IT3021- Data Warehousing & Business Intelligence

Assignment 1 -Report

IT20237554

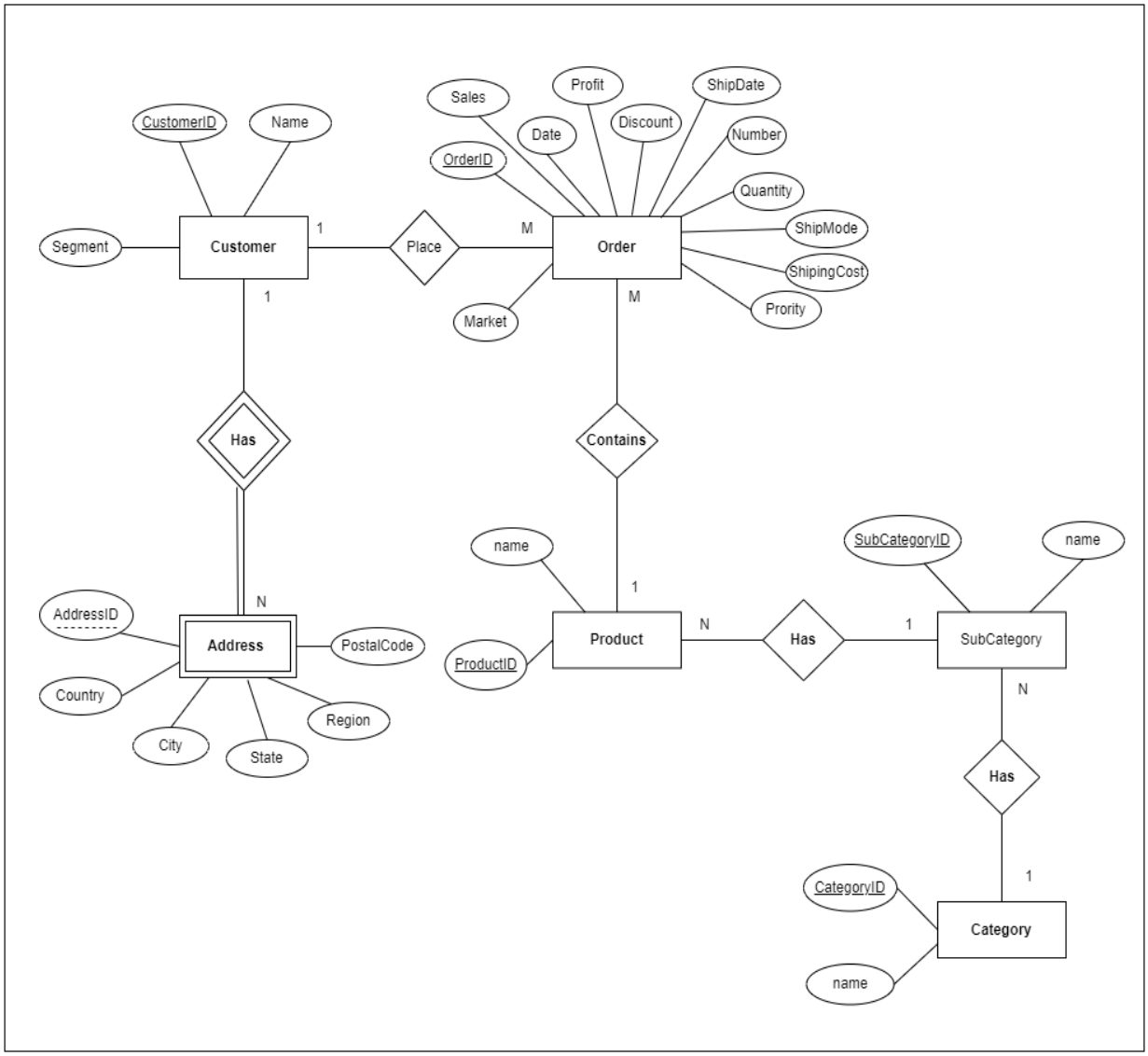
Rathnaweera R.P.W.G

Table of Contents

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.

ER diagram



Preparation of the Data Sources – Step 2

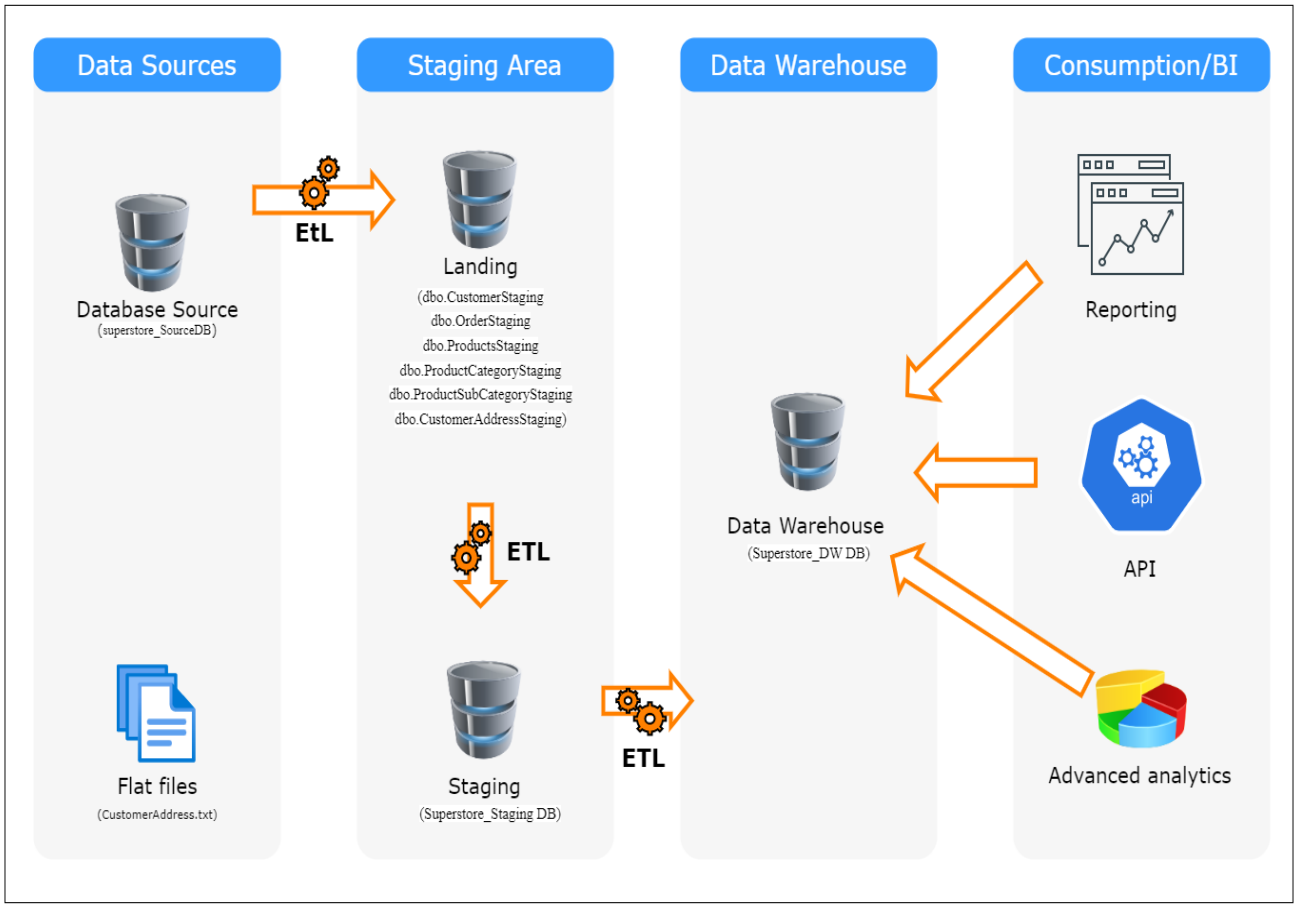
One common csv file in the dataset had been separated into 5 unique csv files and one text(.txt) file. after the separation, created a source database and named it as “Superstore_SourceDB” and loaded the above separated unique 5 csv file into that DB. And the separated text file is directly saved as “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	int	Unique ID to identify the customer
			CustomerName	nvarchar(50)	Customer Name
			CustomerSegment	nvarchar(50)	Segment of the customer
	dbo.Order	Order	OrderID	int	Unique ID to identify the order
			OrderNum	int	Order number
			CustomerID	int	CustomerID
			ProductID	int	ProductID
			OrderDate	datetime	Date of the order
			Sales	money	Price per item
			Profit	money	Profit received from a single order
			Discount	money	Discount given for a single order
			ShipDate	datetime	Shipping date of the ordered products

			Quantity	int	No.of products
			ShipMode	nvarchar(50)	The shipment way
			Market	nvarchar(50)	Market name
			OrderPriority	nvarchar(50)	Priority of the order
			ShippingCost	money	Cost for Shipping products
	dbo.Product	Product	ProductID	int	Unique ID to identify the Product
			ProductName	nvarchar(50)	Name of the product
			SubCategoryID	int	SubCategoryID
	dbo.SubCategory	SubCategory	SubCategoryID	int	Unique ID to identify the Product subcategory
			SubCategoryName	nvarchar(250)	Name of the subcategory
			CategoryID	int	CategoryID
	dbo.Category	Category	CategoryID	int	Unique ID to identify the Product category
			CategoryName	nvarchar(250)	Name of the Category
Text File (.txt)	CustomerAddress.txt	CustomerAddress	AddressID	int	Unique ID to identify the location
			CustomerID	nvarchar(30)	CustomerID
			Country	nvarchar(250)	Country of the customer
			Region	nvarchar(250)	Region of the customer
			State	nvarchar(250)	State of the customer
			City	nvarchar(250)	City of the customer
			PostalCode	nvarchar(250)	PostalCode of the customer

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

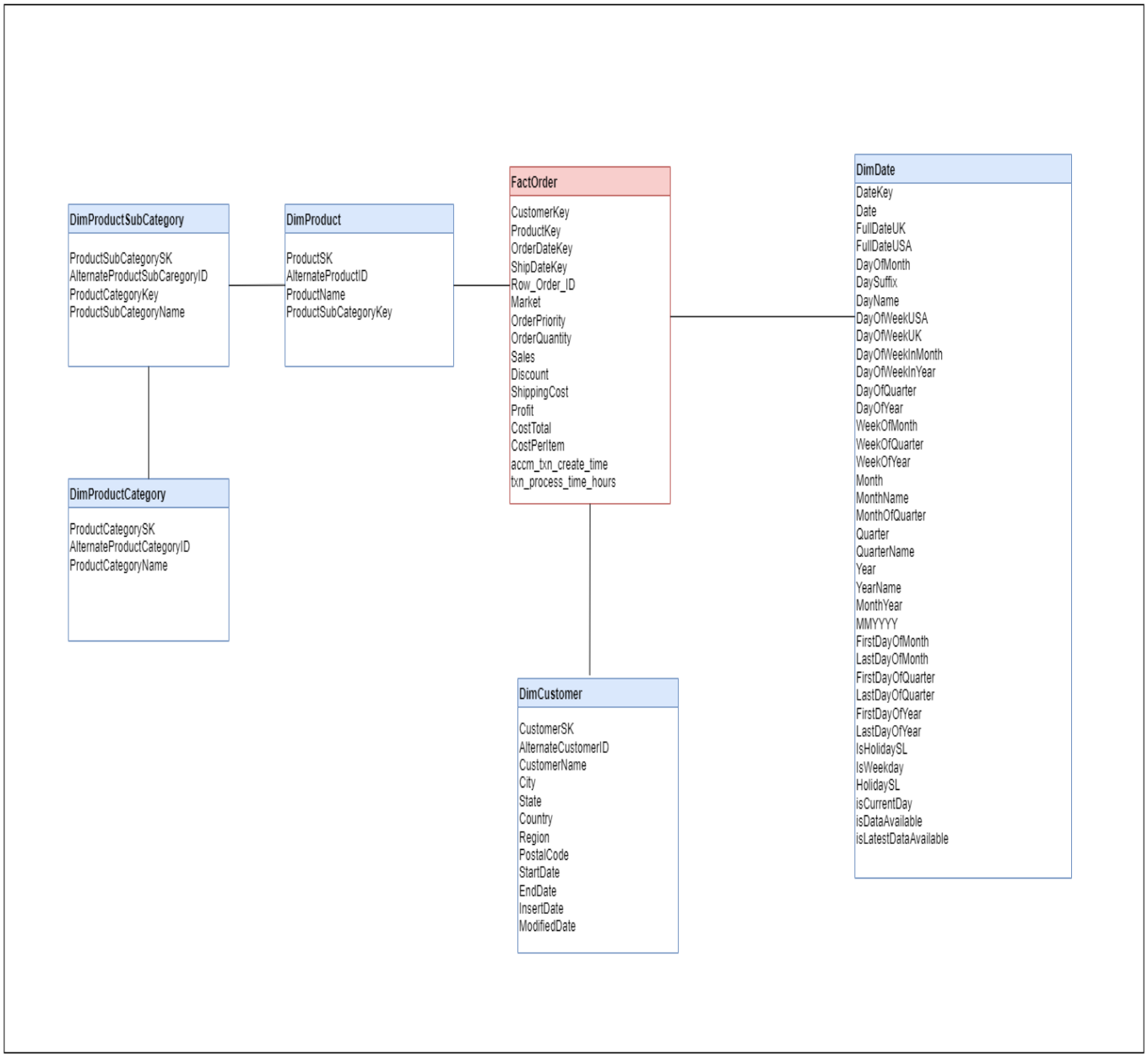
  Using ‘Extract’, ‘Transform' and 'Load,' the data warehouse DB component displays the warehouse's cratering dimension tables.

- 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



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])

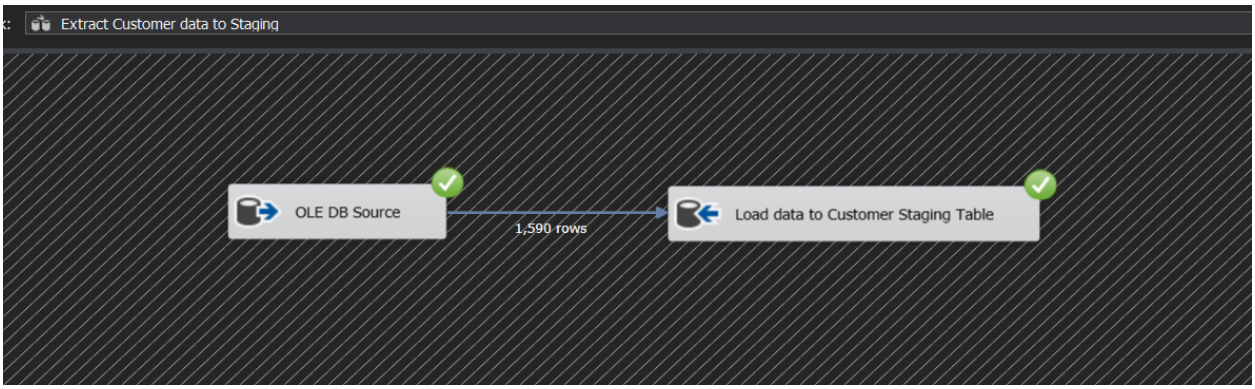
ETL development – Step 5

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

I. First, I had extracted the Customer Data from SuperStore_SourceDB to staging.

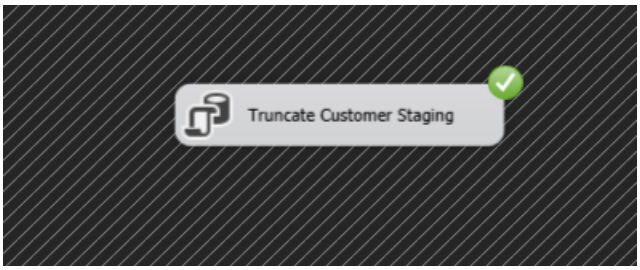
Data Flow task – Extract Customer Data to Staging



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**

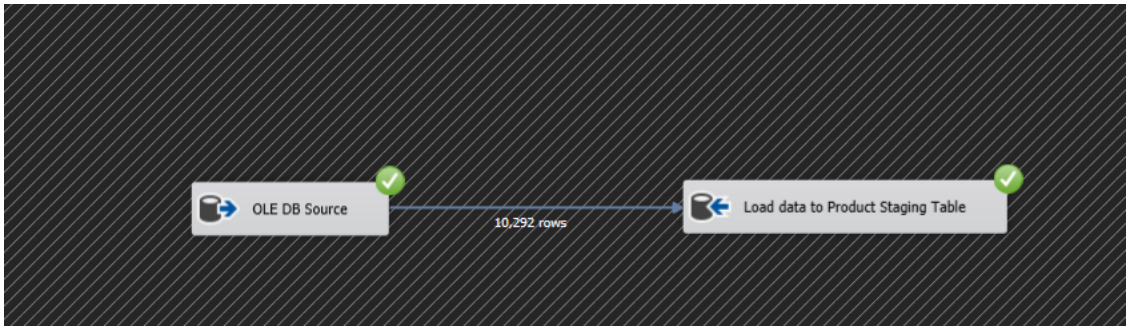
Truncate table [dB].[CustomerStaging]



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

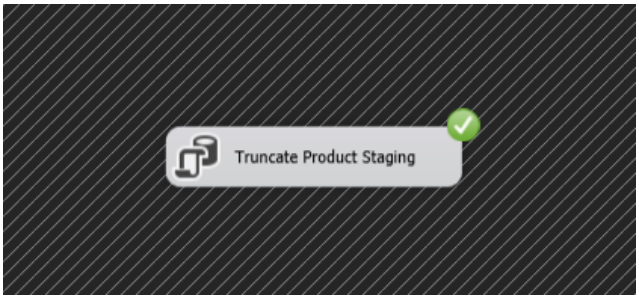
II. Extracted the Product Data from SuperStore_SourceDB to staging

Data Flow task – Extract Product Data to Staging



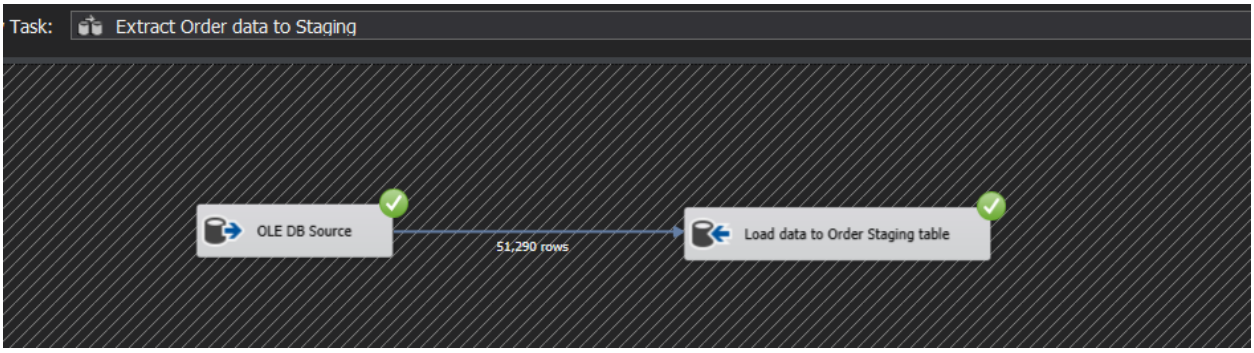
Executable: Extract Product Data to Staging **Event Handler: OnPreExecute**

Truncate table [dB].[ProductStaging]



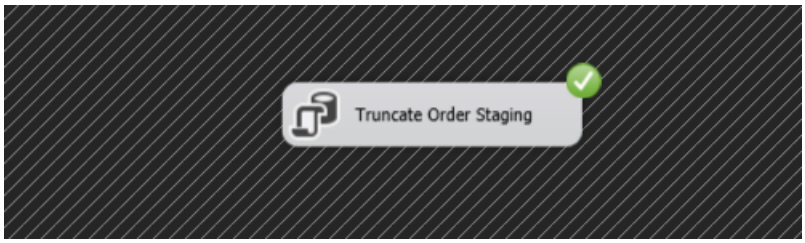
III. Extracted the Order Data from SuperStore_SourceDB to staging

Data Flow task – Extract Order Data to Staging



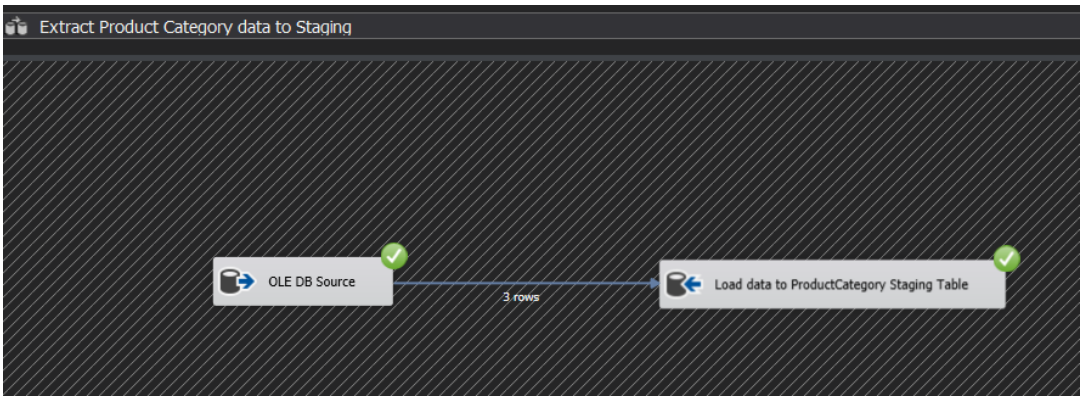
Executable: Extract Order Data to Staging **Event Handler: OnPreExecute**

Truncate table [dB].[OrderStaging]



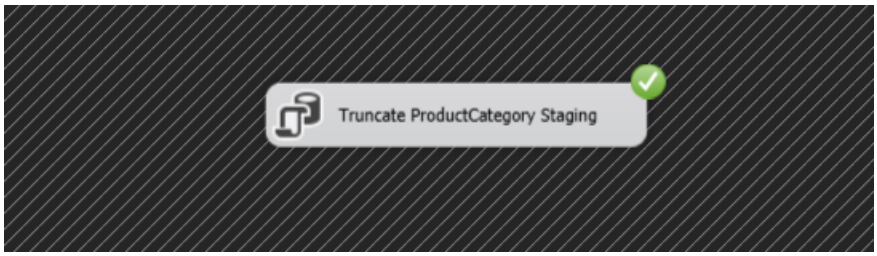
IV. Extracted the Product Category Data from SuperStore_SourceDB to staging

Data Flow task – Extract Product Category Data to Staging



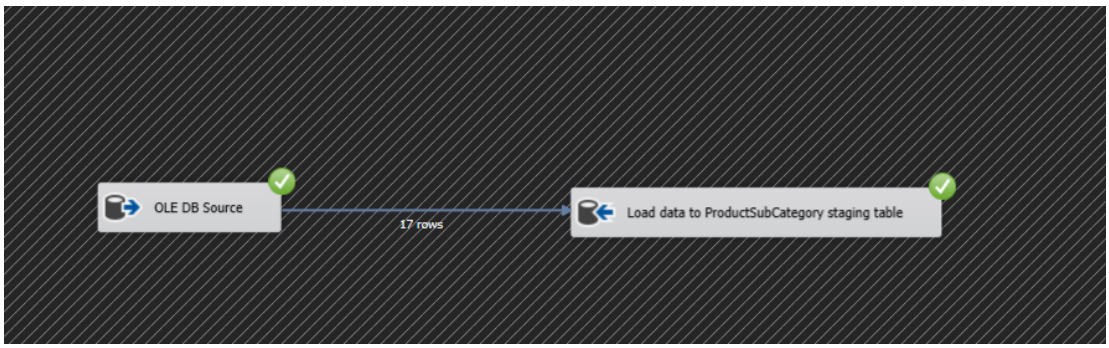
Executable: Extract Product Category Data to Staging **Event Handler: OnPreExecute**

Truncate table [dB].[ProductCategoryStaging]



V. Extracted the Product Sub Category Data from SuperStore_SourceDB to staging

Data Flow task – Extract ProductSubCategory Data to Staging

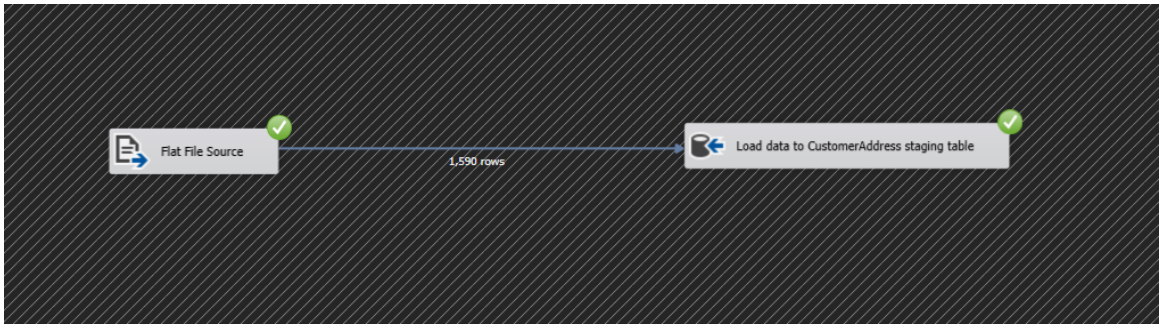


Executable: Extract Product Sub Category Data to Staging **Event Handler: OnPreExecute**

Truncate table [dB].[ProductSubCategoryStaging]

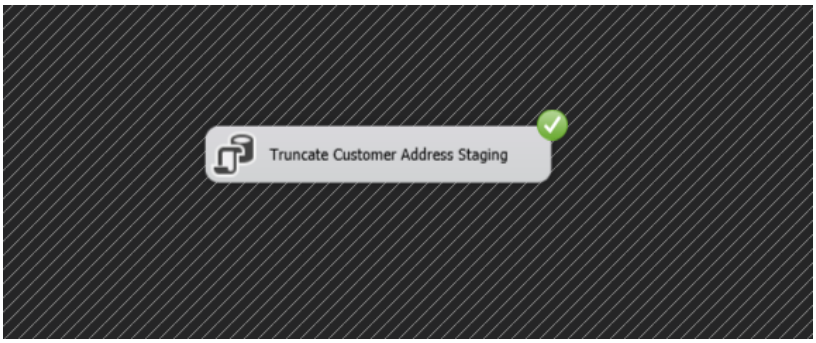
VI. Extracted CustomerAddress Data form CustomerAddress.txt to staging

Data Flow task – Extract CustomerAddress Data to Staging

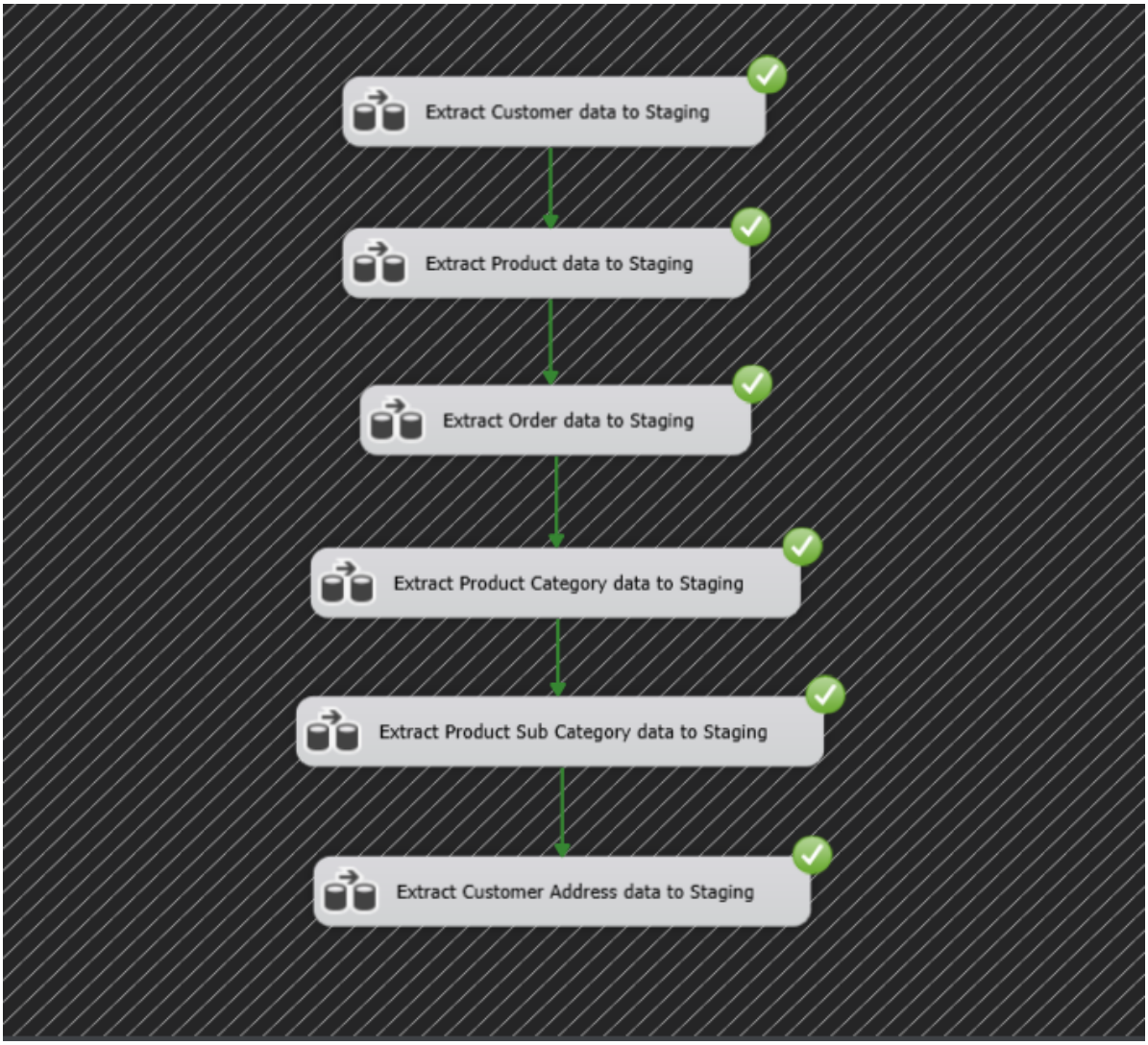


Executable: Extract CustomerAddress Data to Staging **Event Handler: OnPreExecute**

Truncate table [dB].[CustomerAddressStaging]



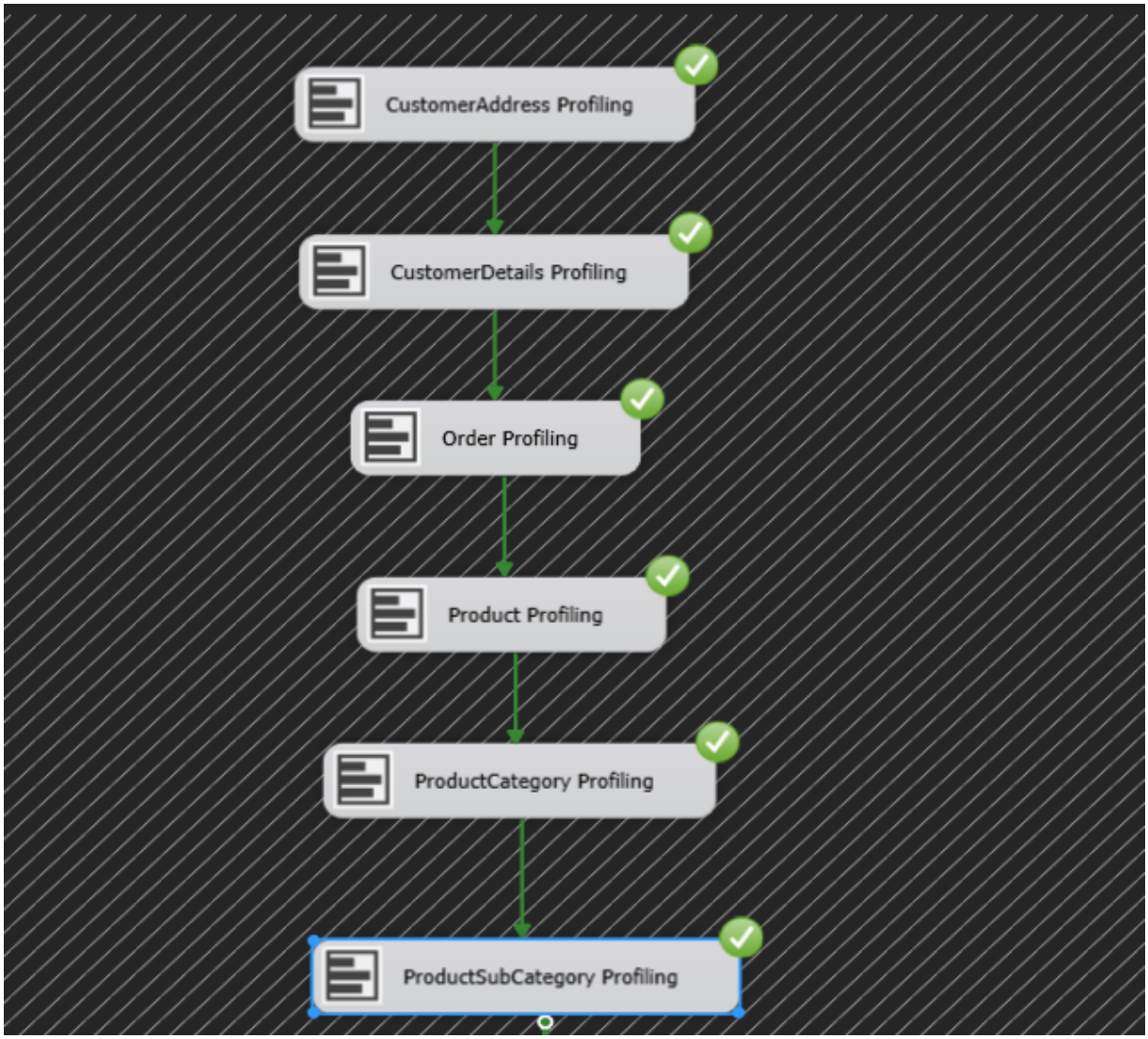
Full Control Flow of extracting data to Staging



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.

Data_Profiling.dtsx

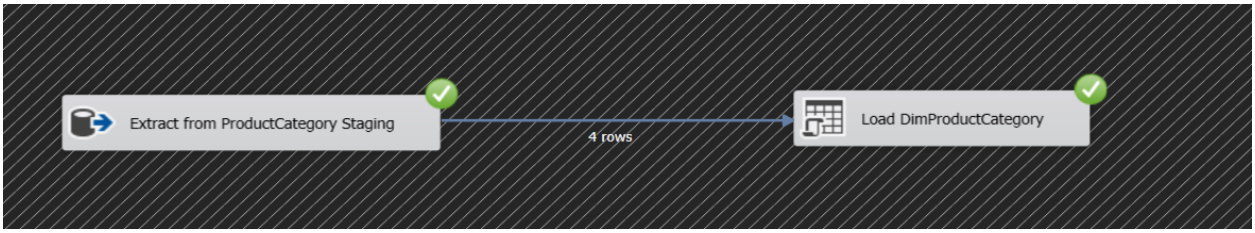


Data Transformations

I. 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.

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.

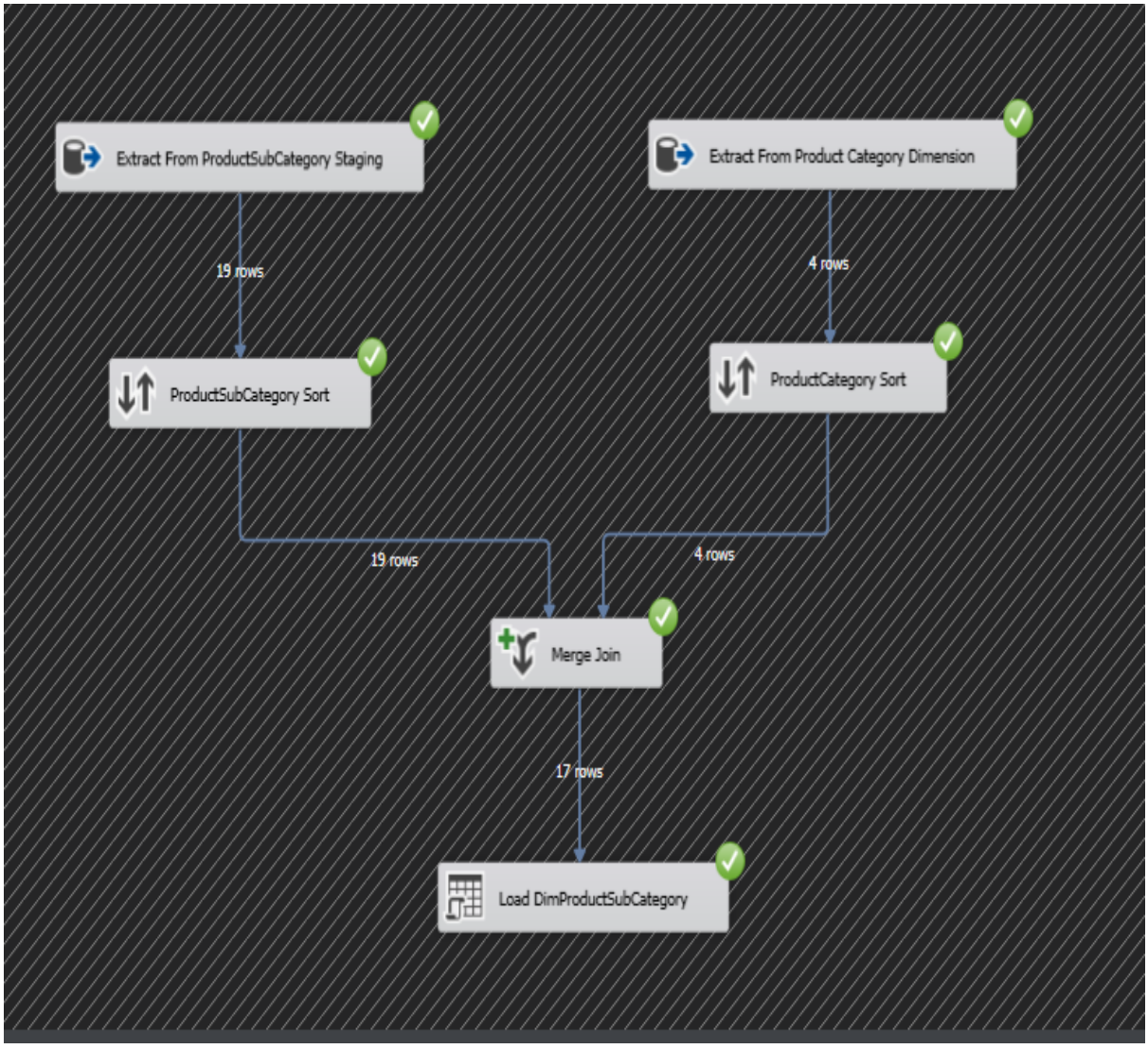
```
CREATE PROCEDURE dbo.UpdateDimProductCategory
@CategoryID int,
@Category nvarchar(50)
AS
BEGIN
IF not exists ( SELECT ProductCategorySK
                FROM dbo.DimProductCategory
                WHERE AlternateProductCategoryID=@CategoryID )
BEGIN
INSERT INTO dbo.DimProductCategory(AlternateProductCategoryID,ProductCategoryName)
VALUES(@CategoryID,@Category)
END;

IF exists (SELECT ProductCategorySK
                FROM dbo.DimProductCategory
                WHERE AlternateProductCategoryID=@CategoryID)
BEGIN
update dbo.DimProductCategory
set ProductCategoryName=@Category
WHERE AlternateProductCategoryID=@CategoryID
END;
END;
```

II. 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.

Transform and Load ProductSubCategory 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.

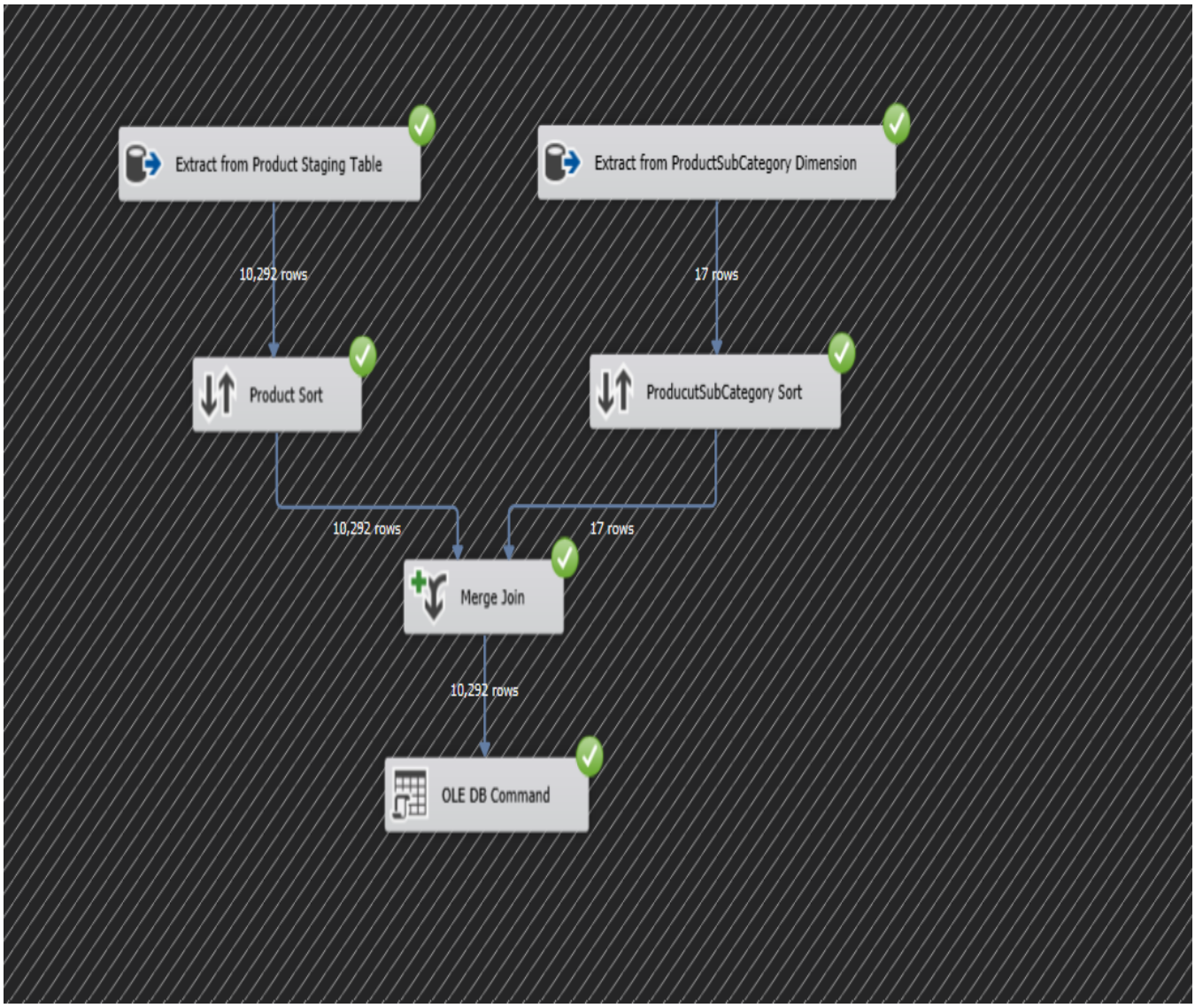
```
CREATE PROCEDURE dbo.UpdateDimProductSubCategory
@Sub_CategoryID int,
@Sub_Category nvarchar(50),
@ProductCategoryKey int
AS
BEGIN
IF not exists ( SELECT ProductSubCategorySK
                FROM  dbo.DimProductSubCategory
                WHERE AlternateProductSubCaregoryID=@Sub_CategoryID)
BEGIN
INSERT INTO dbo.DimProductSubCategory (AlternateProductSubCaregoryID,ProductSubCategoryName,ProductCategoryKey)
VALUES (@Sub_CategoryID,@Sub_Category,@ProductCategoryKey)
END;

IF exists( SELECT ProductSubCategorySK
                FROM  dbo.DimProductSubCategory
                WHERE AlternateProductSubCaregoryID=@Sub_CategoryID)
BEGIN
update dbo.DimProductSubCategory
set ProductSubCategoryName=@Sub_Category,ProductCategoryKey=@ProductCategoryKey
WHERE AlternateProductSubCaregoryID=@Sub_CategoryID
END;

END;
```

III. 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

```
CREATE PROCEDURE dbo.UpdateDimProduct
@Product_ID nvarchar(50),
@Product_Name nvarchar(50),
@Sub_Category int
AS
BEGIN

IF not exists ( SELECT ProductSK
                FROM dbo.DimProduct
                WHERE AlternateProductID=@Product_ID)
BEGIN
INSERT INTO dbo.DimProduct(AlternateProductID,ProductName,ProductSubCategoryKey)
VALUES (@Product_ID,@Product_Name,@Sub_Category)
END;

IF exists (     SELECT ProductSK
                FROM dbo.DimProduct
                WHERE AlternateProductID=@Product_ID)
BEGIN
update dbo.DimProduct
set ProductName=@Product_Name,ProductSubCategoryKey=@Sub_Category
WHERE AlternateProductID=@Product_ID
END;

END;
```

IV. Transform and load Customer Dimension

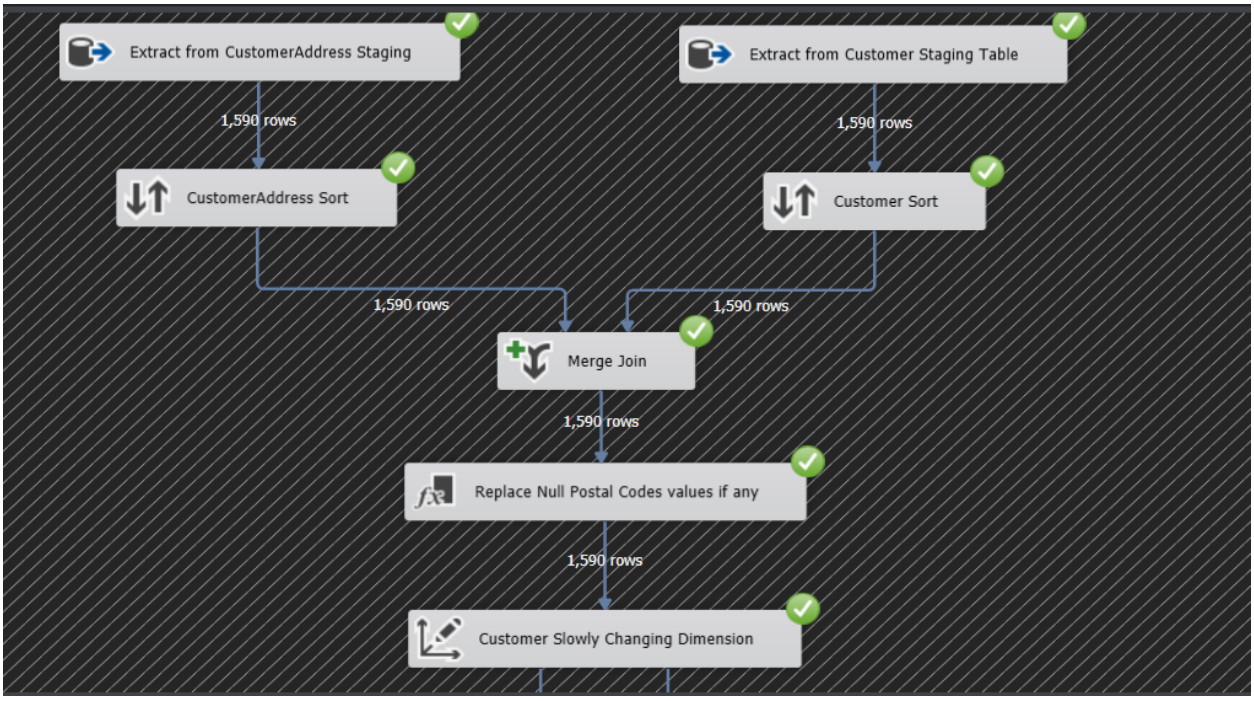
Here CustomerAddress staging table has CustomerID and address related details and Customer Staging table has the customer other details including customer id since the I had used the sort components to sort these and merge them using the merge component. Since I have some null values in the PostalCode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

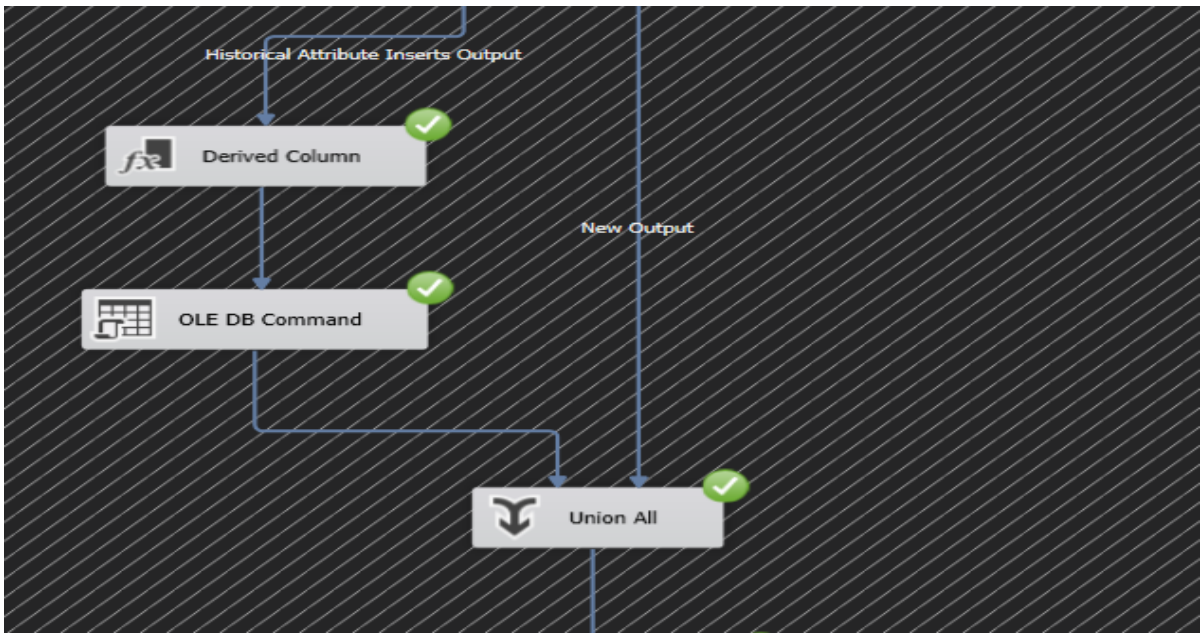
"REPLACENULL(PostalCode,"NA")",

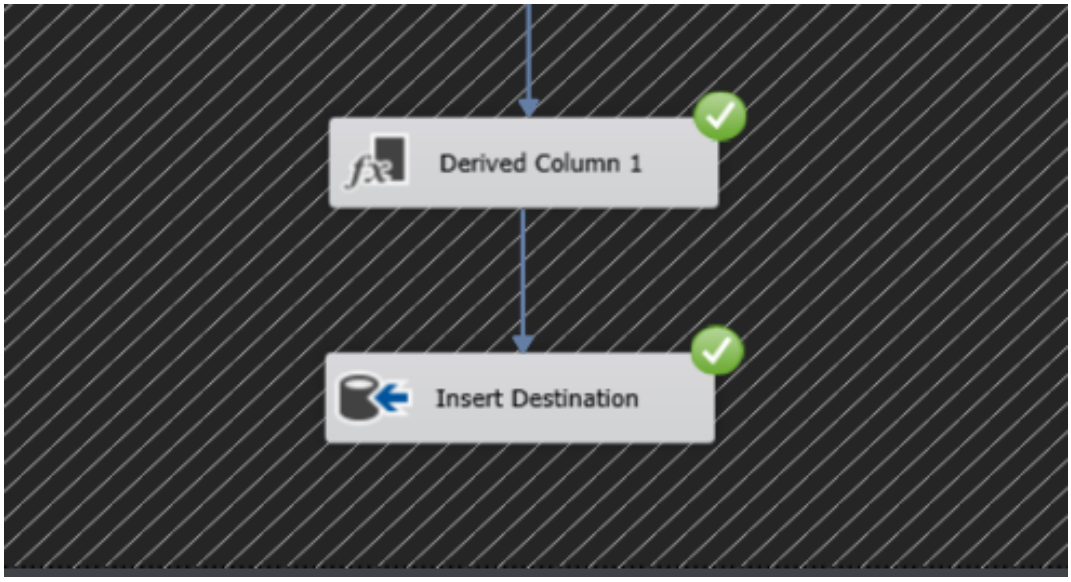
I had replaced those null values with word "NA". Since customer is a Slowly Changing Dimension (SCD) I had to maintain the historical data I had used another derived column to set the EndDate and update the DimCustomer using

"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

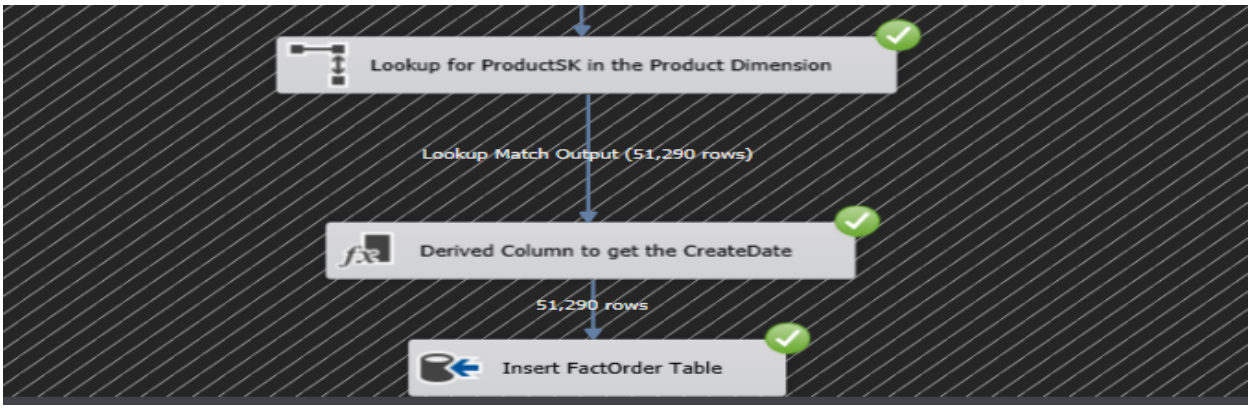



Historical Attribute Inserts Output
Derived Column
New Output
OLE DB Command
Union All

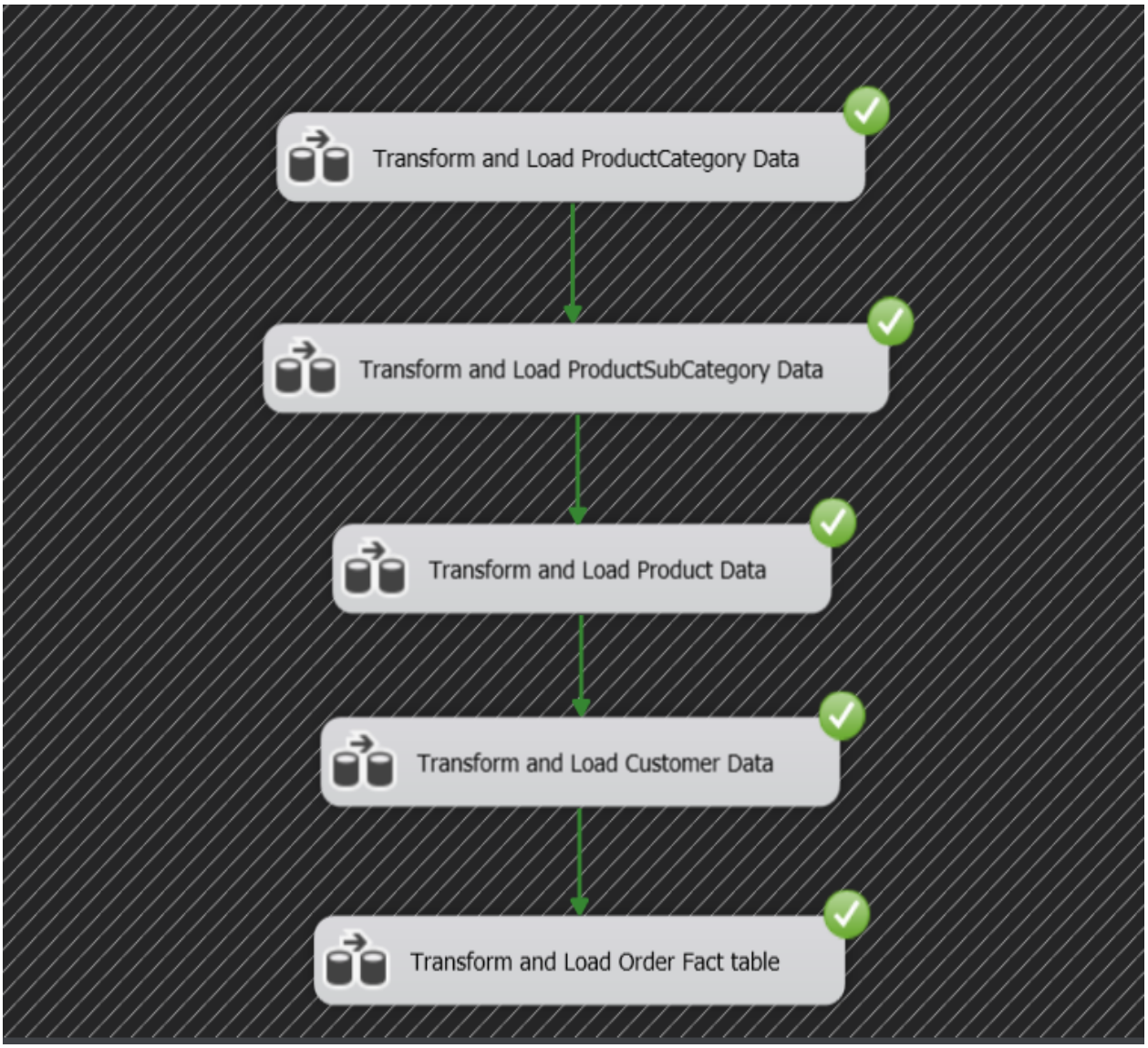

Derived Column 1
Insert Destination

V. 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.



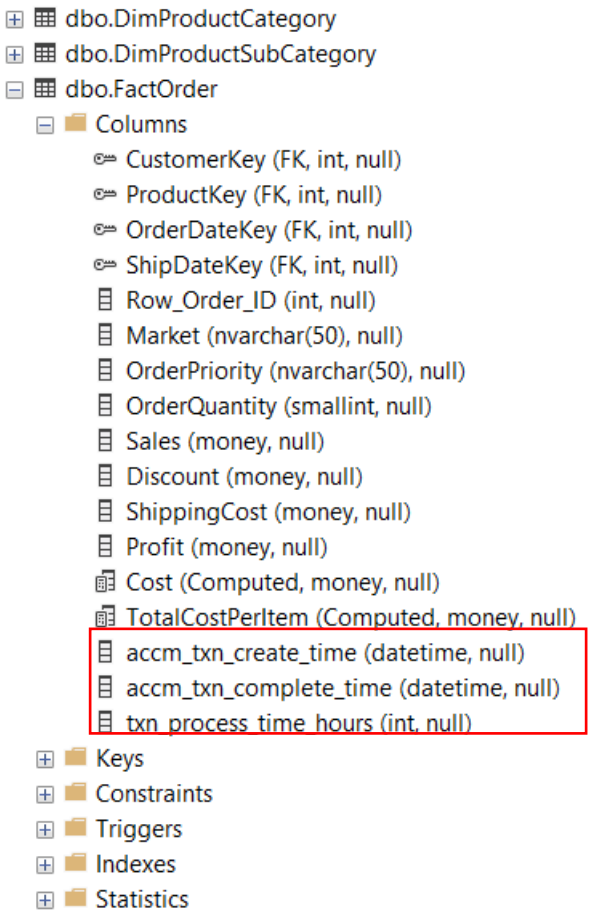
Control Flow of Loading Data from Staging to DW



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

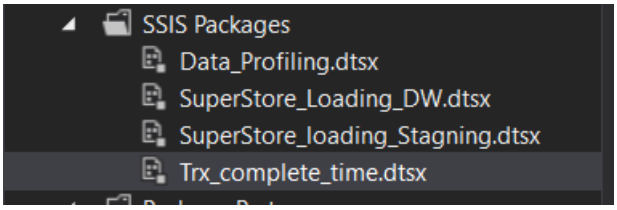


Step 2: Creating a csv file with Business Key and date time.

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.

	A	B	C
1	Row ID	accm_txn_complete_time	
2	42433	5/9/2022 18:48	
3	22253	5/11/2022 5:50	
4	48883	5/9/2022 7:45	
5	11731	5/10/2022 14:01	
6	22255	5/10/2022 23:24	
7	22254	5/10/2022 2:22	
8	21613	5/10/2022 12:57	
9	34662	5/9/2022 20:05	
10	44508	5/9/2022 14:32	
11	23688	5/10/2022 13:20	
12	25293	5/9/2022 15:24	
13	8483	5/10/2022 22:40	
14	41445	5/10/2022 21:27	
15	16727	5/9/2022 1:20	
16	21615	5/9/2022 8:38	
17	8484	5/10/2022 20:48	
18	19796	5/10/2022 16:37	
19	21614	5/11/2022 17:09	
20	21616	5/10/2022 8:30	
21	16726	5/9/2022 4:03	
22	14413	5/9/2022 0:12	

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:

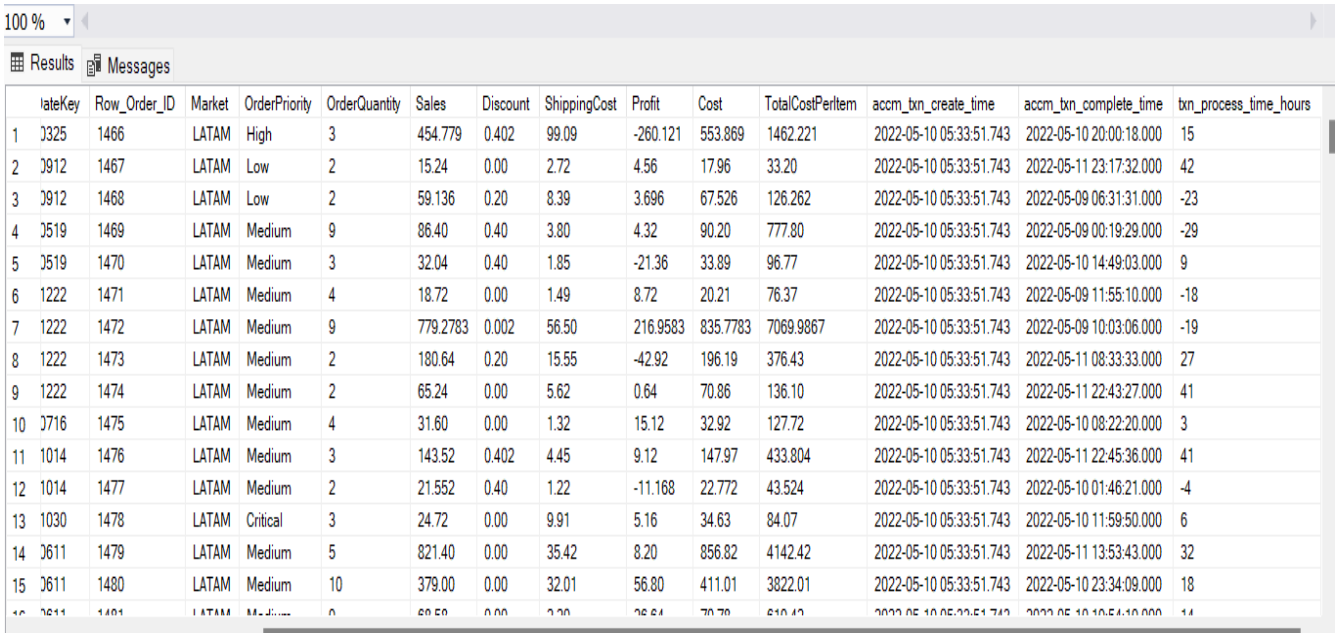
```
UPDATE [dbo].[FactOrder]

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



100 %

Results Messages

	lateKey	Row_Order_ID	Market	OrderPriority	OrderQuantity	Sales	Discount	ShippingCost	Profit	Cost	TotalCostPerItem	accm_txn_create_time	accm_txn_complete_time	txn_process_time_hours
1	0325	1466	LATAM	High	3	454.779	0.402	99.09	-260.121	553.869	1462.221	2022-05-10 05:33:51.743	2022-05-10 20:00:18.000	15
2	0912	1467	LATAM	Low	2	15.24	0.00	2.72	4.56	17.96	33.20	2022-05-10 05:33:51.743	2022-05-11 23:17:32.000	42
3	0912	1468	LATAM	Low	2	59.136	0.20	8.39	3.696	67.526	126.262	2022-05-10 05:33:51.743	2022-05-09 06:31:31.000	-23
4	0519	1469	LATAM	Medium	9	86.40	0.40	3.80	4.32	90.20	777.80	2022-05-10 05:33:51.743	2022-05-09 00:19:29.000	-29
5	0519	1470	LATAM	Medium	3	32.04	0.40	1.85	-21.36	33.89	96.77	2022-05-10 05:33:51.743	2022-05-10 14:49:03.000	9
6	1222	1471	LATAM	Medium	4	18.72	0.00	1.49	8.72	20.21	76.37	2022-05-10 05:33:51.743	2022-05-09 11:55:10.000	-18
7	1222	1472	LATAM	Medium	9	779.2783	0.002	56.50	216.9583	835.7783	7069.9867	2022-05-10 05:33:51.743	2022-05-09 10:03:06.000	-19
8	1222	1473	LATAM	Medium	2	180.64	0.20	15.55	-42.92	196.19	376.43	2022-05-10 05:33:51.743	2022-05-11 08:33:33.000	27
9	1222	1474	LATAM	Medium	2	65.24	0.00	5.62	0.64	70.86	136.10	2022-05-10 05:33:51.743	2022-05-11 22:43:27.000	41
10	0716	1475	LATAM	Medium	4	31.60	0.00	1.32	15.12	32.92	127.72	2022-05-10 05:33:51.743	2022-05-10 08:22:20.000	3
11	1014	1476	LATAM	Medium	3	143.52	0.402	4.45	9.12	147.97	433.804	2022-05-10 05:33:51.743	2022-05-11 22:45:36.000	41
12	1014	1477	LATAM	Medium	2	21.552	0.40	1.22	-11.168	22.772	43.524	2022-05-10 05:33:51.743	2022-05-10 01:46:21.000	-4
13	1030	1478	LATAM	Critical	3	24.72	0.00	9.91	5.16	34.63	84.07	2022-05-10 05:33:51.743	2022-05-10 11:59:50.000	6
14	0611	1479	LATAM	Medium	5	821.40	0.00	35.42	8.20	856.82	4142.42	2022-05-10 05:33:51.743	2022-05-11 13:53:43.000	32
15	0611	1480	LATAM	Medium	10	379.00	0.00	32.01	56.80	411.01	3822.01	2022-05-10 05:33:51.743	2022-05-10 23:34:09.000	18
16	0611	1481	LATAM	Medium	9	69.59	0.00	2.20	26.64	70.79	610.42	2022-05-10 05:33:51.743	2022-05-10 10:54:10.000	14