DWBI- Practical 5

1. Introduction to the Worksheet

In the previous worksheet, we transformed and customer data from the staging database to the data warehouse using SSIS. This worksheet focuses on loading order data (transactional data) into the fact table in the data warehouse with necessary transformations.

When loading fact tables, it is important to link dimension tables with necessary surrogate keys. These keys are required to get the dimension details for the measures (facts) when analyzing data (such as with reports). Note that the process taken to update the fact table depends on the type of the fact table. *If it's a transactional fact table, it is usually an insert that takes place. However, if it's an accumulating fact, it could have both inserts and updates*.

In the use case considered for these practical sessions, we are working on sales data that contains loads of individual transactions. In such a scenario, most of the time we use transactional fact tables. Thus, let's assume that there are no updates to existing transactions. However, you might need to have updates incorporated to other types of fact tables depending on the use case.

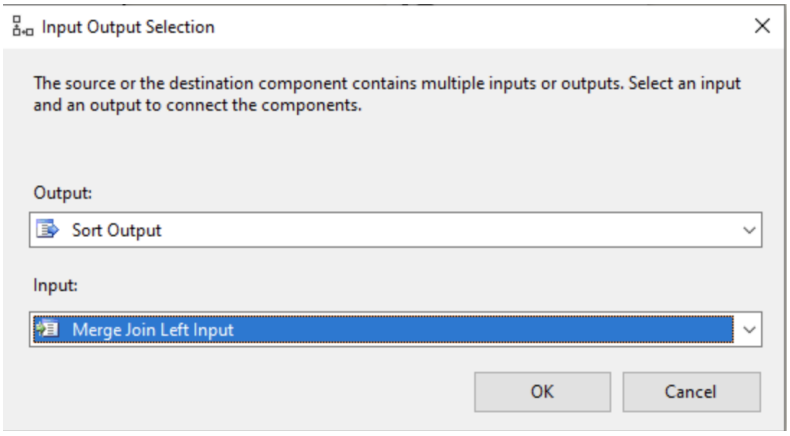
2. Order Transaction Data Transformation and Loading

Follow the step given below:

2.1. Order Data Extraction

1. Open Visual Studio Data Tools in 'Administrator' mode and open previously created solution.
2. Open '**SLIIT_Retail_Load_DW.dtsx**'.
3. Drag and drop a ***Data Flow Task***, rename it as '**Transform and Load Sales Fact Table**' and go to the ***Data Flow*** tab to design the '**Transform and Load Sales Fact Table**' data flow.
4. Drag and drop ***OLE DB Source*****,** rename as '**Extract from Order Header Staging**' and configure it to extract the '**StgOrderHeader**' table. Make sure all the columns are selected.

5. Drag and drop another ***OLE DB Source***, rename as '**Extract from Order Details Staging**' and configure it to extract the '**StgOrderDetails**' table. Make sure all the columns are selected.
6. Drag and drop a ***Sort*** component from ***SSIS Toolbox***, rename it as '**Order Header Sort**' and connect it with the '**Extract from Order Header Staging**' component using the blue line.
7. Double click '**Order Header Sort**' component and select '**SalesOrderId**' as the sorting column by ticking on the checkbox in front of '**SalesOrderId**'. Click ***OK***.
8. Drag and drop another ***Sort*** component, rename it as '**Order Details Sort**' and connect it with the '**Extract from Order Details Staging**' component using the blue line.
9. Double click '**Order Details Sort**' component and select '**SalesOrderID**' as the sorting column. Click ***OK***.
10. Drag and drop ***Merge Join*** component and link the '**Order Header Sort**' component using the blue line.
11. In the ***Input Output Selection*** window, select ***Merge Join Left Input*** as the value for ***Input*** field. Click ***OK***.



12. Link the '**Order Details Sort**' component to ***Merge Join*** using the blue link of '**Order Details Sort**' and double click ***Merge Join*** to configure the join.
13. In the ***Merge Join Transformation Editor*** window, you should be able to see two sources are connected by '**SalesOrderID**' of both '**Order Header Sort**' and '**Order Details Sort**' and the ***Join Type*** is ***Inner Join*****.** Joining columns are automatically picked based on the sort

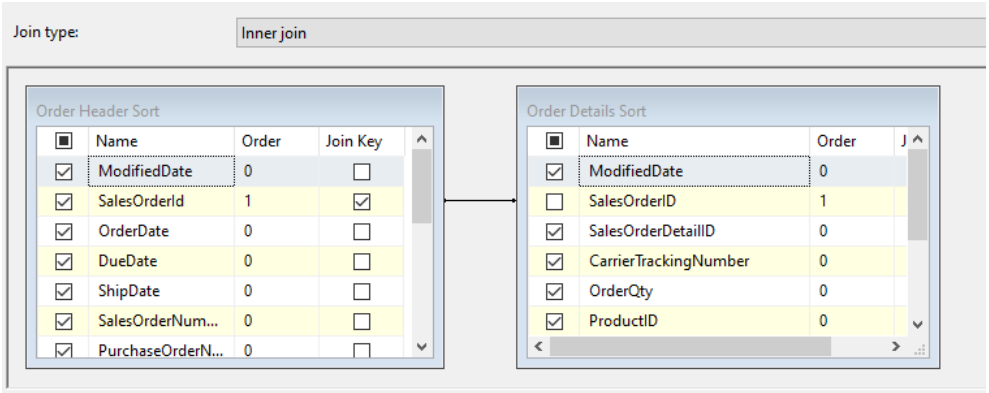
column you provided in the ***Sort*** components. We can assume that for every order header record there is at least one order detail record. Thus, order type can be kept as ***Inner Join***.

14. Select fields from '**Order Header Sort**' and '**Order Details Sort**' (based on the required fields specified in the '**FactSales**' table) as specified in the image below.

    Note that in the column list grid:

    - in the left most column, input table is displayed,
    - center column, input column is displayed and
    - in the right most column, output alias is displayed.

    Note that output alias for 2 'ModifiedDate' columns coming from '**Order Header Sort**' and '**Order Details Sort**' tables are renamed accordingly.



Input	Input Column	Output Alias
Order Header Sort	SalesOrderId	SalesOrderId
Order Header Sort	OrderDate	OrderDate
Order Header Sort	DueDate	DueDate
Order Header Sort	ShipDate	ShipDate
Order Header Sort	SalesOrderNumber	SalesOrderNumber
Order Header Sort	PurchaseOrderNumber	PurchaseOrderNumber
Order Header Sort	CustomerID	CustomerID
Order Details Sort	SalesOrderDetailID	SalesOrderDetailID
Order Details Sort	ProductID	ProductID
Order Details Sort	CarrierTrackingNumber	CarrierTrackingNumber
Order Header Sort	SubTotal	SubTotal
Order Header Sort	TaxAmt	TaxAmt
Order Header Sort	Freight	Freight
Order Header Sort	TotalDue	TotalDue
Order Details Sort	OrderQty	OrderQty
Order Details Sort	UnitPrice	UnitPrice
Order Details Sort	UnitPriceDiscount	UnitPriceDiscount
Order Details Sort	LineTotal	LineTotal
Order Header Sort	ModifiedDate	OrderHeaderModifiedDate
Order Details Sort	ModifiedDate	OrderDetailsModifiedDate

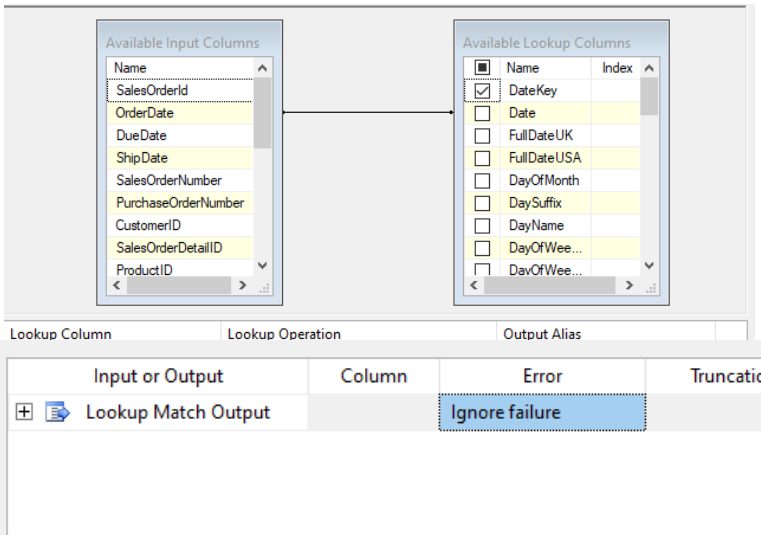
Next steps are to join relevant dimension table in order to obtain corresponding surrogate keys. Surrogate keys are required for '**OrderDate**', '**DueDate**', '**ShipDate**', '**CutomerID**', and '**ProductID**'. This can be done using ***Lookup*** component.

2.2. Order Date Lookup

First, let's connect '**OrderDate**''with the '**DimDate**' and obtain '**DateKey**'.

1. Drag and drop a ***Lookup*** component from ***SSIS Toolbox***, rename it as '**Order Date Lookup**' and connect it with ***Merge Join*** component using the blue line.
2. Double click '**Order Date Lookup**' component to open the ***Lookup Transformation Editor*** window. In ***General*** section, select ***Full cache*** as the ***Cache mode***, and ***OLE DB connection manager*** as the ***Connection type*** and select ***Ignore failure*** from the ***Specify how to handle rows with no matching entries*** drop down list. This specifies what to be done with non-matching records. Note that ignoring is not the ideal option to have; but to keep the scenario simple let's use ignore option for now.
3. In ***Connection*** section (second tab in left), ensure the connection is set to '**SLIIT_Retail_DW**' and select '**DimDate**' as the table. We will connect to '**DimDate**' to lookup for matching '**Date**' and pick the corresponding '**DateKey**'.
4. In **Columns** section, map '**OrderDate**' in the ***Available Input Columns*** box with '**Date**' in the ***Available Lookup Columns*** box and select '**DateKey**'. Once you tick the checkbox, '**DateKey**' will appear in the grid below.
5. In the grid below, set ***Lookup Operation*** as ***<add as new column>*** and provide the ***Output Alias*** as '**OrderDateSK**'; this will take both '**OrderDate**' and '**OrderDateSK**' forward to the next step; but we only need '**OrderDateSK**'.

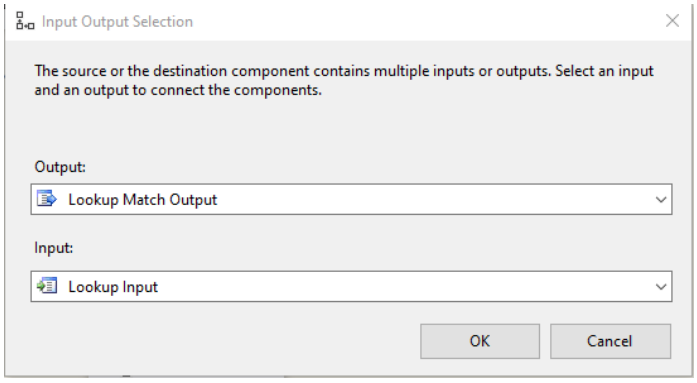
6. In ***Error*** ***Output*** section, in the grid of the window, for ***Lookup Match Output*** entry under ***Input or Output*** column, ensure ***Ignore failure*** is selected as the ***Error*** option.



7. Click ***OK***.

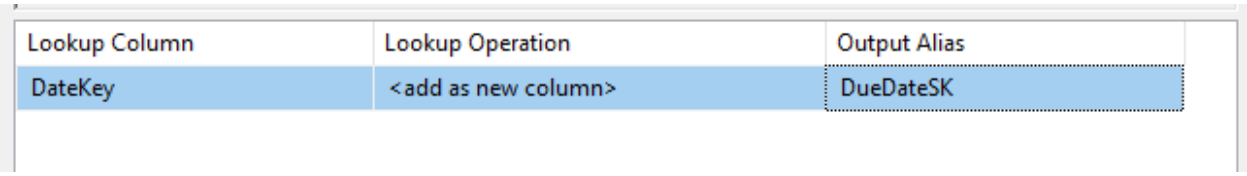
2.3. Due Date Lookup

1. Drag and drop another ***Lookup*** component from ***SSIS Toolbox***, rename it as '**Due Date Lookup**' and connect it with '**Order Date Lookup**' component using the blue line. An ***Input Output Selection*** window will pop-up. Select ***Lookup Match Output*** and click ***OK***.



2. Double click '**Due Date Lookup**' component to open the ***Lookup Transformation Editor*** window. Set the settings in ***General*** section accordingly (follow step 2 in section 2.21).
3. In ***Connection*** section, ensure the connection is set to '**SLIIT_Retail_DW**' and select '**DimDate**' as the table.

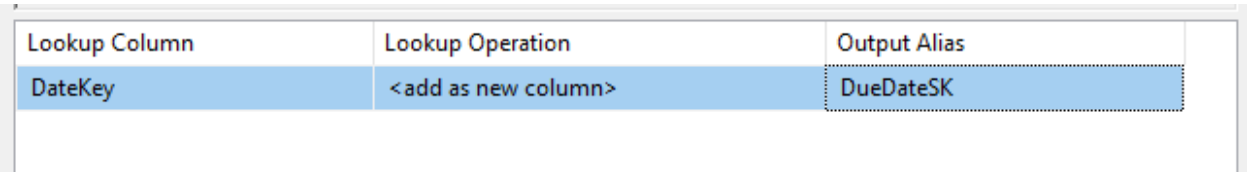
4. In **Columns** section, map '**DueDate**' in the ***Available Input Columns*** box with '**Date**' in the ***Available Lookup Columns*** box and select '**DateKey**'. Once you tick the checkbox, '**DateKey**' will appear in the grid below.
5. In the grid below, set ***Lookup Operation*** as ***<add as new column>*** and provide the ***Output Alias*** as '**DueDateSK**'.



6. In ***Error Output*** section, ensure ***Ignore failure*** is selected as the ***Error*** option for ***Lookup Match Output*** entry.
7. Click ***OK***.

2.4. Ship Date Lookup

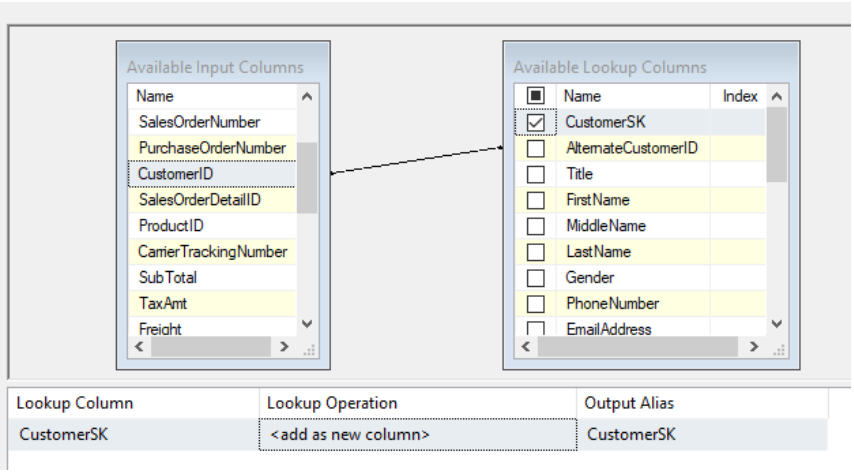
1. Drag and drop another ***Lookup*** component from ***SSIS Toolbox***, rename it as '**Ship Date Lookup**' and connect it with '**Due Date Lookup**' component using the blue line. An ***Input Output Selection*** window will pop-up. Select ***Lookup Match Output*** and click ***OK***.
2. Double click '**Ship Date Lookup**' component to open the ***Lookup Transformation Editor*** window. Set the settings in ***General*** section accordingly (follow step 2 in section 2.21).
3. In ***Connection*** section, ensure the connection is set to '**SLIIT_Retail_DW**' and select '**DimDate**' as the table.
4. In **Columns** section, map '**ShipDate**' in the ***Available Input Columns*** box with '**Date**' in the ***Available Lookup Columns*** box and select '**DateKey**'. Once you tick the checkbox, '**DateKey**' will appear in the grid below.
5. In the grid below, set ***Lookup Operation*** as ***<add as new column>*** and provide the ***Output Alias*** as '**ShipDateSK**'.



6. In ***Error Output*** section, ensure ***Ignore failure*** is selected as the ***Error*** option for ***Lookup Match Output*** entry.
7. Click ***OK***.

2.5. Customer Lookup

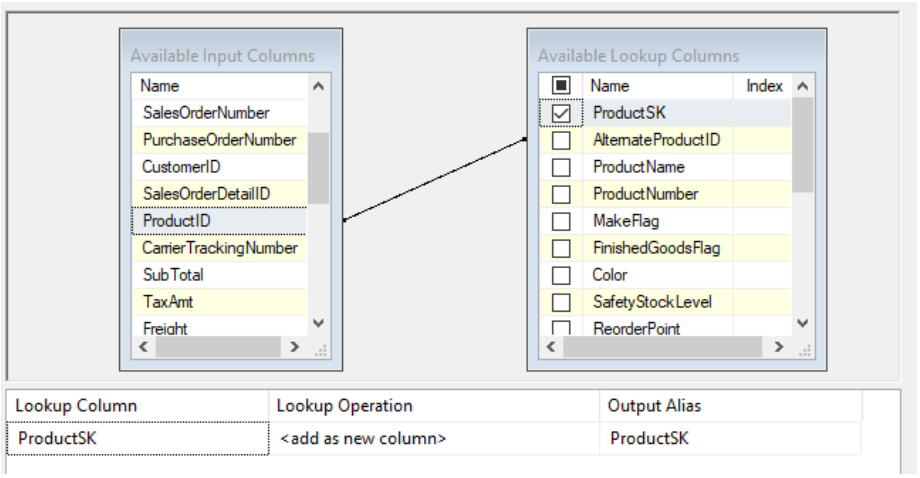
1. Drag and drop another ***Lookup*** component from ***SSIS Toolbox***, rename it as '**Customer Lookup**' and connect it with '**Ship Date Lookup**' component using the blue line. An ***Input Output Selection*** window will pop-up. Select ***Lookup Match Output*** and click ***OK***.
2. Double click '**Customer Lookup**' component to open the ***Lookup Transformation Editor*** window. Set the settings in ***General*** section accordingly (follow step 2 in section 2.21).
3. In ***Connection*** section, ensure the connection is set to '**SLIIT_Retail_DW**' and select '**DimCustomer**' as the table.
4. In **Columns** section, map '**CustomerID**' in the ***Available Input Columns*** box with '**AlternateCustomerID**' in the ***Available Lookup Columns*** box and select '**CustomerSK**'. Once you tick the checkbox, '**CustomerSK**' will appear in the grid below.
5. In the grid below, set ***Lookup Operation*** as ***<add as new column>*** and ensure the ***Output Alias*** is '**CustomerSK**'.



6. In ***Error Output*** section, ensure ***Ignore failure*** is selected as the ***Error*** option for ***Lookup Match Output*** entry.
7. Click ***OK***.

2.6. Product Lookup

1. Drag and drop another ***Lookup*** component from ***SSIS Toolbox***, rename it as '**Product Lookup**' and connect it with '**Customer Lookup**' component using the blue line. An ***Input Output Selection*** window will pop-up. Select ***Lookup Match Output*** and click ***OK***.
2. Double click '**Product Lookup**' component to open the ***Lookup Transformation Editor*** window. Set the settings in ***General*** section accordingly (follow step 2 in section 2.21).
3. In ***Connection*** section, ensure the connection is set to '**SLIIT_Retail_DW**' and select '**DimProduct**' as the table.
4. In **Columns** section, map '**ProductID**' in the ***Available Input Columns*** box with '**AlternateProductID**' in the ***Available Lookup Columns*** box and select '**ProductSK**'. Once you tick the checkbox, '**ProductSK**' will appear in the grid below.
5. In the grid below, set ***Lookup Operation*** as ***<add as new column>*** and ensure the ***Output Alias*** is '**ProductSK**'.



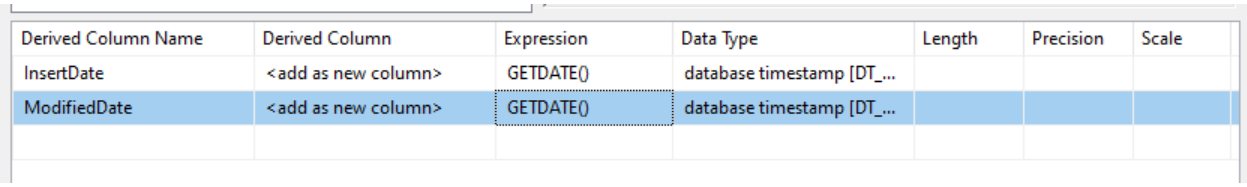
6. In ***Error Output*** section, ensure ***Ignore failure*** is selected as the ***Error*** option for ***Lookup Match Output*** entry.
7. Click ***OK***.

2.7. Load Sales Fact Table

Now we have successfully transformed transactional data. Next step is to insert data to the fact table.

1. Drag and drop a ***Derived Column*** component, rename it as '**Derive Insert and Modified Date Fields**' and connect with the '**Product Lookup**' component.
2. Double click on '**Derive Insert and Modified Date Fields**' to open ***Derived Column Transformation Editor*** window and add 2 new columns as '**InsertDate**' and '**ModifiedDate**' to the grid below.

   Expand ***Date/Time Functions*** folder on the right side box, drag and drop ***GETDATE() function*** to ***Expression*** field of both new columns in the grid. The grid should look like below:



Derived Column Name	Derived Column	Expression	Data Type	Length	Precision	Scale
InsertDate	<add as new column>	GETDATE()	database timestamp [DT_...			
ModifiedDate	<add as new column>	GETDATE()	database timestamp [DT_...			

3. Drag and drop an ***OLE DB Destination*** component from ***SSIS Toolbox***, rename it as '**Load FactSales**' and connect it with '**Derive Insert and Modified Date Fields**' component using the blue line.
4. Double click '**Load FactSales**' component to open the ***OLE DB Destination Editor*** window.
5. In ***Connection Manager*** section, ensure the connection is set to '**SLIIT_Retail_DW**', ***Data access mode*** is selected as ***Table or view – fast load*** and '**FactSales**' is selected from the ***Name of the table or the view*** drop down list.
6. In ***Mappings*** section, map ***Input Columns*** with ***Destination Columns*** as specified in the image below.

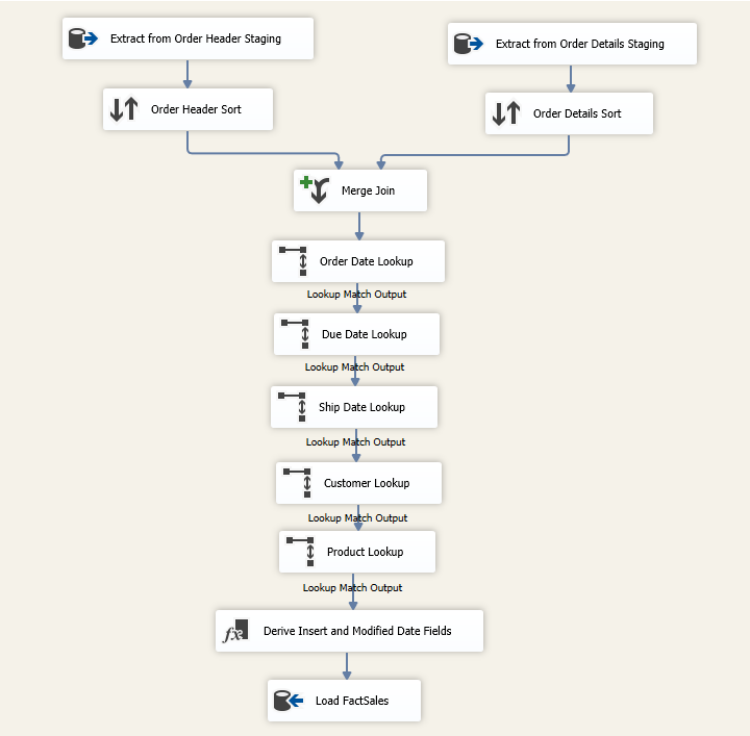
   Note that any values to four calculated fields:

   - '**RemainingPayment**'
   - '**TotalCostPerItem**'
   - '**TotalUnitCostWithTaxFreight**'

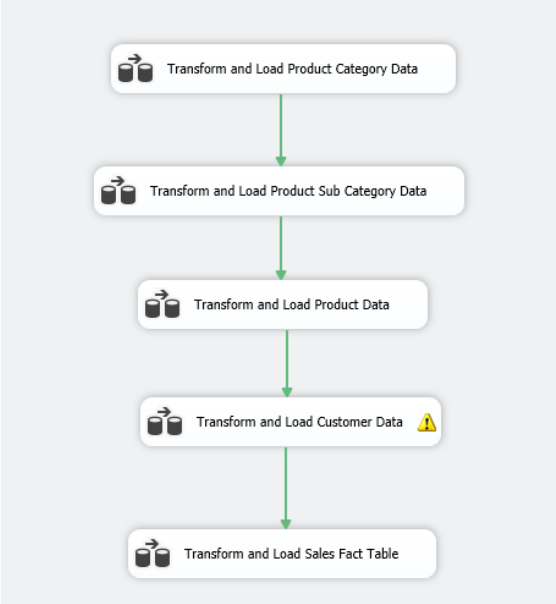
- '**TotalUnitCostBeforeTaxFreight**') are not mapped since those are automatically calculated in the database table level.

Input Column	Destination Column
ShipDateSK	ShipDateKey
SalesOrderNumber	SalesOrderNumber
PurchaseOrderNumber	PurchaseOrderNumber
CustomerSK	CustomerKey
SalesOrderId	SalesOrderID
SalesOrderDetailID	SalesOrderDetailsID
ProductSK	ProductKey
CarrierTrackingNumber	CarrierTrackingNumber
SubTotal	SubTotal
TaxAmt	TaxAmt
Freight	Freight
TotalDue	TotalDue
OrderQty	OrderQty
UnitPrice	UnitPrice
UnitPriceDiscount	UnitPriceDiscount
LineTotal	LineTotal
<ignore>	RemainingPayment
<ignore>	TotalCostPerItem
<ignore>	TotalUnitCostWithTaxFreight
<ignore>	TotalUnitCostBeforeTaxFreight
OrderHeaderModifiedDate	SrcHeaderModifiedDate
OrderDetailsModifiedDate	SrcDetailModifiedDate
InsertDate	InsertDate
ModifiedDate	ModifiedDate

7. Finally, '**Transform and Load Sales Fact Table**' data flow design should look like below:



8. In SSIS Control Flow tab, connect '**Transform and Load Customer Data**' and '**Transform and Load Sales Fact Table**' data flows in a way that sales fact data loading occurs at last.



9. Execute the package to observe how data gets loaded into data warehouse tables.

2.8. Connecting Packages

It is possible to execute one SSIS package from another package.

Since the '**SLIIT_Retail_Load_DW.dtsx**' (which loads data from staging database to data warehouse) package should be executed '**SLIIT_Retail_Load_Staging.dtsx**' (which loads data from sources to staging database), let's execute '**SLIIT_Retail_Load_DW.dtsx**' package as the last step of the '**SLIIT_Retail_Load_Staging.dtsx**' package.

1. Open '**SLIIT_Retail_Load_Staging.dtsx**'.
2. Add an ***Execute Package Task***, rename it as '**Execute SLIIT_Retail_Load_DW**' and connect it to the last **Data Flow Task**; '**Extract Customer Address Data to Staging**' in a way that '**Execute SLIIT_Retail_Load_DW**' component gets executed as the last step.
3. Double click '**Execute SLIIT_Retail_Load_DW**' to open ***Execute Package Task Editor***.
4. In the ***Package*** section, set ReferenceType as Project Reference and select '**SLIIT_Retail_Load_DW.dtsx**' as the ***PackageNameFromProjectReference***. This will execute the '**SLIIT_Retail_Load_DW.dtsx**' package as last step of the '**SLIIT_Retail_Load_Staging.dtsx**'
5. Click **OK** to complete the configuration and save all changes.
6. Execute '**SLIIT_Retail_Load_Staging.dtsx**'.

3. Scheduling with SQL Server Agent Jobs

As a research area try using SQL Server Agent to schedule SSIS packages created for data ingestion from source to data warehouse.

SQL Server Agent job can be found in ***SQL Server Management Studio***.

You can right click on ***SQL Server Agent*** and select ***Start*** to start the agent process.

Expand ***SQL Server Agent***, right click on ***Jobs*** and select ***New Job…*** to create a new scheduled job.

Explore!!!