Sri Lanka Institute of Information Technology

Data warehousing and Business Intelligence

Assignment one

2018

Submitted by:

IT16084896 - L.P.J.Perera

1. About Dataset and Scenario

This Dataset is about a super store. Details were stored in order wise, return wise and user wise.

Inside the Order table there were customer details, order details, product details and also location details. So I have to prepare my dataset by breaking them separately into unique tables.

In this scenario customer orders from super store and the details were stored.

So to understand how the business is performing in different aspects a data warehouse and an analytical platform is developed to measure the business performance.

To analyze the data in the data warehouse design two dimension tables (DimCustomer, DimProduct) were created and the Fact table.

2. Preparation of Data Sources

Source Type	Table name	Column name	Data Type	Description
SQL Database MySourceDB	dbo.CustomerTbl	CustomerID	int	Unique ID.
		CustomerName	nvarchar(50)	Name of the Customer
		CustomerSegment	nvarchar(50)	Under which Segment Customer falls(Home Office, Corporate, Consumer, Small Business)
	dbo.OrdersTbl	OrderID	int	Unique ID.
		QuantityOrderedNew	int	Quantity ordered
		OrderPriority	nvarchar(50)	Priority of the order(Low, Medium, High, else Not Specified)
	dbo.ProductTbl	ProductID	int	Unique ID.
		ProductName	nvarchar(50)	Name of the Product
		ProductBaseMargin	float	Base margin of the product
		ProductSubCategoryID	int	Unique ID.
	dbo.ProductSubCategoryTbl	ProductSubCategoryID	int	Unique ID.
		ProductSubCategoryName	nvarchar(50)	Name of the Product SubCategory
		ProductContainerID	int	Unique ID.
	dbo.ProductContainerTbl	ProductContainerID	int	Unique ID.
		ProductContainerName	nvarchar(50)	Name of the Product Container
		ProductCategoryID	int	Unique ID.

	Dbo.ProductCategoryTbl	ProductCategoryID	int	Unique ID.
		ProductCategoryName	nvarchar(50)	Name of the Product Category
Text file(CustomerLocationTbl.txt)	CustomerLocationTbl	LocationID		Unique ID.
		PostalCode		Postal code
		City		City name
		StateID		Unique ID.
		StateProvince		State name
		RegionID		Unique ID.
		Region		Region name
		CustomerID		Unique ID.

At first I have found the dataset super store as a .bak (backup file) so I have to restore it in the database and then I have imported the database in to a excel file. Then I have prepared the dataset by breaking them in to several tables and saving them into several excel sheets. Again I have imported the prepared excel sheets into my new database. (MySourceDB)

Customer location details excel sheet saved into a text file so now I have two sources to my Data Warehouse.

As explained in the above at first I have prepared my sources to the MySourceDB_DW Data Warehouse.

3. Solution Architecture

This figure shows the high-level BI solution architecture to the Data warehouse design.

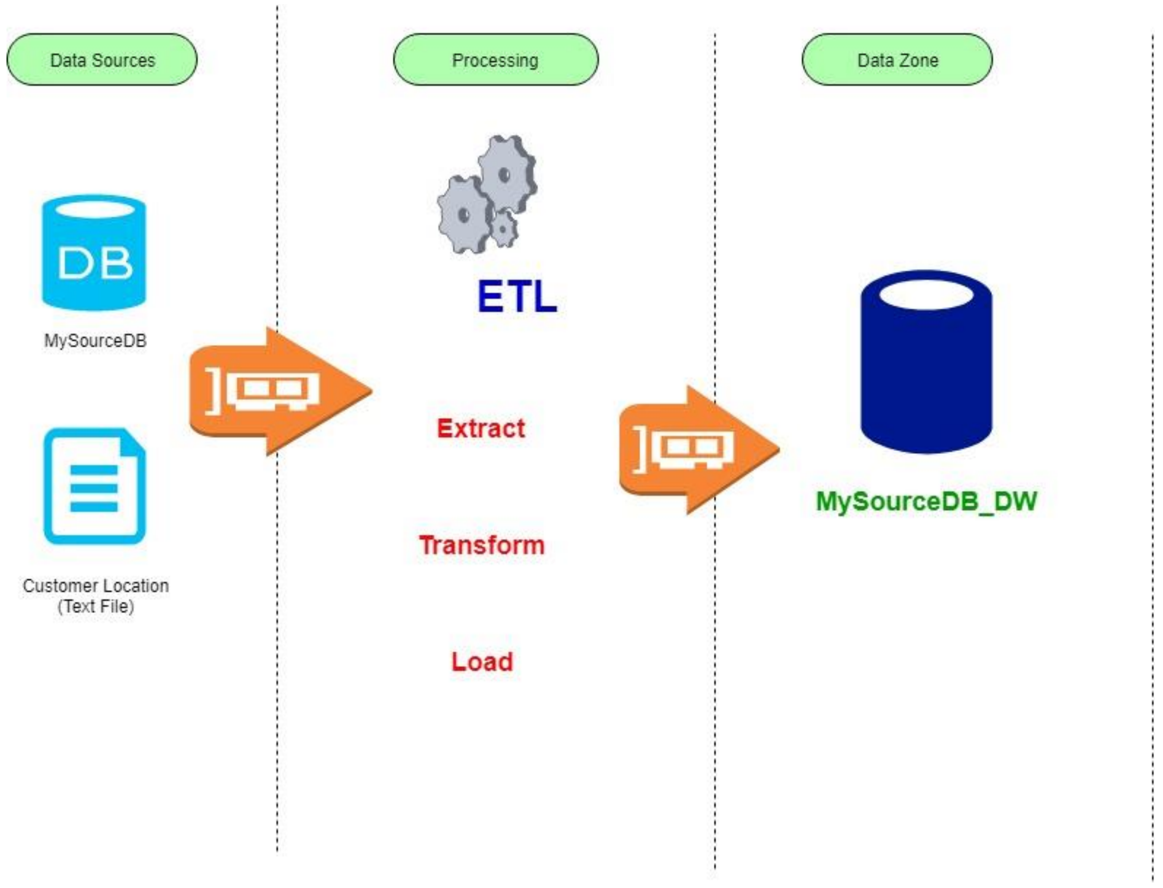


Figure 1.0

1. Data Sources

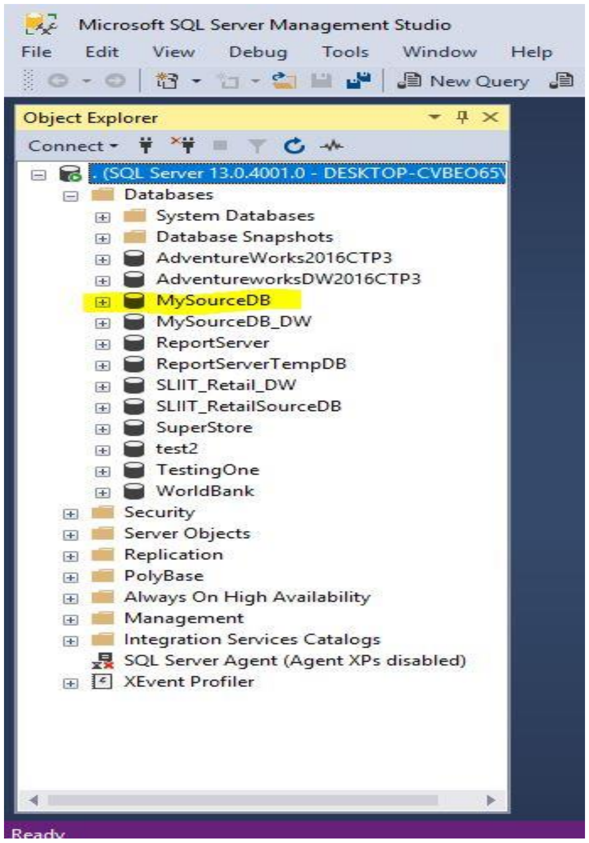
As shown in the figure 1.0 I have used two main data sources to the MySourceDB_DW data warehouse.

I. MySourceDB

This the main source to the MySourceDB_DW data warehouse which is a SQL database which includes six tables.

II. Customer Location table (Text file)

Second source to the MySourceDB_DW data warehouse which is a text file which includes customer location details such as city, state and region.


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
Ready

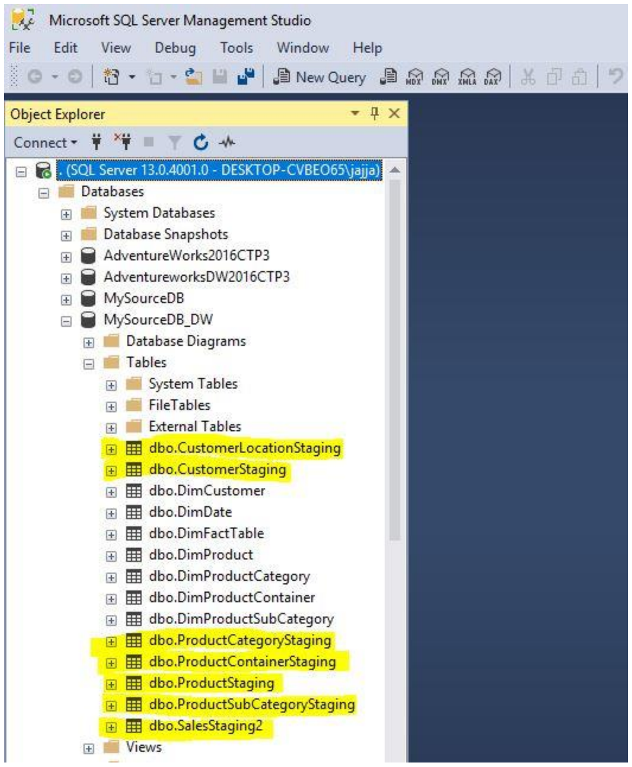
```
. (SQL Server 13.0.4001.0 - DESKTOP-CVBEO65\
  Databases
    System Databases
    Database Snapshots
    AdventureWorks2016CTP3
    AdventureworksDW2016CTP3
    MySourceDB
    MySourceDB_DW
    ReportServer
    ReportServerTempDB
    SLIIT_Retail_DW
    SLIIT_RetailSourceDB
    SuperStore
    test2
    TestingOne
    WorldBank
  Security
  Server Objects
  Replication
  PolyBase
  Always On High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent (Agent XPs disabled)
  XEvent Profiler
```

Figure 1.1

2. Extract Transform and Load

I. Extract

In this extraction process I have not implemented a separate Staging database to extract the data and to store them.
First I have extracted all the tables in the MySourceDB and the Location table text file to separate staging tables which were stored in the MySourceDB_DW data warehouse.


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect

```
. (SQL Server 13.0.4001.0 - DESKTOP-CVBEO65\jajja)
  Databases
    System Databases
    Database Snapshots
    AdventureWorks2016CTP3
    AdventureworksDW2016CTP3
    MySourceDB
    MySourceDB_DW
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        dbo.CustomerLocationStaging
        dbo.CustomerStaging
        dbo.DimCustomer
        dbo.DimDate
        dbo.DimFactTable
        dbo.DimProduct
        dbo.DimProductCategory
        dbo.DimProductContainer
        dbo.DimProductSubCategory
        dbo.ProductCategoryStaging
        dbo.ProductContainerStaging
        dbo.ProductStaging
        dbo.ProductSubCategoryStaging
        dbo.SalesStaging2
      Views
```

Figure 1.2

Extract Transform and Load continued….

II. Transform

As shown in the below figure 1.3 in the transformation process there are seven transformation steps. Order of the execution of these tasks are very important so I have design the order of the execution in correct order. More details will be explained in this report in the ETL development part.

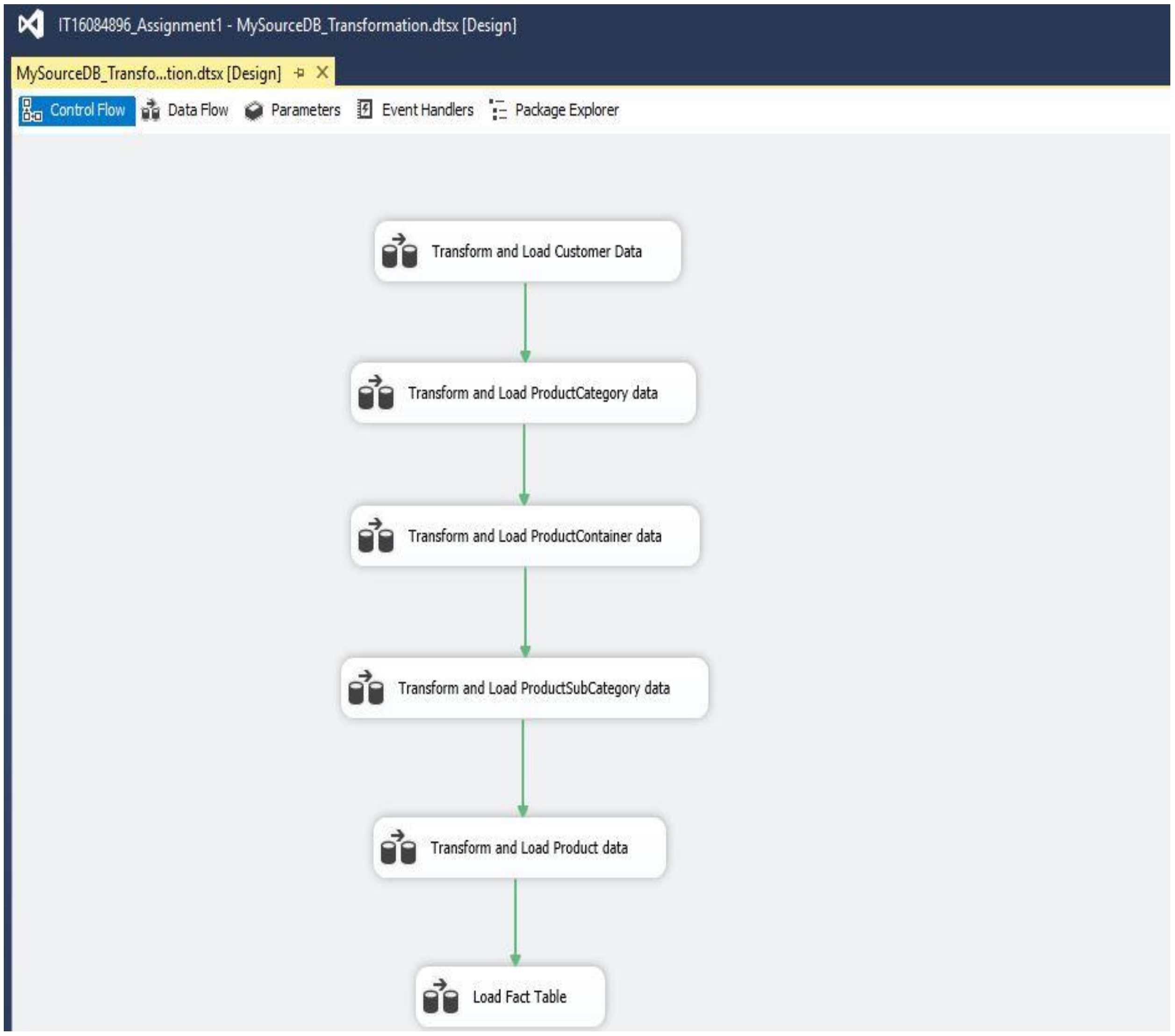

IT16084896_Assignment1 - MySourceDB_Transformation.dtsx [Design]
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Transform and Load Customer Data
Transform and Load ProductCategory data
Transform and Load ProductContainer data
Transform and Load ProductSubCategory data
Transform and Load Product data
Load Fact Table

Figure 1.3

Extract Transform and Load continued….

III. Load

In this loading process we insert our process data to the final destinations to the dimension tables in the data warehouse.

4. Data warehouse design and architecture

There are two types dimension model schemas and MySourceDB_DW data warehouse is a Snowflake schema.

DimProduct dimension table is a normalized table in this snowflake schema because it representing a level in the dimension hierarchy.

We can improve the storage efficiency because we are using the snowflake schema since no redundancy in the snowflake schema.

In the fact table we store the measures of interest. These measures called the **Grain** (Profit, ShippingCost, Discount, and SalesAmount) is stored in the fact table.

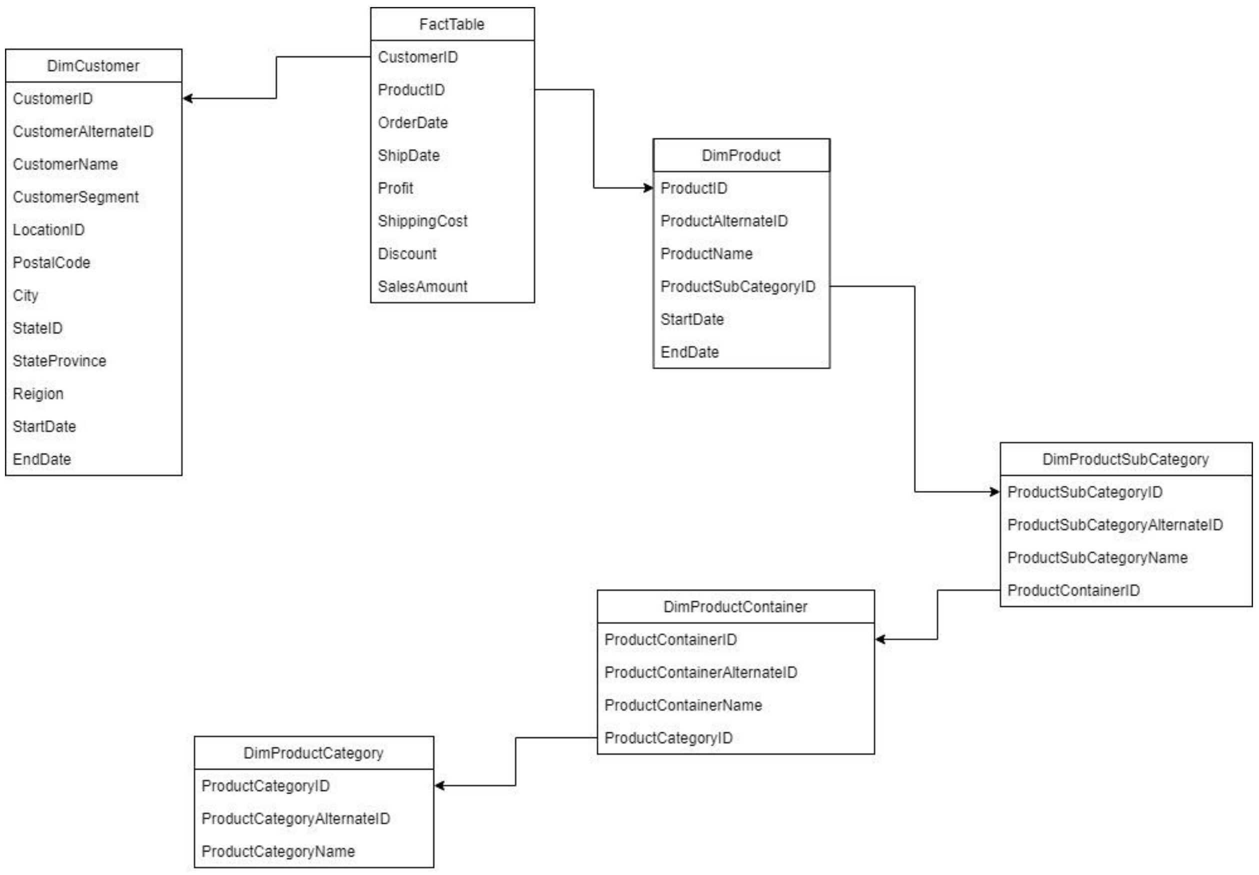


Figure 1.4

L.P.J.Perera
IT16084896

Data warehouse design and the architecture continued…

- In this Data warehouse design it includes two dimension tables which is connecting directly to the fact table except the date dimension. And from this two dimension tables DimProduct dimension tables has a hierarchy as shown in the Figure 1.4.
- And the design has a one fact table and also slowly changing dimensions.

- Customer dimension table (DimCustomer) is a slowly changing dimension table in this design. And it has four historical attributes (City, PostalCode, Region, and StateProvince) which can be changed over time.
- And Customer dimension table is a slowly changing dimension which I have implemented as a type 2 so it will add a new column to the table.
- Because of that when creating the Customer dimension table I added two more columns StartDate and the EndDate, so we can calculate more details of the customer.

5. ETL development

a) Extraction

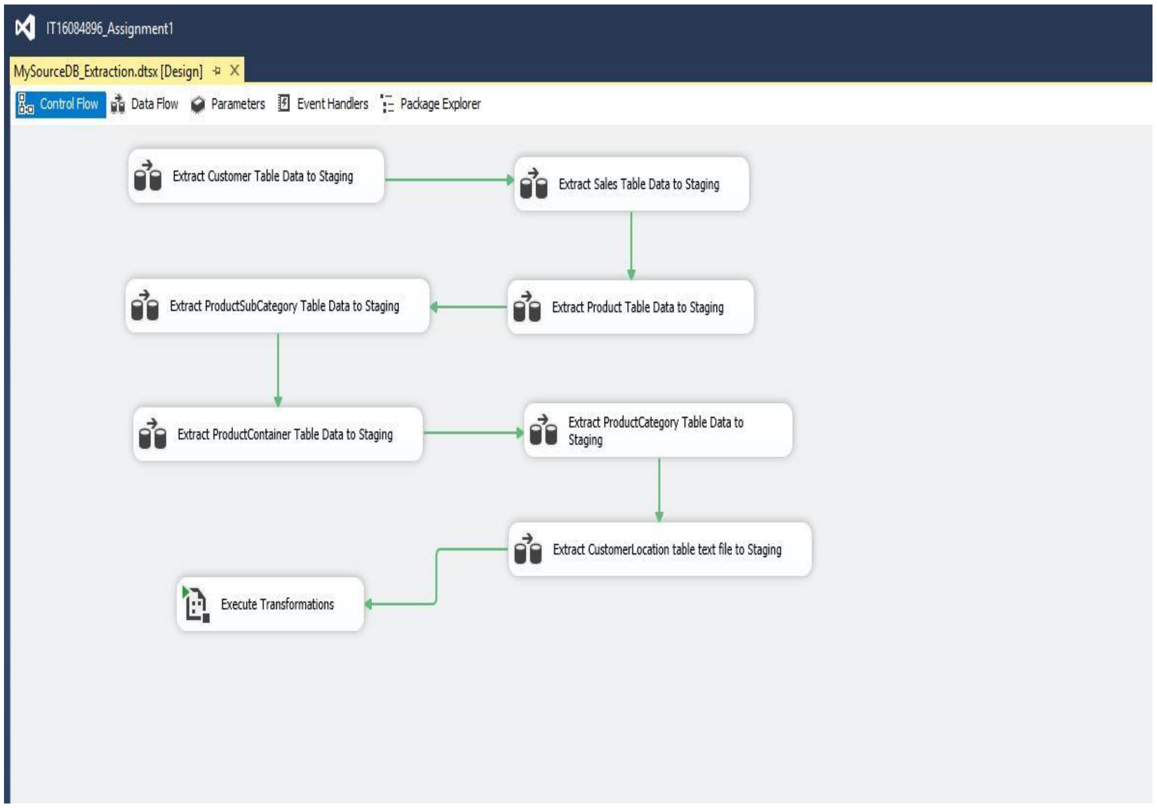

IT16084896_Assignment1
MySourceDB_Extraction.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Extract Customer Table Data to Staging
Extract Sales Table Data to Staging
Extract Product Table Data to Staging
Extract ProductSubCategory Table Data to Staging
Extract ProductContainer Table Data to Staging
Extract ProductCategory Table Data to Staging
Extract CustomerLocation table text file to Staging
Execute Transformations

Figure 1.5

First using the SQL Server Integration Services Software I have extracted all the data from the tables which were in the MySourceDB to separate staging tables as shown in the above figure 1.5.

ETL development continued…

If we go inside to the Data flow task to the data flow,

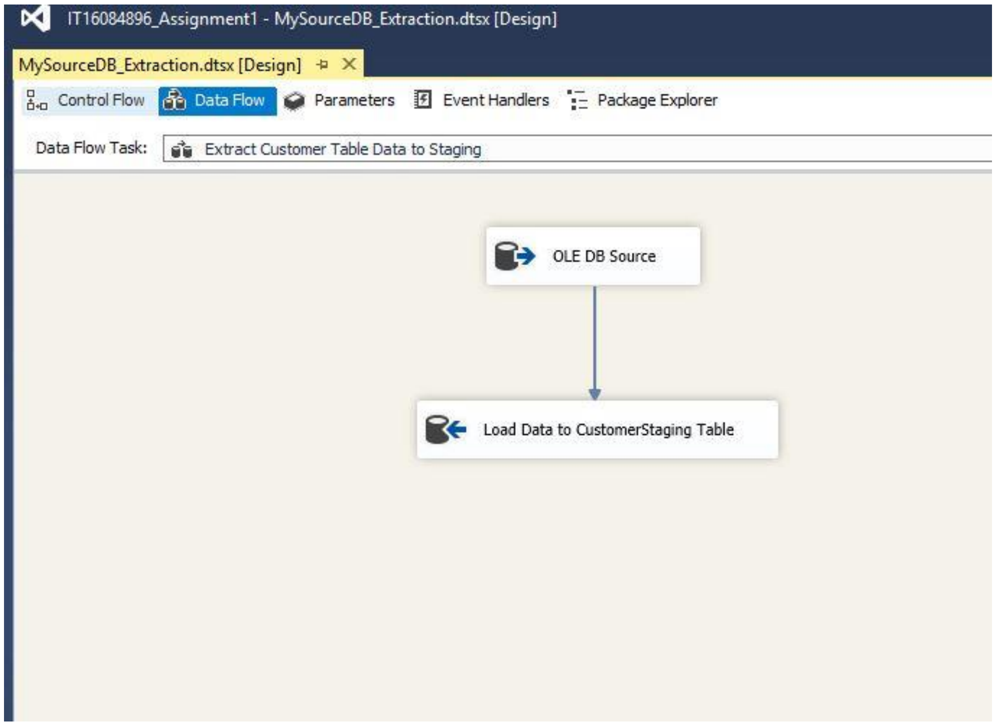

IT16084896_Assignment1 - MySourceDB_Extraction.dtsx [Design]
MySourceDB_Extraction.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
Extract Customer Table Data to Staging
OLE DB Source
Load Data to CustomerStaging Table

Figure 1.6

As shown in the figure 1.6 I used an OLE DB source to select the table that I want to extract data from MySourceDB and to the OLE DB destination I have created a staging table in MySourceDB_DW data warehouse and extracted data was created in that staging table.

And I used an Execute SQL task component to truncate each staging table data each time we run the SSIS.

As I have explained in the above these steps were followed to each table in the MySourceDB and extracted them in to a separate staging table.

b) Transform and load data to the dimension tables

As shown in the figure 1.3 there are seven steps in the data transformation loading process.

1. Transform and load customer data

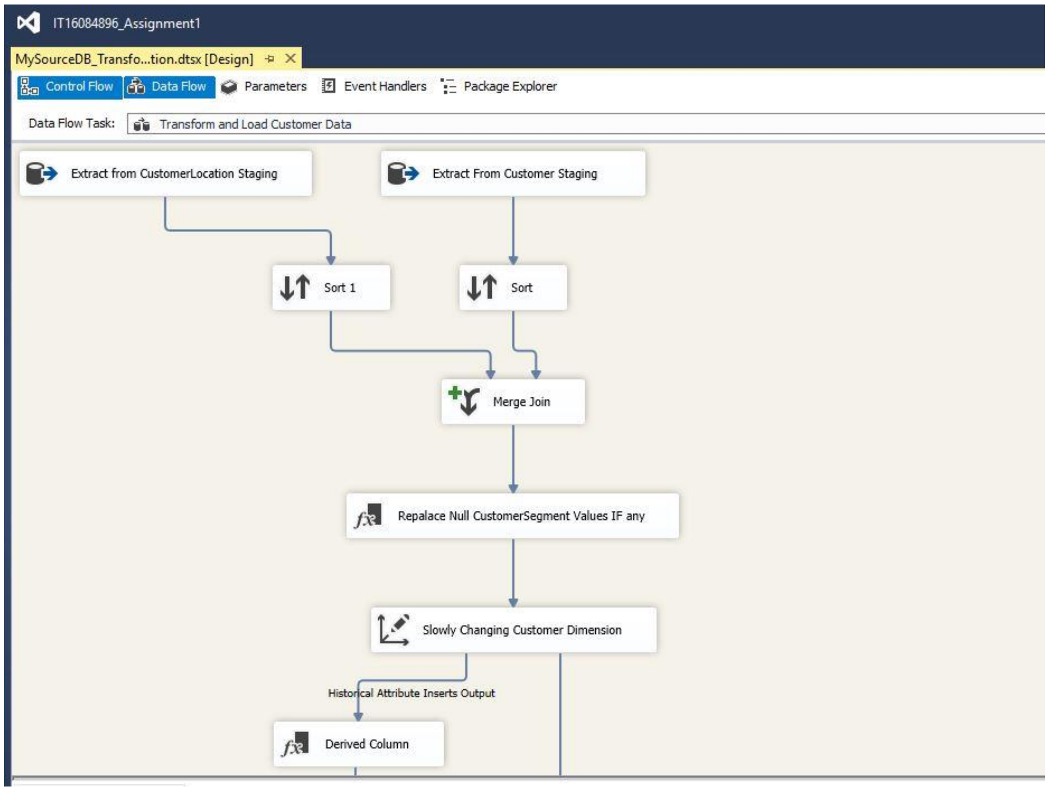

IT16084896_Assignment1
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
Transform and Load Customer Data
Extract from CustomerLocation Staging
Extract From Customer Staging
Sort 1
Sort
Merge Join
Repalace Null CustomerSegment Values IF any
Slowly Changing Customer Dimension
Historical Attribute Inserts Output
Derived Column

Figure 1.7

When loading data to the DimCustomer dimension table I have to get the data from two staging tables. (CustomerLocation staging and Customer Staging)

Then using two sort components using the CustomerID I have merge two tables.
The by using a derived column I have used an expression to insert values for the null values which were in the CustomerSegment column.

Finally a slowly changing dimension component to update the dimension record values.

Then in the transformation process when inserting data to a hierarchy we have to first load the data to the table which is in the end of the hierarchy not the table which is directly connecting to the fact table.

So as shown in the figure 1.3 first I have transform and load data in to the DimProductCategory dimension table.

When loading data to these hierarchies I have used an OLE DB command component and execute a SQL query to load the data to the relevant dimension table.

2. Transform and load Product Category data

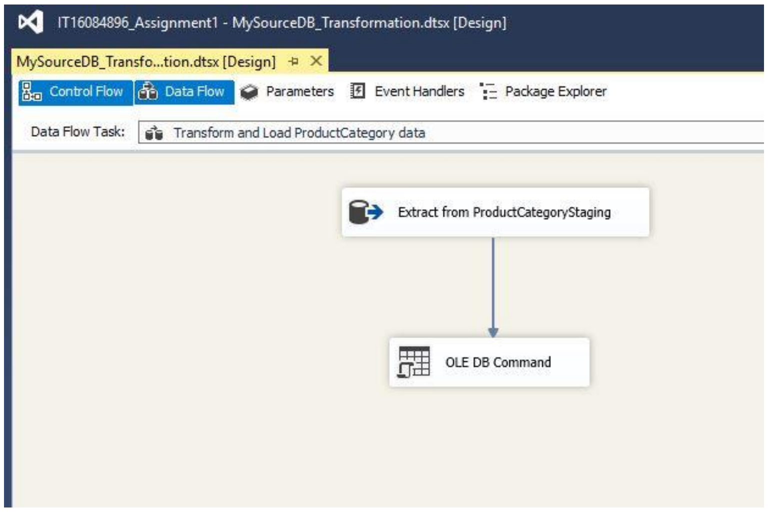

IT16084896_Assignment1 - MySourceDB_Transformation.dtsx [Design]
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
Transform and Load ProductCategory data
Extract from ProductCategoryStaging
OLE DB Command

Figure 1.8

SQL query used to update the product category details…

```
Create Procedure dbo.UpdateProductCategoryDetails
@ProductCategoryID int,
@ProductCategoryName nvarchar(50)
as
Begin
if not exists (select ProductCategoryID
from dbo.DimProductCategory
where ProductCategoryAlternateID = @ProductCategoryID
and ProductCategoryName = @ProductCategoryName )
begin
insert into dbo.DimProductCategory
(ProductCategoryAlternateID, ProductCategoryName)
values
(@ProductCategoryID, @ProductCategoryName)
end;
End;
```

3. Transform and load product container data

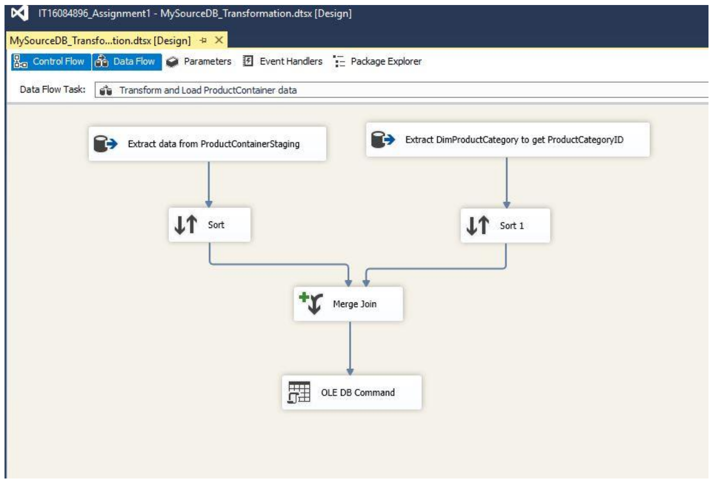

IT16084896_Assignment1 - MySourceDB_Transformation.dtsx [Design]
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
Transform and Load ProductContainer data
Extract data from ProductContainerStaging
Extract DimProductCategory to get ProductCategoryID
Sort
Sort 1
Merge Join
OLE DB Command

Figure 1.9

SQL query used to update the product container details…

```
Create Procedure dbo.UpdateProductContainerDetails
@ProductContainerID int,
@ProductCategoryID int,
@ProductContainerName nvarchar(50)
as
Begin
if not exists (select ProductContainerID
from dbo.DimProductContainer
where ProductContainerAlternateID = @ProductContainerID
and ProductContainerName = @ProductContainerName
and ProductCategoryID = @ProductCategoryID )
begin
insert into dbo.DimProductContainer
(ProductContainerAlternateID, ProductContainerName,
ProductCategoryID)
values
(@ProductContainerID, @ProductContainerName,
@ProductCategoryID)
end;
End;
```

4. Transform and load product sub category data

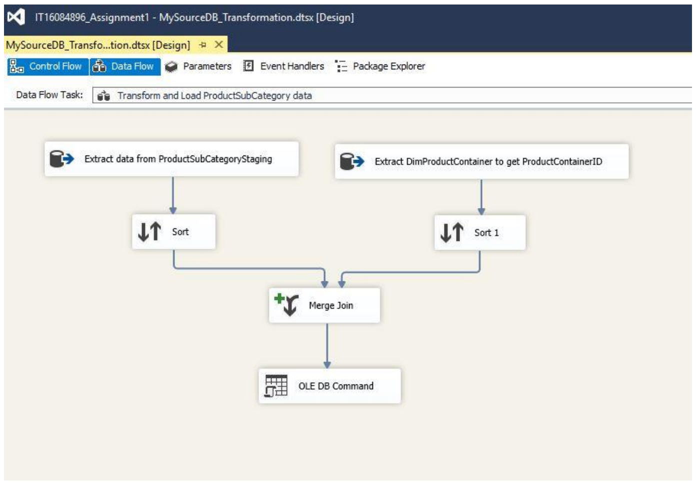

IT16084896_Assignment1 - MySourceDB_Transformation.dtsx [Design]
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
Transform and Load ProductSubCategory data
Extract data from ProductSubCategoryStaging
Extract DimProductContainer to get ProductContainerID
Sort
Sort 1
Merge Join
OLE DB Command

Figure 2.0

SQL query used to update the product sub category details…

```
Create Procedure dbo.UpdateProductSubCategoryDetails
@ProductSubCategoryID int,
@ProductContainerID int,
@ProductSubCategoryName nvarchar(50)
as
Begin
if not exists (select ProductSubCategoryID
from dbo.DimProductSubCategory
where ProductSubCategoryAlternateID =
@ProductSubCategoryID
and ProductSubCategoryName = @ProductSubCategoryName
and ProductContainerID = @ProductContainerID )
begin
insert into dbo.DimProductSubCategory
(ProductSubCategoryAlternateID, ProductSubCategoryName,
ProductContainerID)
values
(@ProductSubCategoryID, @ProductSubCategoryName,
@ProductContainerID)
end;
End;
```

5. Transform and load product data

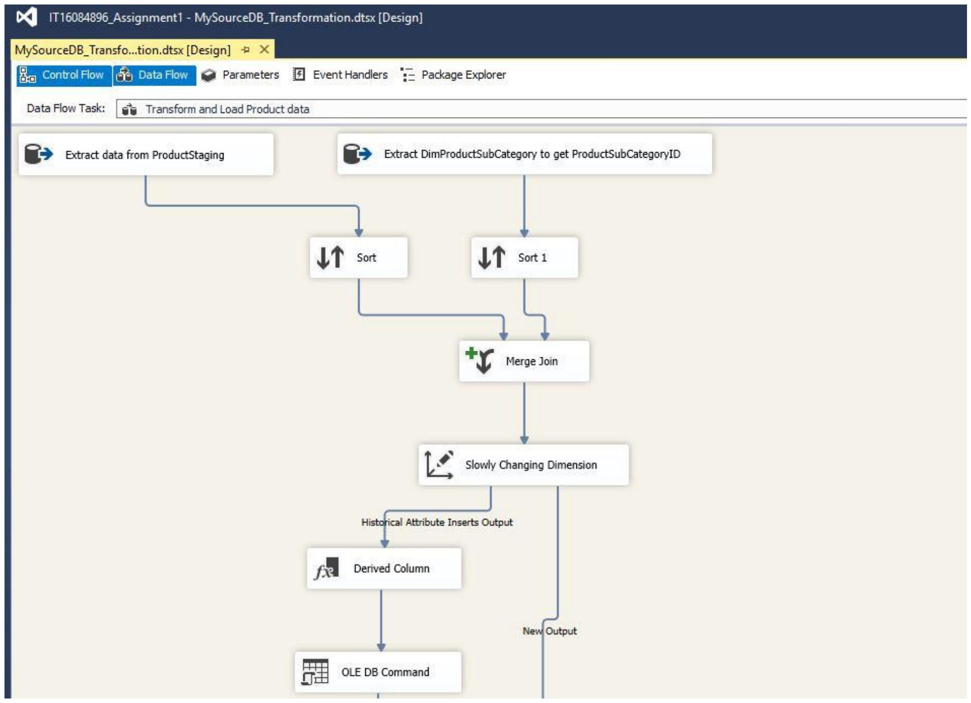

IT16084896_Assignment1 - MySourceDB_Transformation.dtsx [Design]
MySourceDB_Transfo...tion.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task: Transform and Load Product data
Extract data from ProductStaging
Extract DimProductSubCategory to get ProductSubCategoryID
Sort
Sort 1
Merge Join
Slowly Changing Dimension
Historical Attribute Inserts Output
Derived Column
New Output
OLE DB Command

Figure 2.1

6. Finally load the fact table

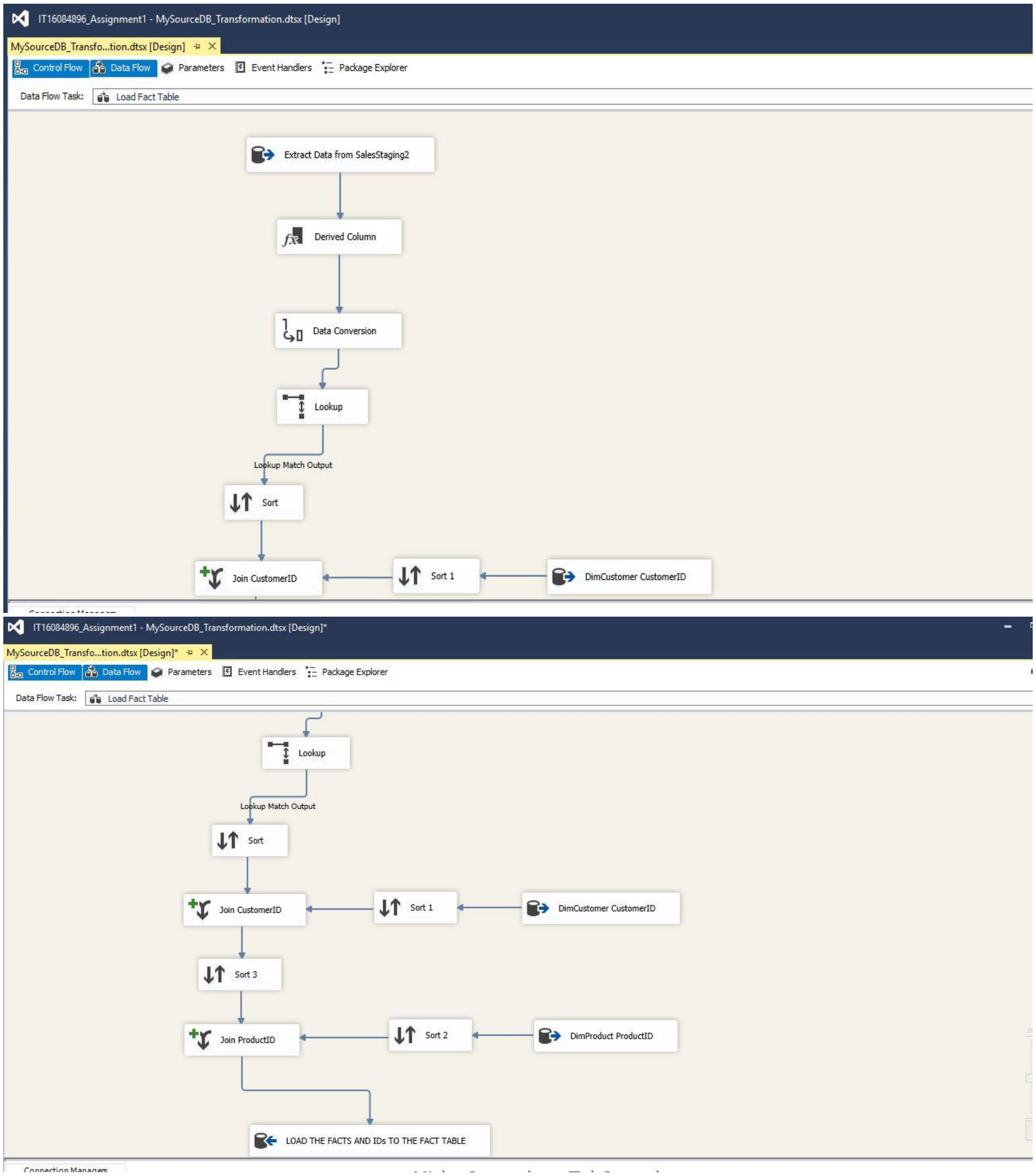


Figure 2.2