DWBI- Practical 2

1. Introduction to the Worksheet

In the previous worksheet, we configured the environment to work on & designed the data warehouse architecture including the staging environment. We will continue from there in this work sheet by designing the ETL to extract data from the source systems and load them into the staging database, and then, extract from the staging database, transform and load them to the data warehouse.

2. Tools Required

Below tools should be available to continue with this tutorial:

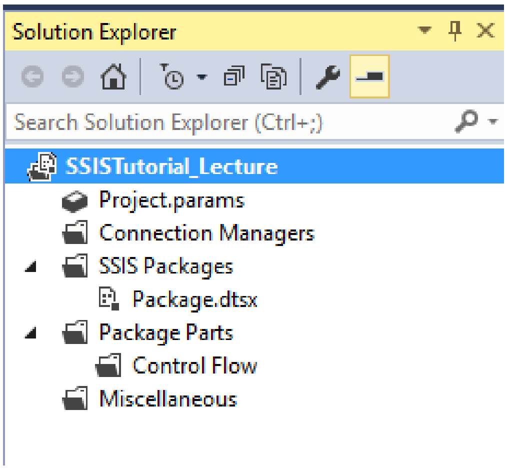
- SQL Server Data Tools
- SQL Server Management Studio

3. Creating the SSIS Project

Before starting with ETL, lets create an SSIS Project using Visual Studio Data Tools.

1. Open Visual Studio Data Tools in 'Administrator' mode.
2. Create an ***Integration Services*** project in the folder you created in C drive to store the '**CustomerAddress.txt**' file.

   (C:\SLIIT Data Warehouse Solution Tutorial\SSISTutorial<Student Number>\).
3. Once the solution is created, you will see a similar folder structure in ***Solution Explorer***.



4. Rename the ‘**Package.dtsx**’ to ‘**SLIIT_Retail_Load_Staging.dtsx**’.
5. Copy and paste the ETL document sheets you prepared to the solution folder.
6. Right click on the solution name in Solution Explorer and select ***Add à Existing Item…***
7. Select the data mapping excel file and attach it to the solution.
8. Double click and open the file to make sure it is attached properly.

Read more to understand the items, tabs & other options available in the SSIS solution interface to understand what each of them do.

4. Extracting Data from Source Database

The first step of the SSIS ETL process is the extraction of data from source systems. In the scenario discussed in practical sessions, we have two data sources (SQL database source & addresses text file). First, let’s consider the SQL Server database with below tables:

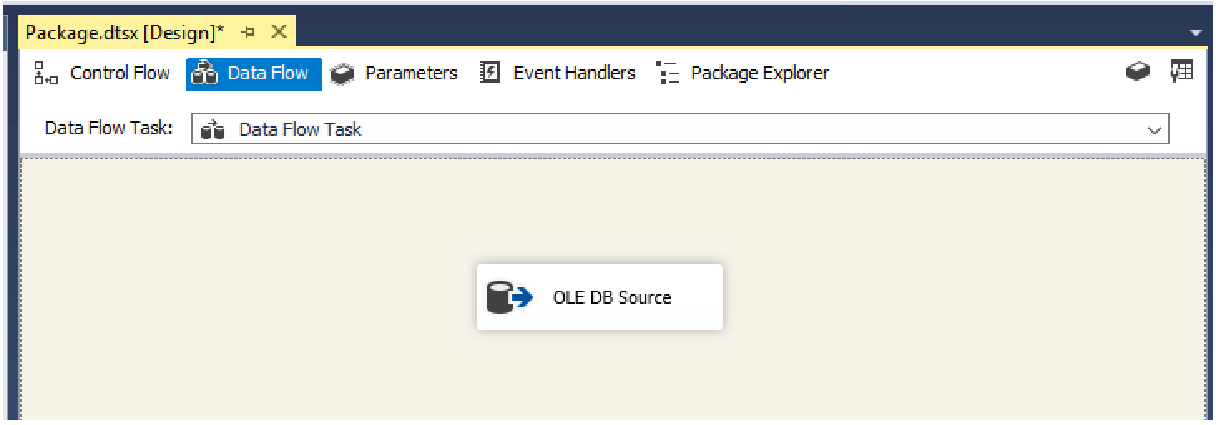
- IndividualCustomer
- Product
- ProductSubCategory
- ProductCategory
- SalesOrderDetails
- SalesOrderHeader

Let’s first extract all the data in these tables to staging area tables. Right now, we do not have any staging tables, but only the staging database, ‘**SLIIT_Retail_Staging**’. We will create them through SSIS. For each source table, we will have a separate staging table in the staging database. Let’s start with customer data.

1. Double click on ‘**SLIIT Retail Load Staging.dtsx**’ file to open the design surface if it is not already open.
2. Note that there are multiple options in the ***SSIS Toolbox***. Read more to understand what each item does.
3. First, we need to extract data from the source tables. To do so, drag and drop a ***Data Flow Task*** item to the ***Control Flow*** area.
4. Right click the ***Data Flow Task*** and select ***Rename*** and rename the item as ‘**Extract Customer Data to Staging**’.

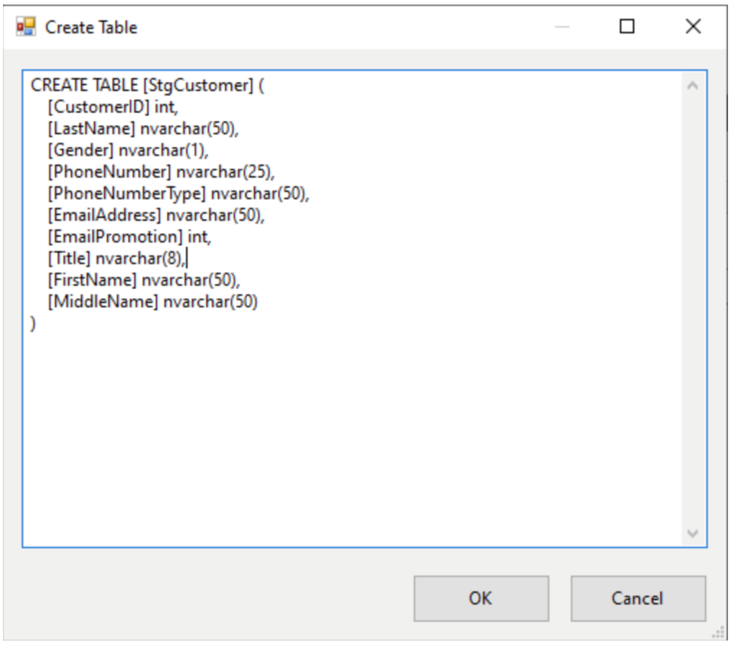
5. Double click on the '**Extract Customer Data to Staging**' to configure the data flow task.
6. On the ***Data Flow*** design surface, Drag and drop an ***OLE DB Source*** component from ***SSIS Toolbox*** under ***Other Sources***.
7. Double click ***OLE DB Source*** to open the ***OLE DB Source Editor*** dialog box.
8. Click ***New*** button to create a new database connection. In the ***Configure OLE DB Connection Manager*** click on the ***New*** button to create a new connection.
9. Provide the ***Server Name***, provide credentials to log on to the server (SQL Server Authentication is recommended over Windows Authentication as once scheduled, scheduler cannot access windows user credentials). Select the database as '**SLIIT_RetailSourceDB**'. Finally, click on ***Test Connection*** to check the connection attempt.
10. If the attempt is successful, click on ***OK*** button to return to the ***OLE DB Source Editor***.
11. In ***OLE DB Source Editor***, select the '**IndividualCustomer**' table from the table dropdown.
12. Click on ***Columns*** and make sure that all the columns are selected and click on the ***OK*** button to complete the configuration.

Now the ***Data Flow*** surface should look similar to below image:



13. Now to configure the destination, drag and drop an ***OLE DB Destination*** item from ***SSIS Toolbox*** under ***Other Destinations*** and link the ***OLE DB Source*** with ***OLE DB Destination*** item using the blue color line.
14. Right click on ***OLE DB Destination*** and click **Rename** to rename the item as '**Load data to Customer Staging table**'.
15. Double click to open the ***OLE DB Destination Editor***.

16. The connection manager shows your source database. Since we are planning to load data to the '**SLIIT_Retail_Staging**' database, create a new connection to point to the staging database following step 9.
17. Since the staging table is not yet created, click on the ***New*** button to open the ***Create Table*** window.
18. Rename the table name as '**StgCustomer**'.

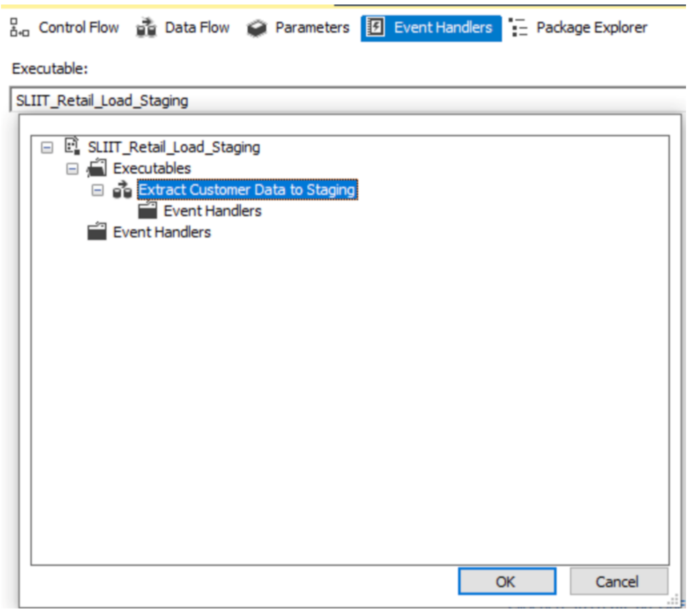


```
CREATE TABLE [StgCustomer] (
    [CustomerID] int,
    [LastName] nvarchar(50),
    [Gender] nvarchar(1),
    [PhoneNumber] nvarchar(25),
    [PhoneNumberType] nvarchar(50),
    [EmailAddress] nvarchar(50),
    [EmailPromotion] int,
    [Title] nvarchar(8),
    [FirstName] nvarchar(50),
    [MiddleName] nvarchar(50)
)
```

19. Click on the ***OK*** button to create the table in the staging database.
20. Click ***Mappings*** and make sure all the fields are mapped to the destination staging table.
21. Click on ***OK*** to complete the configuration.

At this stage, if we run the package, it will extract the data and store in the staging table. However, if we run the process multiple times, the staging table will be repeatedly loaded with customer data without truncating the data already available in the table. To handle this, do the following:

1. Click on ***Event Handlers***.
2. Under the ***Executable*** dropdown, select the '**Extract Customer Data to Staging**' and click ***OK***.



3. Select ***OnPreExecute*** from the ***Event Handler*** drop down and click on the link on the event handler body to create an 'OnPreExecute' event for the data flow task.

   NOTE: Executables are components in Control Flow area. Against these executables, events can't be triggered on different conditions such as ***OnPreExcute***, ***OnPostExecute***, ***OnError***.

4. On the event handler design surface, drag and drop an ***Execute SQL Task*** item from the ***SSIS Toolbox*** and rename it as '**Truncate Customer Staging Table**'.
5. Double click to open ***Execute SQL Task Editor***.
6. In the editor window, specify the ***Connection*** by selecting '**SLIIT_Retail_Staging**' as the database.
7. In the same window, use the below code for the ***SQL Statement*** to truncate the table.

```
truncate table dbo.StgCustomer
```

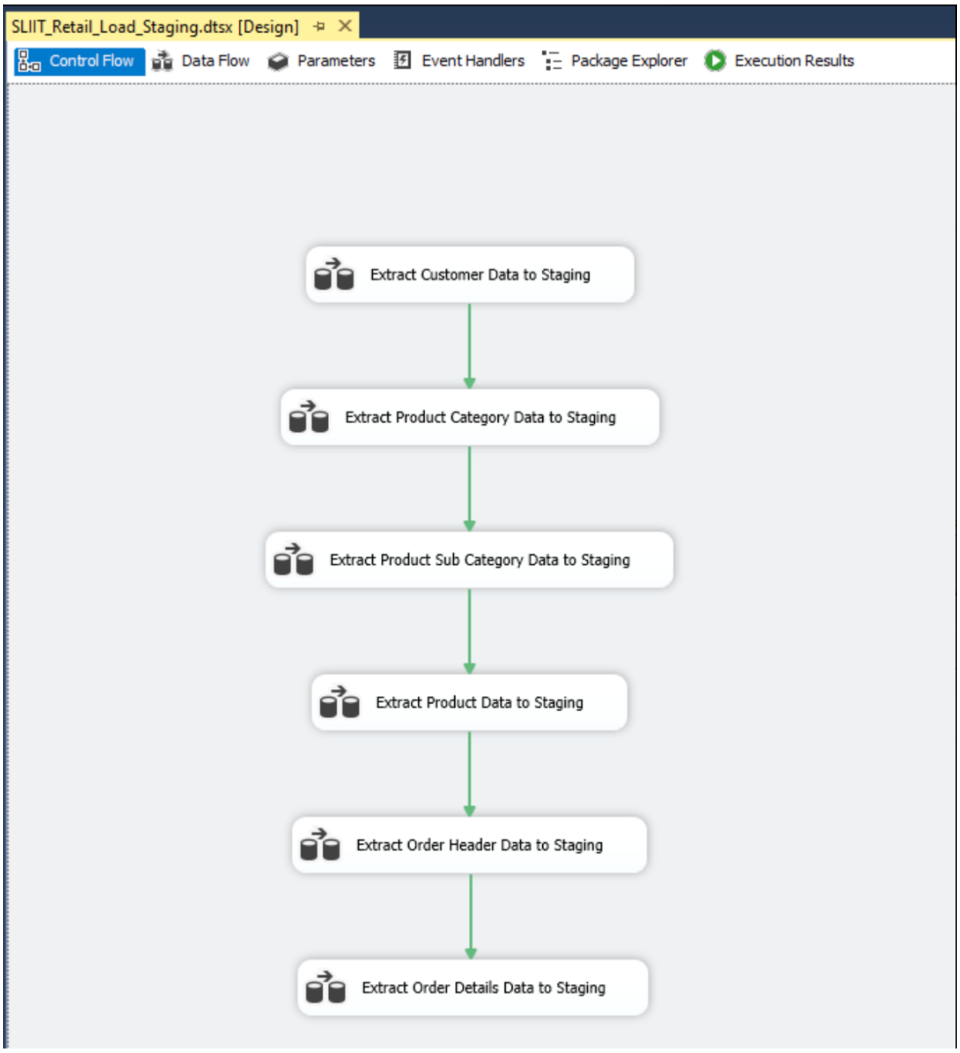
8. Click on ***OK*** button to complete the configuration.
9. Follow the same process for other source tables to delete the content in the target table before it is being loaded with data extracted from the source table. For each table, add a separate ***Data Flow Task*** and configure it accordingly.
   a. All the ***Data Flow Tasks*** should be connected to each other in a serial way in order to execute them in a serial order.

NOTE: Loading data from '**CustomerAddress.txt**' to staging database is discussed in the next section. Address data should go to a separate staging table. Address and customer data are combined when loading them from staging database to the data warehouse.

You may use following names as necessary when loading other tables from source database to staging database.

Source Table	***Data Flow Task***	***OLE DB Destination Task***	***Staging Table***	***Event Handler SQL Task***
IndividualCustomer	Extract Customer Data to Staging	Load data to Customer Staging table	StgCustomer	Truncate Customer Staging Table
Product	Extract Product Data to Staging	Load data to Product Staging table	StgProduct	Truncate Product Staging Table
ProductSubCategory	Extract Product Sub Category Data to Staging	Load data to Product Sub Category Staging table	StgProductSubCategory	Truncate Product Sub Category Staging Table
ProductCategory	Extract Product Category Data to Staging	Load data to Product Category Staging table	StgProductCategory	Truncate Product Category Staging Table
SalesOrderDetails	Extract Order Details Data to Staging	Load data to Order Details Staging table	StgOrderDetails	Truncate Order Details Staging Table
SalesOrderHeader	Extract Order Header Data to Staging	Load data to Order Header Staging table	StgOrderHeader	Truncate Order Header Staging Table

When you have completed up to this point, ***Control Flow*** design area should look like below:



Extract Customer Data to Staging

Extract Product Category Data to Staging

Extract Product Sub Category Data to Staging

Extract Product Data to Staging

Extract Order Header Data to Staging

Extract Order Details Data to Staging

5. Extracting Data from Text File

To extract data from ‘**CustomerAddress.txt**’ file, we will use a ***Flat File Source*** in ***SSIS Toolbox*** under ***Other Sources***.

1. Drag and drop a ***Data Flow Task*** item to the ***Control Flow*** area and connect it.
2. Right click the ***Data Flow Task*** and select ***Rename*** and rename the item as ‘**Extract Customer Address Data to Staging**’.
3. Double click on the ‘**Extract Customer Address Data to Staging**’ to configure the data flow task.
4. Drag and drop a ***Flat File Source***.
5. Double click ***on Flat File Source*** task to open the ***File Source Editor***.
6. In the Flat ***File Source Editor***, click the ***New*** button and provide a connection to the text file by giving the path to the file.

7. Select the ***Format*** as ***Delimited*****.**

8. Click on ***Columns*** to view and understand whether it seperates the columns accurately. Make sure the ***Row delimiter*** is set to ***{CR}{LF}*** and ***Column Delimiter*** is set to ***Tab {t}***.

9. Click on ***Advanced*** option to configure the data type for each column.

10. Select ***CustomerID*** column and set its data type to ***Numeric [DT_NUMERIC]***.

11. Set the data type of all the other fields to ***Unicode String [DT_WSTR]***.

Read more to understand the difference between ***String [DT_STR]*** and ***Unicode String [DT_WSTR]***.

10. Set the ***OutputColumnWidth*** for text column as follows.

    a. Address Type – 50
    b. Address Line 1 & 2 – 60
    c. City – 30
    d. State Province Name – 50
    e. Postal Code – 15
    f. Country Region – 50

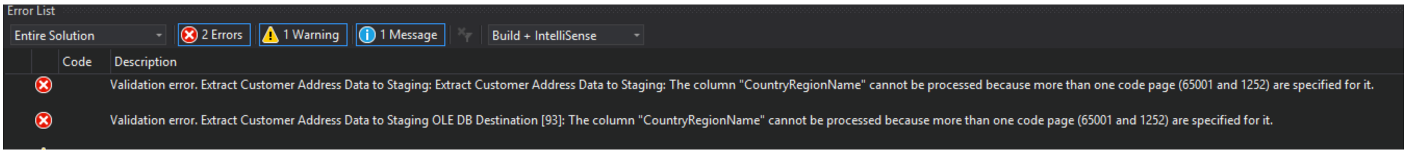
11. Click ***OK*** to complete the configuration.

12. Check the box that says ***Retail null values from the source as null values in the data flow***. (Change this and see what happens when loading data!)

13. Click ***OK*** to complete the configuration.

14. Add an ***OLE DB Destination*** task, rename it as '**Load data to Customer Addresses Staging table**' and configure it the way it was done previously. Please remember to change the data type of **CustomerID** to **INT** when creating the table. You may name the staging table as '**StgCustomerAddresses**'.

15. After configuring you might see an error as below.

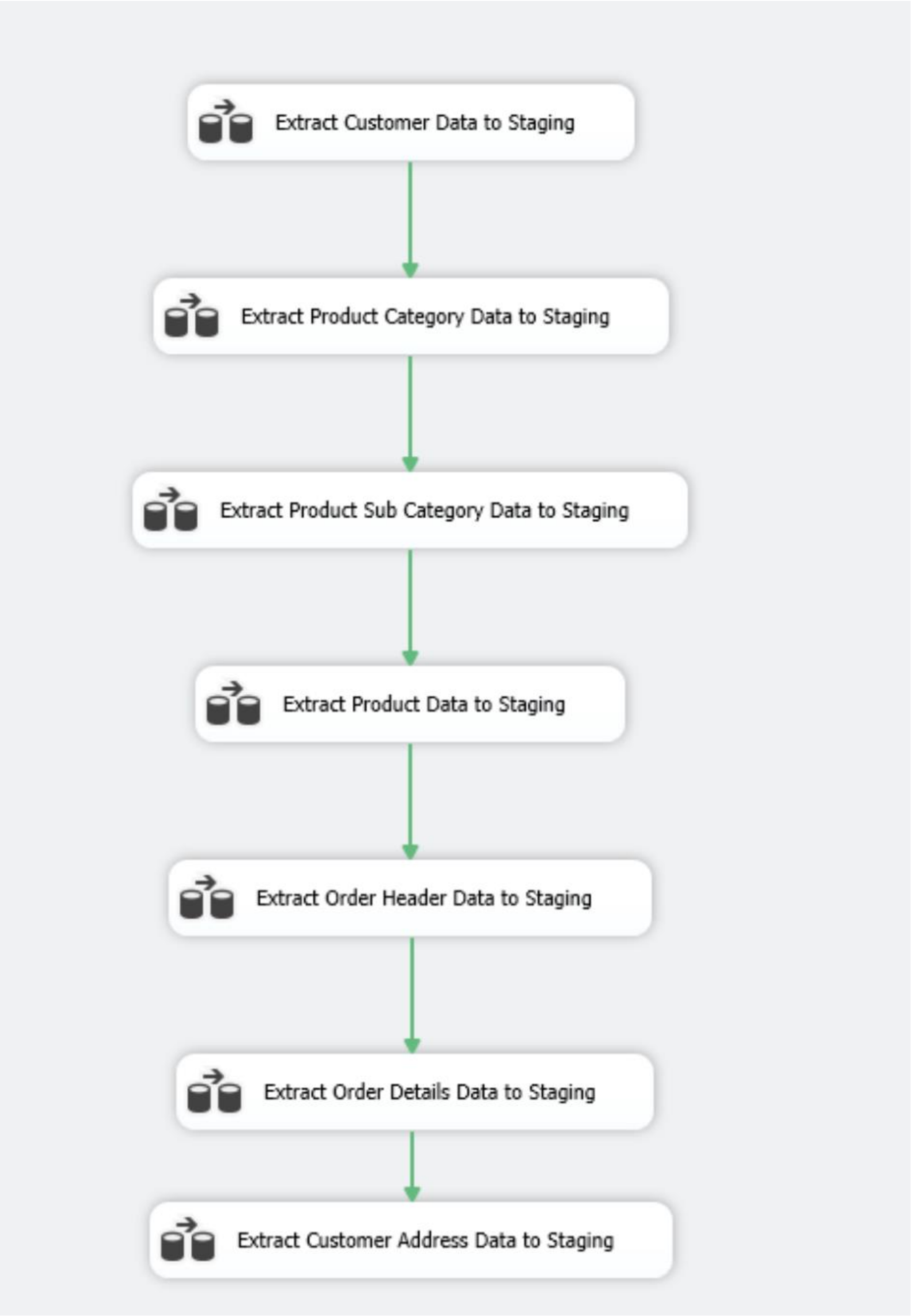


```
Validation error. Extract Customer Address Data to Staging: Extract Customer Address Data to Staging: The column "CountryRegionName" cannot be processed because more than one code page (65001 and 1252) are specified for it.
Validation error. Extract Customer Address Data to Staging OLE DB Destination [93]: The column "CountryRegionName" cannot be processed because more than one code page (65001 and 1252) are specified for it.
```

    a. If so, set the value of ***DefaultCodePage*** to **65001** and set the value of ***AlwaysUseDefaultCodePage*** property to **True** (these are properties of '**Load data to Customer Addresses Staging table**' task).

16. Go to ***Event Handlers*** tab and configure the ***OnPreExecute*** handler to truncate the customer addresses staging table. You may rename the ***Execute SQL Task*** item as '**Truncate Customer Addresses Staging Table**'.

When you have completed up to this point, ***Control Flow*** design area should look like below:



Extract Customer Data to Staging
Extract Product Category Data to Staging
Extract Product Sub Category Data to Staging
Extract Product Data to Staging
Extract Order Header Data to Staging
Extract Order Details Data to Staging
Extract Customer Address Data to Staging

6. Data Profiling

Since we have source data in our staging tables, now we can use the staging table data to analyze how the data looks like to determine what type of transformations we need to perform on the data.

Create a new package and rename it as ‘**Data_Profiling.dtsx**’.

1. Right click on ***SSIS Packages*** and select ***New SSIS Package*** and rename it as ‘**Data Profiling.dtsx**’.
2. In the ***Control Flow*** of ‘**Data Profiling Pack.dtsx**’, drag and drop a ***Data Profiling Task*** and double click to open ***Data Profiling Task Editor*** window.
3. Click on ***Quick Profile*** button to open up ***Single Table Quick Profile Form***.
4. Click on ***New*** button and create a connection to ‘**SLIIT_Retail_Staging**’.
5. From the ***Table or View*** dropdown, select ‘**StgCustomer**’ table.
6. Select all the check boxes and click on ***OK*** button to complete the configuration.
7. Back in ***Data Profiling Task Editor*** window go to ***General*** page.
8. For the ***Destination*****,** select ***<New File Connection…>***.
9. In the ***File Connection Manager Editor*** window, select ***Create File*** as the value for ***Usage type*** and provide the file path to any location as you prefer and click ***OK*** to complete the configuration.
10. Save the package and right clicking on the ‘**Data_Profiling.dtsx**’ and select ***Execute Package*** to execute ‘**Data_Profiling.dtsx**’ package in order to profile the customer data.
11. Once the package is executed, double click the ***Data Profiling Task*** and click on ***Open Profile Viewer…*** button to view the output.

Read more on how to profile data and how to understand the data analysis. Do the same for all staging tables to analyze the data by yourselves.