**Sri Lanka Institute of Information Technology**

**Data warehousing and Business Intelligence**

**Assignment 1**

A picture containing text, clipart, vector graphics

Description automatically generated

**Student Registration No: IT20186142**

**Student Name: Wijesooriya H.M.A.H.**

**Step 1: Data Set Selection**

This data set contains shopping online analytics of a famous super store which centers are distributed globally. They reach out to customers who do orders online. Each online order has a relevant customer, a product, a market and a shipment mode. Through this data set, it is able to profile the customers based on their of purchases and to profile countries based on the sales.

This data set includes details about 20,000 orders which were happened throughout 4 years and the number of customers involved in these orders are over 1500. In those orders there are about 2600 significant products which varies with category and subcategory.

This dataset contains Super store details,

* Customer details
* Customers segment details
* Customer addresses
* Order details
* Market details
* Product details
* Shipment details
* Product category details
* Product subcategory details

Also, there are some added details to this database.

Following ER- diagram will describe the scenario of the selected dataset

Diagram

Description automatically generated

**Step 2: Preparation of Data Sources**

The whole of data was in ‘csv’ file type and they were separated into the following data sources, Database, Text, Excel and csv. And they were used to create the following,

**1.Database(.bak)**

Category.xls, MarketDetails.xls, OrderPriorityDetails.xls, Products.xls, SubCategory.xls and ShipmentDetails.xls files were imported to the Globa\_Super\_Store Database.

**2.Text(.txt)**

CustomerAddress.txt was used directly.

**3.Excel(.xls)**

Orders.xls was used.

**4.Comma Separated Values (.csv)**

Customers.csv was used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Source Type** | **Source Name** | **Column Name** | **Data Type** | **Description** |
| Database  File (.bak) | dbo.Products | Product\_ID | nvarchar(50) | Unique ID |
| Product\_Name | nvarchar(150) | Name of Product |
| CategoryID | int | Product Category ID |
| Sub\_CategoryID | int | Product Subcategory ID |
| dbo.Category | CategoryID | int | Unique ID |
| Category | nvarchar(50) | Product Category name |
| Dbo.SubCategory | Sub\_CategoryID | int | Unique ID |
| Sub\_Category | nvarchar(50) | Product SubCategory name |
| dbo.MarketDetails | MarketID | int | Unique ID |
| Market | nvarchar(50) | Market Name |
| Region | nvarchar(50) | Region of market |
| dbo.OrderPriorityDetails | OrderPriorityID | int | Unique ID |
| OrderPriority | nvarchar(50) | Order Priority types |
| Dbo.ShipmentDetails | ShipmentID | int | Unique ID |
| Ship\_Mode | nvarchar(50) | Shipment types |
| ShipmentDescriptions | nvarchar(100) | Shipment type descriptions |
| Excel File | Orders.xls | Row ID | nvarchar(50) | Unique ID |
| OrderID | nvarchar(50) | ID of the order |
| OrderDate | date | Order placed date |
| OrderPriority ID | int | ID of Order Priority |
| Customer ID | nvarchar(50) | ID of Customer |
| Sales | money | Sales |
| Quantity | int | Quantity |
| Discount | decimal(4, 3) | Discount of Product |
| Profit | money | Profit |
| Shipping Cost | money | Shipping Cost |
| MarketID | int | ID of Market |
| Product ID | nvarchar(50) | ID of Product |
| ShipementID | int | ID of Shipment mode |
| ShipmentDate | date | Shipment placed date |
| CSV File | Customers.csv | Customer ID | varchar(50) | Unique ID |
| Customer Name | varchar(50) |  |
| Segment | varchar(50) |  |
| Text File | CustomerAddresses.txt | Customer ID | varchar(50) | Unique ID |
| Country | varchar(50) | Customer’s Country |
| State | varchar(50) | Customer’s State |
| City | varchar(50) | Customer’s City |
| ZIP | varchar(50) | ZIP code of the Customer |
| Address | varchar(60) | Customer’s Address |

**Step 3: Solution Architecture**

**Graphical user interface, application

Description automatically generated**

Above architecture shows the high-level BI solution to the warehouse design.

**Data Sources**

‘.txt’ component represents Text files, ‘.xls’ component is used to represent Excel files, ‘.csv’ component is used to display Comma Separated files and ‘.bak’ component represents database files.

**Staging Area**

Loading DB component represents the process of the creating database tables. Test, Patient, TestPrices, AddmissionFees and Attendance text files was imported to the database and was used to create the tables. And these tables were used as the DB source data.

Staging DB component represents creating staging level tables through the ‘Extract’.

**Data Warehouse**

Data warehouse DB component is used display the cratering dimension tables in the warehouse using ‘Transform’ and ‘Load.’

**Step 4: Data Warehouse Design & Development**

Following figure will show how the fact table and dimension tables was combined in a rational manner.

Diagram

Description automatically generated

**Schema Type**

For this scenario, snowflake schema type was used.

**Dimension Types**

* Hierarchical Dimension
  + Date – all the hierarchies in date
  + Product – product 🡪 category 🡪 subcategory
  + CustomerAddress – country 🡪 state 🡪 city 🡪 ZIP 🡪 address
* Slowly Changing Dimension
  + Customer – used type 2.
  + Following columns were set as changing attributes.
    - Segment
    - Country
    - State
    - City
    - ZIP code
    - Address
* Fact Table
  + Numbers – Sales, Quantity, Discount, Profit, ShippingCost, FinalCost
  + FK - CustomerKey, OrderDateKey, OrderPriorityKey, MarketKey, ProductKey, ShipmentKey

**Assumptions**

* Customer dimension was considered as a slowly changing dimension.

**Step 5: ETL development**

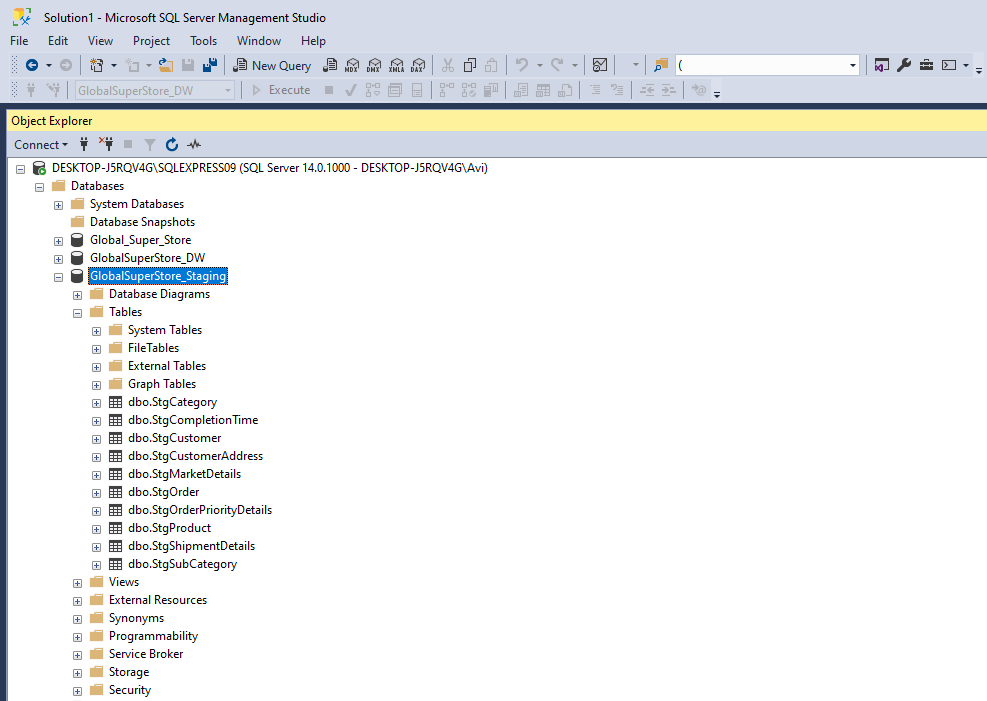
**1.Extract**

In this step, All the data sources were imported to the staging tables by using the relevant Data connection.

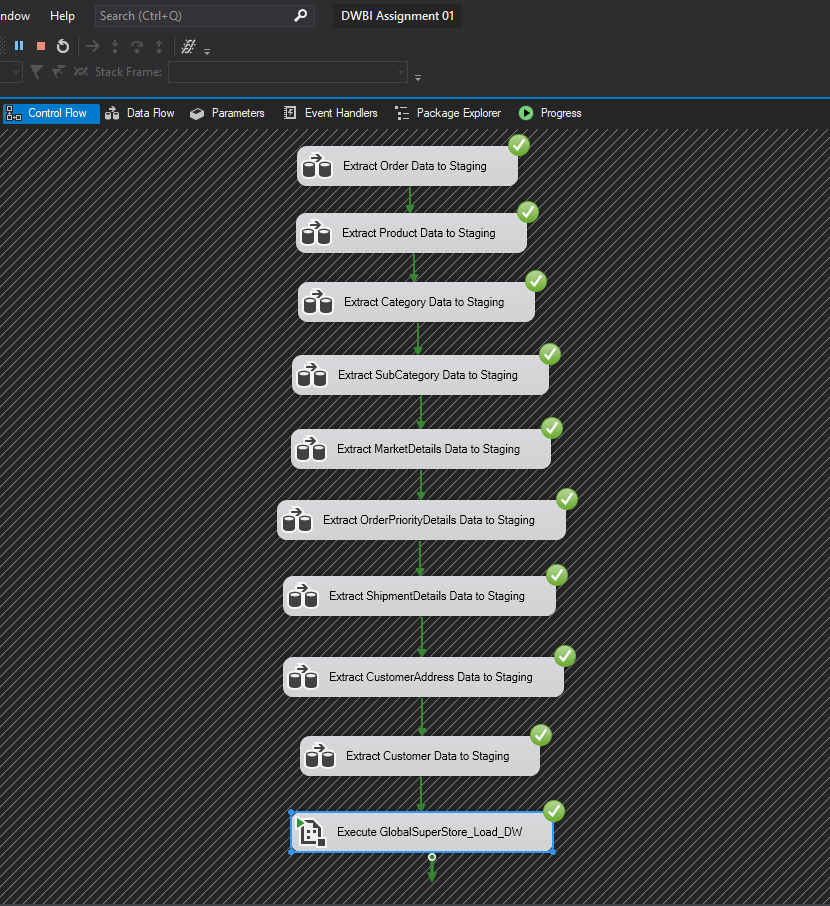
Flat file connection was used for text files and csv files, Excel file connections for excel file, DB source connection for DB file. All those tables were imported to the GlobalSuperStore\_Staging DB, which contains the below tables,

1. StgCategory
2. StgCustomer
3. StgCustomerAddress
4. StgMarketDetails
5. StgOrder
6. StgOrderPriorityDetails
7. StgProduct
8. StgShipmentDetails
9. StgSubCategory
10. StgCompletionTime

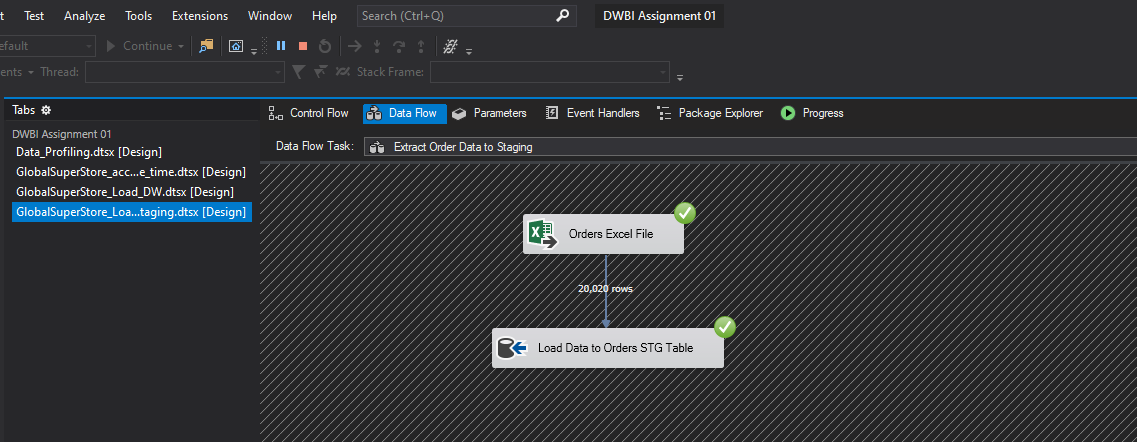
* **Snapshot of SSMS Staging Database**

****

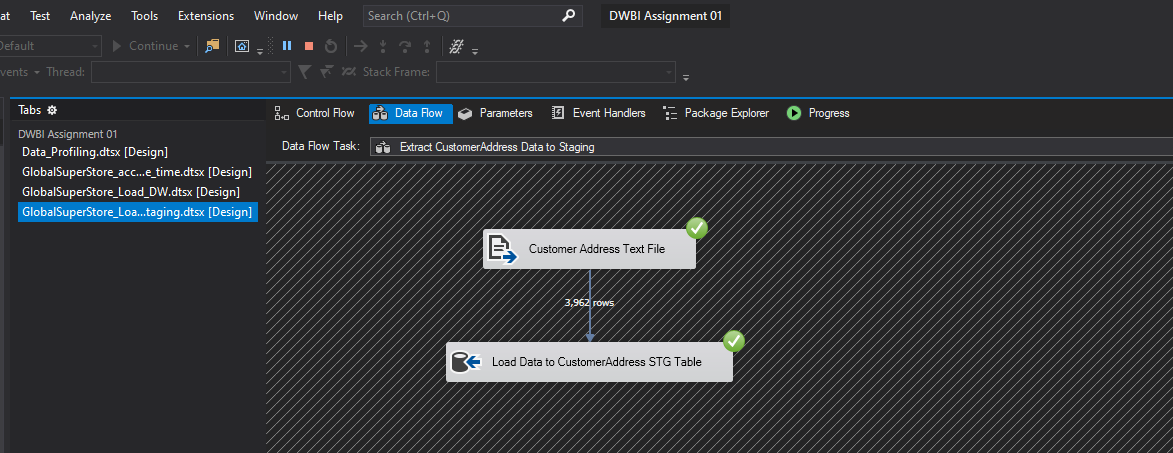
* **Snapshot of Visual Studio Control Flow of Extract**

****

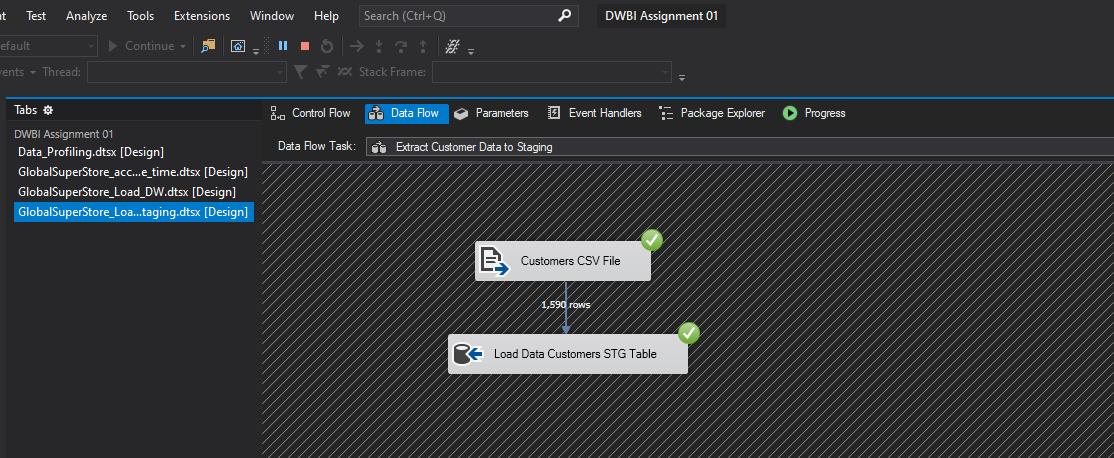
* **Snapshots of several data types of Data Flows**



Orders Data Staging



Customer Address Data Staging



Customers Data Staging

* **Event Handling (Truncate Staging Data)**

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, website

Description automatically generated

* **Data profiling**

Used Data\_Profiling package to profiling the staging tables

Table

Description automatically generated with medium confidenceText

Description automatically generated

**3.Transform & Load**

In this step, both the ‘Transform’ and ‘Load’ are done. Firstly, The Dimension tables in the Datawarehouse DB data were created. Then, using the relevant components, data from the staging tables was loaded into the warehouse tables, GlobalSuperStore\_DW, which contains the below tables.

1. DimCategory
2. DimCustomer
3. DimDate
4. DimMarketDetails
5. DimOrderPriorityDetails
6. DimProduct
7. DimShipmentDetails
8. DimSubCategory
9. FactOrders

**Used Transformation Tasks**

1. Lookups

DimCustomer’s CustomerID is looked when loading using DimOrder

DimProduct’s ProductSK is looked when loading using DimMarket

1. Derived Columns

Derived column is used in FactOrders to derive both StartDate and EndDate by using GETDATE() expression and to derive the FinalCost too.

1. Union

Union is used in the Extract step to combine and get all the data from data files.

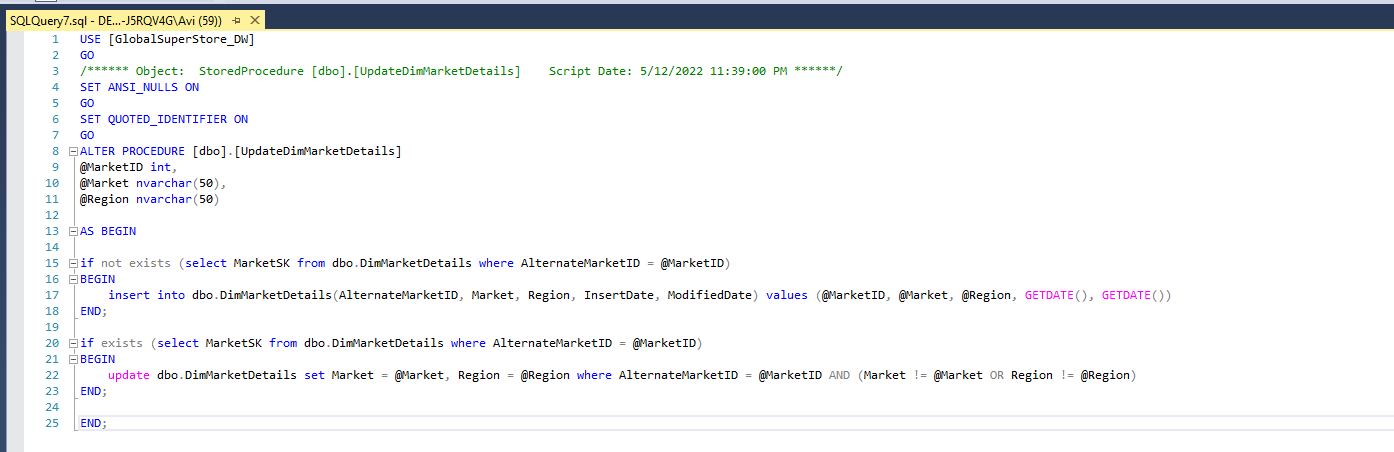
1. Sort and Merge

‘Sort’ is used in DimProduct to sort out the Product and Category data and they are merged using CategoryID.

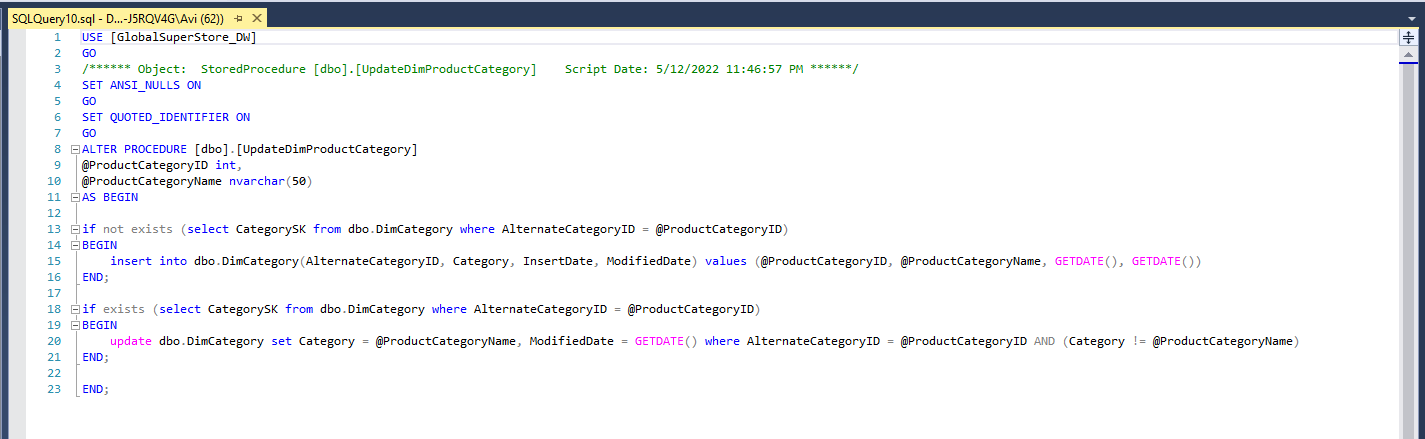
‘Sort’ is used sort out the Product and Sub-Category data and they are merged using SubCategoryID.

**Update Functions**

* DimMarketDetails

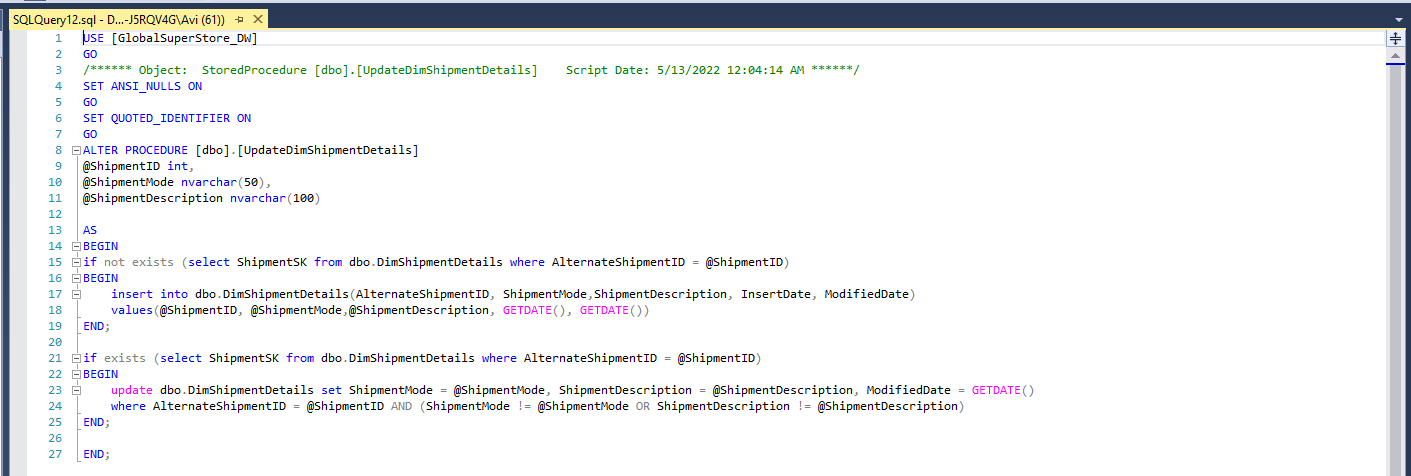


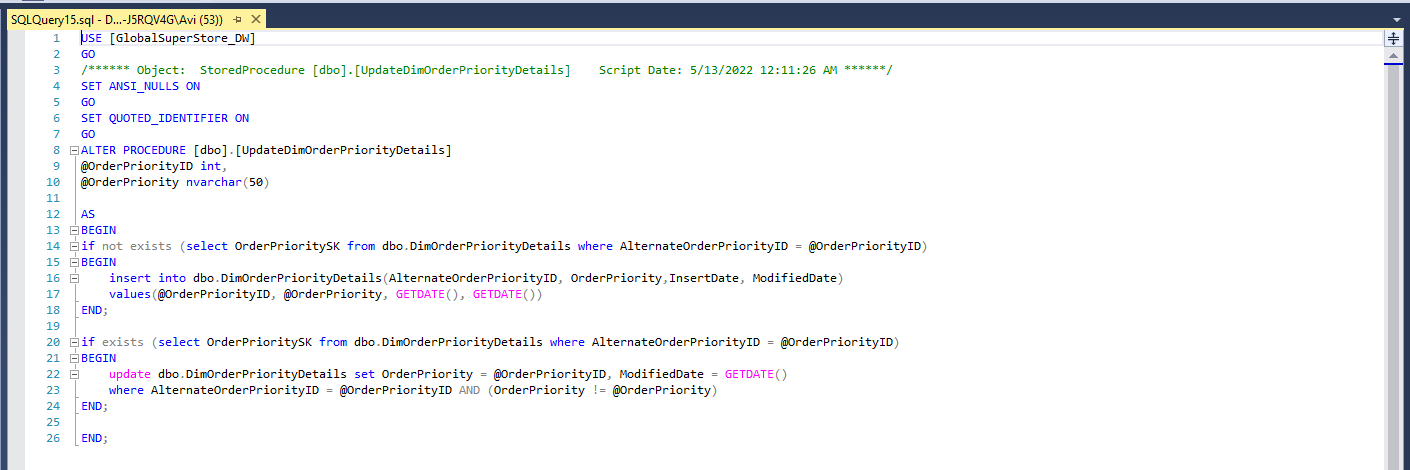
* DimProductCategory

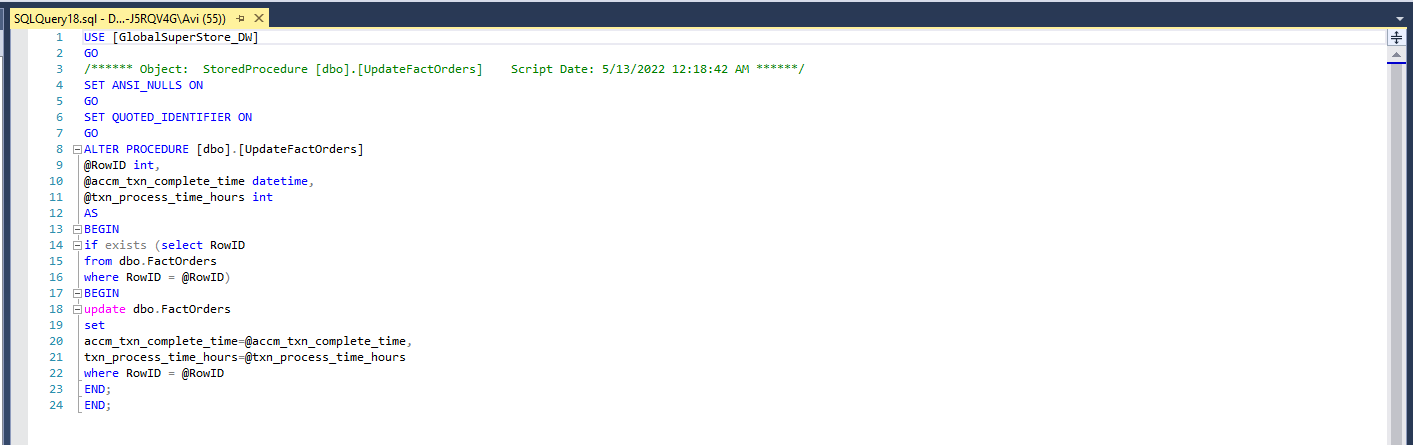


* Graphical user interface, text, application, email

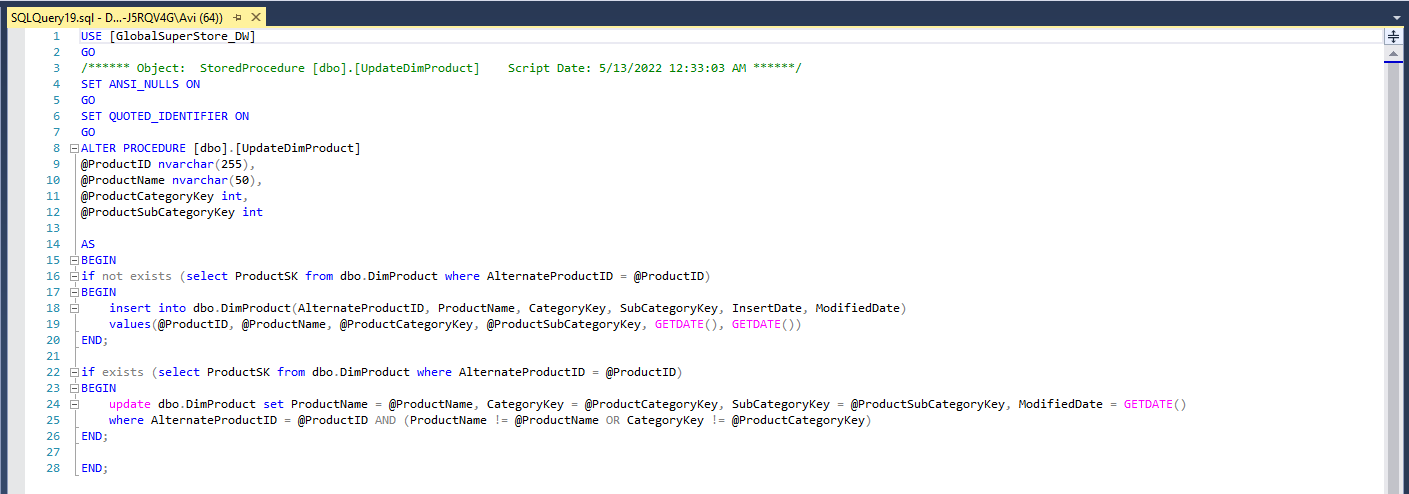
  Description automatically generatedDimProductSubCategory
* DimShipmentDetails



* DimOrderPriorityDetails
* FactOrders



* DimProduct



* **Snapshot of SQL server Data warehouse Database**

Graphical user interface, text, application

Description automatically generated

* **Snapshot of Visual Studio Control Flow of Extraction**

A screenshot of a computer

Description automatically generated with medium confidence

* **Product Category Data Transform and Load**

Graphical user interface, application

Description automatically generated

* **Product Subcategory Data Transform and Load**

Graphical user interface, text, application

Description automatically generated

* **Market Details Data Transform and Load**

Graphical user interface, application

Description automatically generated

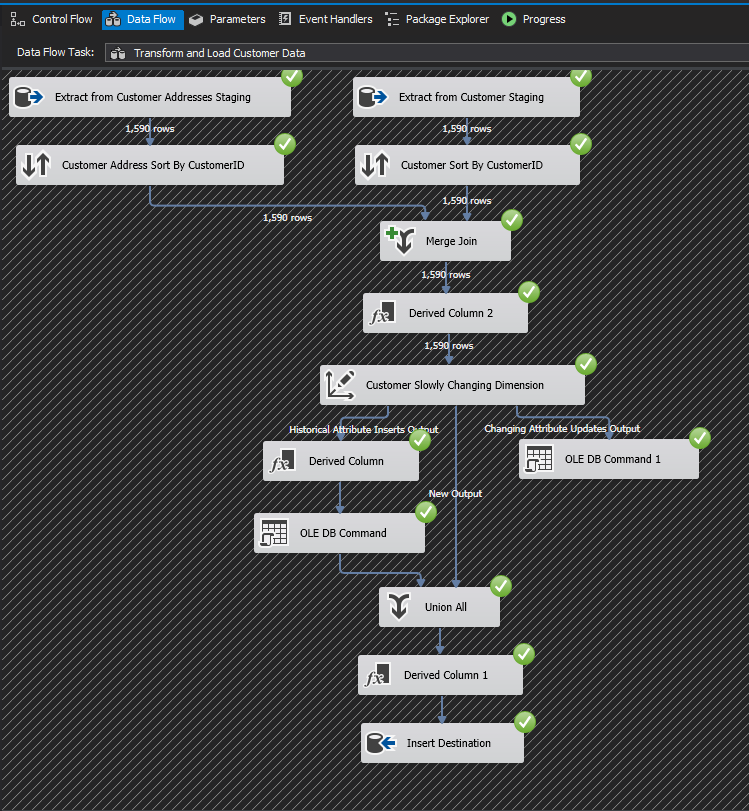
* **Product Data Transform and Load**

**Graphical user interface, diagram, application

Description automatically generated**

* A screenshot of a computer

  Description automatically generated with medium confidence **Shipment Details Data Transform and Load**
* **Customer Data Transform and Load**



* **Accumulating Fact Table**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generated