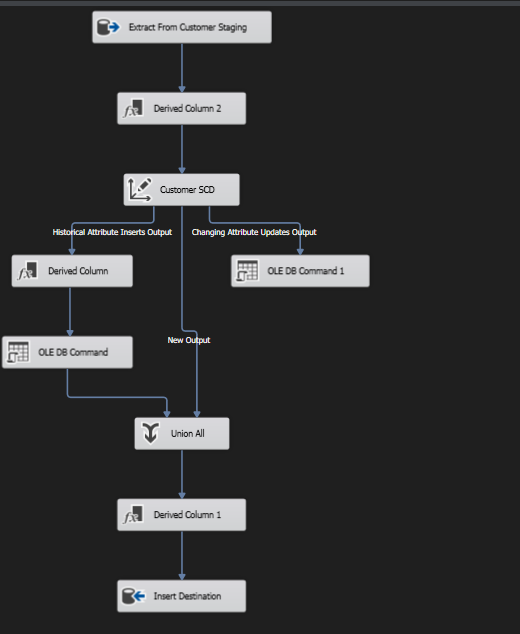
After extracting from the customer table and using a derived column to get the insert date and start date I dragged and dropped a SCD. I then set the following configurations.







Once it has all been configured the rest will be automatically be generated as seen below.



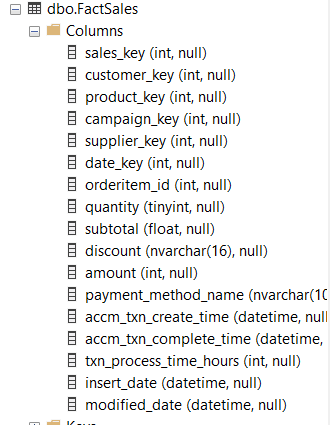
**STEP 06: ETL Development -Accumulating Fact Table**

First, I extended my sales fact table with following 03 columns.

accm\_txn\_create\_time

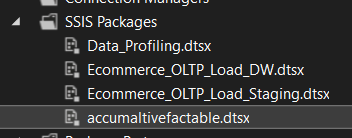
accm\_txn\_complete\_time

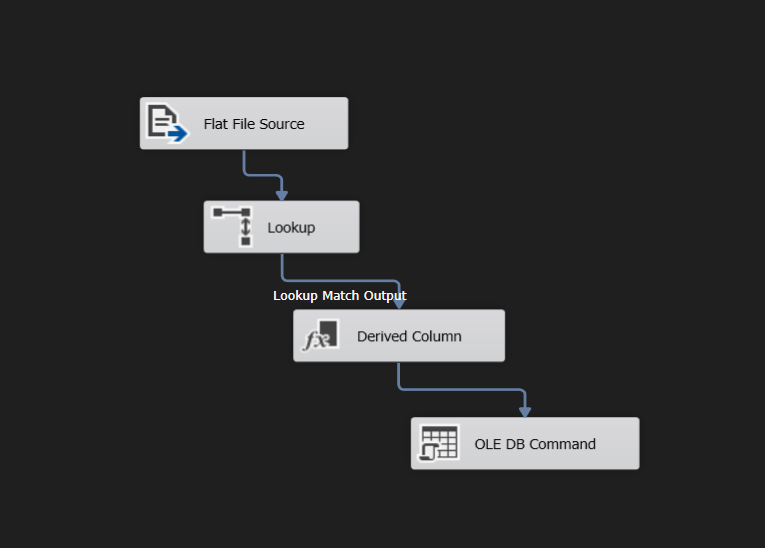
txn\_process\_time\_hours



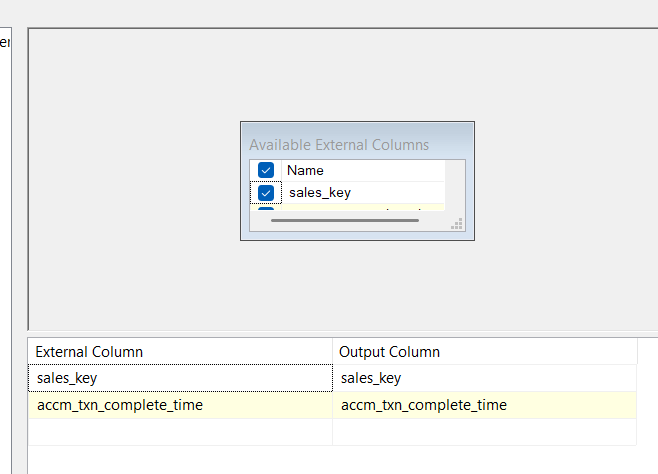
I then prepared a separate dataset for the complete time.

After that I created a new package in which I created a new dataflow task. This will be to receive updates and update the aam\_txn\_complete\_time accordingly.

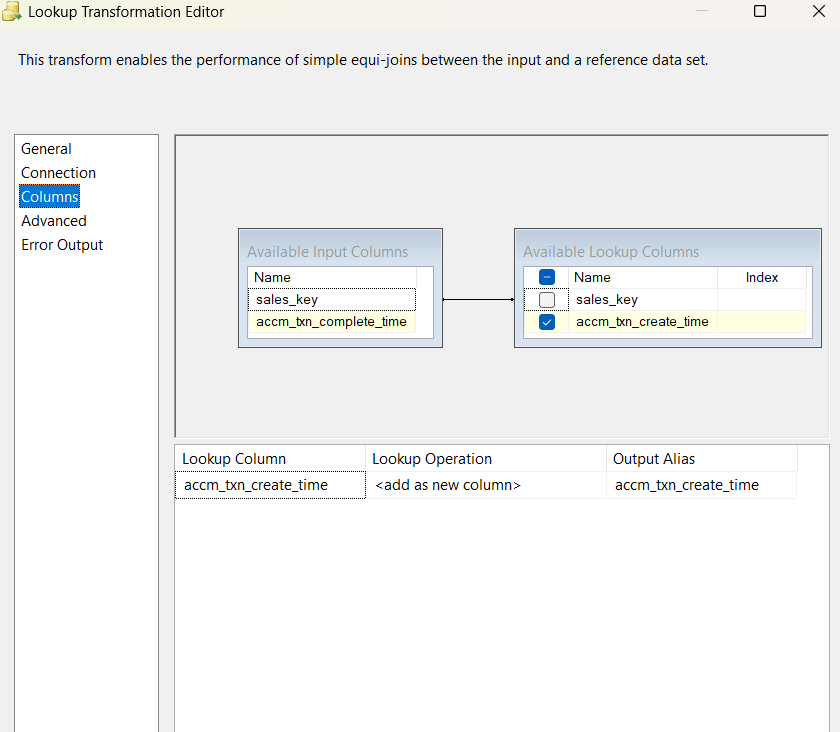


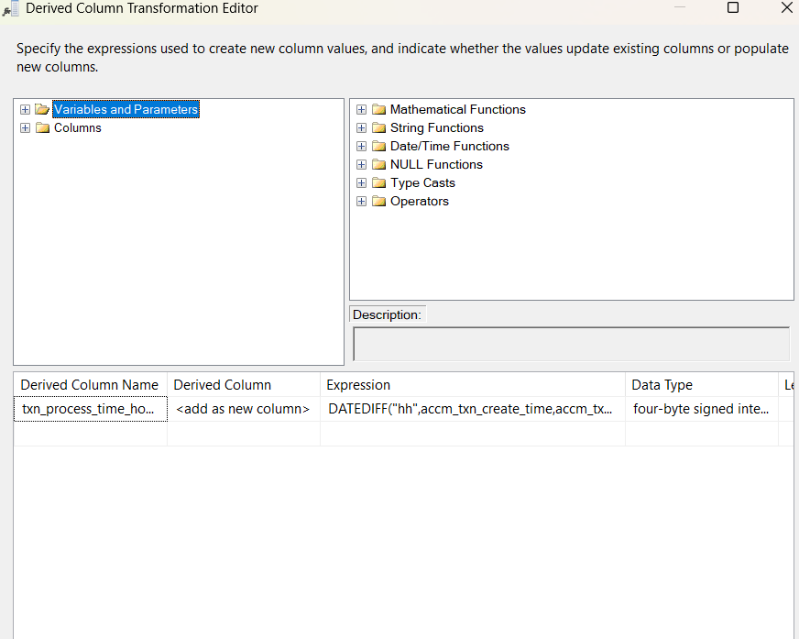


I extracted the data from the flat file

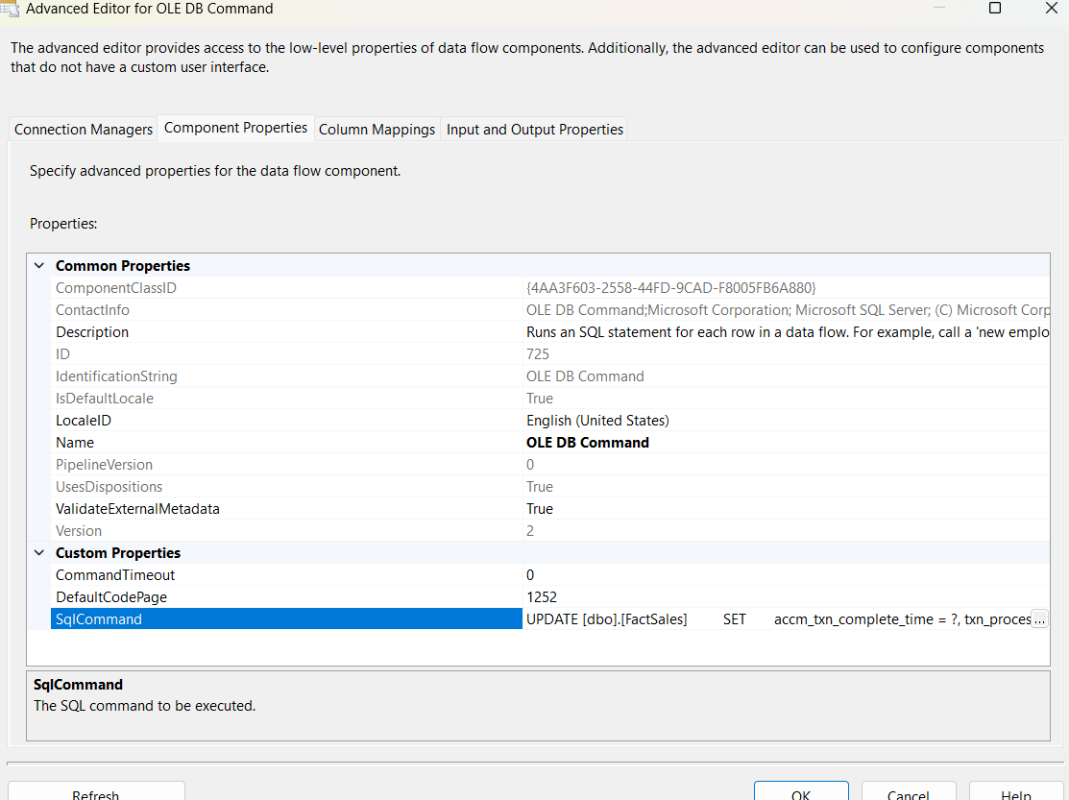


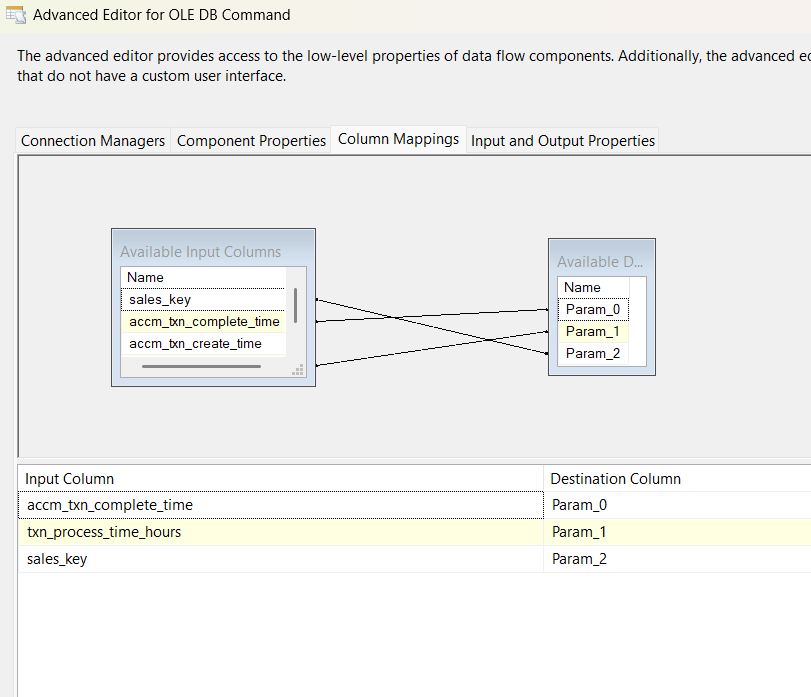
Used the lookup to match with the factSales table using the sales\_key and to retrieve the accm\_txn\_create\_time value which we need for the calculation of txn\_process\_time\_hours.

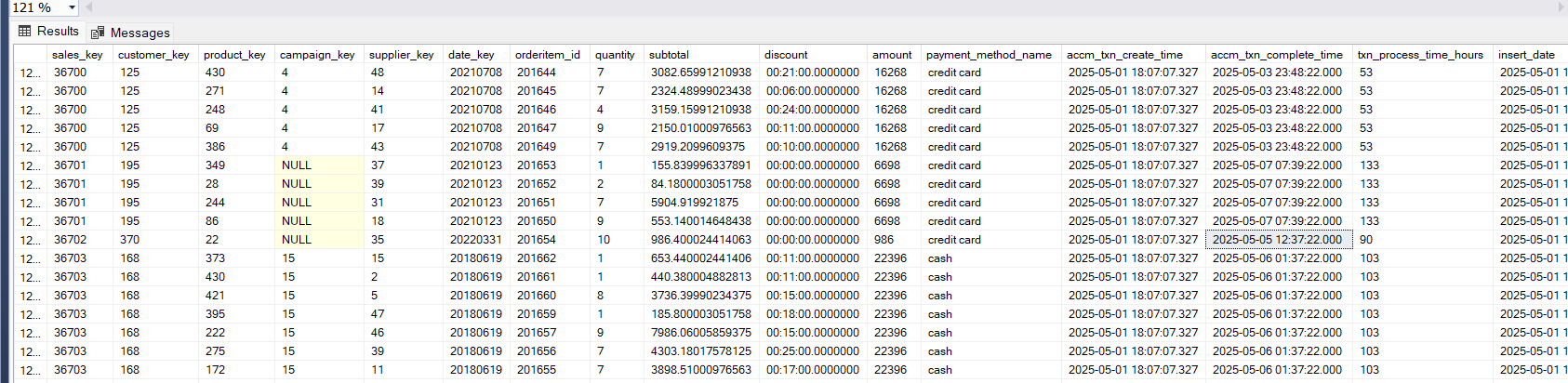


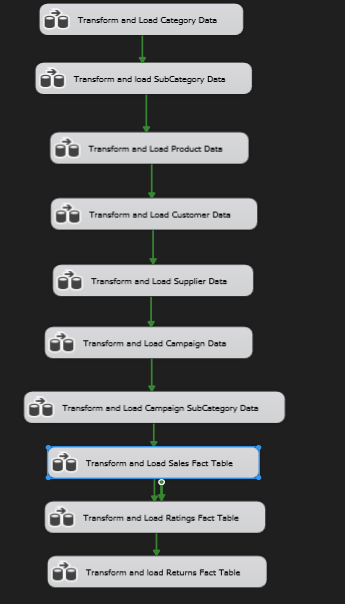
I then used the derived column to generate the value for the txn\_process\_time\_hours and to add the column.

Finally in the OLE DB Command we connected it to the Data Warehouse database and gave an sql command to update the factsales table columns accordingly and mapped the necessary parameters.





The results

**Final Control flow of the Data Warehouse**