Logo, company name

Description automatically generated

**IT3021- Data Warehousing & Business Intelligence**

**Assignment 1 -Report**

**IT20237554**

**Rathnaweera R.P.W.G**

Table of Contents

[Data set selection -Step 1 3](#_Toc103072729)

[Preparation of the Data Sources – Step 2 4](#_Toc103072730)

[Solution Architecture – Step 3 6](#_Toc103072731)

[Data warehouse design & development -Step 4 8](#_Toc103072732)

[ETL development – Step 5 10](#_Toc103072733)

[Extract Data from Source Files 10](#_Toc103072734)

[Data Profiling 15](#_Toc103072735)

[Data Transformations 16](#_Toc103072736)

[ETL development: Accumulating Fact tables – Step 6 25](#_Toc103072737)

[**Step 1:** Extended the fact table with the below columns, 25](#_Toc103072738)

[Step 2: Creating a csv file with Business Key and date time. 25](#_Toc103072739)

[Step 3: Creating Separate package in SSIS 26](#_Toc103072740)

[Step 4: Creating a control Flow 26](#_Toc103072741)

[Step 5: Updating FactOrder table with necessary columns 27](#_Toc103072742)

# Data set selection -Step 1

I had selected a Retail dataset of a global superstore with records for 4 years (2011-2014). The data set consists of different details such as order details, address details, customer details, product details. All the details are stored in one .csv file with 25 columns. For the above data set I had chosen a scenario where customer order a product and those order details including customer details will be stored.

Diagram

Description automatically generated**ER diagram**

# Preparation of the Data Sources – Step 2

One common csv file in the dataset had been separated into 5 unique csv files and one text(.txt) file. after the separation, created a source database and named it as “Superstore\_SourceDB” and loaded the above separated unique 5 csv file into that DB. And the separated text file is directly saved as “CustomerAddress.txt”.

Source table “Superstore\_SourceDB” details are as below,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Source Type | Source Name | Description | Column Name | Data Type | Description |
| Database  File (.bak) | dbo.CustomerDetails | CustomerDetails | CustomerID  CustomerName  CustomerSegment | int  nvarchar(50)  nvarchar(50) | Unique ID to identify the customer  Customer Name  Segment of the customer |
|  | dbo.Order | Order | OrderID  OrderNum  CustomerID  ProductID  OrderDate  Sales  Profit  Discount  ShipDate  Quantity  ShipMode  Market  OrderPriority  ShippingCost | int  int  int  int  datetime  money  money  money  datetime  int  nvarchar(50)  nvarchar(50)  nvarchar(50)  money | Unique ID to identify the order  Order number  CustomerID  ProductID  Date of the order  Price per item  Profit received from a single order  Discount given for a single order  Shipping date of the ordered products  No.of products  The shipment way  Market name  Priority of the order  Cost for Shipping products |
|  | dbo.Product | Product | ProductID  ProductName  SubCategoryID | int  nvarchar(50)  int | Unique ID to identify the Product  Name of the product  SubCategoryID |
|  | dbo.SubCategory | SubCategory | SubCategoryID  SubCategoryName  CategoryID | int  nvarchar(250)  int | Unique ID to identify the Product subcategory  Name of the subcategory  CategoryID |
|  | dbo.Category | Category | CategoryID  CategoryName | int  nvarchar(250) | Unique ID to identify the Product category  Name of the Category |
| Text File  (.txt) | CustomerAddress.txt | CustomerAddress | AddressID  CustomerID  Country  Region  State  City  PostalCode | int  nvarchar(30)  nvarchar(250)  nvarchar(250)  nvarchar(250)  nvarchar(250)  nvarchar(250) | Unique ID to identify the location  CustomerID  Country of the customer  Region of the customer  State of the customer  City of the customer  PostalCode of the customer |

# A picture containing timeline Description automatically generated Solution Architecture – Step 3

As per the above show High-level BI solution architecture.

* **Data Sources:**

Data sources are also knowing as source system layer where this contains all sources that need to develop the database as in the above scenario superstore\_sourceDB and the Customeraddress text file

* **Staging Area:**

The above data is extracted from the source system to the staging are using ELT, in here the loading component shows the process of creating database tables, in my scenario Customer, Order, Products, Category, SubCategory and the CustomerAddress text files were used to import the database and to create tables.

On this staging level the “Extract” of ETL process happens. In the “Transform”, identified the dimension tables and developed transform process to those table and the last step is “Load”, Loading data into the target warehouse is done.

* Data Warehouse

Using ‘Extract’, ‘Transform' and 'Load,' the data warehouse DB component displays the warehouse's cratering dimension tables.

* Consumption / BI

This layer is also known as Presentation layer where all the interactions of the end users are done in here. The presentation layer consists of different tools.

As an example, they can access the warehouse using client access tools such as Data mining tools, online analytical processing tools executive information system tools, application development tools and report & query tools.

# Data warehouse design & development -Step 4

**Diagram, schematic

Description automatically generated**

In a DW Fact table consist of measurements (grain) metrices or facts about the business process in here in the dimension model Fact table located at the center of the schema, and it’s surrounded by all the dimension tables such as DimCustomer, DimProduct, DimDate. For the above selected scenario, I had chosen the concept of Snowflake Schema to develop the dimension tables and the fact table since snowflake is an extension of star schema and because I have added a normalized table named DimProduct which related another dimension like DimProductSubCategory.

In the Data Warehouse two-dimension tables named DimCustomer, DimProduct is directly connected to the fact table and one common dimension named DimDate.

DimDate dimension consists of 37 attributes and the DateKey is used as the Surrogate Key

DimProduct dimension consist of 4 attributes, ProductSK is taken as the Surrogate Key and ProductSubCategoryKey is a Foreign Key(FK) Which refers to DimProductSubCategory

DimProductSubCategory consist of 4 attributes, ProductSubCategorySK is taken as the Surrogate Key and ProductCategoryKey is a Foreign Key (FK) which refers to DimProductCategory

DimProductCategory consists of 3 attributes, ProductCategorySK is taken as the Surrogate Key.

FactOrder consists of 16 attributes, CustomerKey, ProductKey, DateKey are the Foreign Keys (FK) and Row\_Order\_ID is taken as the business key (natural Key) for the FactOrder table

**Assumptions:**

* Assumed that the slowly Changing Dimension (SCD) of the Data Warehouse as the Customer Dimension since this table must have historical attributes which can be change over the time and we must maintain a history about them, and it has been designed with the Type 2 slowly changing dimension. I have added 2 columns as StartDate and EndDate to the SCD.

Following columns were set as changing attributes:

City

Country

PostalCode

Region

State

**Calculations:**

* For the Cost, TotalCostPerItem columns in the FactOrder table I had used some calculations,
  + - Cost = ([Sales]+[ShippingCost])
    - TotalCostPerItem = (([Sales]-[discount]) \* [OrderQuantity]+[ShippingCost])

# ETL development – Step 5

## Extract Data from Source Files

First step of SSIS ETL process is extracting data from sources in this scenario I had used 2 data sources, Source Database and a Flat file

1. First, I had extracted the Customer Data from SuperStore\_SourceDB to staging.

**Data Flow task – Extract Customer Data to Staging**

Graphical user interface, application

Description automatically generated

If I execute this task for multiple times the staging table will be loaded with customer details without truncating data. (Data will be duplicate). Therefore, I have added an Event Handler to truncate data when loading to the staging table.

**Executable: Extract Customer Data to Staging Event Handler: OnPreExecute**

**Truncate table [dB].[CustomerStaging]**

Graphical user interface, text, application

Description automatically generated

I had used the Event Handler to truncate the Data for all the Stating tables.

1. Extracted the Product Data from SuperStore\_SourceDB to staging

**Data Flow task – Extract Product Data to Staging**

Graphical user interface

Description automatically generated

**Executable: Extract Product Data to Staging Event Handler: OnPreExecute**

**Truncate table [dB].[ProductStaging]**

**Graphical user interface, website

Description automatically generated**

1. Extracted the Order Data from SuperStore\_SourceDB to staging

**Graphical user interface, application

Description automatically generatedData Flow task – Extract Order Data to Staging**

**Executable: Extract Order Data to Staging Event Handler: OnPreExecute**

**Graphical user interface, application

Description automatically generatedTruncate table [dB].[OrderStaging]**

1. Extracted the Product Category Data from SuperStore\_SourceDB to staging

**A screenshot of a computer

Description automatically generatedData Flow task – Extract Product Category Data to Staging**

**Executable: Extract Product Category Data to Staging Event Handler: OnPreExecute**

**Truncate table [dB].[ProductCategoryStaging]**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

1. Extracted the Product Sub Category Data from SuperStore\_SourceDB to staging

**Data Flow task – Extract ProductSubCategory Data to Staging**

Graphical user interface

Description automatically generated

**Executable: Extract Product Sub Category Data to Staging Event Handler: OnPreExecute**

Graphical user interface, application

Description automatically generated**Truncate table [dB].[ProductSubCategoryStaging]**

1. Extracted CustomerAddress Data form CustomerAddress.txt to staging

A screenshot of a computer

Description automatically generated with medium confidence**Data Flow task – Extract CustomerAddress Data to Staging**

**Executable: Extract CustomerAddress Data to Staging Event Handler: OnPreExecute**

**Truncate table [dB].[CustomerAddressStaging]**

**Graphical user interface, text

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidenceFull Control Flow of extracting data to Staging**

## Data Profiling

After performing the data staging part, I had done the data profiling since when profiling data, I can use the staging tables to analyze and determine what kind of transformation that we need to do for the data.

Graphical user interface, application

Description automatically generated**Data\_Profiling.dtsx**

## Data Transformations

1. Transform and Load ProductCategory Dimension

First, I had used ProductCategory Staging table and load that Data into DimProductCategory dimension since other tables has references, I created this table first. Since no foreign keys are in the product category dimension.

Diagram

Description automatically generated with medium confidence**Transform and Load ProductCategory Data Flow**

I had used a Stored Procedure to check whether the inputted data is exists or not if it exists it will update else it will insert as usual.

Graphical user interface, text, application, email

Description automatically generated

1. Transform and load ProductSubCategory Dimension

Here the product Subcategory staging table has the productCategoryKey where it refers to the product Category Dimension therefore, I had got data from Subcategory staging table and product category dimenstion table and sort the ids using Sort component. After sorting I used merge sort component to merge those data and finally inserted to the DimProductSubCategory dimension table.

**Graphical user interface, diagram, application

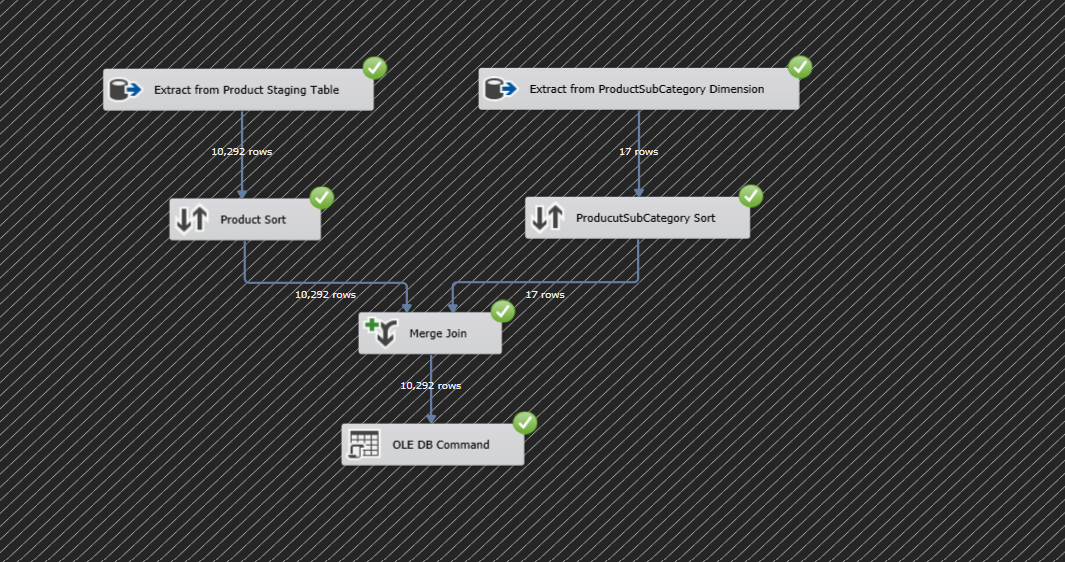
Description automatically generated** **Transform and Load ProductSubCategory Data Flow**

Graphical user interface, text, application, email

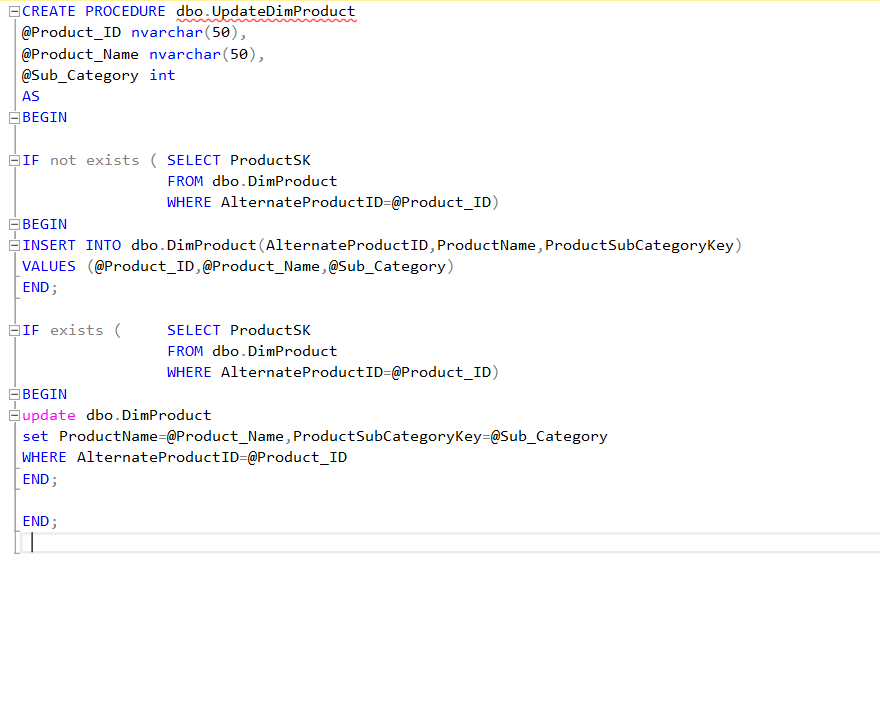
Description automatically generatedI had used a Stored Procedure to check whether the inputted data is exists or not if it exists it will update else it will insert as usual.

1. Transform and load Product Dimension

Here product staging table has the ProductSubCategoryKey where it refers to the ProductSubCategory Dimension therefore, I had got data from Product staging table and productSubCategory dimension table and sort the ids using Sort component. After sorting I used merge sort component to merge those data and finally inserted to the DimProduct dimension table.



I had used a Stored Procedure to check whether the inputted data is exists or not if it exists it will update else it will insert as usual



1. Transform and load Customer Dimension

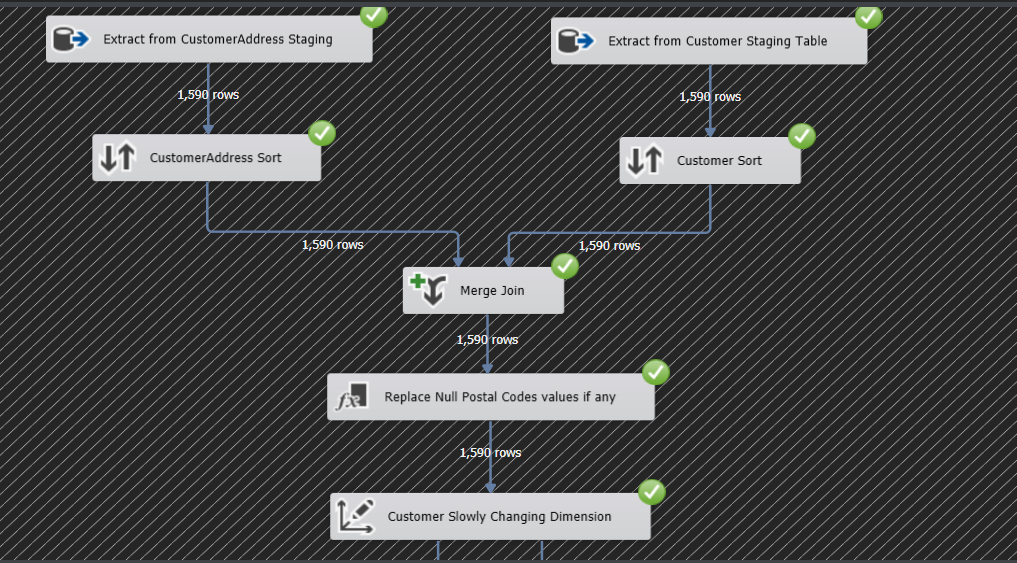
Here CustomerAddress staging table has CustomerID and address related details and Customer Staging table has the customer other details including customer id since the I had used the sort components to sort these and merge them using the merge component. Since I have some null values in the PostalCode column I had used a derived column and using the function,

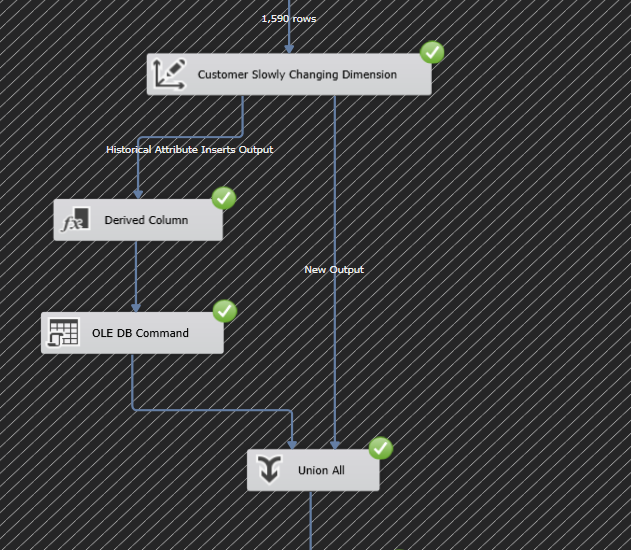
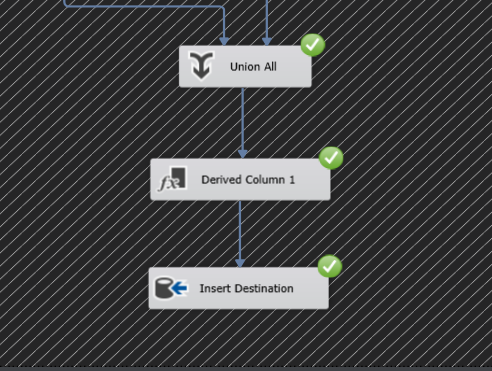
“REPLACENULL(PostalCode,"NA")”,

I had replaced those null values with word “NA”. Since customer is a Slowly Changing Dimension (SCD) I had to maintain the historical data I had used another derived column to set the EndDate and update the DimCustomer using

“UPDATE [dbo].[DimCustomer] SET [EndDate] = ? , ModifiedDate = GETDATE() WHERE [AlternateCustomerID] = ? AND [EndDate] IS NULL” this sql statement.

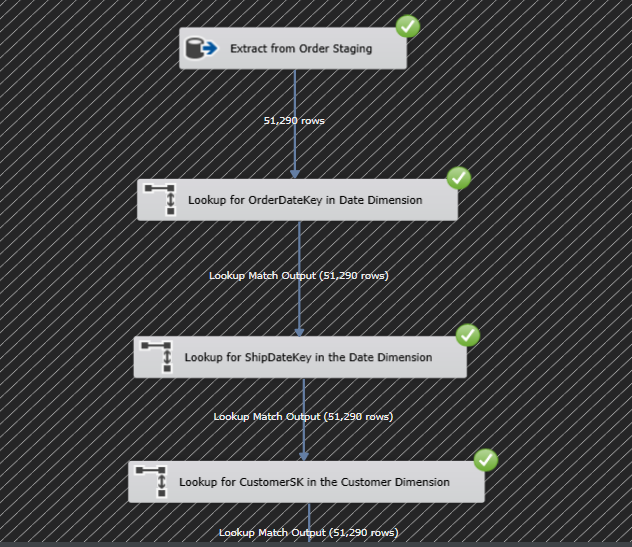
And union all the above-mentioned components together using union component, Used another derived column to set the StartDate and Insert the Data into DimCustomer Dimension table in the Data Warehouse

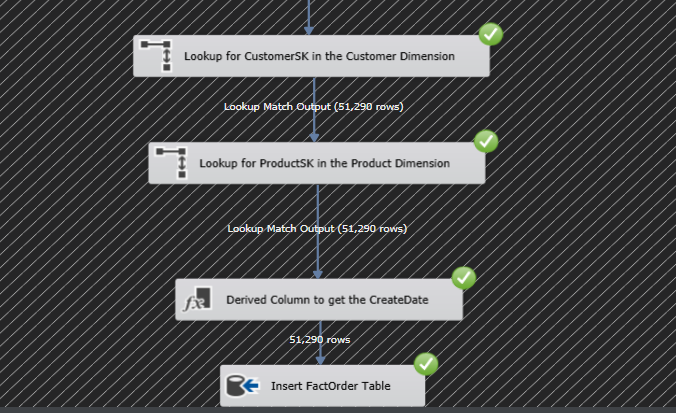




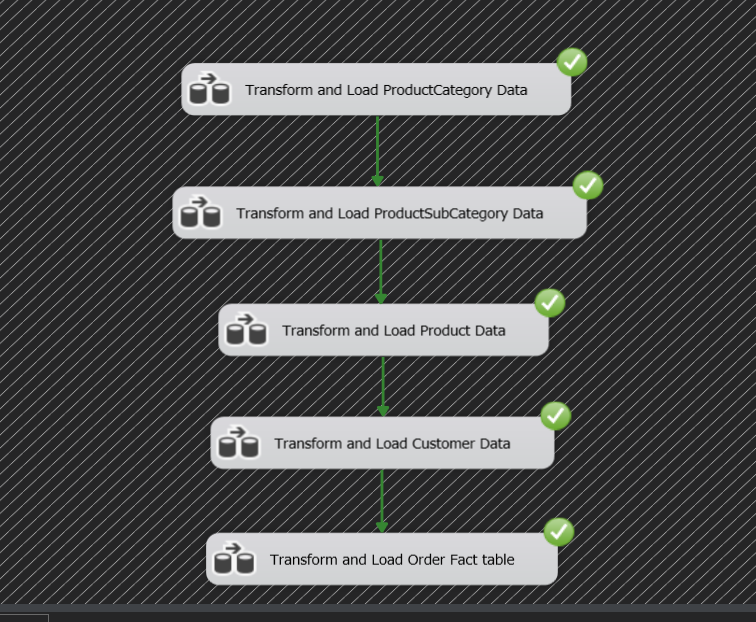
1. Transform and Load FactOrder Table.

First Extracted Data from Order Dimension then using a Lookup component I had gathered the OrderDateKey from the DimDate Dimension, ShipDateKey from the DimDate Dimension, CustomerSK from the DimCustomer dimension, ProductSK from the DimProduct Dimension Since all these are FK where it refers to the mentioned dimensions And I had used a Derived column where to get the CreateDate using the GETDATE () function. and finally, I had inserted all these data to the FactOrder Table in the Data Warehouse.





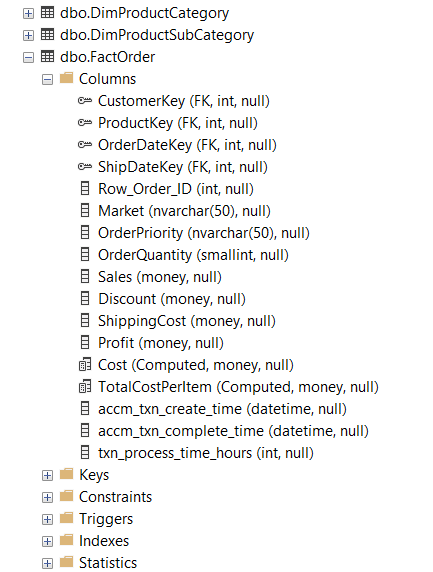
**Control Flow of Loading Data from Staging to DW**

****

# ETL development: Accumulating Fact tables – Step 6

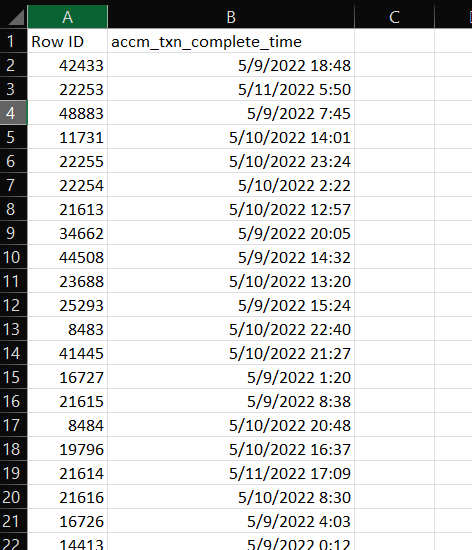
## **Step 1:** Extended the fact table with the below columns,

* accm\_txn\_create\_time
* accm\_txn\_complete\_time
* txn\_process\_time\_hours

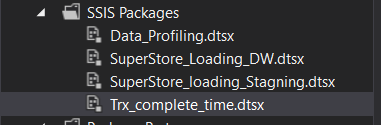


## Step 2: Creating a csv file with Business Key and date time.

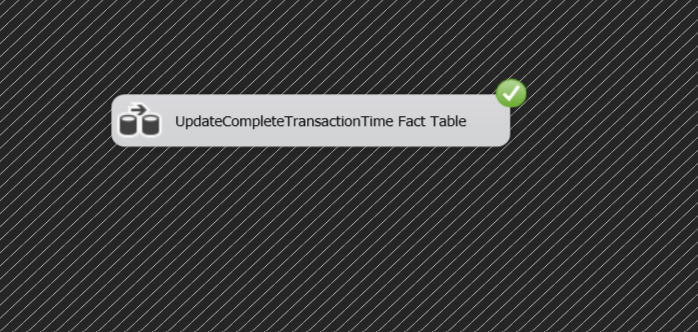
I have chosen a Business Key (Natural Key) named “Row\_Order\_ID” in the FactOrder table which were loaded from the OrderStaging. By using this Business Key, I had created a separated csv file named “TRNX” which contains 2 columns as Row ID and accm\_txn\_complete\_time since it had mentioned that hypothetically, it will take couple of days for your transaction to be completed. So That I had generated Date with 2 days using DATE and RANDBETWEEN function.



## Step 3: Creating Separate package in SSIS

****

## Step 4: Creating a control Flow

****

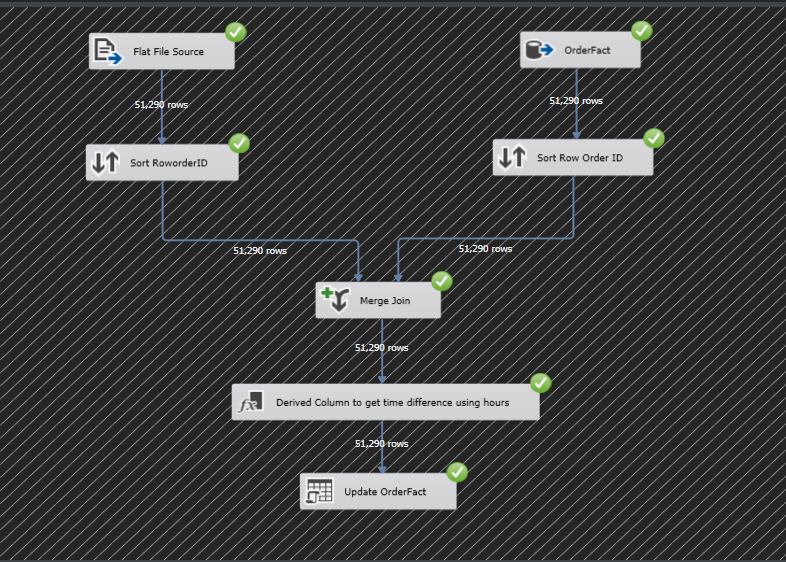
## Step 5: Updating FactOrder table with necessary columns

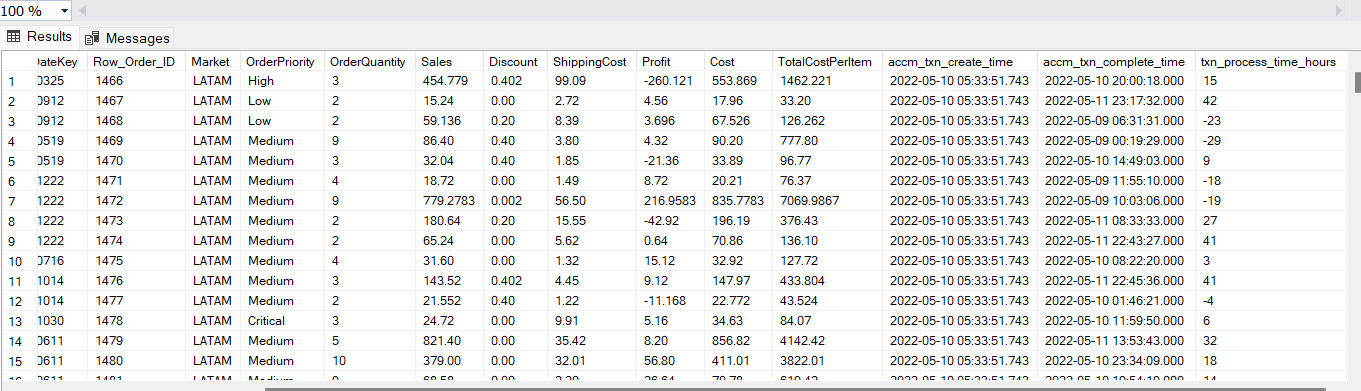
Loading the Fact source file “TRNX” which was created above with 2 columns and sort that flat file with the RowID using sort component and loaded the FactOrder table and sort that table with the same unique RowID using sort component, after sorting I had used a merge sort component to merge those 2 ID s. and I had used Derived column component to get the difference and calculate the process hours so that i had used DATEDIFF() by passing 3 params as “hh” to get the hours, “create time”, “complete time”. After getting the results I had updated the FactOrder table using the below sql command:

UPDATE [dbo].[FactOrder]

SET [accm\_txn\_complete\_time] = ?, [txn\_process\_time\_hours]=?

WHERE [Row\_Order\_ID] = ?

** Data flow of the whole process**

 **Updated FactOrder table with complete time and hours to complete the task**