



Data Warehousing & Business Intelligent (IT)

3rd Year, 1st Semester

Assignment 1

Submitted to

Sri Lanka Institute of Information Technology

Bachelor of Science Special Honors Degree in Data Science

IT17167710

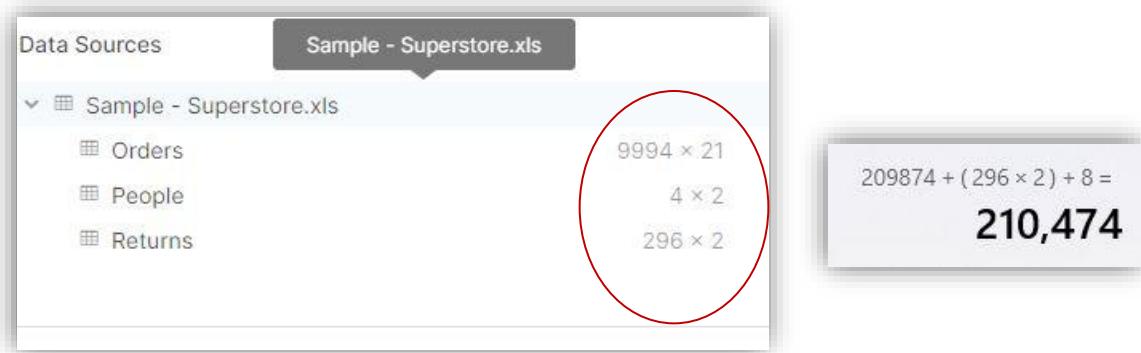
D.M.J.Prathapa

Weekday Batch

Step 1: Data set selection

I selected the superstore as the data set. It consists of a large CSV file along with two small CSV files. Furthermore, I have partitioned the main large CSV file into small sub CSV files. The sub CSV files consist of new IDs. The process of the CSV file partitioning was uploaded to the drive as a zip file. The zip consists of small video clips that reveal how I have separated the CSV files.

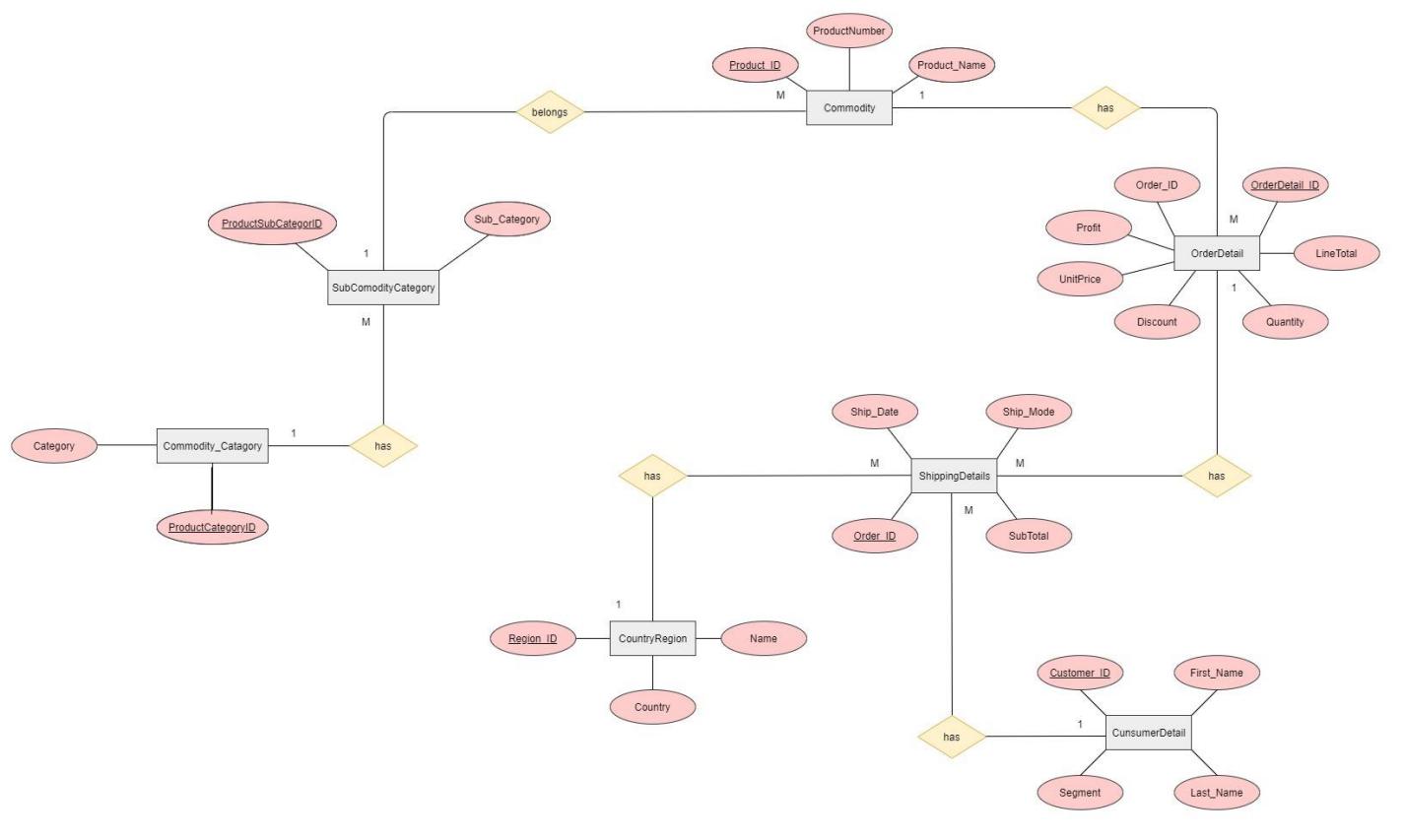
The data set was initiated with sufficient data, according to the assignment principles. The approval of choosing this data set was taken by the respected lecturer. Moreover, the superstore data set is created according to the conceptualized theoretical procedures including more than 210,000 data.



Data set Link

<https://www.kaggle.com/pruthvi1995/superstore-sales>

Er diagram



Step 2: Preparation of Data Sources

This implement data warehouse used data set view, it is Store Sale transaction data set, it partitioned into separate Sources files to implement data warehouse, such as Commodity.csv, Commodity_Catagory.csv, ConsumerDetails.csv, CountryRegion.csv, OrderDetail.csv, ShippingDetails.csv, Sub_Commodity_Catagory.csv, salesRegionRepresentative.csv and ConsumerLocationDetails.txt Text file

	A	B	C	D
1	Customer	First Name	Last Name	Segment
2	CG-12520	Claire	Gute	Consumer
3	DV-13045	Darrin	Van	Corporate
4	SO-20335	Sean	O'Donnell	Consumer
5	BH-11710	Brosina	Hoffman	Consumer
6	AA-10480	Andrew	Allen	Consumer
7	IM-15070	Irene	Maddox	Consumer
8	HP-14815	Harold	Pawlan	Home Office
9	PK-19075	Pete	Kriz	Consumer
10	AG-10270	Alejandro	Grove	Consumer
11	ZD-21925	Zuschuss	Donatelli	Consumer
12	KB-16585	Ken	Black	Corporate
13	SF-20065	Sandra	Flanagan	Consumer
14	EB-13870	Emily	Burns	Consumer
15	EH-13945	Eric	Hoffmann	Consumer
16	TB-21520	Tracy	Blumstein	Consumer
17	MA-17560	Matt	Abelman	Home Office
18	GH-14485	Gene	Hale	Corporate
19	SN-20710	Steve	Nguyen	Home Office
20	LC-16930	Linda	Cazamias	Corporate

ConsumerDetails.csv

A	B	C	D	E	F	G
Order ID	Order Date	Ship Date	Ship Mode	Customer ID	SubTotal	RegionID
CA-2014-1000	09/07/2014	9/13/2014	Standard Class	DK-13375	377.97	2
CA-2014-1000	07/08/2014	07/12/2014	Standard Class	EB-13705	502.488	2
CA-2014-1002	3/14/2014	3/18/2014	Standard Class	NF-18475	91.056	3
CA-2014-1003	1/28/2014	02/03/2014	Standard Class	JC-15340	3.928	1
CA-2014-1003	04/08/2014	4/15/2014	Standard Class	JM-15655	2.368	2
CA-2014-1003	5/25/2014	5/29/2014	Standard Class	BW-11065	14.62	4
CA-2014-1006	4/18/2014	4/22/2014	Standard Class	KM-16720	2.688	1
CA-2014-1007	12/16/2014	12/18/2014	Second Class	LE-16810	99.98	1
CA-2014-1007	11/24/2014	11/29/2014	Standard Class	NG-18355	151.92	2
CA-2014-1008	3/26/2014	3/30/2014	Second Class	CS-12505	18.75	4
CA-2014-1008	10/19/2014	10/24/2014	Standard Class	EH-14125	321.552	3
CA-2014-1008	3/28/2014	04/01/2014	Standard Class	DR-12940	302.376	2
CA-2014-1008	06/02/2014	06/06/2014	Standard Class	SV-20785	8.56	1
CA-2014-1009	10/21/2014	10/26/2014	Standard Class	FH-14275	194.7	3
CA-2014-1009	11/19/2014	11/24/2014	Second Class	DB-13360	166.44	2
CA-2014-1011	12/02/2014	12/04/2014	First Class	MC-17575	2.394	2
CA-2014-1011	12/09/2014	12/14/2014	Standard Class	DM-12955	100.704	3
CA-2014-1012	8/27/2014	8/30/2014	Second Class	MM-17920	13.36	1
CA-2014-1013	12/22/2014	12/26/2014	Standard Class	TW-21025	296.712	1

ShippingDetails.csv

A	B	C	D	E	F
Product ID	ProductNumber	ProductSubCat	Product Name		
1	FUR-BO-10000112		1 Bush Birmingham Collection Bo		
2	FUR-BO-10000330		1 Sauder Camden County Barriste		
3	FUR-BO-10000362		1 Sauder Inglewood Library Book		
4	FUR-BO-10000468		1 O'Sullivan 2-Shelf Heavy-Duty E		
5	FUR-BO-10000711		1 Hon Metal Bookcases, Gray		
6	FUR-BO-10000780		1 O'Sullivan Plantations 2-Door Li		
7	FUR-BO-10001337		1 O'Sullivan Living Dimensions 2-		
8	FUR-BO-10001519		1 O'Sullivan 3-Shelf Heavy-Duty E		
9	FUR-BO-10001567		1 Bush Westfield Collection Book		
10	FUR-BO-10001601		1 Sauder Mission Library with Do		
11	FUR-BO-10001608		1 Hon Metal Bookcases, Black		
12	FUR-BO-10001619		1 O'Sullivan Cherrywood Estates		
13	FUR-BO-10001798		1 Bush Somerset Collection Book		
14	FUR-BO-10001811		1 Atlantic Metals Mobile 5-Shelf I		
15	FUR-BO-10001918		1 Sauder Forest Hills Library with		
16	FUR-BO-10001972		1 O'Sullivan 4-Shelf Bookcase in C		
17	FUR-BO-10002202		1 Atlantic Metals Mobile 2-Shelf I		
18	FUR-BO-10002206		1 Bush Saratoga Collection 5-Shel		
19	FUR-BO-10002213		1 DMI Elcine Executive Suite Bo		

Commodity.csv

A	B	C	D
ProductSubCategory	ProductCategory	Sub-Category	
1	1	Bookcases	
2	1	Chairs	
3	1	Furnishings	
4	1	Tables	
5	2	Appliances	
6	2	Art	
7	2	Binders	
8	2	Envelopes	
9	2	Fasteners	
10	2	Labels	
11	2	Paper	
12	2	Storage	
13	2	Supplies	
14	3	Accessories	
15	3	Copiers	
16	3	Machines	
17	3	Phones	

Sub_Commodity_Catagory.csv

A	B	C	D	E	F	G	H
OrderDetail	Order ID	Product ID	LineTotal	Quantity	Discount	UnitPrice	Profit
1	CA-2016-152156	13	261.96	2	0	130.98	41.9136
2	CA-2016-152156	56	731.94	3	0	243.98	219.582
3	CA-2016-138688	947	14.62	2	0	7.31	6.8714
4	US-2015-108966	320	957.5775	5	0.45	348.21	-383.031
5	US-2015-108966	1317	22.368	2	0.2	13.98	2.5164
6	CA-2014-115812	186	48.86	7	0	6.98	14.1694
7	CA-2014-115812	563	7.28	4	0	1.82	1.9656
8	CA-2014-115812	1762	907.152	6	0.2	188.99	90.7152
9	CA-2014-115812	795	18.504	3	0.2	7.71	5.7825
10	CA-2014-115812	438	114.9	5	0	22.98	34.47
11	CA-2014-115812	329	1706.184	9	0.2	236.97	85.3092
12	CA-2014-115812	1751	911.424	4	0.2	284.82	68.3568
13	CA-2017-114412	1168	15.552	3	0.2	6.48	5.4432
14	CA-2016-161389	782	407.976	3	0.2	169.99	132.5922
15	US-2015-118983	422	68.81	5	0.8	68.81	-123.858
16	US-2015-118983	664	2.544	3	0.8	4.24	-3.816
17	CA-2014-105893	1409	665.88	6	0	110.98	13.3176
18	CA-2014-167164	1295	55.5	2	0	27.75	9.99
19	CA-2014-143336	568	8.56	2	0	4.28	2.4824

OrderDetail.csv

A	B	C	D
RegionID	Name	Country	
1	West	United States	
2	East	United States	
3	Central	United States	
4	South	United States	

A	B	C	D	E
RegionID	Name	Country	Person	
1	West	United State	Anna Andreadi	
2	East	United State	Chuck Magee	
3	Central	United State	Kelly Williams	
4	South	United State	Cassandra Brandow	

CountryRegion.csv

salesRegionRepresentative.csv

A	B	C
ProductCatagor	Category	
1	Furniture	
2	Office Supplies	
3	Technology	

Commodity_Catagory.csv

A	B	C	D	E	F
Customer ID	Country	City	State	Postal Code	Region
2 CG-12520	United States	Henderson	Kentucky	42420	South
3 DV-13045	United States	Los Angeles	California	90036	West
4 SO-20335	United States	Fort Lauderdale	Florida	33311	South
5 BH-11710	United States	Los Angeles	California	90032	West
6 AA-10480	United States	Concord	North Carolina	28027	South
7 IM-15070	United States	Seattle	Washington	98103	West
8 HP-14815	United States	Fort Worth	Texas	76106	Central
9 PK-19075	United States	Madison	Wisconsin	53711	Central
10 AG-10270	United States	West Jordan	Utah	84084	West
11 ZD-21925	United States	San Francisco	California	94109	West
12 KB-16585	United States	Fremont	Nebraska	68025	Central
13 SF-20065	United States	Philadelphia	Pennsylvania	19140	East
14 EB-13870	United States	Orem	Utah	84057	West
15 EH-13945	United States	Los Angeles	California	90049	West
16 TB-21520	United States	Philadelphia	Pennsylvania	19140	East
17 MA-17560	United States	Houston	Texas	77095	Central
18 GH-14485	United States	Richardson	Texas	75080	Central
19 SN-20710	United States	Houston	Texas	77041	Central
20 LC-16930	United States	Naperville	Illinois	60540	Central
21 RA-19885	United States	Los Angeles	California	90049	West

ConsumerLoationDetails.csv

I have loaded Commodity.csv, Commodity_Catagory.csv, ConsumerDetails.csv, CountryRegion.csv, OrderDetail.csv, ShippingDetails.csv and Sub_Commodity_Catagory.csv files to the DB called [IT17167710_DataSource](#)

Then I Have loaded salesRegionRepresentative.csv to another DB called [IT17167710_DataSource2](#)

IT17167710_DataSource

- IT17167710_DataSource
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Commodity
 - dbo.Commodity_Catagory
 - dbo.ConsumerDetails
 - dbo.CountryRegion
 - dbo.OrderDetail
 - dbo.ShippingDetails
 - dbo.Sub_Commodity_Catagory
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security

IT17167710_DataSource2

- IT17167710_DataSource2
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.salesRegionRepresentative
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security

I have Converted ConsumerLoationDetails.csv to Text file so that I can Use Flat file Source in SSIS tool, to extract ConsumerLoationDetails.txt data.

	A	B	C	D	E	F
1	Customer ID	Country	City	State	Postal Code	Region
2	CG-12520	United States	Henderson	Kentucky	42420	South
3	DV-13045	United States	Los Angeles	California	90036	West
4	SO-2035	United States	Fort Lauderdale	Florida	33311	South
5	BH-11710	United States	Los Angeles	California	90032	West
6	AA-10480	United States	Concord	North Carolina	28027	South
7	IM-15070	United States	Seattle	Washington	98103	West
8	HP-14815	United States	Fort Worth	Texas	76106	Central
9	PK-19075	United States	Madison	Wisconsin	53711	Central
10	AG-10270	United States	West Jordan	Utah	84084	West
11	ZD-21925	United States	San Francisco	California	94109	West
12	KB-16585	United States	Fremont	Nebraska	68025	Central
13	SF-20065	United States	Philadelphia	Pennsylvania	19140	East
14	EB-13870	United States	Orem	Utah	84057	West
15	EH-13945	United States	Los Angeles	California	90049	West
16	TB-21520	United States	Philadelphia	Pennsylvania	19140	East
17	MA-17560	United States	Houston	Texas	77095	Central
18	GH-14485	United States	Richardson	Texas	75080	Central
19	SN-20710	United States	Houston	Texas	77041	Central
20	LC-16930	United States	Naperville	Illinois	60540	Central
21	RA-19885	United States	Los Angeles	California	90049	West

Customer location Details

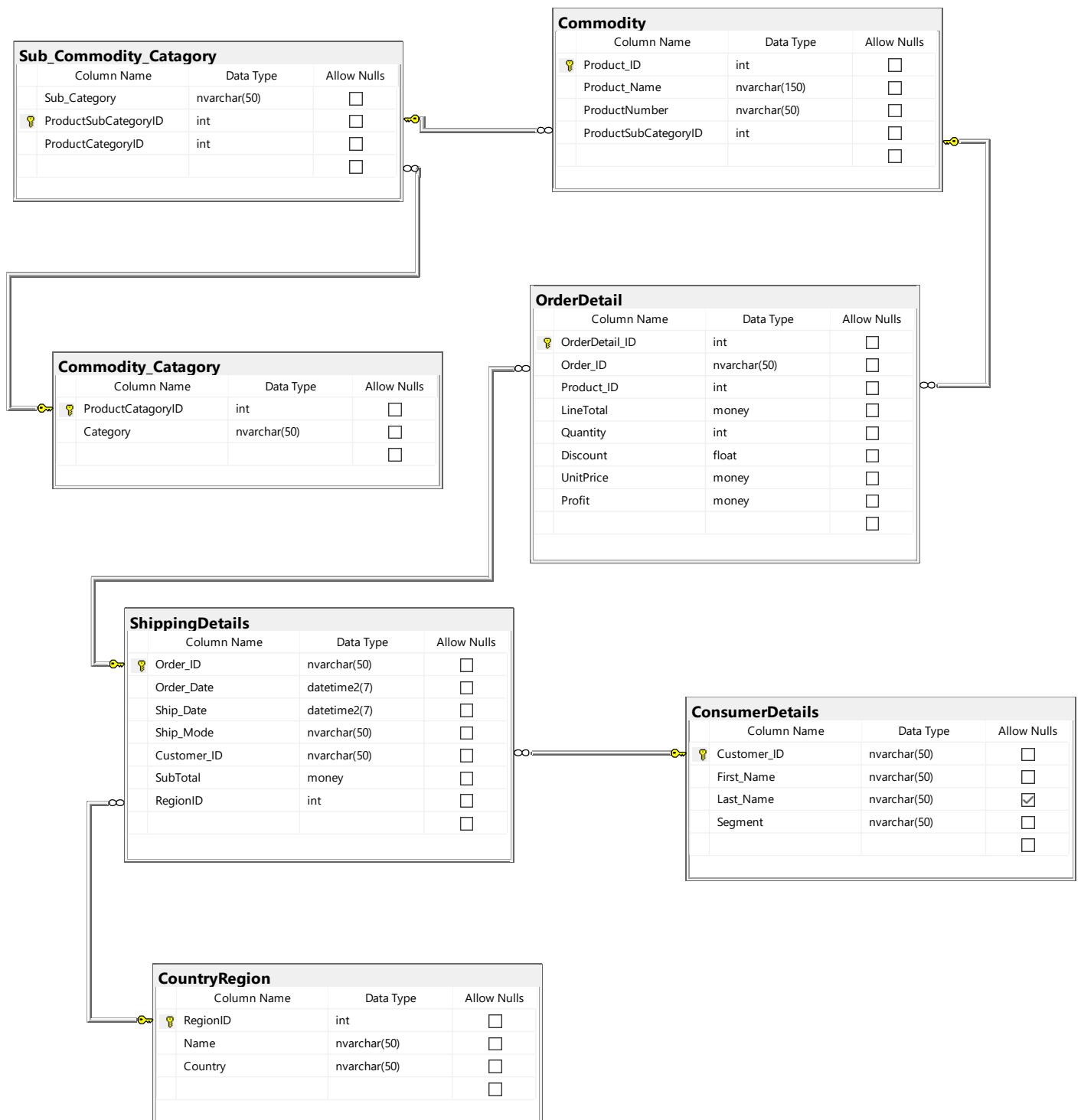
ConsumerLoationDetails.csv



ConsumerLocationDetails - Notepad						
Customer ID	Country	City	State	Postal Code	Region	
CG-12520	United States	Henderson	Kentucky	42420	South	
DV-13045	United States	Los Angeles	California	90036	West	
SO-2035	United States	Fort Lauderdale	Florida	33311	South	
BH-11710	United States	Seattle	Washington	98103	West	
AA-10480	United States	Fort Worth	Texas	76106	Central	
IM-15070	United States	Madison	Wisconsin	53711	Central	
HP-14815	United States	West Jordan	Utah	84057	West	
PK-19075	United States	San Francisco	California	94109	West	
AG-10270	United States	Fremont	Nebraska	68025	Central	
ZD-21520	United States	Philadelphia	Pennsylvania	19140	East	
KB-16585	United States	Orem	Utah	84057	West	
SF-20065	United States	Houston	Texas	77095	Central	
EB-13870	United States	Naperville	Illinois	60540	Central	
EH-13945	United States	Philadelphia	Pennsylvania	19140	East	
TB-21520	United States	Houston	Texas	77095	Central	
MA-17560	United States	Richardson	Texas	75080	Central	
GH-14485	United States	Houston	Texas	77041	Central	
SN-20710	United States	Naperville	Illinois	60540	Central	
LC-16930	United States	Philadelphia	Pennsylvania	19140	East	
RA-19885	United States	Houston	Texas	77095	Central	
ES-14080	United States	Naperville	Illinois	60540	Central	
ON-18715	United States	Philadelphia	Pennsylvania	19140	East	
PO-18865	United States	Houston	Texas	77041	Central	
LH-16900	United States	Naperville	Illinois	60540	Central	
DP-13000	United States	Philadelphia	Pennsylvania	19140	East	
JM-15265	United States	Houston	Texas	77095	Central	
TB-21055	United States	Naperville	Illinois	60540	Central	
KM-16720	United States	Philadelphia	Pennsylvania	19140	East	
PS-18970	United States	Houston	Texas	77041	Central	
BS-11590	United States	Naperville	Illinois	60540	Central	
KD-16270	United States	Philadelphia	Pennsylvania	19140	East	
HM-14980	United States	Houston	Texas	77041	Central	
JE-15745	United States	Naperville	Illinois	60540	Central	
KB-16600	United States	Philadelphia	Pennsylvania	19140	East	
SC-20770	United States	Houston	Texas	77041	Central	

ConsumerLoationDetails.txt

To Describe the sources, I have use Database Diagrams.



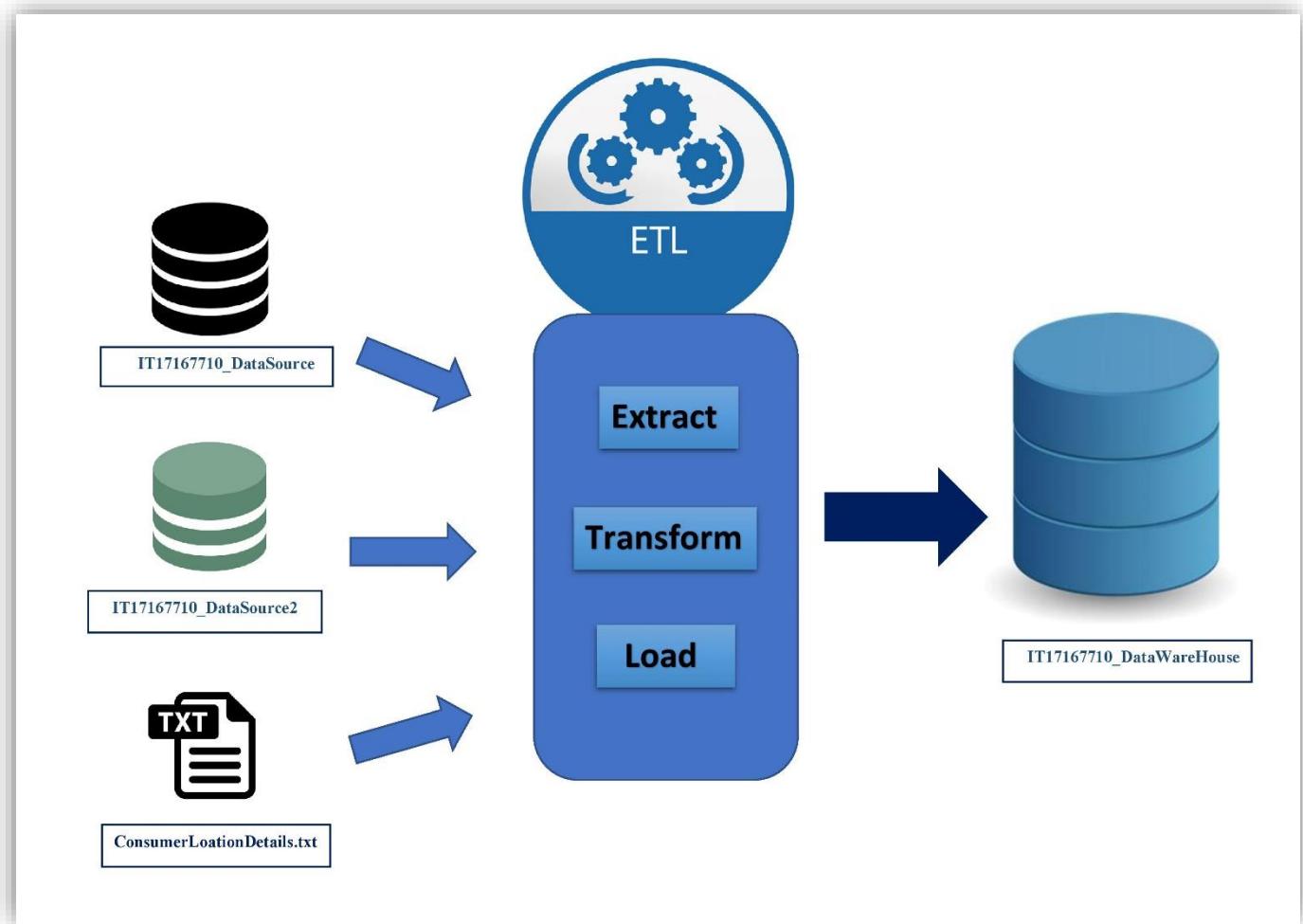
IT17167710_DataSource

salesRegionRepresentative			
	Column Name	Data Type	Allow Nulls
!	RegionID	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
	Country	nvarchar(50)	<input type="checkbox"/>
	Person	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

IT17167710_DataSource2

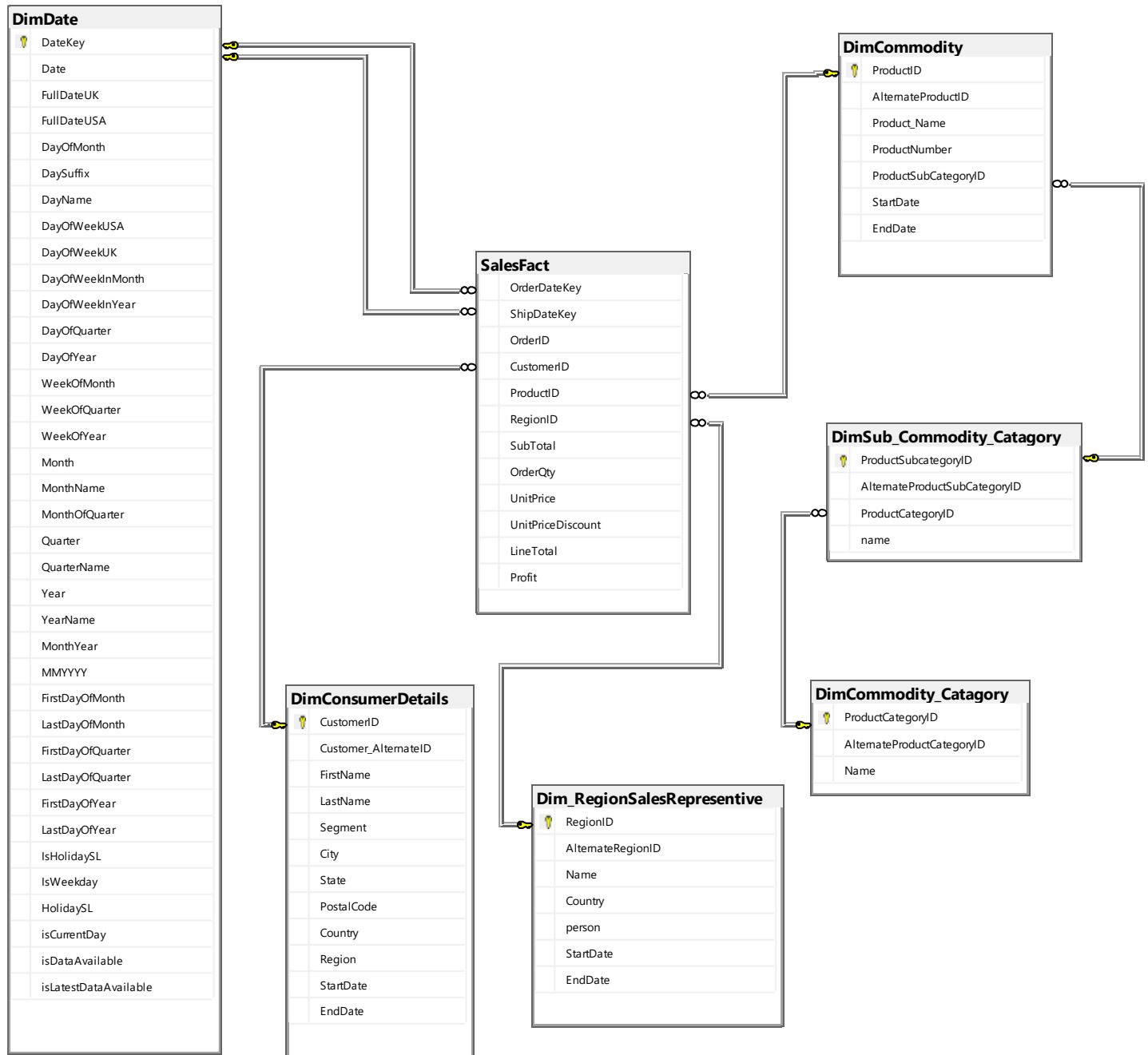
Step 3: Solution Architecture

architectural diagram



The data warehouse is the core of the BI system. A data warehouse is a database built for the purpose of data analysis and reporting. This purpose changes the design of this database as well.

Step 4: Data warehouse design & development



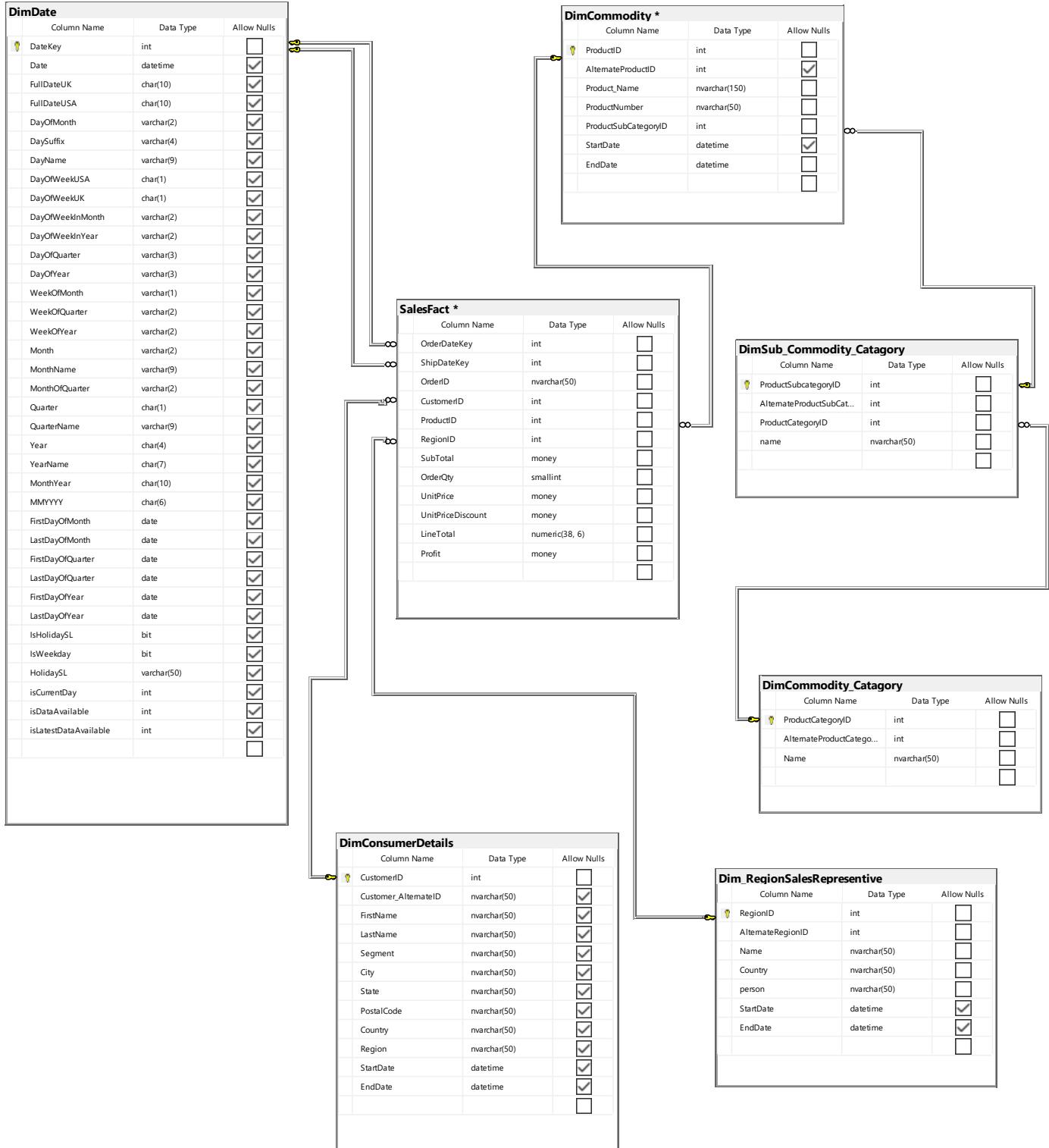
Assumptions

I have taken Dim Consumer Details as slowly changing dimension, Consumer addresses can change time to time, and we need to keep track of their historical address.

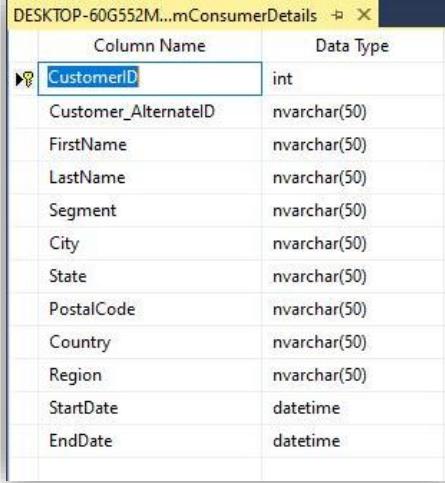
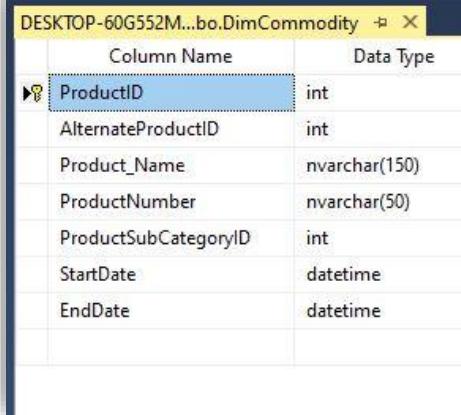
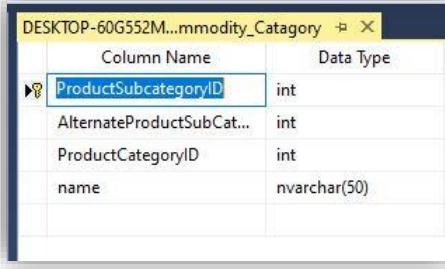
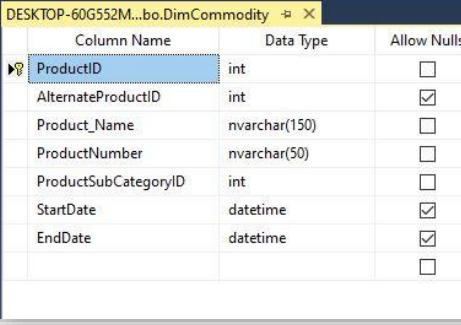
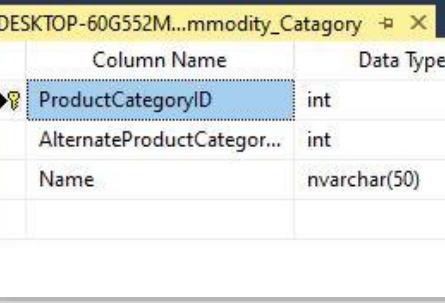
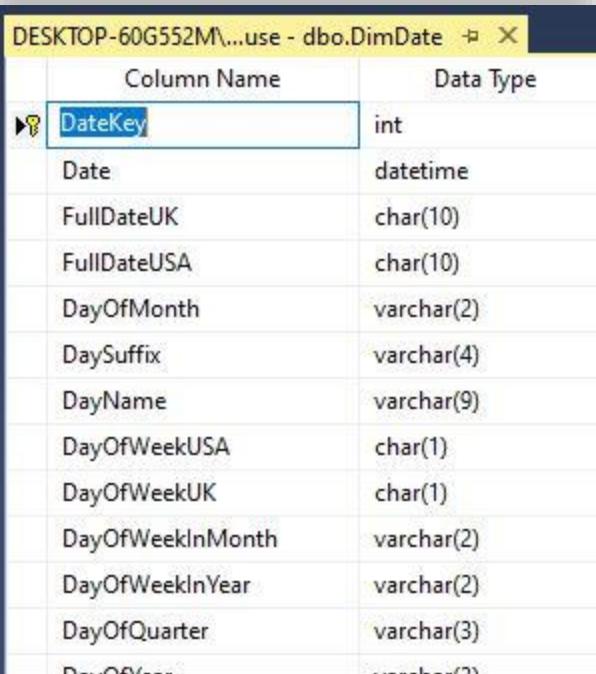
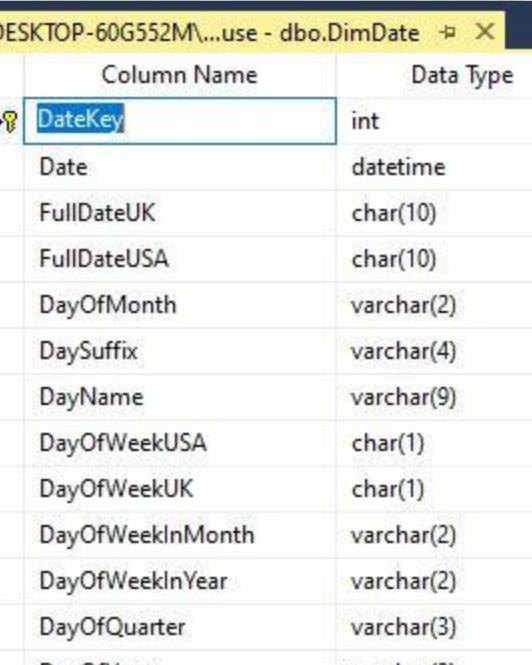
In the data set product name is too large So assume that if Store owner wants to change the product name and we need to keep track of their historical name so that I have taken Dim Commodity as slowly changing dimension,

The reason for taking Dim Region Sales Representative as slowly changing dimension, is Sales Representative's state can be change time to time, and we should have to keep track of their historical Representative record. Additionally, if the Region Sales Representative changes, we should replace the old Representative with the new Representative.

Data Warehouse Data Type



Before creating the sales fact table & Other dimensions, start by creating the Date Dimension.
For that I use the code in the file ‘DateMaster.sql’ file.

Once done, press CTRL + S to save the table. In the popup, provide the table name.

For dimensions, name the table using ‘Dim<TableName>’ format

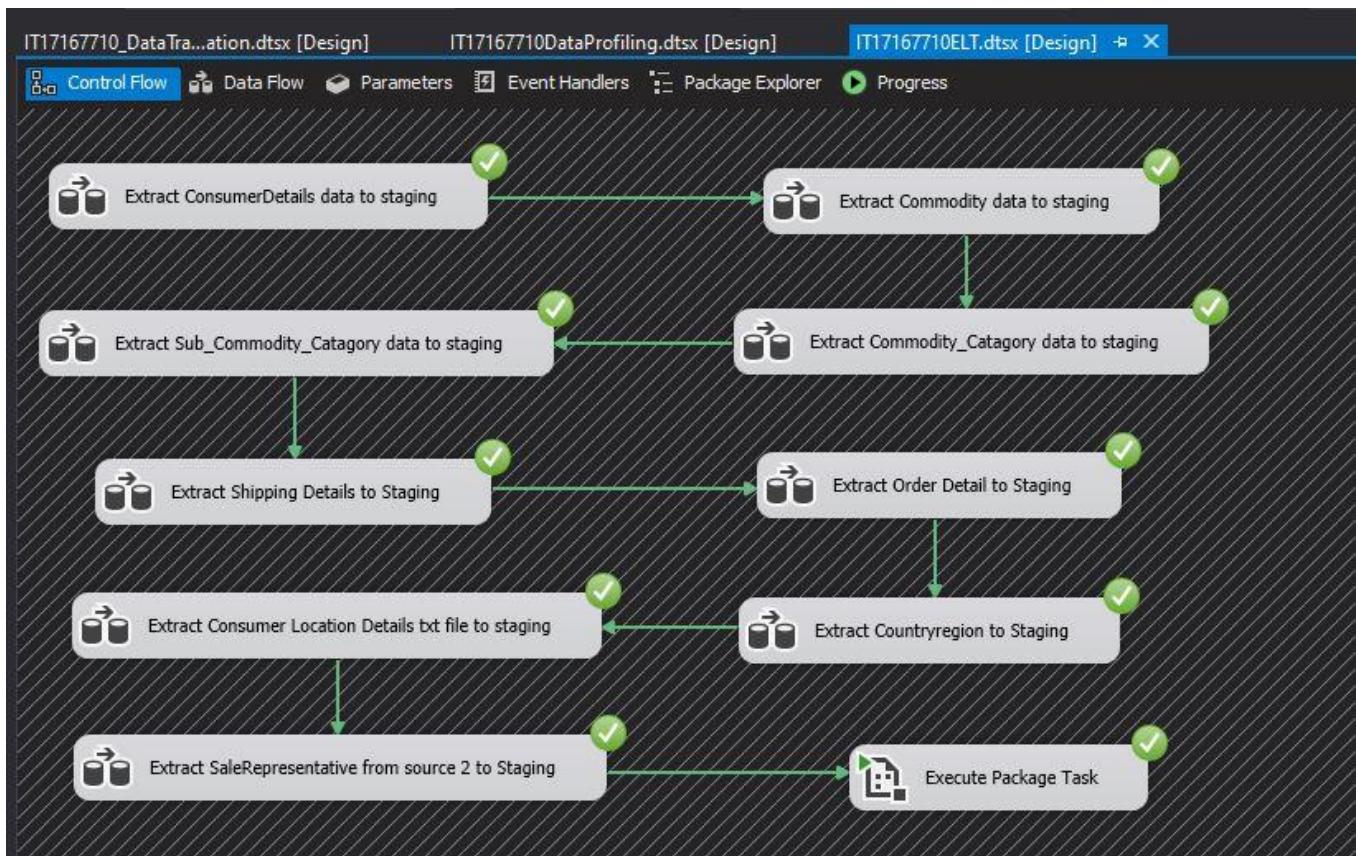
For Facts, name the table using ‘<TableName>Fact’ format

Scope	<p>1. <u>Data Completeness Testing</u></p> <p>Test whether all expected source data is loaded into target Dimension tables by Comparing record counts between source, staging and DW Dimension tables</p> <p>2. <u>Data Duplication testing</u></p> <p>Ensure that there are no duplicate values exist in DW Dimension tables</p> <p>3. <u>Metadata Testing</u></p> <p>i. Data Type Testing</p> <p>Verify that the table and column data type definitions are same in the Source, Staging And DW.</p> <p>ii. Data Lengths Testing</p> <p>Verify that the data lengths of Source, Staging and DW are same</p> <p>iii. Index / Constraint Testing</p> <p>Verify that proper constraints defined on the Source tables and DW Dimensions as per the Data Warehouse Design (Snowflakes Schema).</p> <p>4. <u>Incremental ETL testing</u></p> <p>Check Whether Slowly Changing Dimensions Work as Expected</p> <ul style="list-style-type: none"> - Are the old records end dated appropriately? - Are the New records start dated appropriately?
Out of Scope	<ul style="list-style-type: none"> - ETL Regression Testing - Not consider about Transfer Speed - Date Validation
Assumptions	<ul style="list-style-type: none"> - The ETL will be Tested by using SQL queries, if the output shows correctly as Expected, it will be assumed that the ETL is working properly. - There is no environment downtime during test due to outages or defect fixes.

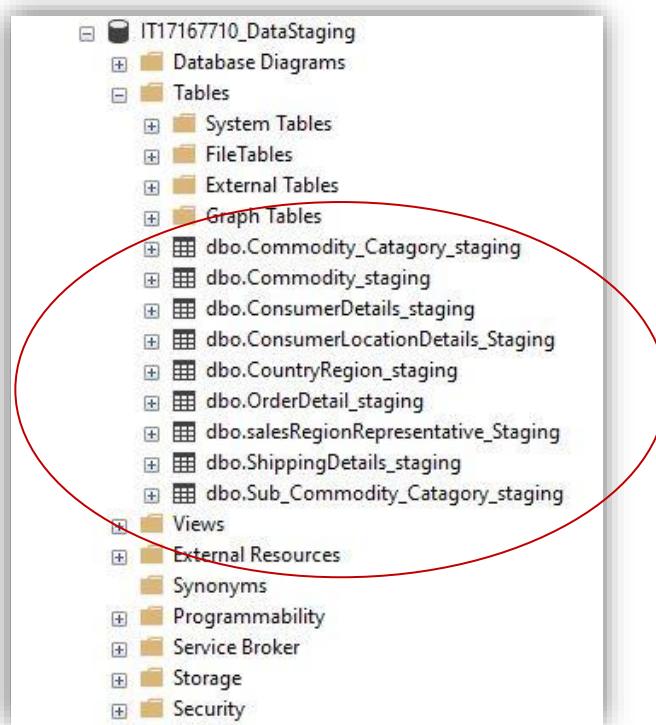
Schedules:	<ul style="list-style-type: none"> - One testing Cycle Start date: 27/04/2020 End date: 29/04/2020 - Cycle 2 will be Started if there are more failures found in Cycle 1. <table border="1" data-bbox="541 480 1283 705"> <thead> <tr> <th data-bbox="541 480 917 518">Task</th><th data-bbox="917 480 1283 518">Estimate effort</th></tr> </thead> <tbody> <tr> <td data-bbox="541 518 917 557">Create the Test plan</td><td data-bbox="917 518 1283 557">3 man-hours</td></tr> <tr> <td data-bbox="541 557 917 595">Create Test Cases</td><td data-bbox="917 557 1283 595">4 man-hours</td></tr> <tr> <td data-bbox="541 595 917 633">Perform Test Execution</td><td data-bbox="917 595 1283 633">3 man-hours</td></tr> <tr> <td data-bbox="541 633 917 671">Test Summary</td><td data-bbox="917 633 1283 671">2 man-hours</td></tr> <tr> <td data-bbox="541 671 917 705">Total</td><td data-bbox="917 671 1283 705">12 man-hours</td></tr> </tbody> </table>	Task	Estimate effort	Create the Test plan	3 man-hours	Create Test Cases	4 man-hours	Perform Test Execution	3 man-hours	Test Summary	2 man-hours	Total	12 man-hours
Task	Estimate effort												
Create the Test plan	3 man-hours												
Create Test Cases	4 man-hours												
Perform Test Execution	3 man-hours												
Test Summary	2 man-hours												
Total	12 man-hours												
Roles and Responsibilities	<p>Person: D.M.J Prathapa</p> <p>Duties: create test plane / Create test cases / execute test cases / Create test summary</p>												
Test Deliverables	<ul style="list-style-type: none"> - Test Plan - Test Cases - Test Summary <p>Deadline: 30/04/2020</p>												
Test Environment	<ul style="list-style-type: none"> - Database Server: SQL Server Management Studio v18.4 Version 15.0.18296.0 - Operating system: Windows 10 												
Test Tools	<ul style="list-style-type: none"> - Microsoft SQL Server Data Tools for Visual Studio 2017(SSDT) Version 15.9.21 												
Exit Criteria	Execution of all test cases.												
Terms/Acronyms	<ul style="list-style-type: none"> - ETL = (extract, transform, load) - DW = Data warehouse 												

The SQL script was uploaded to the drive in the zip file.

Step 6: ETL development



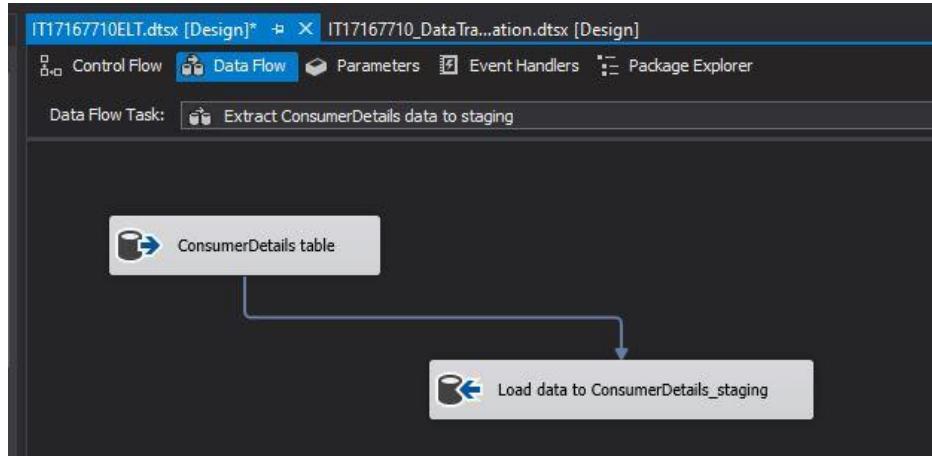
First using the SQL Server Integration Services Software I have extracted all the data from the tables which were in the [IT17167710_DataSource](#), [IT17167710_DataSource2](#) and Extract [ConsumerLoationDetails.txt](#) to separate staging DB called [IT17167710_DataStaging](#) as shown in the below .



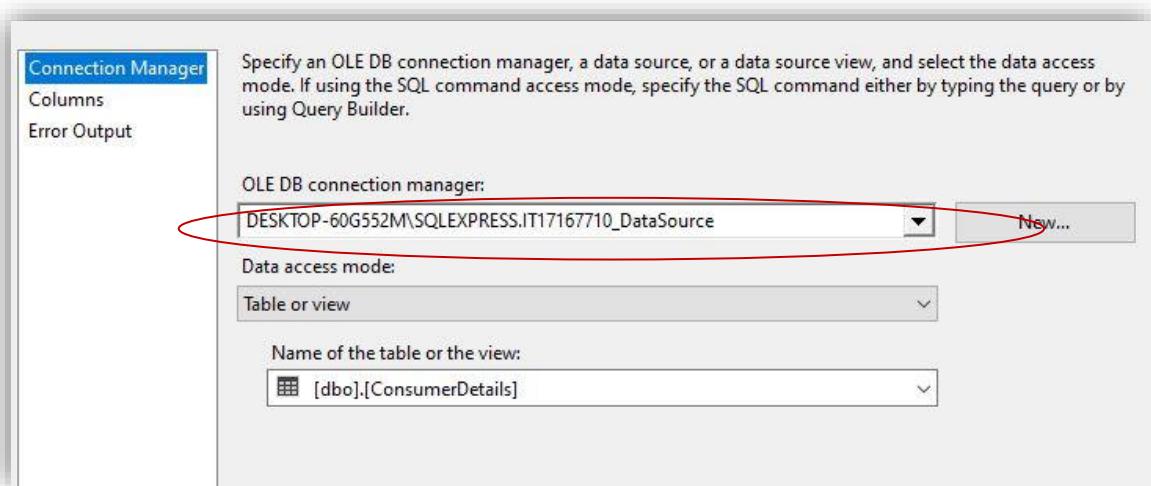
IT17167710_DataStaging

Extract Consumer Details

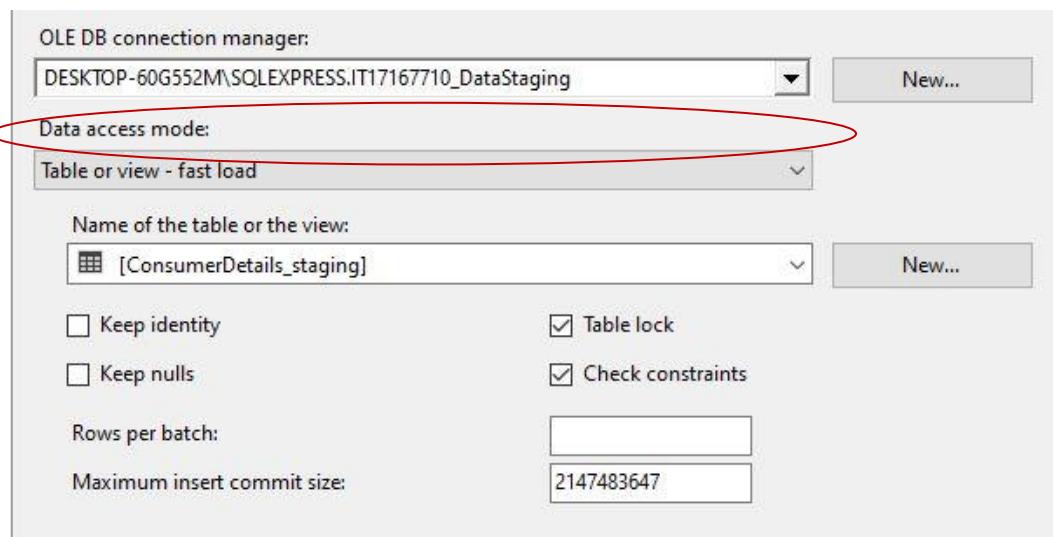
(Data flow) 1.1



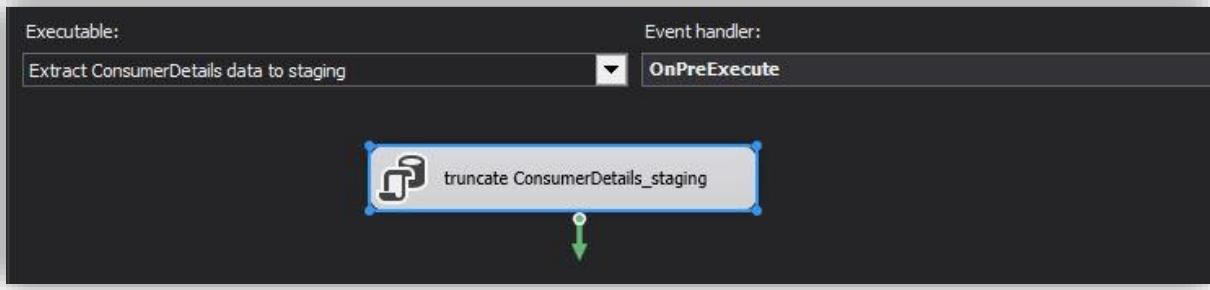
Used OLE DB Source as dbo.ConsumerDetails table in [IT17167710_DataSource](#)



OLE DB Destination for create new table as ConsumerDetails _staging in the [IT17167710_DataStaging](#) database.



(Event Handlers) 1.2



Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.ConsumerDetails _staging in [IT17167710_DataStaging Database](#)

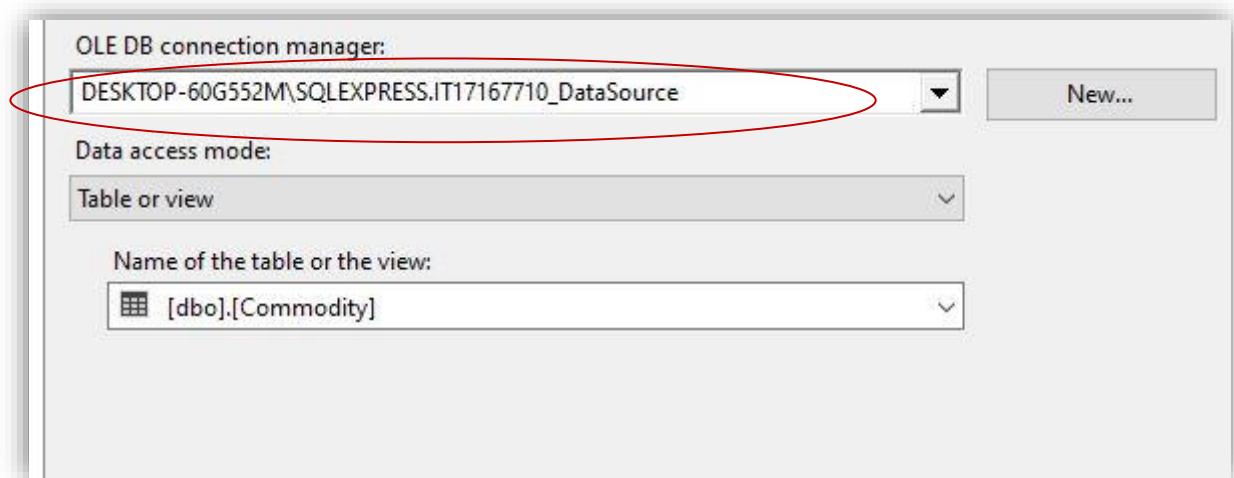
PROPERTY	VALUE
ConnectionType	OLE DB
Connection	DESKTOP-60G552M\SQLEXPRESS.IT17167710_DataStaging
SQLSourceType	Direct input
SQLStatement	<code>truncate table dbo.ConsumerDetails_staging</code>
IsQueryStoredProcedure	False
BypassPrepare	True
Name	

Extract Commodity data to staging

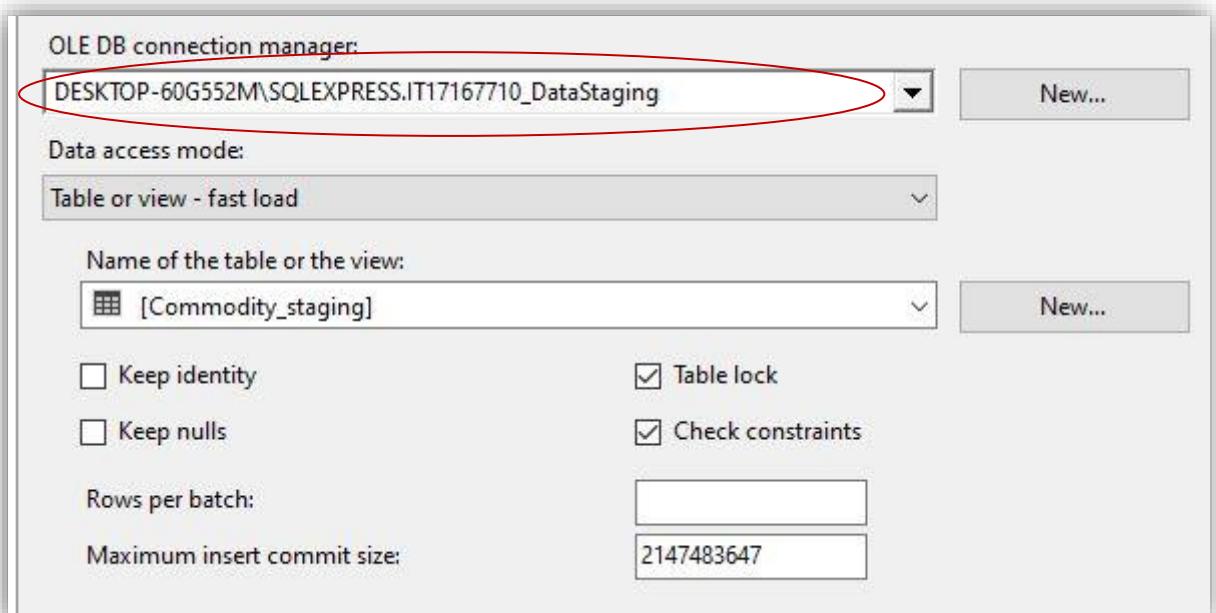
(Data flow)



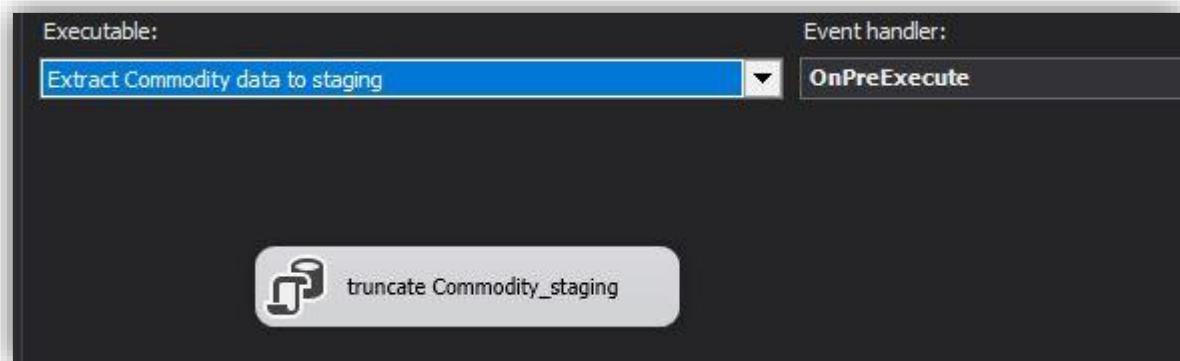
Used OLE DB Source as dbo.Commodity table in [IT17167710_DataSource](#)



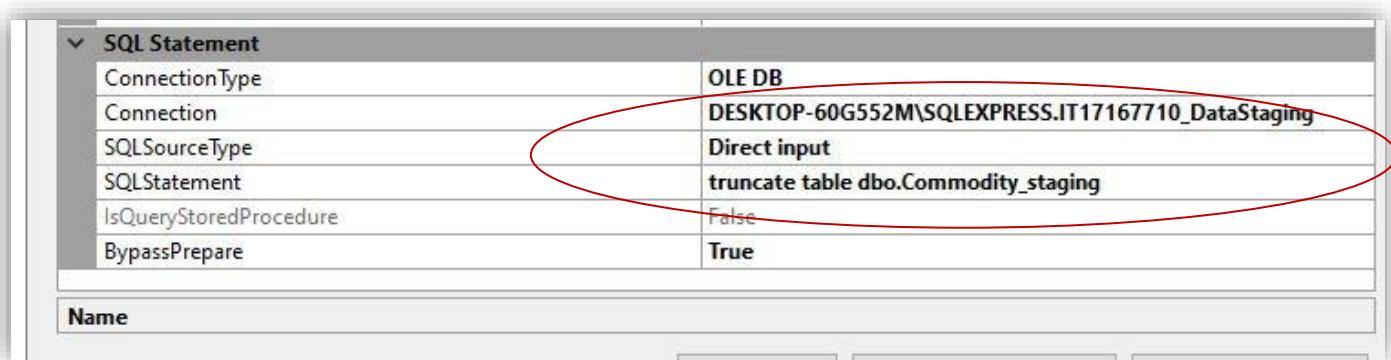
OLE DB Destination for create new table as dbo.Commodity_staging in the [IT17167710_DataStaging](#) database.



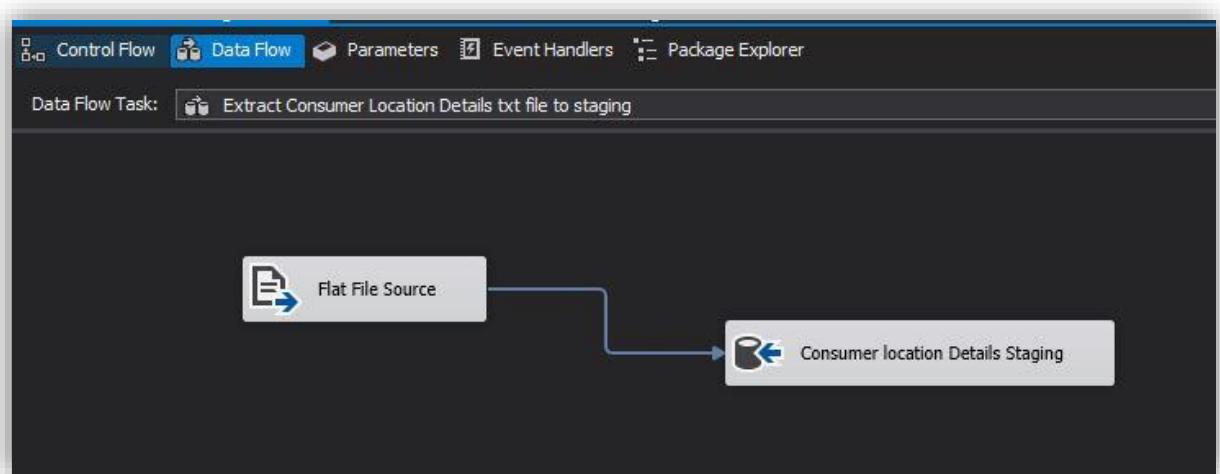
(Event Handlers) 1.2



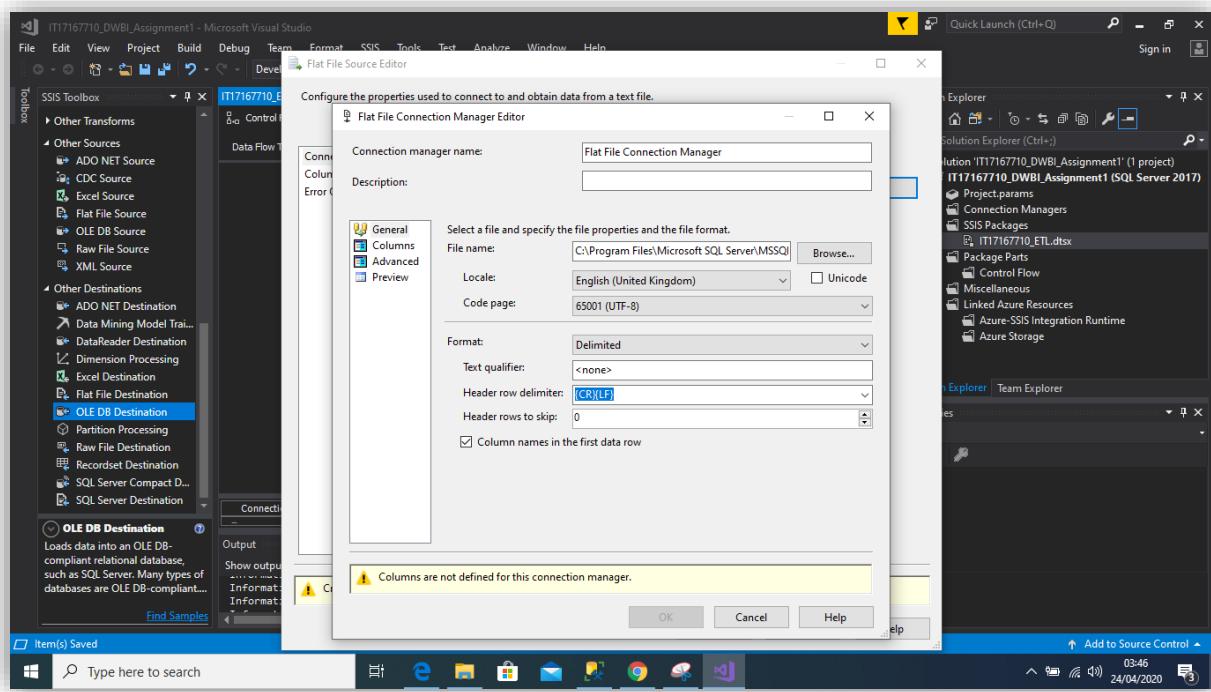
Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table
dbo.Commodity_staging in IT17167710_DataStaging Database



Customer Address Details Extraction (Data flow)

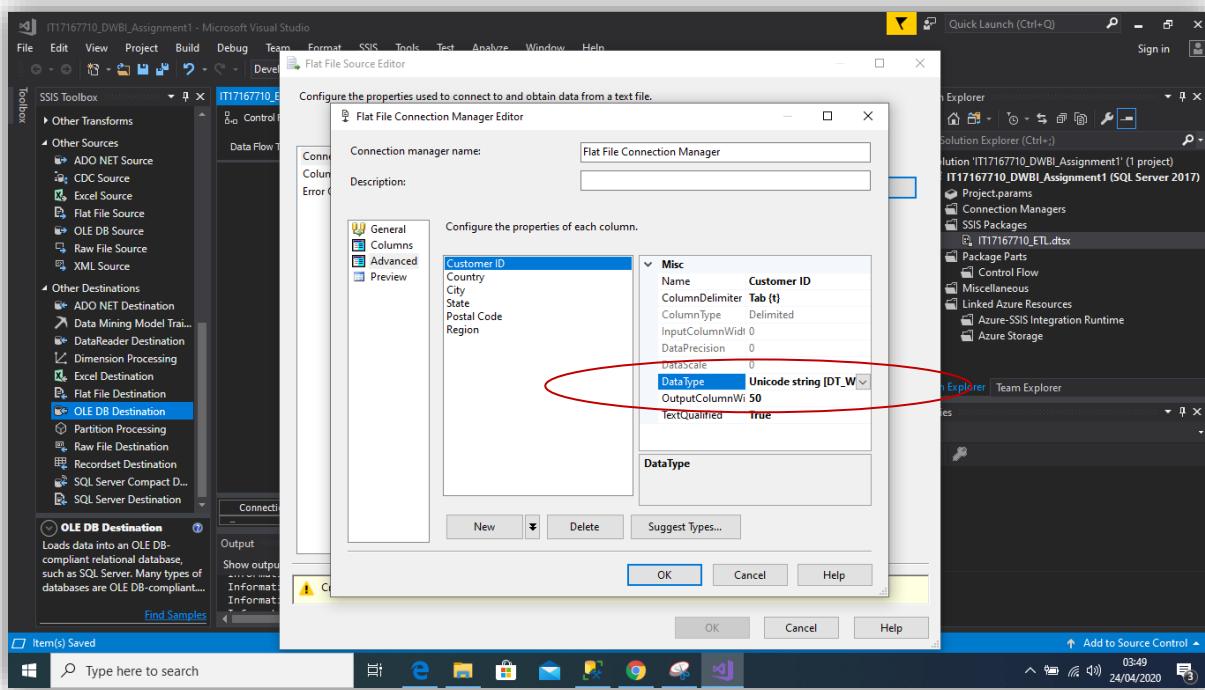


Used Flat file Source SSIS tool, to extract CustomerLocationDetail.txt data .

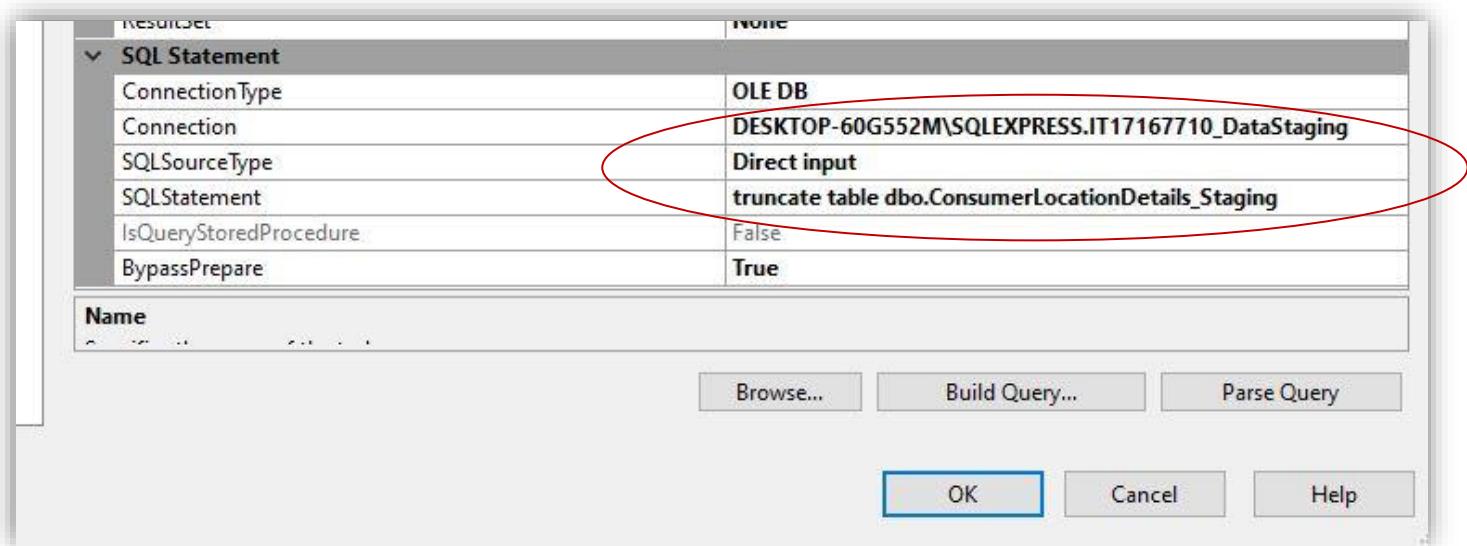


Used OLE DB Destination SSIS tool to create new table as
ConsumerLocationDetails_Staging load text file data to IT17167710_DataStaging Database

In Advanced tab, text file data has columns then every column data type select as Unicode string[DT_WSTR]



Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table ConsumerLocationDetails_Staging in [IT17167710_DataStaging](#)

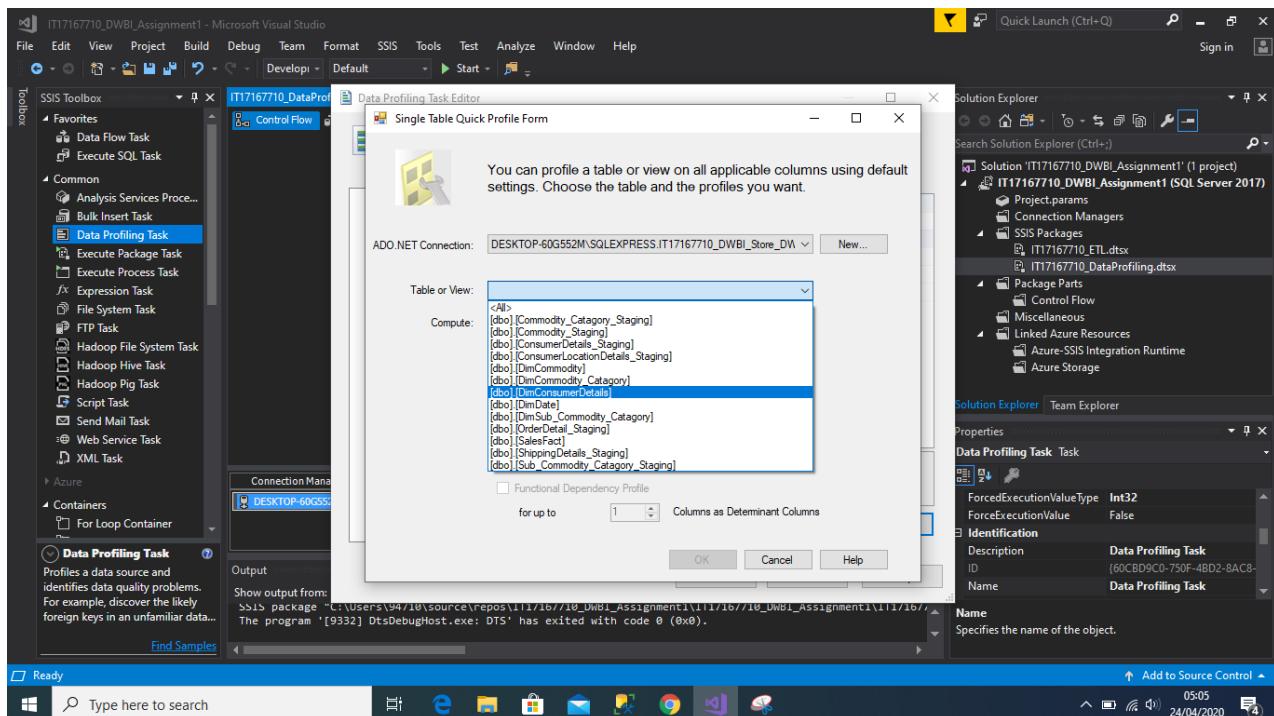


followed same process to extract other source table data to Staging

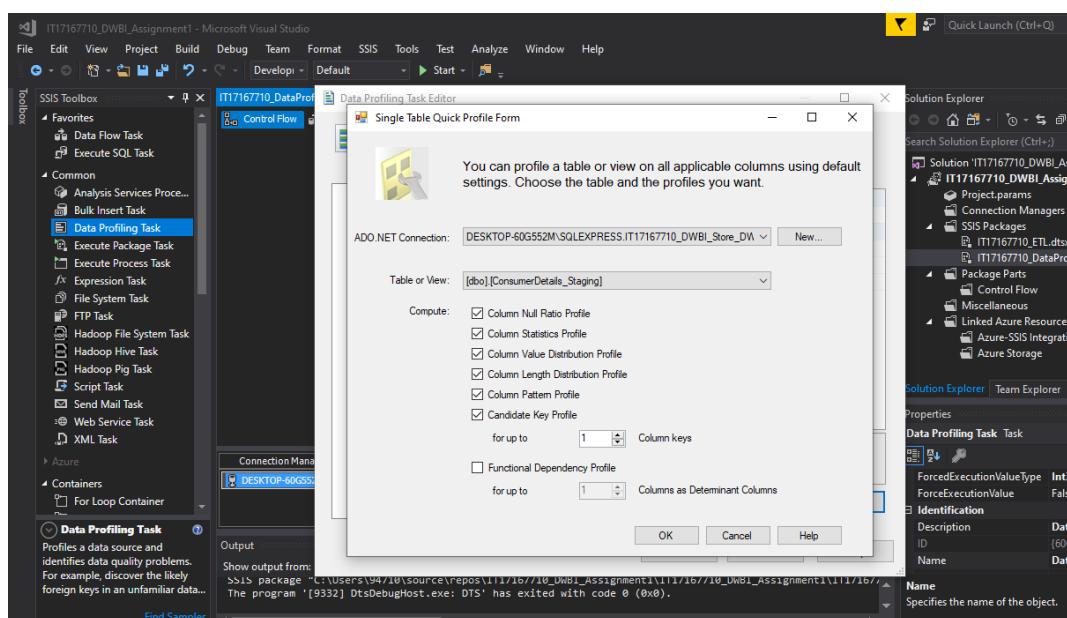
Data Profiling

I used the staging table data to analyse how the data looks like to determine what type of transformations I need to perform on the data.

- Right click on SSIS Packages and New SSIS Package selected.
- In the Control Flow of Data Profiling Pack.dtsx, drag and drop the Data Profiling Task and double click to open the configuration.
- Clicked on Quick Profile button to open up Single Table Quick Profile Form.
- Clicked on New button and create the connection to **IT17167710_DataStaging**
- From the Table or View dropdown, ConsumerDetails _staging table selected.



- Selected the all check boxes and click on OK button to complete the configuration.



- Save the package and Run the Data Profiling Pack to profile the ConsumerDetails
- Once the green tick appeared, double click the Data Profiling Task and click on Open Profile Viewer to view the analysed data.

Data Profile Viewer - Profiles (Table View)

Open Refresh

Profiles (Table View)

Data Sources
DESKTOP-60G552M\SQLEXPRESS
Databases
IT17167710_DWBI_Store_DW
Tables
[dbo].[ConsumerDetails_Staging]
Candidate Key Profiles
Column Length Distribution Profiles
Column Null Ratio Profiles
Column Pattern Profiles
Column Value Distribution Profiles
Functional Dependency Profiles

Candidate Key Profiles - [dbo].[ConsumerDetails_Staging]

Key Columns	Key Strength
Customer_ID	100.0000 %
Customer_ID	100.0000 %

Data Profile Viewer - Profiles (Table View)

Open Refresh

Profiles (Table View)

Data Sources
DESKTOP-60G552M\SQLEXPRESS
Databases
IT17167710_DWBI_Store_DW
Tables
[dbo].[ConsumerDetails_Staging]
Candidate Key Profiles
Column Length Distribution Profiles
Column Null Ratio Profiles
Column Pattern Profiles
Column Value Distribution Profiles
Functional Dependency Profiles

Column Null Ratio Profiles - [dbo].[ConsumerDetails_Staging]

Column	Null Count	Null Percentage
Customer_ID	0	0.0000 %
Customer_ID	0	0.0000 %
First_Name	0	0.0000 %
First_Name	0	0.0000 %
Last_Name	1	0.1261 %
Last_Name	1	0.1261 %
Segment	0	0.0000 %
Segment	0	0.0000 %

Data Profile Viewer - Profiles (Table View)

Open Refresh

Profiles (Table View)

Data Sources
DESKTOP-60G552M\SQLEXPRESS
Databases
IT17167710_DWBI_Store_DW
Tables
[dbo].[ConsumerDetails_Staging]
Candidate Key Profiles
Column Length Distribution Profiles
Column Null Ratio Profiles
Column Pattern Profiles
Column Value Distribution Profiles
Functional Dependency Profiles

Functional Dependency Profiles - [dbo].[ConsumerDetails_Staging]

Determinant Columns	Dependent Column	Functional Dependency Strength
Customer_ID	Segment	100.0000 %
Customer_ID	Last_Name	100.0000 %
Customer_ID	First_Name	100.0000 %
Customer_ID	Segment	100.0000 %
Customer_ID	Last_Name	100.0000 %
Customer_ID	First_Name	100.0000 %

Functional Dependency Violations

Successfully loaded data profile from ...

Data Profile Viewer - Profiles (Table View)

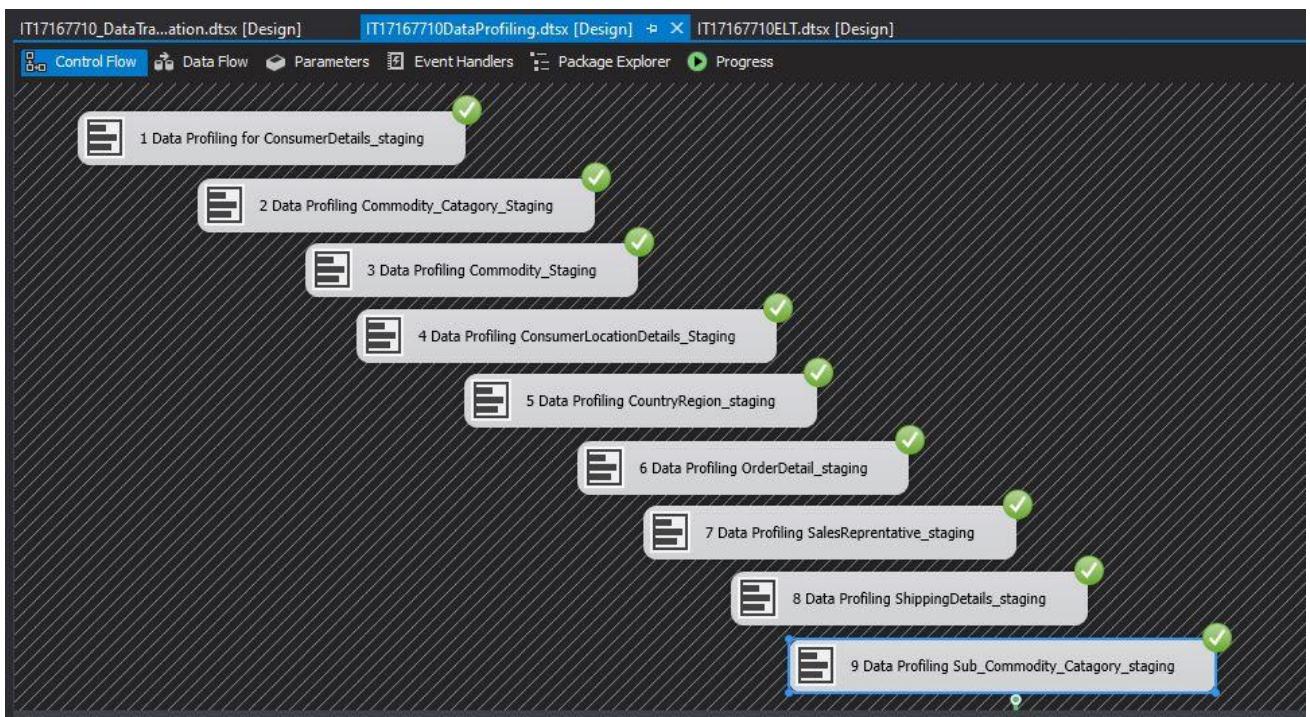
Open Refresh

Profiles (Table View)

Data Sources
DESKTOP-60G552M\SQLEXPRESS
Databases
IT17167710_DWBI_Store_DW
Tables
[dbo].[ConsumerDetails_Staging]
Candidate Key Profiles
Column Length Distribution Profiles
Column Null Ratio Profiles
Column Pattern Profiles
Column Value Distribution Profiles
Functional Dependency Profiles

Column Length Distribution Profiles - [dbo].[ConsumerDetails_Staging]

Column	Minimum Length	Maximum Length	Ignore Leading Spaces	Ignore Trailing Spaces
Customer_ID	8	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Customer_ID	8	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>
First_Name	2	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>
First_Name	2	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Last_Name	3	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Last_Name	3	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Segment	8	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Segment	8	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>

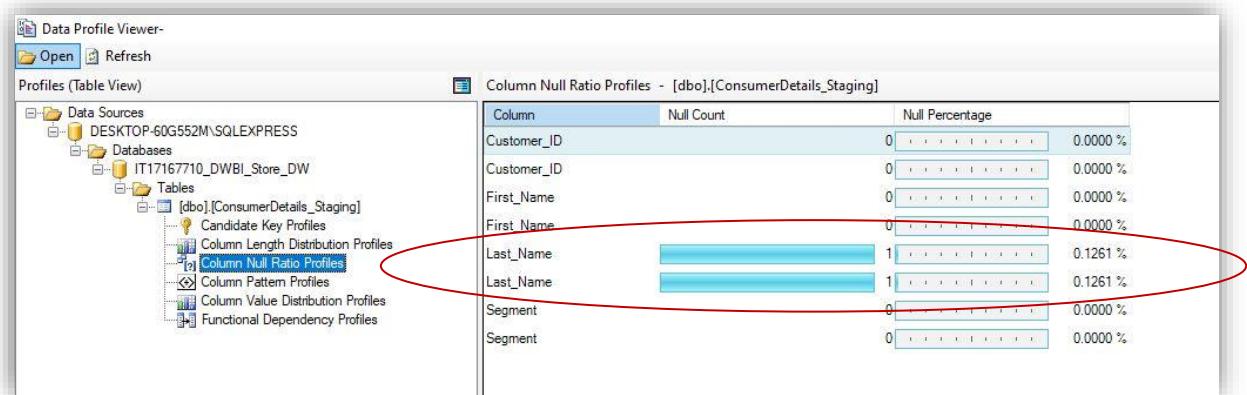


Data Transformation

Transform and Load Consumer Details

I created Consumer Data Transformation by below mentioned steps

- Created new package called IT17167710_DataTransformation.dtsx.
- Then Dragged and dropped a Data Flow Task, renamed it as Transform and Load Consumer Details and go the Data Flow tab.
- Dragged and dropped OLE DB Source, renamed as Extract from Consumer Details Staging and configure it to access the Consumer Details Staging table.
- And I used another OLE DB Source, renamed as Extract from Consumer location staging and configure it to access the ConsumerLocationDetails_Staging table. And selected all the columns.
- Then I Dragged and dropped two Sort items and connect each OLE DB Source to them.
- After that I Double click Sort that is connected to Extract from Consumer Details Staging and select Customer_ID as the sort option by ticking on the checkbox in front of Customer_ID. Then I did the same for the other Sort item connected to Consumer Location Details Staging.
- Dragged and dropped Merge Join and link the Sort item connected to Extract from Consumer location staging table using.
- In the Input Output Selection popup, I have selected Merge Join Left Input.
- Then I Dragged and dropped Derived Column item and connect the Merge Join item to Derived Column because I have found in the data profiling part Consumer's last name Contain one null value.



- Then I Replace the Replace null by using the Derived Column

Derived Column Name	Derived Column	Expression	Data Type
Last_Name	Replace 'Last_Name'	REPLACENULL(Last_Name, "NA")	Unicode string [DT_WS...]

- After that I dragged and dropped Slowly Changing Dimension item and connect the last Derived Column to that.
- In the SCD Configuration Wizard I set the configurations as below

Select a Dimension Table and Keys

Select a dimension table to load and map columns in the transformation input to columns in the dimension table.

Connection manager: DESKTOP-60G552M\SQLEXPRESS.IT17167710_DataWareHouse

Table or view: [dbo].[DimConsumerDetails]

Input Columns	Dimension Columns	Key Type
City	City	Not a key column
Country	Country	Not a key column
Customer_ID	Customer_Altimate...	Business key
	EndDate	
First_Name	FirstName	Not a key column
Last_Name	LastName	Not a key column
Postal Code	PostalCode	Not a key column
Region	Region	Not a key column
Segment	Segment	Not a key column
	StartDate	
State	State	Not a key column

Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

Dimension Columns	Change Type
City	Historical att...
Country	Historical att...
PostalCode	Historical att...
Region	Historical att...
Segment	Historical att...
State	Historical att...

Fixed Attribute

Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute

Select this type when changed values should overwrite existing values. This is a Type 1 change.

Historical Attribute

Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

SCD Configuration Wizard

Use start and end dates to identify current and expired records

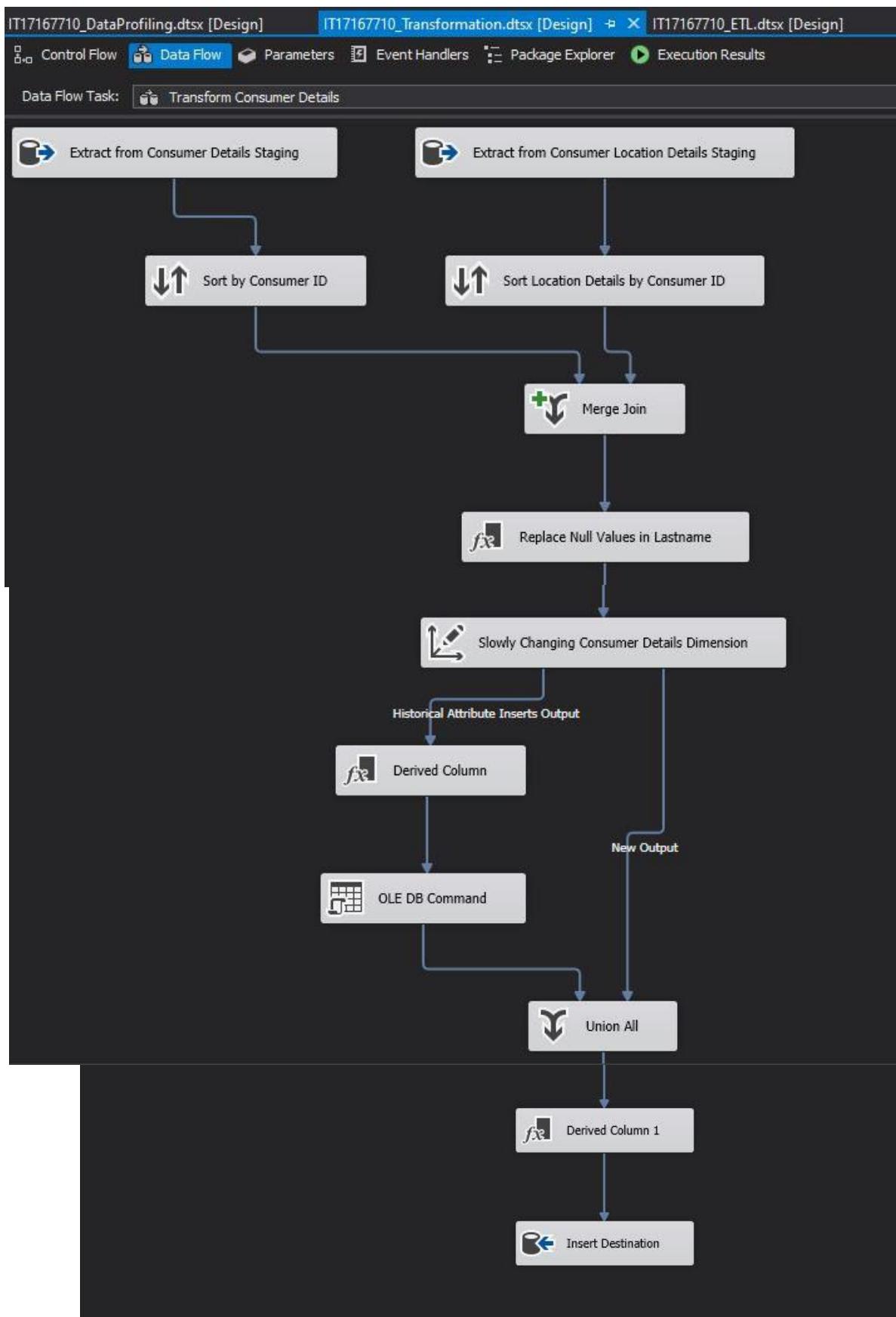
Start date column: StartDate

End date column: EndDate

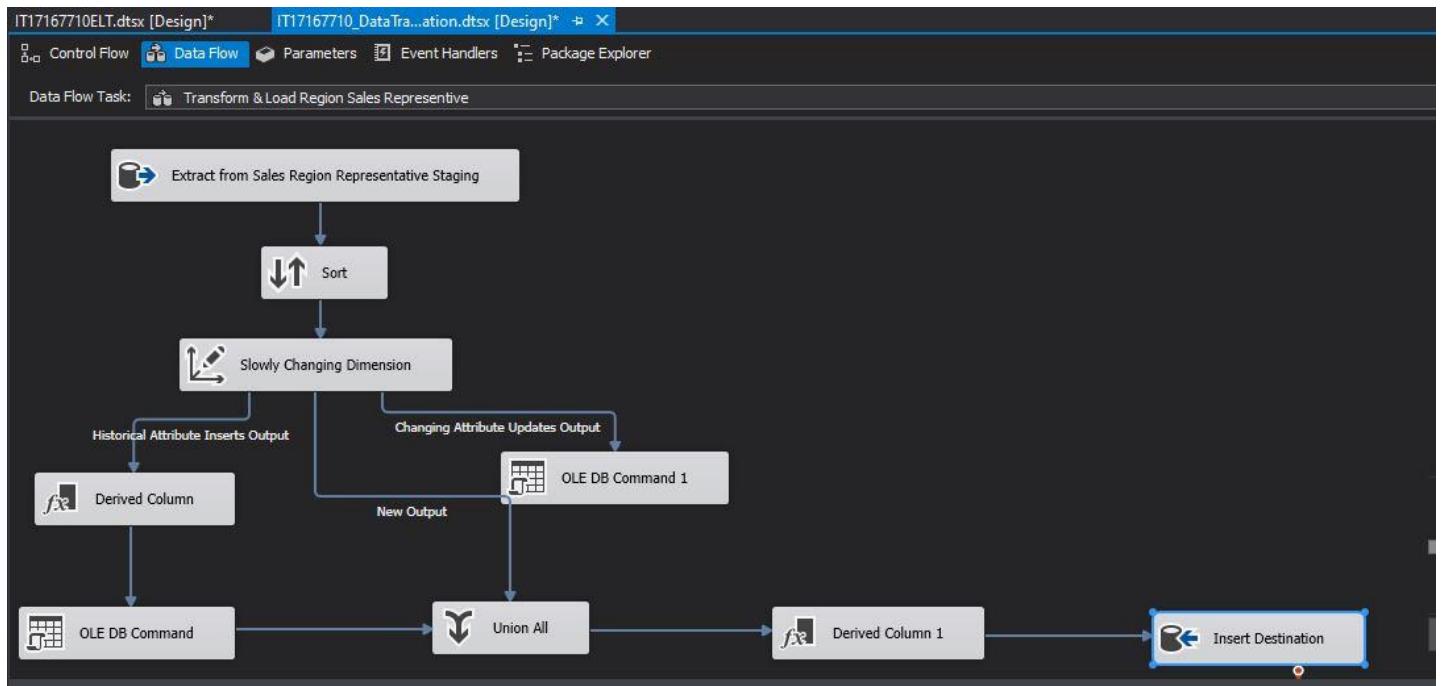
Variable to set date values: System::StartTime

Buttons: Help, < Back, Next >, Finish >>, Cancel

Once All Configurations done properly, it will automatically create the Slowly Changing Dimension as shown below.



Transform & Load Region Sales Representative



The reason for taking Dim Region Sales Representative as slowly changing dimension, is Sales Representative's state can be change time to time, and we should have to keep track of their historical Representative record. Additionally, if the Region Sales Representative changes, we should replace the old Representative with the new Representative.

Slowly Changing Dimension Wizard

Select a Dimension Table and Keys
Select a dimension table to load and map columns in the transformation input to columns in the dimension table.

Connection manager: DESKTOP-60G552M\SQLEXPRESS.IT17167710_DataWarehouse

Table or view: [dbo].[Dim_RegionSalesRepresentive]

Input Columns	Dimension Columns	Key Type
RegionID	AlternateRegionID	Business key
Country	Country	Not a key column
	EndDate	
Name	Name	Not a key column
Person	person	Not a key column
	StartDate	

Slowly Changing Dimension Wizard

Slowly Changing Dimension Columns
Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

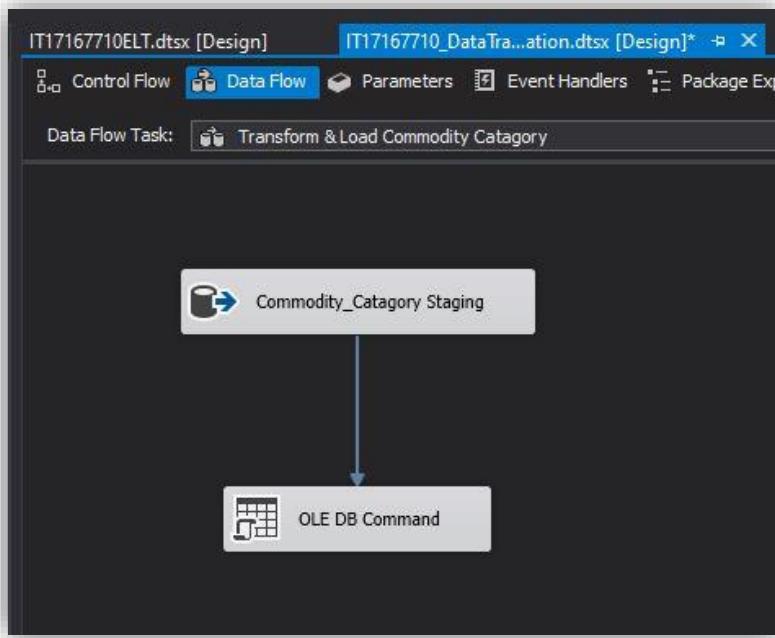
Fixed Attribute
Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute
Select this type when changed values should overwrite existing values. This is a Type 1 change.

Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
Name	Historical att...
person	Changing at...

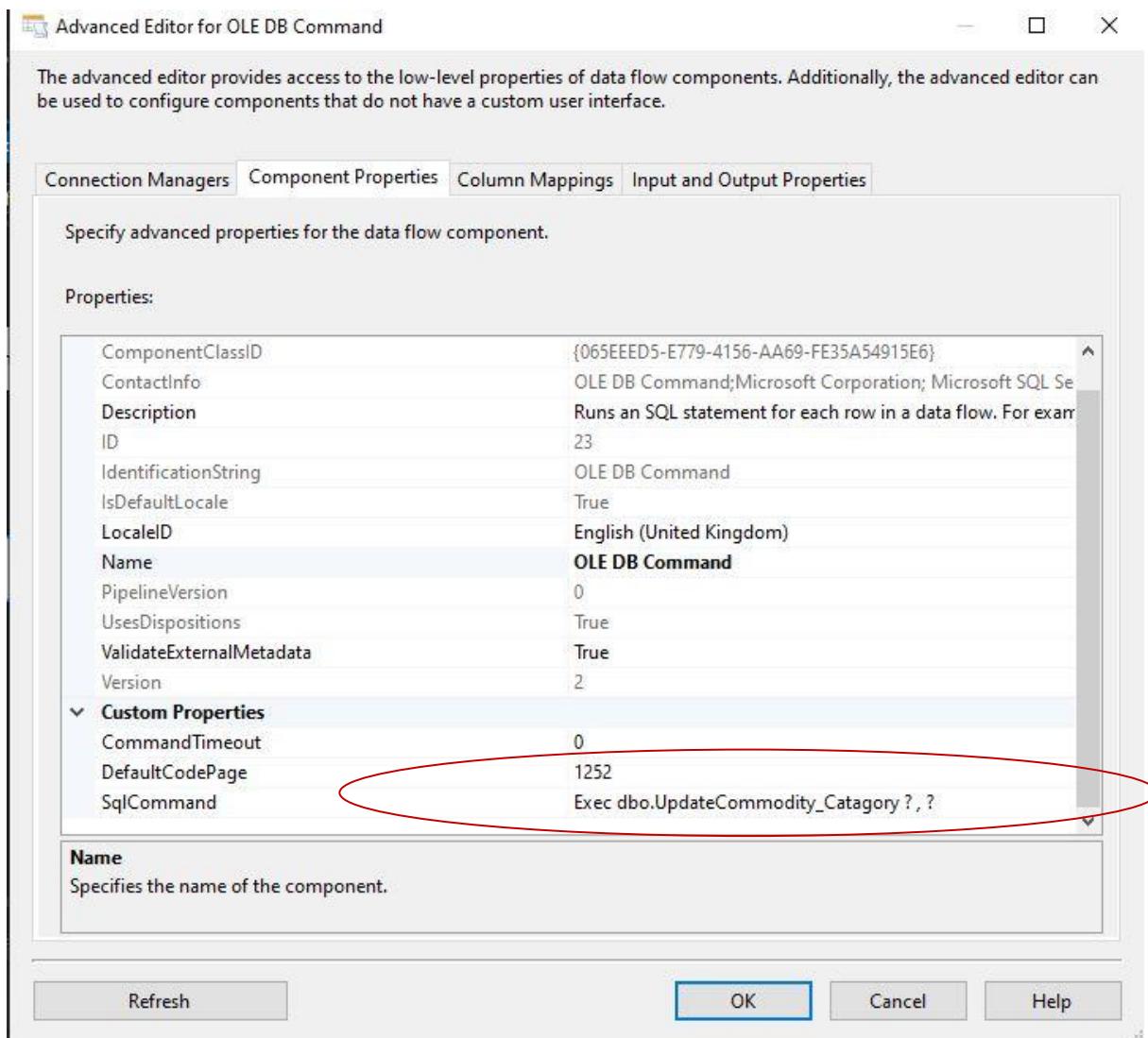
Transform & Load Commodity Category



First have create procedure called **UpdateCommodity_Catagory** and executed in the **IT17167710_DataWareHouse** database

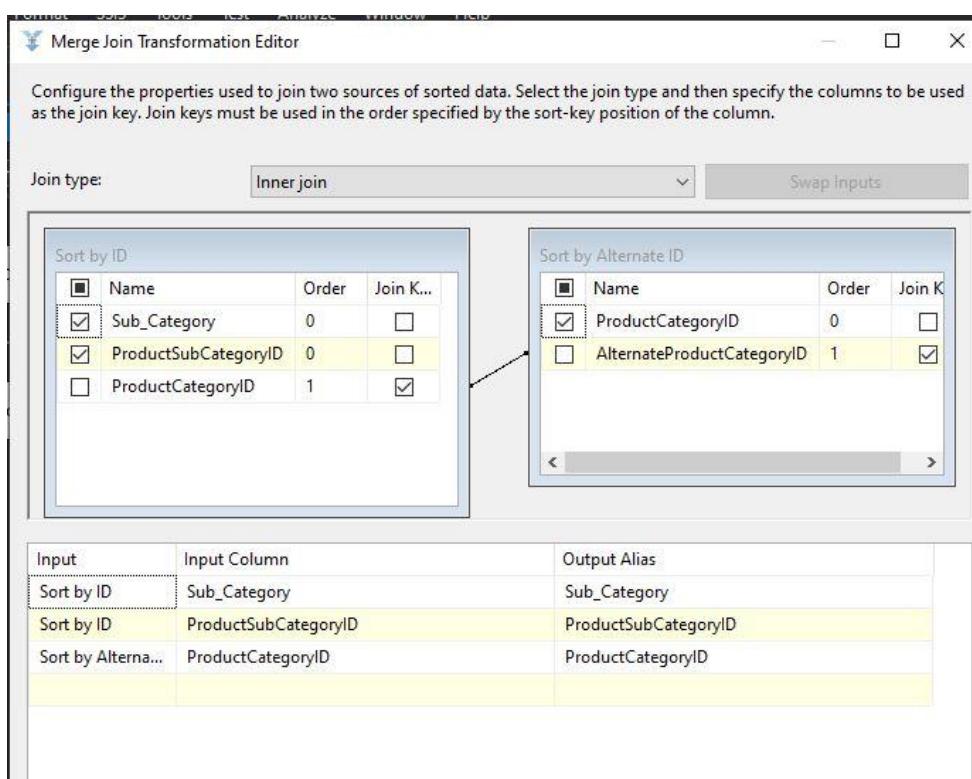
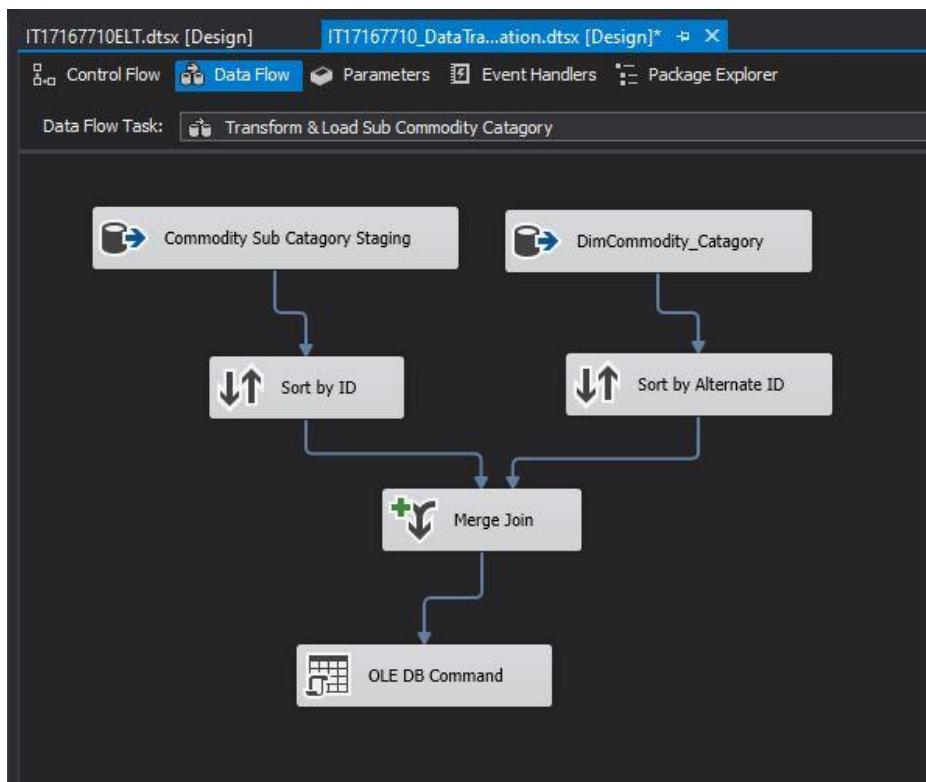
```
CREATE PROCEDURE dbo.UpdateCommodity_Catagory
@CommodityCategoryID int,
@CommodityCategoryName nvarchar(50)
AS
BEGIN
IF NOT EXISTS (
    SELECT ProductCategoryID
    FROM dbo.DimCommodity_Catagory
    WHERE AlternateProductCategoryID = @CommodityCategoryID
    AND
    Name = @CommodityCategoryName
)
BEGIN
    INSERT INTO dbo.DimCommodity_Catagory(AlternateProductCategoryID,Name)
        VALUES(@CommodityCategoryID,@CommodityCategoryName)
END;
END;
```

OLE DB Command SSIS tool used to execute, **UpdateCommodity_Catagory** procedure, it used to insert data into **Commodity_Catagory** Staging to Dim**Commodity_Catagory** without data duplication.



Transform & Load Sub Commodity Category

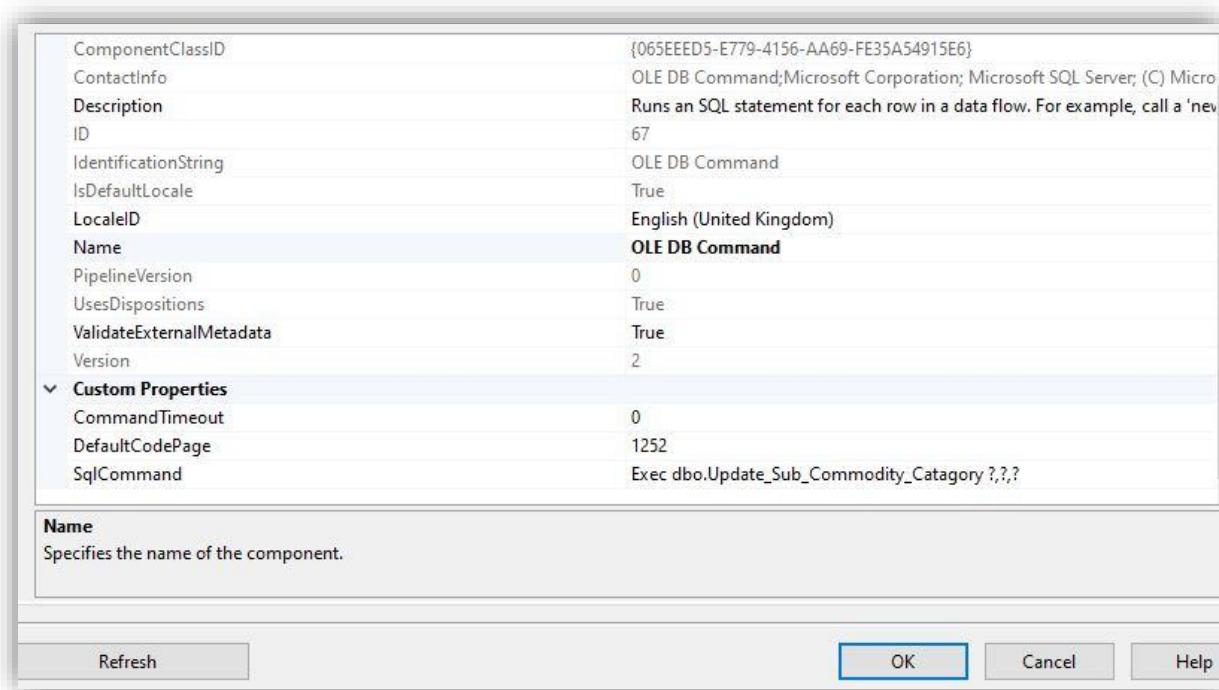
Use two OLE DB SOURCE to Commodity Sub category Staging and DimCommodity_Catagory in **IT17167710_DataWareHouse** database after use each source to Sort, SSIS tool and sort by ProductCategoryID in Commodity Subcategory Staging table and DimCommodity_Catagory Dimession table by AlternateProductCategoryID



Then I have created another procedure in **IT17167710_DataWareHouse** database as **Update_Sub_Commodity_Catagory**

- OLE DB Command SSIS item used to execute used **Update_Sub_Commodity_Catagory** procedure , it used to Sorted using ProductCategoryID data of Commodity Sub category Staging and DimCommodity_Catagory after Merging insert to DimSub_Commodity_Catagory without any data duplication

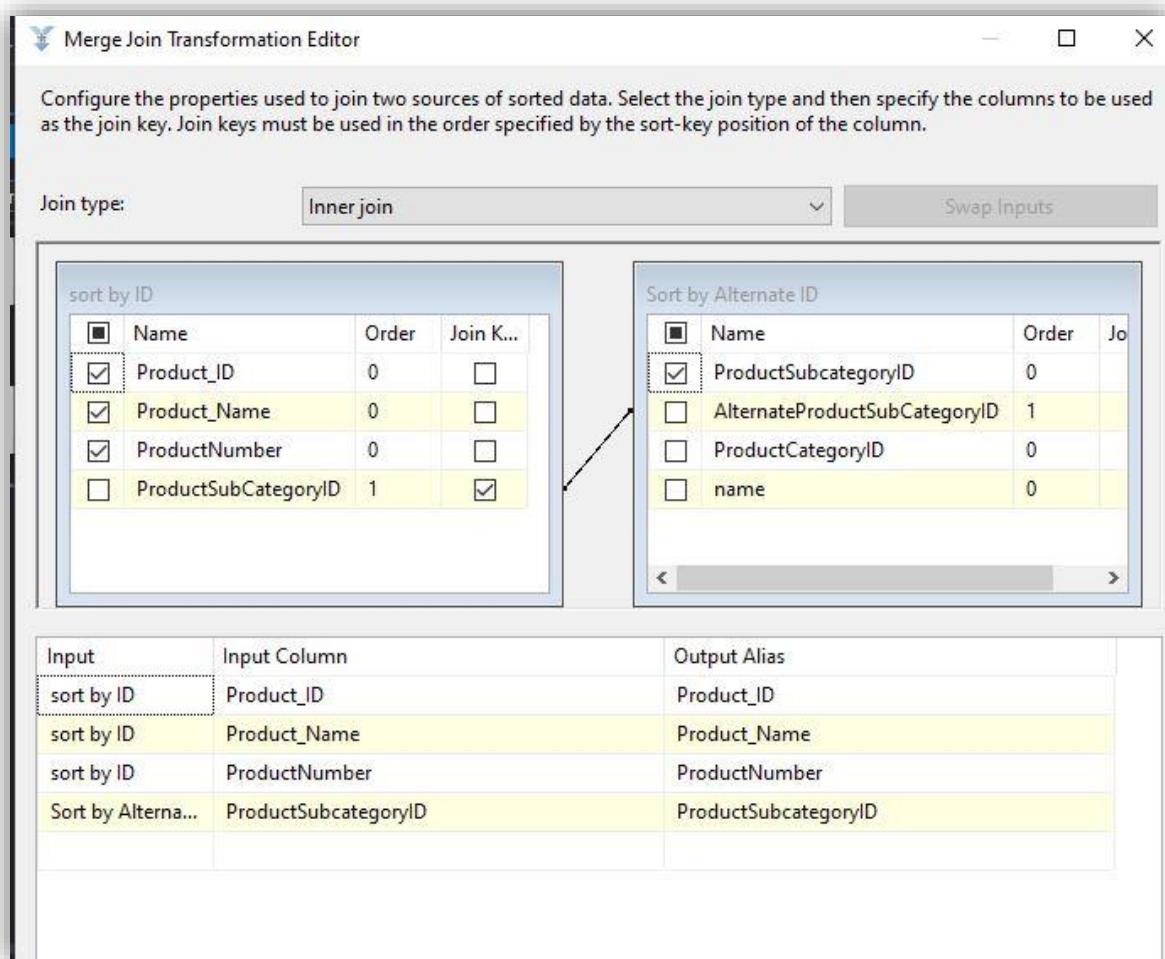
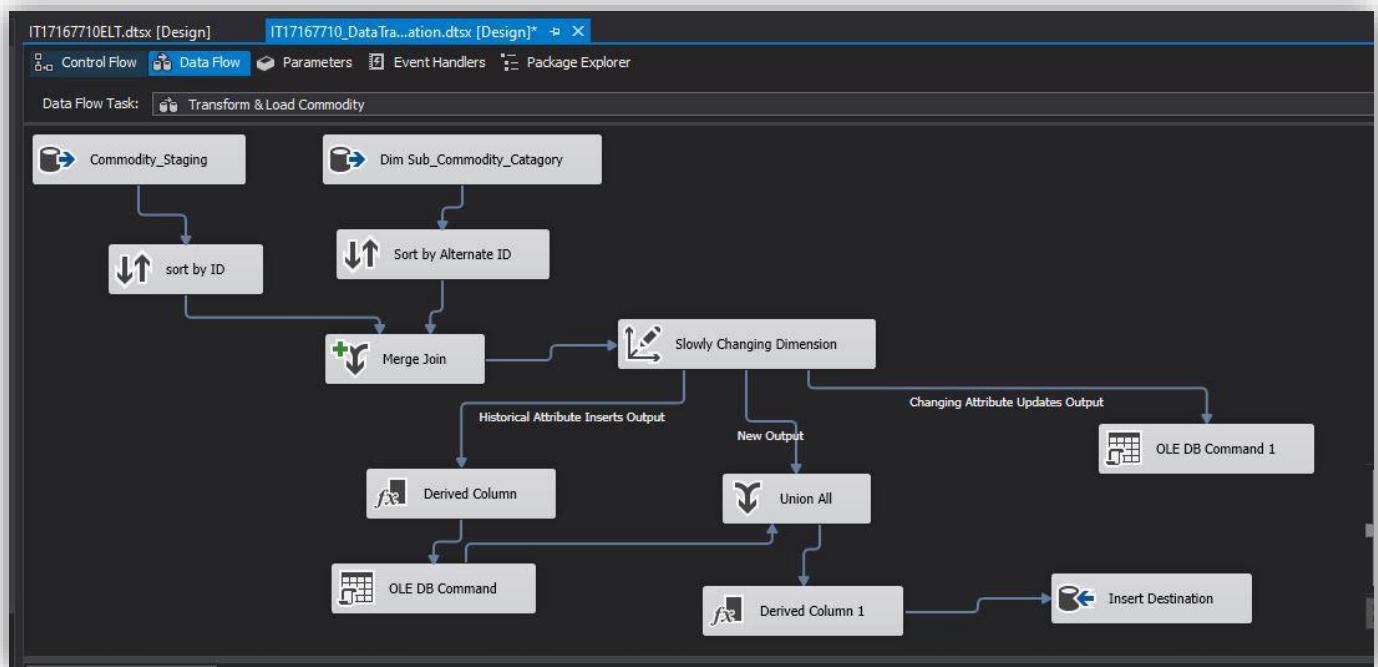
```
CREATE PROCEDURE dbo.Update_Sub_Commodity_Catagory
@Sub_CommodityCategoryID int,
@Sub_CommodityCategoryName nvarchar(50),
@CommodityCategoryID int
AS
BEGIN
    IF NOT EXISTS (
        SELECT ProductSubcategoryID
        FROM dbo.DimSub_Commodity_Catagory
        WHERE AlternateProductSubCategoryID = @Sub_CommodityCategoryID
            AND
            name = @Sub_CommodityCategoryName
            AND
            ProductCategoryID=@CommodityCategoryID
    )
    BEGIN
        INSERT INTO
        dbo.DimSub_Commodity_Catagory(AlternateProductSubCategoryID, name, ProductCategoryID)
        VALUES(@Sub_CommodityCategoryID, @Sub_CommodityCategoryName, @CommodityCategoryID)
    END;
END;
```



Transform & Load Commodity

Use two OLE DB SOURCE to Commodity_staging and DimSub_Commodity_Catagory in **IT17167710_DataWareHouse** database after use each source to Sort ,SSIS tool and sort by ProductSubCategoryID Commodity_staging table and DimSub_Commodity_Catagory table

AlternateProductSubCategoryID



Load to Sales Fact

In the IT17167710Transformations.dtsx, add another Data Flow Task and join the previous data flow task with the new data flow task.

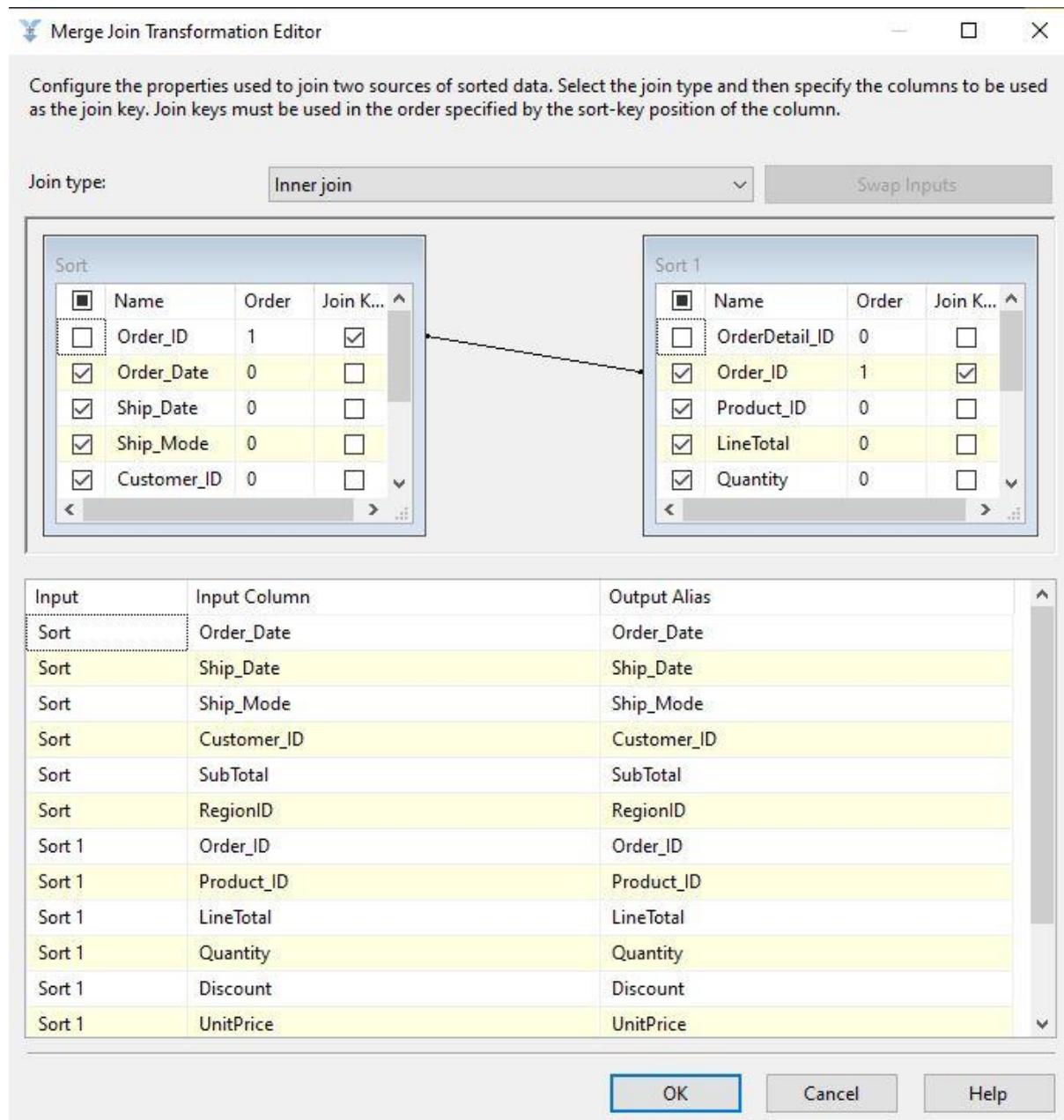
Renamed the new Data Flow Task as Load Sales Fact

Then I Dragged and dropped the OLE DB Source and configure it to fetch data from table.

Then I Used another OLE DB Source and configure it to fetch data from dbo.ShippingDetails_Staging table.

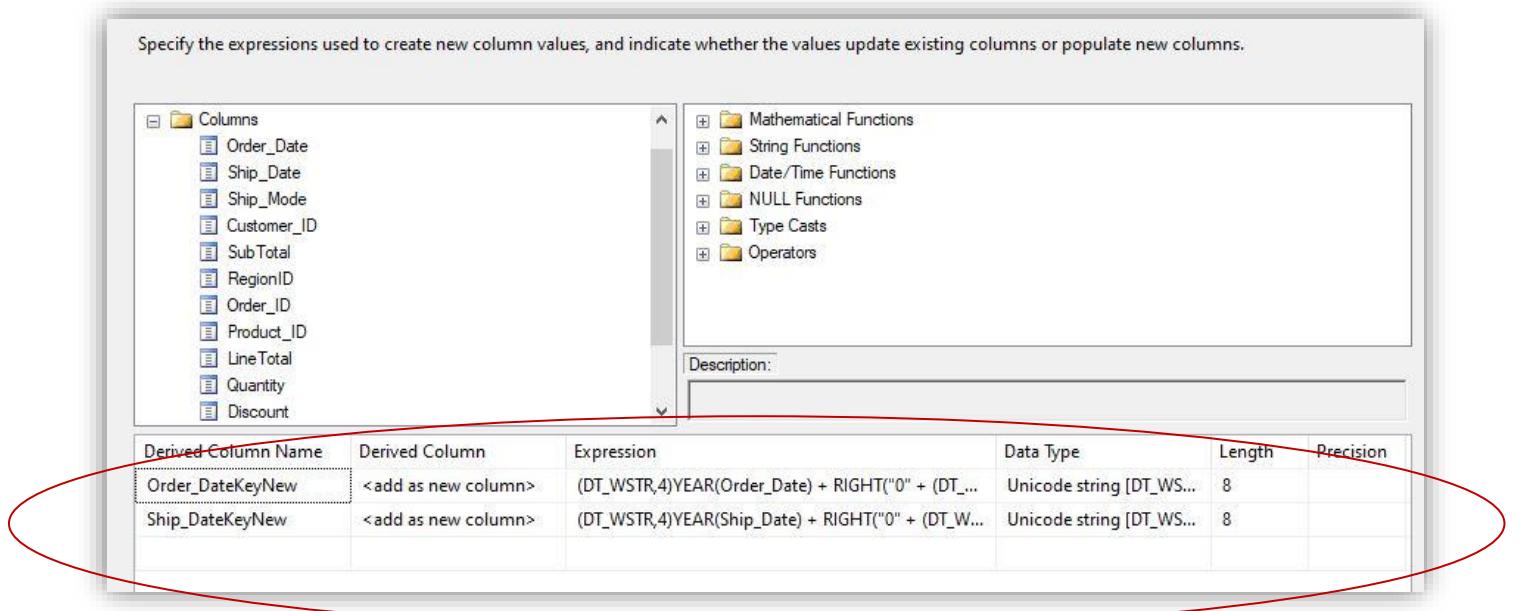
After that I Used a Merge Join to combine the two sources.

Then Used a Sort item in between Sources and Merge Join and sort the datasets from the field used to link the two datasets

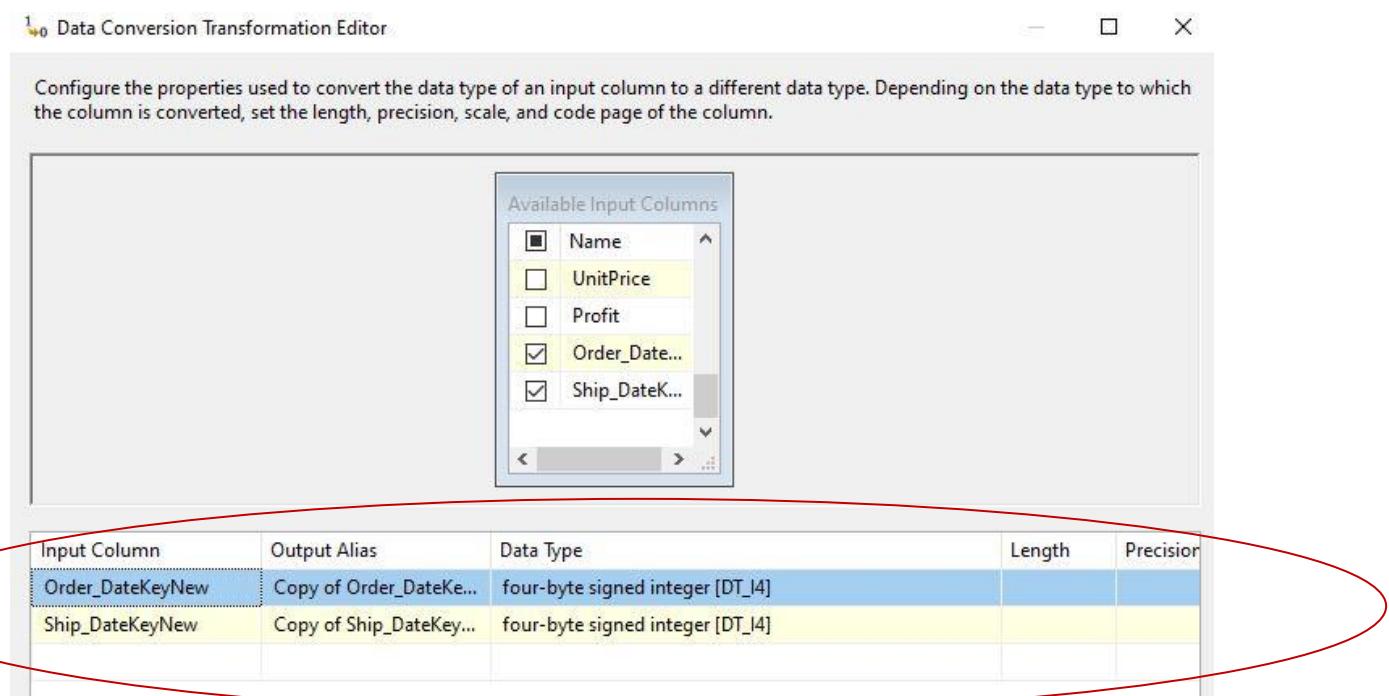


Then I Added a Derived Column and join the Merge Join item to Derived Column.

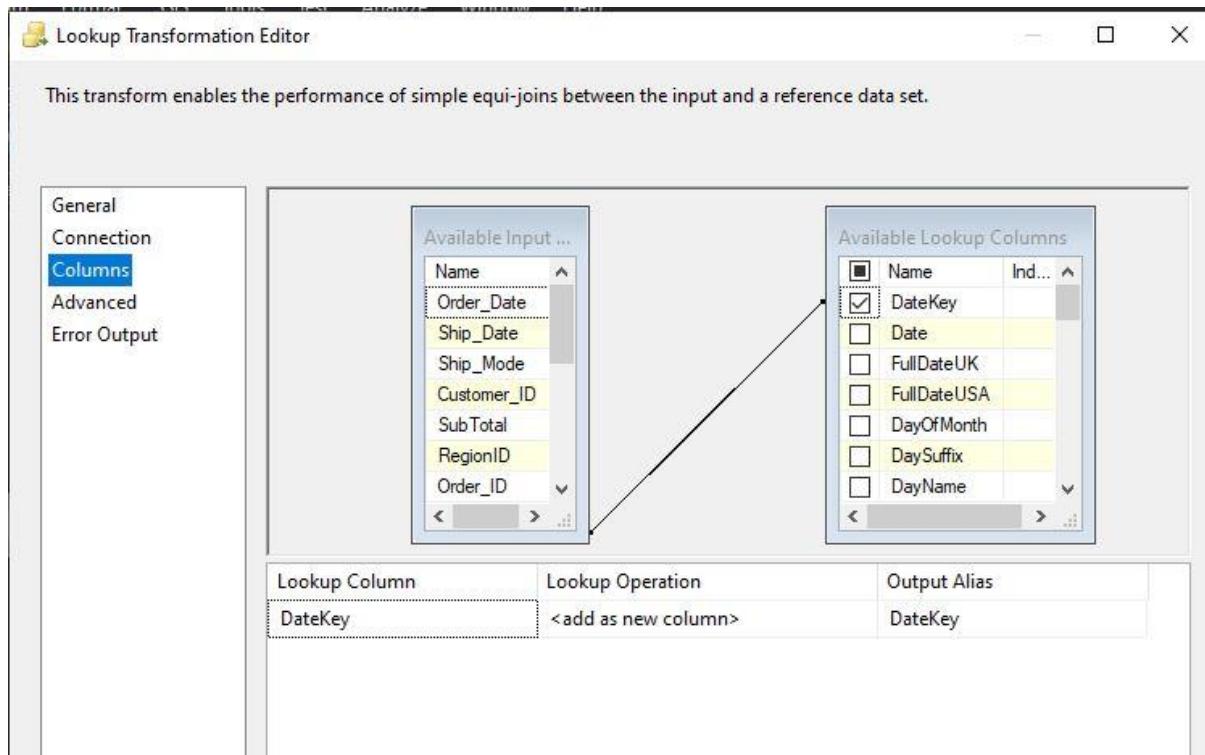
After that I Added a Data Conversion item & link the Derived Column item to Data Transformation item.



Then I Converted dates into numeric format by using Data Conversion

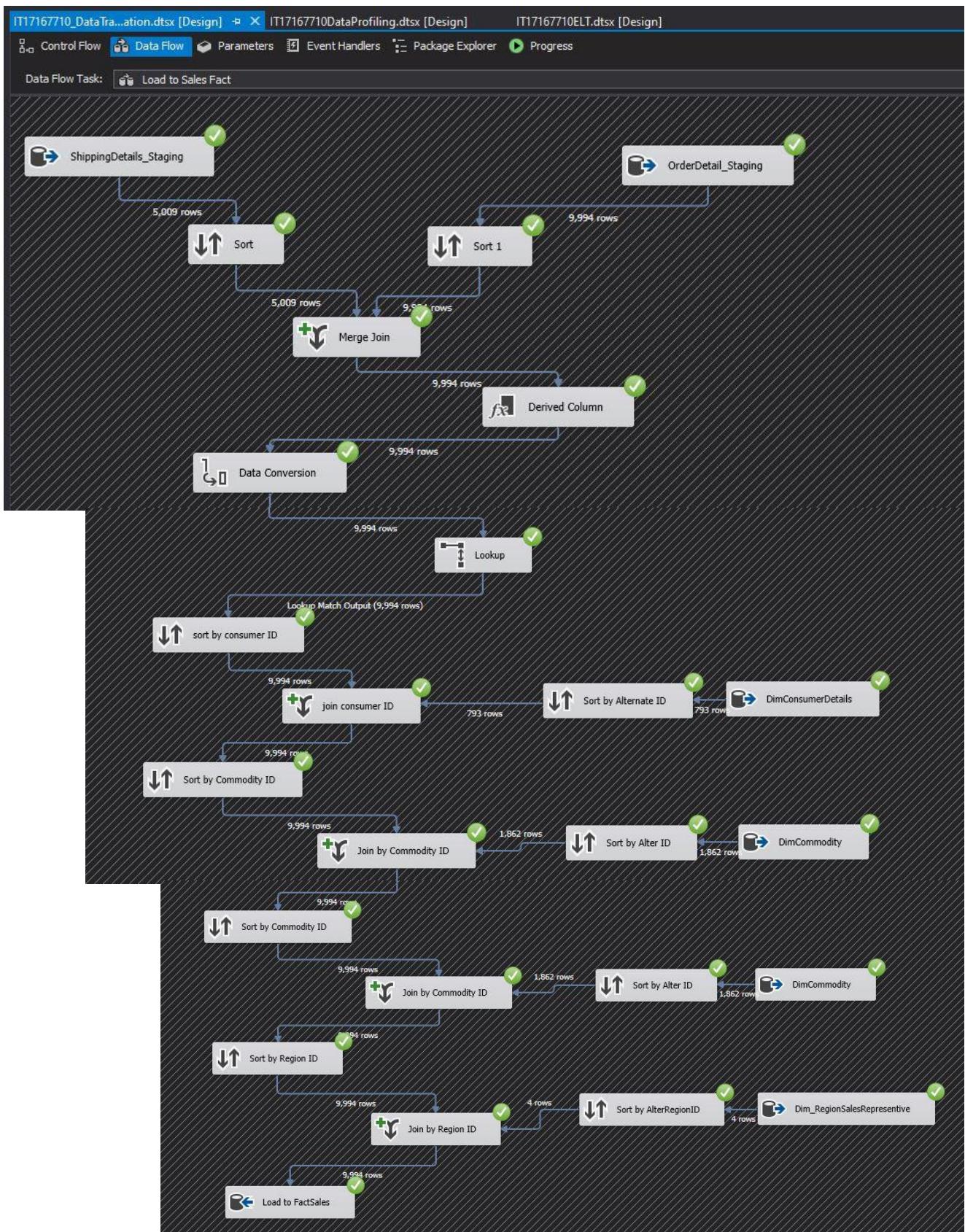


Then I Added a Lookup item and connect the Data Conversion item to it.



We have 3 foreign keys (CustomerID, ProductID,RegionID). Let's use OLE DB Source to point to their respective dimension tables and will use Merge Join to link the tables with the main data flow. Configure each accordingly to get the primary keys of each dimension by joining the data flows using the alternate key.

This section of your data flow should look similar to below



Step 7: Execution of Test Cases and create a Test Summary Report

Data Completeness Testing

Test Case ID	1		
Test Case Description	Test whether the all expected source data is loaded into staging tables and target Dimension tables.	Test Priority	High
Pre-Requisite	data loading process should be executed	Post-Requisite	NA

Test Execution Steps: execute the test case in SQL Server Management Studio.

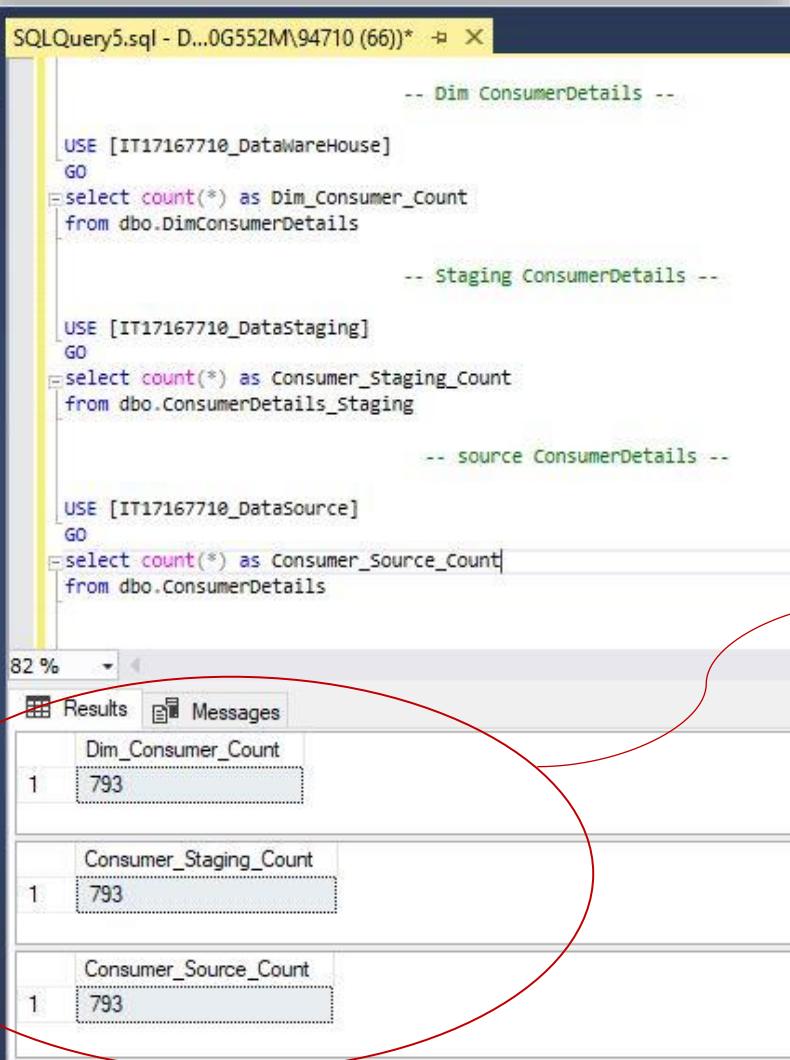
Check whether the test case is pass by Comparing record counts between source, staging and DW Dimension tables

No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	Check whether the Consumer source data is Loaded into staging table and target Dimension table Without data any lost	Data count of Consumer Source/Staging And Dim Table Should be Same Count = 793	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 1.1	Pass	Consumer details data count was equal to consumer staging and Dim consumer table in Data warehouse
2	Check whether the Commodity source data is Loaded into staging table and target Dimension table Without data any lost	Data count of Commodity Source/Staging And Dim Table Should be Same Count = 1862	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 1.2	Pass	Commodity details data count was equal to Commodity staging and Dim Commodity table in Data warehouse
3	Check whether the Order Detail source data is Loaded into staging table and target Fact Sales table Without data any lost	Data count of Order Detail Source/Staging And Dim Table Should be Same Count = 9994	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 1.3	Pass	Order Detail details data count was equal to Order Detail staging and FactSales table in Data warehouse
4	Check whether the of Sub Commodity Category data is Loaded into staging table and target Dimension table Without data any lost	Data count of Sub Commodity Category Source/Staging And Dim Table Should be Same Count = 17	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 1.4	Pass	Sub Commodity Category count was equal to Sub Commodity Category staging and Dim Sub Commodity Category table in Data warehouse

SQL Queries and Screen shots - Test Case ID: 1

- figure 1.1

```
-- Dim ConsumerDetails count--  
  
USE [IT17167710_DataWareHouse]  
GO  
select count(*) as Dim_Consumer_Count  
from dbo.DimConsumerDetails  
  
-- Staging ConsumerDetails count--  
  
USE [IT17167710_DataStaging]  
GO  
select count(*) as Consumer_Staging_Count  
from dbo.ConsumerDetails_Staging  
  
-- source ConsumerDetails count--  
  
USE [IT17167710_DataSource]  
GO  
select count(*) as Consumer_Source_Count  
from dbo.ConsumerDetails
```



The screenshot shows the SQL Query window with three distinct sections of T-SQL code. The first section counts records in the DimConsumerDetails table. The second section counts records in the ConsumerDetails_Staging table. The third section counts records in the ConsumerDetails table. All three sections return a single row with a value of 793.

Dim_Consumer_Count
793

Consumer_Staging_Count
793

Consumer_Source_Count
793

Each table contain same number of Consumer records, which means Consumer details are successfully extract to the staging table and Load to the target Dimension table Without data lost

- figure 1.2

```

-- Fact Sales --
USE [IT17167710_DataWareHouse]
GO
select count( Distinct ProductID) as Fact_Table_Commodity_Count
from dbo.SalesFact

-- Dim Commodity --

USE [IT17167710_DataWareHouse]
GO
select count(ProductID) as Dim_Commodity_Count
from dbo.DimCommodity

-- Staging Commodity --

USE [IT17167710_DataStaging]
GO
select count(Product_ID) as Staging_Commodity_Count
from dbo.Commodity_staging

-- Source Commodity --

USE [IT17167710_DataSource]
GO
select count(Product_ID) as Source_Commodity_Count
from dbo.Commodity

```

SQLQuery5.sql - D...0G552M\94710 (59)* ➔ X

```

----- Commodity -----
-- Fact Sales --
USE [IT17167710_DatawareHouse]
GO
select count( Distinct ProductID) as Fact_Table_Commodity_Count
from dbo.SalesFact

-- Dim Commodity --

USE [IT17167710_DataWareHouse]
GO
select count(ProductID) as Dim_Commodity_Count
from dbo.DimCommodity

-- Staging Commodity --

USE [IT17167710_DataStaging]
GO
select count(Product_ID) as Staging_Commodity_Count
from dbo.Commodity_staging

-- Source Commodity --

USE [IT17167710_DataSource]
GO
select count(Product_ID) as Source_Commodity_Count
from dbo.Commodity

```

32 %

	Results	Messages
Fact_Table_Commodity_Count	1 1862	
Dim_Commodity_Count	1 1862	
Staging_Commodity_Count	1 1862	
Source_Commodity_Count	1 1862	

- figure 1.3

```
-- Fact Sales --
USE [IT17167710_DataWarehouse]
GO
select count(OrderID) as Fact_Table_OrderCount
from dbo.SalesFact

-- Staging OrderDetail--
USE [IT17167710_DataStaging]
GO
select count(Order_ID) as Staging_OrderCount
from dbo.OrderDetail_staging

-- Source OrderDetail --
USE [IT17167710_DataSource]
GO
select count(Order_ID) as Source_OrderCount
from dbo.OrderDetail
```

SQLQuery5.sql - D...0G552M\94710 (59)*

```
-- Fact Sales --
USE [IT17167710_DataWarehouse]
GO
select count(OrderID) as Fact_Table_OrderCount
from dbo.SalesFact

-- Staging OrderDetail--

USE [IT17167710_DataStaging]
GO
select count(Order_ID) as Staging_OrderCount
from dbo.OrderDetail_staging

-- Source OrderDetail --

USE [IT17167710_DataSource]
GO
select count(Order_ID) as Source_OrderCount
from dbo.OrderDetail]
```

82 %

	Results	Messages
Fact_Table_OrderCount	1 9994	
Staging_OrderCount	1 9994	
Source_OrderCount	1 9994	

Query executed successfully.

- figure 1.4

```
-- Dim Sub_Commodity_Catagory --
USE [IT17167710_DataWarehouse]
GO
select count(*) as Dim_Sub_Commodity_Catagorye_Count
from dbo.DimSub_Commodity_Catagory

-- Staging Sub_Commodity_Catagory --
USE [IT17167710_DataStaging]
GO
select count(*) as Staging_Sub_Commodity_Catagory_Count
from dbo.Sub_Commodity_Catagory_staging

-- Scource Sub_Commodity_Catagory --
USE [IT17167710_DataSource]
GO
select count(*) as Source_Sub_Commodity_Catagory_Count
from dbo.Sub_Commodity_Catagory
```

SQLQuery5.sql - D...0G552M\94710 (59)*

```
-- Dim Sub_Commodity_Catagory --
USE [IT17167710_DataWarehouse]
GO
select count(*) as Dim_Sub_Commodity_Catagorye_Count
from dbo.DimSub_Commodity_Catagory

-- Staging Sub_Commodity_Catagory --
USE [IT17167710_DataStaging]
GO
select count(*) as Staging_Sub_Commodity_Catagory_Count
from dbo.Sub_Commodity_Catagory_staging

-- Scource Sub_Commodity_Catagory --
USE [IT17167710_DataSource]
GO
select count(*) as Source_Sub_Commodity_Catagory_Count
from dbo.Sub_Commodity_Catagory]
```

82 %

	Results	Messages
Dim_Sub_Commodity_Catagorye_Count	1 17	
Staging_Sub_Commodity_Catagory_Count	1 17	
Source_Sub_Commodity_Catagory_Count	1 17	

Data Duplication testing

Test Case ID	2					
Test Case Description	Ensure that there are no duplicate values exist in DW Dimension tables	Test Priority	High			
Pre-Requisite	data loading process should be executed	Post-Requisite	NA			
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	test whether the duplicate values are available in Consumer source table, staging table and target Dimension table	Duplicate Data count of Consumer Source/Staging And Dim Table Should be 0	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 2.1	Pass	Duplicated data count of Consumer details source, staging and Dim consumer table is 0 Which mean There is no data redundancy
2	test whether the duplicate values are available in Commodity source table, staging table and target Dimension table	Duplicate Data count of Commodity Source/Staging And Dim Table Should be 0	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 2.2	Pass	Duplicated data count of Commodity source, staging and Dim Commodity table is 0 Which mean There is no data redundancy
3	test whether the duplicate values are available in Sales Region Representative source table, staging table and target Dimension table	Duplicate Data count of Sales Region Representative Source/Staging And Dim Table Should be 0	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 2.3	Pass	Duplicated data count of Sales Region Representative source, staging and Dim Sales Region Representative table is 0 Which mean There is no data redundancy
4	test whether the duplicate values are available in Sub Commodity Category source table, staging table and target Dimension table	Duplicate Data count of Sub Commodity Category Source/Staging And Dim Table Should be 0	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 2.4	Pass	Duplicated data count of Sub Commodity Category source, staging and Dim Sub Commodity Category table is 0 Which mean There is no data redundancy

5	test whether the duplicate values are available in Commodity Category source table, staging table and target Dimension table	Duplicate Data count of Commodity Category Source/Staging And Dim Table Should be 0	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 2.5	Pass	Duplicated data count of Commodity Category source, staging and Dim of Commodity Category table is 0 Which mean There is no data redundancy
---	--	---	-------------------------	---	------	--

SQL Queries and Screen shots - Test Case ID: 2

- figure 2.1

```
-----Dim Consumer Details-----  
USE[IT17167710_DataWareHouse]  
GO  
select count (*) as Duplicate_values_In_Dimension  
from DimConsumerDetails c  
group by c.CustomerID  
having count (*)>1  
  
-----Consumer Details Staging-----  
USE[IT17167710_DataStaging]  
GO  
select count (*) as Duplicate_values_IN_Staging  
from ConsumerDetails_staging c  
group by c.Customer_ID  
having count (*)>1  
  
-----Consumer Details Source-----  
USE[IT17167710_DataSource]  
GO  
select count (*) as Duplicate_values_IN_Source  
from ConsumerDetails c  
group by c.Customer_ID  
having count (*)>1
```

SQLQuery5.sql - D...0G552M\94710 (75)* X

```
===== Data Duplicate =====  
-----Dim Consumer Details-----  
USE[IT17167710_DataWareHouse]  
GO  
select count (*) as Duplicate_values_In_Dimension  
from DimConsumerDetails c  
group by c.CustomerID  
having count (*)>1  
-----Consumer Details Staging-----  
USE[IT17167710_DataStaging]  
GO  
select count (*) as Duplicate_values_IN_Staging  
from ConsumerDetails_staging c  
group by c.Customer_ID  
having count (*)>1  
-----Consumer Details Source-----  
USE[IT17167710_DataSource]  
GO  
select count (*) as Duplicate_values_IN_Source  
from ConsumerDetails c  
group by c.Customer_ID  
having count (*)>1
```

Results Messages

Duplicate_values_In_Dimension
Duplicate_values_IN_Staging
Duplicate_values_IN_Source

Query executed successfully. DESKTOP-60G552M\SC

There are no Duplicated values in DimConsumerDetails, ConsumerDetails_staging and ConsumerDetails tables

- figure 2.2

```

-----Dim Commodity Details-----
USE[IT17167710_DataWarehouse]
GO
select count (*) as Duplicate_values_In_Commodity_Dimension
from DimCommodity c
group by c.ProductID
having count (*)>1
-----Commodity Staging-----

USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_Commodity_Staging
from Commodity_staging c
group by c.Product_ID
having count (*)>1

-----Commodity Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Commodity_Source
from Commodity c
group by c.Product_ID
having count (*)>1

```

SQLQuery5.sql - D...0G552M\94710 (75)* ✎ X

```

===== Data Duplicate =====
-----Dim Commodity Details-----
USE[IT17167710_DataWarehouse]
GO
select count (*) as Duplicate_values_In_Commodity_Dimension
from DimCommodity c
group by c.ProductID
having count (*)>1
-----Commodity Staging-----

USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_Commodity_Staging
from Commodity_staging c
group by c.Product_ID
having count (*)>1

-----Commodity Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Commodity_Source
from Commodity c
group by c.Product_ID
having count (*)>1

```

There are no Duplicated values in DimCommodity, Commodity_staging and Commodity tables

82 %

Results Messages

Duplicate_values_In_Commodity_Dimension

Duplicate_values_IN_Commodity_Staging

Duplicate_values_IN_Commodity_Source

Query executed successfully.

DESKTOP-60G552M\S

- figure 2.3

```

-----Dim_RegionSalesRepresentive Details-----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_Dim_RegionSalesRepresentive
from Dim_RegionSalesRepresentive c
group by c.RegionID
having count (*)>1

-----RegionSalesRepresentive Staging-----
USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_RegionSalesRepresentive_Staging
from salesRegionRepresentative_Staging c
group by c.RegionID
having count (*)>1

-----RegionSalesRepresentive Source-----
USE[IT17167710_DataSource2]
GO
select count (*) as Duplicate_values_IN_RegionSalesRepresentive_Source
from salesRegionRepresentative c
group by c.RegionID
having count (*)>1

```

SQLQuery5.sql - D...0G552M\94710 (75) * X

```

-----Data Commodity Sub Catagory in Consumer -----
-----Dim_RegionSalesRepresentive Details-----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_Dim_RegionSalesRepresentive
from Dim_RegionSalesRepresentive c
group by c.RegionID
having count (*)>1

-----RegionSalesRepresentive Staging-----
USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_RegionSalesRepresentive_Staging
from salesRegionRepresentative_Staging c
group by c.RegionID
having count (*)>1

-----RegionSalesRepresentive Source-----
USE[IT17167710_DataSource2]
GO
select count (*) as Duplicate_values_IN_RegionSalesRepresentive_Source
from salesRegionRepresentative c
group by c.RegionID
having count (*)>1

```

82 %

Results Messages

Duplicate_values_In_Dim_RegionSalesRepresentive
Duplicate_values_IN_RegionSalesRepresentive_Staging
Duplicate_values_IN_RegionSalesRepresentive_Source

Query executed successfully. | DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\94710 ...

- figure 2.4

```

USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_DimSub_Commodity_Catagory
from DimSub_Commodity_Catagory c
group by c.ProductSubcategoryID
having count (*)>1
-----Commodity Sub CatagoryStaging-----

USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_Sub_Commodity_Catagory_Staging
from Sub_Commodity_Catagory_staging c
group by c.ProductSubCategoryID
having count (*)>1

-----Commodity Sub Catagory Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Sub_Commodity_Catagory_Source
from Sub_Commodity_Catagory c
group by c.ProductSubCategoryID
having count (*)>1

```

SQLQuery5.sql - D...0G552M(94710 (75))*

```

-----Data Commodity Sub Catagory in Consumer -----
-----Dim sub Commodity Catagory -----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_DimSub_Commodity_Catagory
from DimSub_Commodity_Catagory c
group by c.ProductSubcategoryID
having count (*)>1
-----Commodity Sub CatagoryStaging-----

USE[IT17167710_Datastaging]
GO
select count (*) as Duplicate_values_IN_Sub_Commodity_Catagory_Staging
from Sub_Commodity_Catagory_staging c
group by c.ProductSubCategoryID
having count (*)>1

-----Commodity Sub Catagory Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Sub_Commodity_Catagory_Source
from Sub_Commodity_Catagory c
group by c.ProductsubCategoryID
having count (*)>1

```

82 %

Results Messages

Duplicate_values_In_DimSub_Commodity_Catagory
Duplicate_values_IN_Sub_Commodity_Catagory_Staging
Duplicate_values_IN_Sub_Commodity_Catagory_Source

Query executed successfully. DESKTOP-60G552M\SQLEXPRESS ... DESKTO

- figure 2.5

```

-----Dim Commodity Catagory-----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_Commodity_Catagory_Dimension
from DimCommodity_Catagory c
group by c.ProductCategoryID
having count (*)>1
----- Commodity Catagory Staging-----

USE[IT17167710_DataStaging]
GO
select count (*) as Duplicate_values_IN_Commodity_Catagory_Staging
from Commodity_Catagory_staging c
group by c.ProductCatagoryID
having count (*)>1

----- Commodity Catagory Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Commodity_Catagory_Source
from Commodity_Catagory c
group by c.ProductCatagoryID
having count (*)>1

```

SQLQuery5.sql - D...0G552M\94710 (75)* ✎ X

```

===== Data Duplicate =====
-----Dim Commodity Details-----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_Commodity_Dimension
from DimCommodity c
group by c.ProductID
having count (*)>1
-----Commodity Staging-----

USE[IT17167710_Datastaging]
GO
select count (*) as Duplicate_values_IN_Commodity_Staging
from Commodity_staging c
group by c.Product_ID
having count (*)>1

-----Commodity Source-----
USE[IT17167710_DataSource]
GO
select count (*) as Duplicate_values_IN_Commodity_Source
from Commodity c
group by c.Product_ID
having count (*)>1

```

82 %

Results	Messages
Duplicate_values_In_Commodity_Dimension	
Duplicate_values_IN_Commodity_Staging	
Duplicate_values_IN_Commodity_Source	

Query executed successfully. | DESKTOP-60G552M\S

3 Metadata Testing

Data Type Testing

Test Case ID	3.1					
Test Case Description	Verify that the table and column data type definitions are same in the Source, Staging And DW.		Test Priority	High		
Pre-Requisite	data loading process should be executed		Post-Requisite	NA		
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	Ensure that the table and column data type definitions are same in the Consumer Details Source, Staging And DW.	All the column data type definitions are should be same in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.1.1	Pass	All the column data type definitions are same in the Source, Staging And DW.
2	Ensure that the table and column data type definitions are same in the Commodity Details Source, Staging And DW.	All the column data type definitions are should be same in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.1.2	Pass	All the column data type definitions are same in the Source, Staging And DW.
3	Ensure that the table and column data type definitions are same in the sales Region Representative Details Source, Staging And DW.	All the column data type definitions are should be same in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.1.3	Pass	All the column data type definitions are same in the Source, Staging And DW.

SQL Queries and Screen shots - Test Case ID: 3.1

- figure 3.1.1

```
-- Source ConsumerDetails --
USE [IT17167710_DataSource]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION,
DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='ConsumerDetails'

-- Staging ConsumerDetails --
USE [IT17167710_DataStaging]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION,
DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='ConsumerDetails_staging'

-- Dim ConsumerDetails --
USE [IT17167710_DataWareHouse]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION,
DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='DimConsumerDetails'
```

SQLQuery5.sql - D...0G552M\94710 (59)* ✘ X

```
----- meta Data Testing -----
-- source ConsumerDetails --
USE [IT17167710_DataSource]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='ConsumerDetails'

-- Staging ConsumerDetails --
USE [IT17167710_DataStaging]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='ConsumerDetails_staging'

-- Dim ConsumerDetails --
USE [IT17167710_DataWareHouse]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='DimConsumerDetails'
```

82 %

COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1 Customer_ID	nvarchar	50	NULL	NULL	NO
2 First_Name	nvarchar	50	NULL	NULL	NO
3 Last_Name	nvarchar	50	NULL	NULL	YES
4 Segment	nvarchar	50	NULL	NULL	NO

COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1 Customer_ID	nvarchar	50	NULL	NULL	YES
2 First_Name	nvarchar	50	NULL	NULL	YES
3 Last_Name	nvarchar	50	NULL	NULL	YES
4 Segment	nvarchar	50	NULL	NULL	YES

COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1 CustomerID	int	NULL	10	NULL	NO
2 Customer_AlternateID	nvarchar	50	NULL	NULL	YES
3 FirstName	nvarchar	50	NULL	NULL	YES
4 LastName	nvarchar	50	NULL	NULL	YES
5 Segment	nvarchar	50	NULL	NULL	YES

Query executed successfully.

All the column data type definitions are same in the Source, Staging And DW.

I used above mentioned (figure 3.1.1) SQL Queries by changing table name for Verify the table and column data type definitions are same in the Source, Staging And DW.

- figure 3.1.2

TestingSQL_Script....0G552M\94710 (75)* -> X

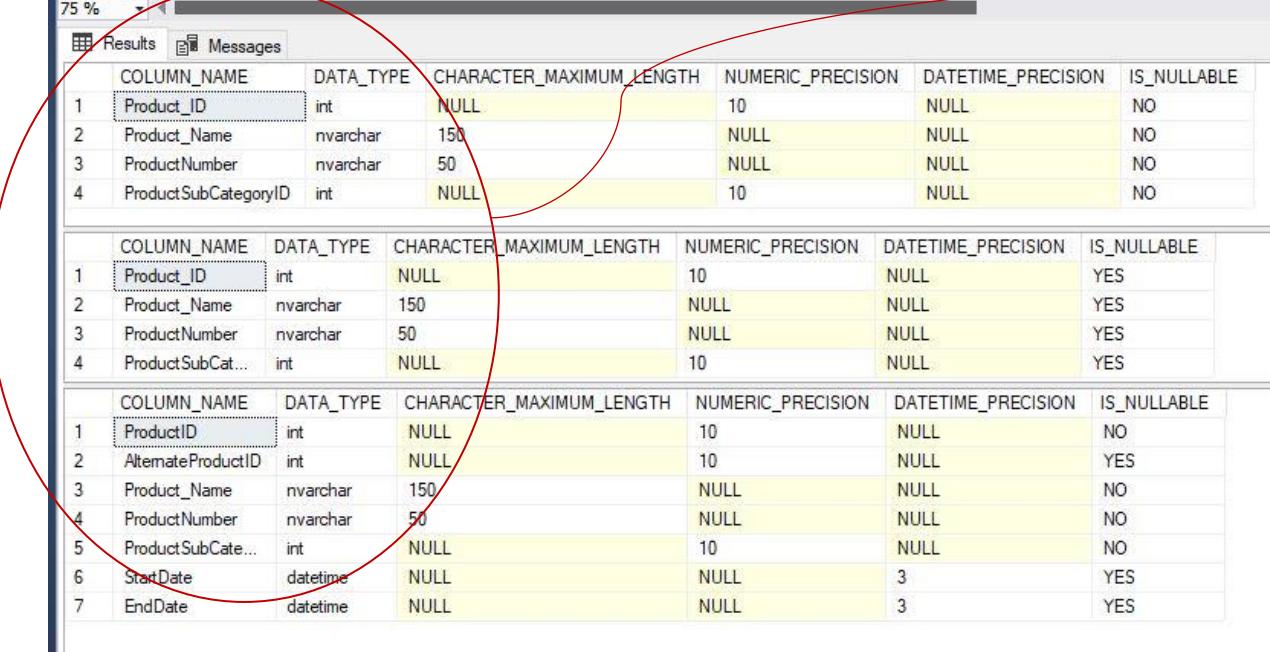
```
-- meta Data Testing (Data type testing)---

-- source Commodity --
USE [IT17167710_DataSource]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='Commodity'

-- Staging Commodity --
USE [IT17167710_DataStaging]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='Commodity_staging'

-- Dim Commodity --
USE [IT17167710_DataWareHouse]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, I
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='DimCommodity'
```

All the column data type definitions are same in the Source, Staging And DW.



	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	Product_ID	int	NULL	10	NULL	NO
2	Product_Name	nvarchar	150	NULL	NULL	NO
3	ProductNumber	nvarchar	50	NULL	NULL	NO
4	ProductSubCategoryID	int	NULL	10	NULL	NO

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	Product_ID	int	NULL	10	NULL	YES
2	Product_Name	nvarchar	150	NULL	NULL	YES
3	ProductNumber	nvarchar	50	NULL	NULL	YES
4	ProductSubCat...	int	NULL	10	NULL	YES

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	ProductID	int	NULL	10	NULL	NO
2	AlternateProductID	int	NULL	10	NULL	YES
3	Product_Name	nvarchar	150	NULL	NULL	NO
4	ProductNumber	nvarchar	50	NULL	NULL	NO
5	ProductSubCate...	int	NULL	10	NULL	NO
6	StartDate	datetime	NULL	NULL	3	YES
7	EndDate	datetime	NULL	NULL	3	YES

- figure 3.1.3

TestingSQL_Script....0G552M\94710 (75)* meta Data Testing (Data type testing) ----- --

```
-- source sales Region Representative --
USE [IT17167710_DataSource2]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='salesRegionRepresentative'
-- Staging sales Region Representative --

USE [IT17167710_DataStaging]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='salesRegionRepresentative_Staging'
-- Dim sales Region Representative --
USE [IT17167710_DataWareHouse]
GO
select COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, NUMERIC_PRECISION, DATETIME_PRECISION, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='Dim_RegionSalesRepresentative'
```

80 %

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	RegionID	int	NULL	10	NULL	NO
2	Name	nvarchar	50	NULL	NULL	NO
3	Country	nvarchar	50	NULL	NULL	NO
4	Person	nvarchar	50	NULL	NULL	NO

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	Person	nvarchar	50	NULL	NULL	YES
2	RegionID	int	NULL	10	NULL	YES
3	Name	nvarchar	50	NULL	NULL	YES
4	Country	nvarchar	50	NULL	NULL	YES

	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	DATETIME_PRECISION	IS_NULLABLE
1	RegionID	int	NULL	10	NULL	NO
2	AlternateRegionID	int	NULL	10	NULL	NO
3	Name	nvarchar	50	NULL	NULL	NO
4	Country	nvarchar	50	NULL	NULL	NO
5	person	nvarchar	50	NULL	NULL	NO

Query executed successfully. | DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\94710 ... | IT17167710_DataWareHouse | 00:00:00

3 Metadata Testing

Data Lengths Testing

In this testing I have randomly choose some rows execute the test cases

Test Case ID	3.2					
Test Case Description	Verify that the data lengths of Source, Staging and DW are same		Test Priority	High		
Pre-Requisite	data loading process should be executed		Post-Requisite	NA		
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	Test Whether the ' Adrian Barton ' Consumer's Data lengths are same in Source, Staging and DW are same.	All the data lengths: (First Name, Last Name, Segment) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.1	Pass	All data lengths are equals in source, staging & DW which means I can expect that Consumer details are accurately loaded and transformed as expected.
2	Test Whether the ' Alex Avila ' Consumer's Data lengths are same in Source, Staging and DW are same.	All the data lengths: (First Name, Last Name, Segment) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.2	Pass	All data lengths are equals in source, Staging & DW which means I can expect that Consumer details are accurately loaded and transformed as expected.
3	Test Whether the Product Number = ' FUR-BO-0000112 ' th Commodity Data lengths are same in Source, Staging and DW are same.	All the data lengths: (Product Name, Product Number, Product Subcategory ID) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.3	Pass	All data lengths are equals in source, Staging & which means we can assume that Commodity data are accurately loaded and transformed as expected.

4	Test Whether the Product Number = ' FUR-BO-10000330 ' Commodity Data lengths are same in Source, Staging and DW are same.	All the data lengths: (Product Name, Product Number, Product Subcategory ID) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.4	Pass	All data lengths are equals in source, Staging & which means we can assume that Commodity data are accurately loaded and transformed as expected.
5	Test Whether the ProductCategoryID=2 Commodity category Data lengths are same in Source, Staging and DW are same.	All the data lengths: (Category Name, Product Category ID) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.5	Pass	All data lengths are equals in source, Staging & DW which means we can assume that Commodity category data are accurately loaded and transformed as expected.
6	Test Whether the ProductCategoryID=1 Commodity category Data lengths are same in Source, Staging and DW are same.	All the data lengths: (Category Name, Product Category ID) should be equal in the Source, Staging And DW.	Same as Expected Output	Screen shots And SQL queries for the particular section is attached as - figure 3.2.6	Pass	All data lengths are equals in source, Staging & DW which means we can assume that Commodity category data are accurately loaded and transformed as expected.

SQL Queries and Screen shots - Test Case ID: 3.2

- figure 3.2.1

```
--Dim Consumer Details ---  
USE [IT17167710_DataWarehouse]  
GO  
SELECT DATALENGTH(c.FirstName) as Fname_length ,DATALENGTH(c.LastName) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From DimConsumerDetails c  
where c.FirstName='Adrian' AND c.LastName='Barton';  
  
---Dim Consumer Details ---  
USE [IT17167710_DataStaging]  
GO  
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From ConsumerDetails_staging c  
where c.First_Name='Adrian' AND c.Last_Name='Barton';  
  
---Source Consumer Details ---  
  
USE [IT17167710_DataSource]  
GO  
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From ConsumerDetails c  
where c.First_Name='Adrian' AND c.Last_Name='Barton';
```

SQLQuery5.sql - D...0G552M\94710 (75)* ↵ X

```
--Dim Consumer Details ---  
USE [IT17167710_DataWarehouse]  
GO  
SELECT DATALENGTH(c.FirstName) as Fname_length ,DATALENGTH(c.LastName) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From DimConsumerDetails c  
where c.FirstName='Adrian' AND c.LastName='Barton';  
  
---Dim Consumer Details ---  
USE [IT17167710_DataStaging]  
GO  
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From ConsumerDetails_staging c  
where c.First_Name='Adrian' AND c.Last_Name='Barton';  
  
---Source Consumer Details ---  
  
USE [IT17167710_DataSource]  
GO  
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length  
From ConsumerDetails c  
where c.First_Name='Adrian' AND c.Last_Name='Barton';
```

82 % ↵

Results Messages

	Fname_length	Lname_length	Segment_length
1	12	12	16

	Fname_length	Lname_length	Segment_length
1	12	12	16

	Fname_length	Lname_length	Segment_length
1	12	12	16

All the data lengths eg:
(First Name, Last Name, Segment)
should be equal in the Source,
Staging And DW.

Query executed successfully. | DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\94710 ... | IT17167710_DataS

- figure 3.2.2

```

---Dim Consumer Details ---
USE [IT17167710_DataWarehouse]
GO
SELECT DATALENGTH(c.FirstName) as Fname_length ,DATALENGTH(c.LastName) as
Lname_length ,DATALENGTH(c.Segment) as Segment_length
From DimConsumerDetails c
where c.FirstName='Alex' AND c.LastName='Avila';

---Dim Consumer Details ---
USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as
Lname_length ,DATALENGTH(c.Segment) as Segment_length
From ConsumerDetails_staging c
where c.First_Name='Alex' AND c.Last_Name='Avila';

---Source Consumer Details ---
USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as
Lname_length ,DATALENGTH(c.Segment) as Segment_length
From ConsumerDetails c
where c.First_Name='Alex' AND c.Last_Name='Avila';

```

SQLQuery5.sql - D...0G552M\94710 (75)* ↗ X

```

---Dim Consumer Details ---
USE [IT17167710_DataWarehouse]
GO
SELECT DATALENGTH(c.FirstName) as Fname_length ,DATALENGTH(c.LastName) as Lname_length ,DATALENGTH(c.Segment) as Segment_length
From DimConsumerDetails c
where c.FirstName='Alex' AND c.LastName='Avila';

---Dim Consumer Details ---
USE [IT17167710_Datastaging]
GO
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length
From ConsumerDetails_staging c
where c.First_Name='Alex' AND c.Last_Name='Avila';

---Source Consumer Details ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.First_Name) as Fname_length ,DATALENGTH(c.Last_Name) as Lname_length ,DATALENGTH(c.Segment) as Segment_length
From ConsumerDetails c
where c.First_Name='Alex' AND c.Last_Name='Avila';

```

82 % ↻

Results Messages

Fname_length	Lname_length	Segment_length	
1	8	10	16

Fname_length	Lname_length	Segment_length	
1	8	10	16

Fname_length	Lname_length	Segment_length	
1	8	10	16

Query executed successfully. | DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\94710 ... | IT17167710_DataSource

- figure 3.2.3

```

--Dim Commodity data length ---
USE [IT17167710_DataWarehouse]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From DimCommodity c
where c.ProductNumber='FUR-BO-10000112' ;

--staging Commodity data length ---
USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From Commodity_staging c
where c.ProductNumber='FUR-BO-10000112' ;
--Source Commodity Details ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From Commodity c
where c.ProductNumber='FUR-BO-10000112' ;

```

SQLQuery5.sql - D..0G552M\94710 (75)*

```

--Dim Commodity data length ---
USE [IT17167710_DataWarehouse]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From DimCommodity c
where c.ProductNumber='FUR-BO-10000112' ;

--staging Commodity data length ---
USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From Commodity_staging c
where c.ProductNumber='FUR-BO-10000112' ;
--Source Commodity Details ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From Commodity c
where c.ProductNumber='FUR-BO-10000112' ;

```

82 %

	Product_Name_length	ProductNumber_length	ProductSubCategoryID_length
1	96	30	4

	Product_Name_length	ProductNumber_length	ProductSubCategoryID_length
1	96	30	4

	Product_Name_length	ProductNumber_length	ProductSubCategoryID_length
1	96	30	4

All the data lengths eg:
 (Product Name, Product Number, Product Subcategory ID) should be equal in the Source, Staging And DW.

Query executed successfully. DESKTOP-60G552M\SQLEXPRESS ... DESKTOP-60G552M\94710 ... IT17167710_DataSource | 00:00:00 | 3 rows

- figure 3.2.4

---Dim Commodity data length ---

```
USE [IT17167710_DataWareHouse]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From DimCommodity c
where c.ProductNumber='FUR-BO-10000330' ;
```

---staging Commodity data length ---

```
USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From Commodity_staging c
where c.ProductNumber='FUR-BO-10000330' ;
---Source Commodity Details ---
```

```
USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber)
as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as
ProductSubCategoryID_length
From Commodity c
where c.ProductNumber='FUR-BO-10000330' ;
```

SQLQuery5.sql - D...0G552M\94710 (75))*

```
--Dim Commodity data length ---
USE [IT17167710_DataWareHouse]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From DimCommodity c
where c.ProductNumber='FUR-BO-10000112' ;

---staging Commodity data length ---
USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From Commodity_staging c
where c.ProductNumber='FUR-BO-10000112' ;
---Source Commodity Details ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Product_Name) as Product_Name_length ,DATALENGTH(c.ProductNumber) as ProductNumber_length ,DATALENGTH(c.ProductSubCategoryID) as ProductSubCategoryID_length
From Commodity c
where c.ProductNumber='FUR-BO-10000112' ;
```

82 %

	Product_Name_Length	ProductNumber_Length	ProductSubCategoryID_Length
1	124	30	4

	Product_Name_Length	ProductNumber_Length	ProductSubCategoryID_Length
1	124	30	4

	Product_Name_Length	ProductNumber_Length	ProductSubCategoryID_Length
1	124	30	4

Query executed successfully.

DESKTOP-60G552M\SQLEXPRESS ... DESKTOP-60G552M\94710 ... IT17167710_DataSource 00:00:00 3 rows

- figure 3.2.5

```

---Dim Commodity_Catagory Data Length---

USE [IT17167710_DataWareHouse]
GO
SELECT DATALENGTH(c.Name) as CommodityName_length
,DATALENGTH(c.ProductCategoryID) as CommodityID_length
From DimCommodity_Catagory c
where c.ProductCategoryID=2;

---Dim Commodity_Catagory Data Length---

USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length
,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory_staging c
where c.ProductCatagoryID=2;

---Source Commodity_Catagory Data Length ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length
,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCatagoryID=2;

```

SQLQuery5.sql - D...0G552M\94710 (75)* X

```

---Dim Commodity_Catagory Data Length---
USE [IT17167710_DataWareHouse]
GO
SELECT DATALENGTH(c.Name) as CommodityName_length ,DATALENGTH(c.ProductCategoryID) as CommodityID_length
From DimCommodity_Catagory c
where c.ProductCategoryID=2;

---Dim Commodity_Catagory Data Length ---
USE [IT17167710_Datastaging]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length ,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory_staging c
where c.ProductCatagoryID=2;

---Source Commodity_Catagory Data Length ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length ,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCatagoryID=2;

```

82 % ▾

	CommodityName_length	CommodityID_length
1	30	4

	CommodityName_length	CommodityID_length
1	30	4

	CommodityName_length	CommodityID_length
1	30	4

Query executed successfully. | DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\947

- figure 3.2.6

```

---Dim Commodity_Catagory Data Length---

USE [IT17167710_DataWareHouse]
GO
SELECT DATALENGTH(c.Name) as CommodityName_length
,DATALENGTH(c.ProductCategoryID) as CommodityID_length
From DimCommodity_Catagory c
where c.ProductCategoryID=1;

---Dim Commodity_Catagory Data Length---

USE [IT17167710_DataStaging]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length
,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory_staging c
where c.ProductCatagoryID=1;

---Source Commodity_Catagory Data Length ---

USE [IT17167710_DataSource]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length
,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCatagoryID=1;

```

SQLQuery5.sql - D...0G552M\94710 (75)*

```

--Dim Commodity_Catagory Data Length---
USE [IT17167710_DatawareHouse]
GO
SELECT DATALENGTH(c.Name) as CommodityName_length ,DATALENGTH(c.ProductCategoryID) as CommodityID_length
From DimCommodity_Catagory c
where c.ProductCategoryID=1;

--Dim Commodity_Catagory Data Length ---
USE [IT17167710_Datastaging]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length ,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory_staging c
where c.ProductCatagoryID=1;

--Source Commodity_Catagory Data Length ---

USE [IT17167710_Datasource]
GO
SELECT DATALENGTH(c.Category) as CommodityName_length ,DATALENGTH(c.ProductCatagoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCatagoryID=1;

```

82 %

	CommodityName_length	CommodityID_length
1	18	4

	CommodityName_length	CommodityID_length
1	18	4

	CommodityName_length	CommodityID_length
1	18	4

Query executed successfully.

DESKTOP-60G552M\SQLEXPRESS ... | DESKTOP-60G552M\94710

3 Metadata Testing

Index / Constraint Testing

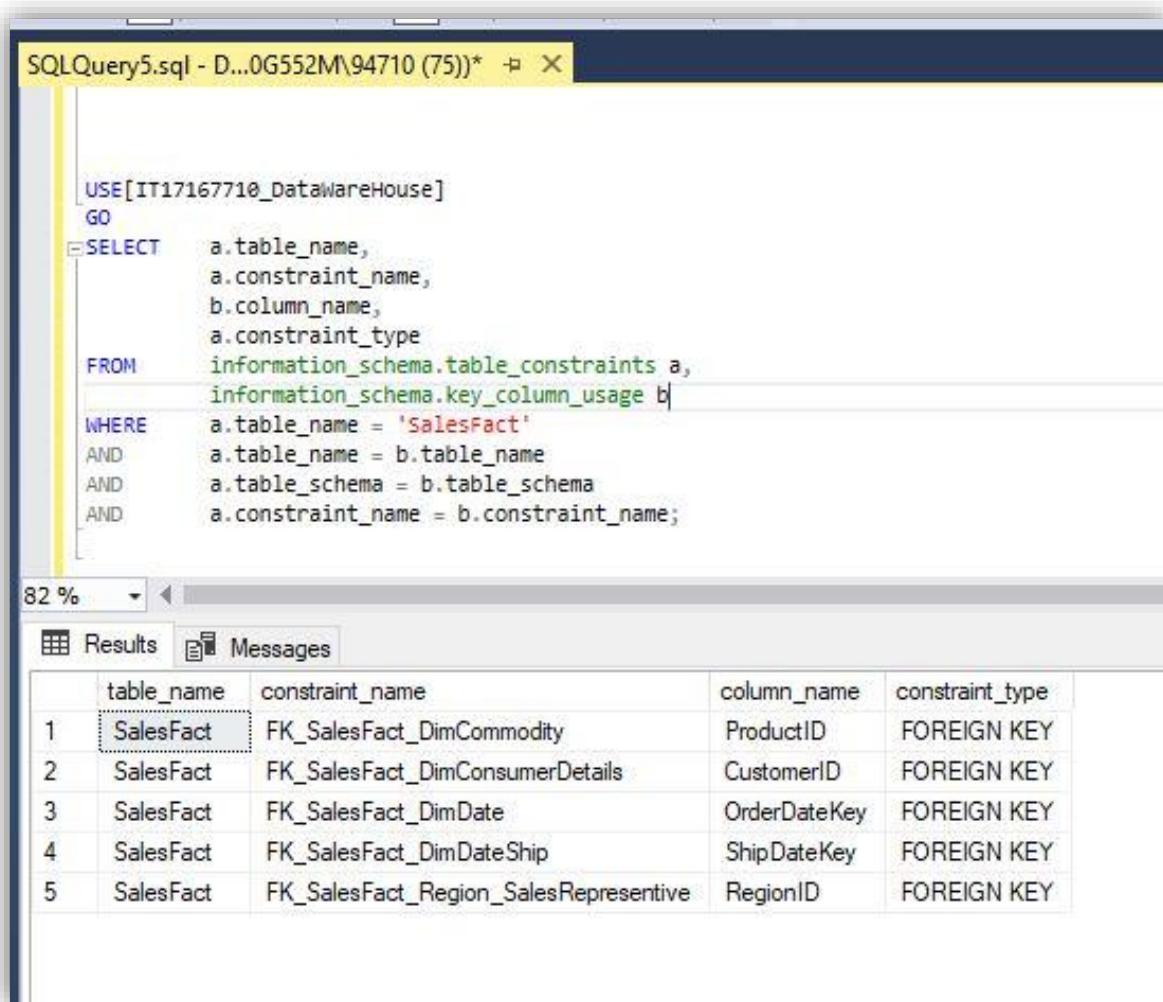
Test Case ID	3.3					
Test Case Description	Verify that proper constraints defined on the and DW Dimensions as per the Data Warehouse Design		Test Priority	High		
Pre-Requisite	NA		Post-Requisite	NA		
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	Ensure that the proper constraints defined on the ' SalesFact ' table	All the constraints Should be defined on the fact table according to the Data Warehouse Design	Output is same as Data Warehouse Design	Screen shots And SQL queries for the particular section is attached as - figure 3.3.1	Pass	According to the output the Fact Table constraints are defined as expected
2	Ensure that the proper constraints defined on the Dim_RegionSales Representative table	All the constraints Should be defined on the Dim RegionSales Representative table according to the Data Warehouse Design	Output is same as Data Warehouse Design	Screen shots And SQL queries for the particular section is attached as - figure 3.3.2	Pass	According to the output the Dim RegionSales Representative Table constraints are defined as expected
3	Ensure that the proper constraints defined on the Dim Commodity table	All the constraints Should be defined on the Dim Commodity table according to the Data Warehouse Design	Output is same as Data Warehouse Design	Screen shots And SQL queries for the particular section is attached as - figure 3.3.3	Pass	According to the output the Dim Commodity Table constraints are defined as expected
4	Ensure that the proper constraints defined on the Dim Commodity table	All the constraints Should be defined on the Dim Commodity table according to the Data Warehouse Design	Output is same as Data Warehouse Design	Screen shots And SQL queries for the particular section is attached as - figure 3.3.4	Pass	According to the output the Dim Commodity Table constraints are defined as expected

5	<p>Ensure that the proper constraints defined on the 'Dim Commodity Category' table</p>	<p>All the constraints Should be defined on the 'Dim Commodity Category' table according to the Data Warehouse Design</p>	<p>Output is same as Data Warehouse Design</p>	<p>Screen shots And SQL queries for the particular section is attached as - figure 3.3.5</p>	<p>Pass</p>	<p>According to the output the 'Dim Commodity Category 'Table constraints are defined as expected</p>
6	<p>Ensure that the proper constraints defined on the 'Dim Sub Commodity Category' table</p>	<p>All the constraints Should be defined on the 'Dim Commodity Category' table according to the Data Warehouse Design</p>	<p>Output is same as Data Warehouse Design</p>	<p>Screen shots And SQL queries for the particular section is attached as - figure 3.3.5</p>	<p>Pass</p>	<p>According to the output the 'Dim Commodity Category 'Table constraints are defined as expected</p>

SQL Queries and Screen shots - Test Case ID: 3.3

- figure 3.3.1

```
----- constraints SalesFact-----  
  
USE[IT17167710_DataWareHouse]  
GO  
SELECT      a.table_name,  
                    a.constraint_name,  
                    b.column_name,  
                    a.constraint_type  
FROM        information_schema.table_constraints a,  
                    information_schema.key_column_usage b  
WHERE       a.table_name = 'SalesFact'  
                    AND      a.table_name = b.table_name  
                    AND      a.table_schema = b.table_schema  
                    AND      a.constraint_name = b.constraint_name;
```



The screenshot shows a SQL query window titled "SQLQuery5.sql - D...0G552M\94710 (75)*". The query listed is identical to the one shown above. Below the query, the results pane displays a table with five rows of data, corresponding to the five foreign key constraints found in the SalesFact table.

	table_name	constraint_name	column_name	constraint_type
1	SalesFact	FK_SalesFact_DimCommodity	ProductID	FOREIGN KEY
2	SalesFact	FK_SalesFact_DimConsumerDetails	CustomerID	FOREIGN KEY
3	SalesFact	FK_SalesFact_DimDate	OrderDateKey	FOREIGN KEY
4	SalesFact	FK_SalesFact_DimDateShip	ShipDateKey	FOREIGN KEY
5	SalesFact	FK_SalesFact_Region_SalesRepresentive	RegionID	FOREIGN KEY

I used above mentioned (figure 3.3.1) SQL Query by changing table name to Verify that proper constraints defined on DW Dimensions as per the Data Warehouse Design

- figure 3.3.2

The screenshot shows a SQL query window with the following code:

```
SELECT      a.table_name,
            a.constraint_name,
            b.column_name,
            a.constraint_type
FROM        information_schema.table_constraints a,
            information_schema.key_column_usage b
WHERE       a.table_name = 'Dim_RegionSalesRepresentive'
            AND a.table_name = b.table_name
            AND a.table_schema = b.table_schema
            AND a.constraint_name = b.constraint_name;
```

The results grid displays one row of data:

	table_name	constraint_name	column_name	constraint_type
1	Dim_RegionSalesRepresentive	PK_Dim_RegionSalesRepresentive	RegionID	PRIMARY KEY

- figure 3.3.3

The screenshot shows a SQL query window with the following code:

```
SELECT      a.table_name,
            a.constraint_name,
            b.column_name,
            a.constraint_type
FROM        information_schema.table_constraints a,
            information_schema.key_column_usage b
WHERE       a.table_name = 'DimCommodity'
            AND a.table_name = b.table_name
            AND a.table_schema = b.table_schema
            AND a.constraint_name = b.constraint_name;
```

The results grid displays two rows of data:

	table_name	constraint_name	column_name	constraint_type
1	DimCommodity	FK_DimCommodity_DimSub_Commodity_Catagory	ProductSubCategoryID	FOREIGN KEY
2	DimCommodity	PK_DimCommodity	ProductID	PRIMARY KEY

- figure 3.3.4

```

SELECT      a.table_name,
            a.constraint_name,
            b.column_name,
            a.constraint_type
FROM        information_schema.table_constraints a,
            information_schema.key_column_usage b
WHERE       a.table_name = 'DimCommodity_Catagory'
            AND a.table_name = b.table_name
            AND a.table_schema = b.table_schema
            AND a.constraint_name = b.constraint_name;

```

82 %

Results Messages

	table_name	constraint_name	column_name	constraint_type
1	DimCommodity_Catagory	PK_DimCommo_3224ECEE4BA1AFD8	ProductCategoryID	PRIMARY KEY

- figure 3.3.5

```

SELECT      a.table_name,
            a.constraint_name,
            b.column_name,
            a.constraint_type
FROM        information_schema.table_constraints a,
            information_schema.key_column_usage b
WHERE       a.table_name = 'DimSub_Commodity_Catagory'
            AND a.table_name = b.table_name
            AND a.table_schema = b.table_schema
            AND a.constraint_name = b.constraint_name;

```

82 %

Results Messages

	table_name	constraint_name	column_name	constraint_type
1	DimSub_Commodity_Catagory	FK_Sub_Commodity_Catagory_Commodity_Catagory	ProductCategoryID	FOREIGN KEY
2	DimSub_Commodity_Catagory	ProductSubcategoryID	ProductSubcategoryID	PRIMARY KEY

4. Incremental ETL testing

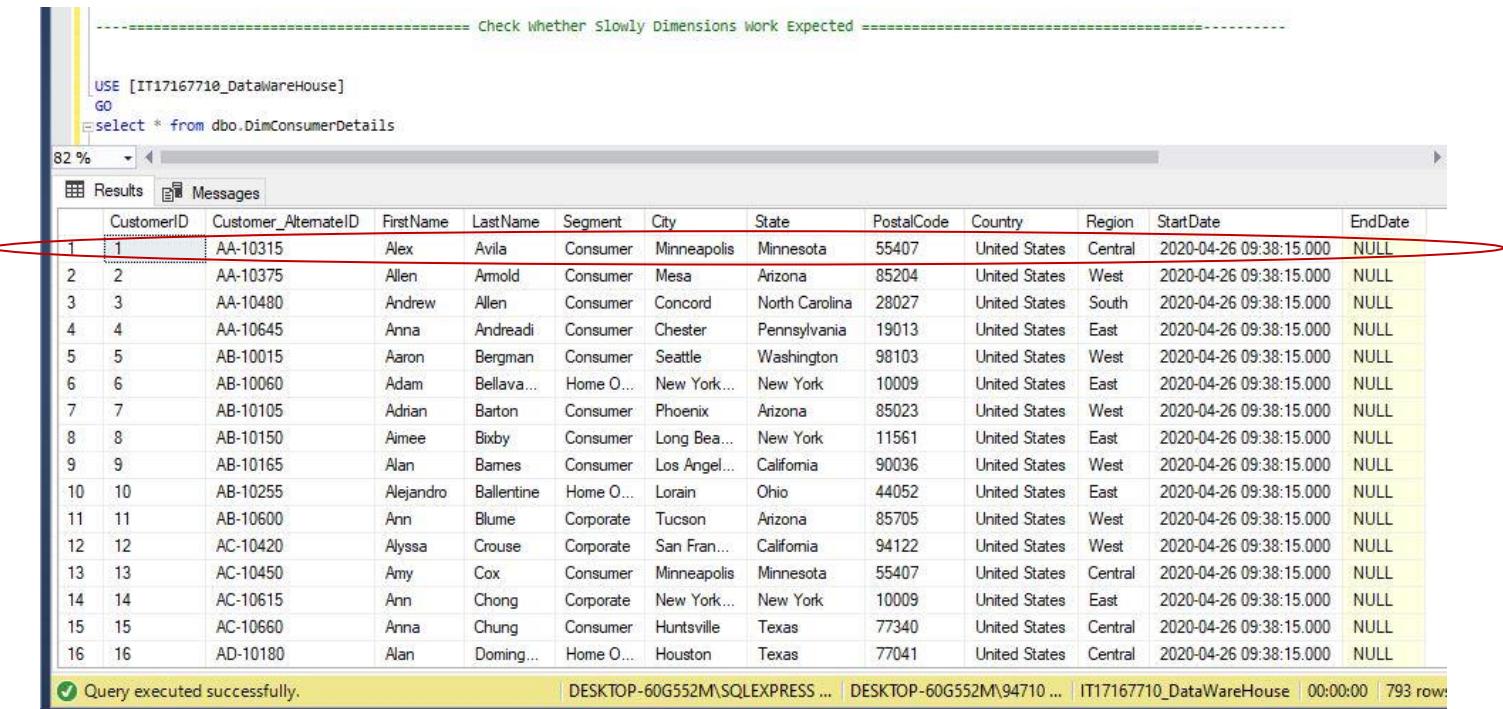
Test Case ID	3.1					
Test Case Description	Check Whether Slowly Changing Dimensions Work as Expected *Are the old records end dated appropriately? *Are the New records start dated appropriately		Test Priority	High		
Pre-Requisite	Dimension table should have historical Attributes		Post-Requisite	NA		
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	Ensure that the Consumer Details Slowly Changing dimension is Works properly By updating one consumer record.	Consumer Data should be successfully updated the current record Should be closed A new record Should be created with the changed data values and this new record Should become the current record	Same as Expected	Screen shots And SQL queries for the particular section is attached as - figure 4.1.	Pass	Consumer Details Slowly Changing dimension is Works properly As expected.
2	Ensure that the Commodity Slowly Changing dimension is Works properly By updating one Commodity record.	Commodity Data should be successfully updated the current record Should be closed A new record Should be created with the changed data values and this new record Should become the current record	Same as Expected	Screen shots And SQL queries for the particular section is attached as - figure 4.2	Pass	commodity Slowly Changing dimension is Works properly As expected.

3	<p>Ensure that the Sales Region Representative Slowly Changing dimension is Works properly By updating one Commodity record.</p>	<p>Sales Region Representative Data should be successfully updated the current record Should be closed A new record Should be created with the changed data values and this new record Should become the current record</p>	Same as Expected	<p>Screen shots And SQL queries for the particular section is attached as - figure 4.3</p>	Fail	<p>This test case failed because Sales Region Representative dimension doesn't have any historical attributes, I only assigned changing attributes. So that it wont Closed the current record.</p>
---	--	---	------------------	--	------	--

SQL Queries and Screen shots - Test Case ID: 4.1

- figure 4.1

Before update the recode



```
-- Check Whether Slowly Dimensions Work Expected
USE [IT17167710_DataWarehouse]
GO
SELECT * FROM dbo.DimConsumerDetails
```

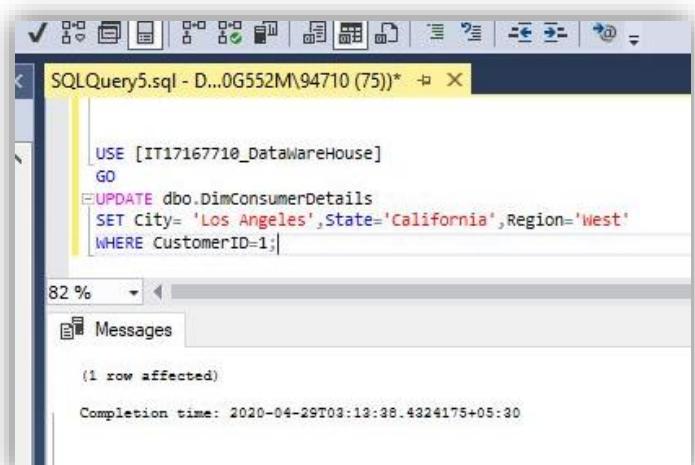
CustomerID	Customer_AlternateID	FirstName	LastName	Segment	City	State	PostalCode	Country	Region	StartDate	EndDate
1	AA-10315	Alex	Avila	Consumer	Minneapolis	Minnesota	55407	United States	Central	2020-04-26 09:38:15.000	NULL
2	AA-10375	Allen	Arnold	Consumer	Mesa	Arizona	85204	United States	West	2020-04-26 09:38:15.000	NULL
3	AA-10480	Andrew	Allen	Consumer	Concord	North Carolina	28027	United States	South	2020-04-26 09:38:15.000	NULL
4	AA-10645	Anna	Andreadi	Consumer	Chester	Pennsylvania	19013	United States	East	2020-04-26 09:38:15.000	NULL
5	AB-10015	Aaron	Bergman	Consumer	Seattle	Washington	98103	United States	West	2020-04-26 09:38:15.000	NULL
6	AB-10060	Adam	Bellav... Home O...	New York...	New York	10009	United States	East	2020-04-26 09:38:15.000	NULL	
7	AB-10105	Adrian	Barton	Consumer	Phoenix	Arizona	85023	United States	West	2020-04-26 09:38:15.000	NULL
8	AB-10150	Aimee	Bixby	Consumer	Long Bea...	New York	11561	United States	East	2020-04-26 09:38:15.000	NULL
9	AB-10165	Alan	Bames	Consumer	Los Angel...	California	90036	United States	West	2020-04-26 09:38:15.000	NULL
10	AB-10255	Alejandro	Ballentine	Home O...	Lorain	Ohio	44052	United States	East	2020-04-26 09:38:15.000	NULL
11	AB-10600	Ann	Blume	Corporate	Tucson	Arizona	85705	United States	West	2020-04-26 09:38:15.000	NULL
12	AC-10420	Alyssa	Crouse	Corporate	San Fran...	California	94122	United States	West	2020-04-26 09:38:15.000	NULL
13	AC-10450	Amy	Cox	Consumer	Minneapolis	Minnesota	55407	United States	Central	2020-04-26 09:38:15.000	NULL
14	AC-10615	Ann	Chong	Corporate	New York...	New York	10009	United States	East	2020-04-26 09:38:15.000	NULL
15	AC-10660	Anna	Chung	Consumer	Huntsville	Texas	77340	United States	Central	2020-04-26 09:38:15.000	NULL
16	AD-10180	Alan	Doming...	Home O...	Houston	Texas	77041	United States	Central	2020-04-26 09:38:15.000	NULL

Query executed successfully.

Assume that Alex Avla moved from Minnesota to California,

So, I updated the state and city to California and Los Angeles.

```
USE [IT17167710_DataWarehouse]
GO
UPDATE dbo.DimConsumerDetails
SET City= 'Los Angeles', State='California',
Region='West'
WHERE CustomerID=1;
```



```
SQLQuery5.sql - D...OG552M\94710 (75)*
```

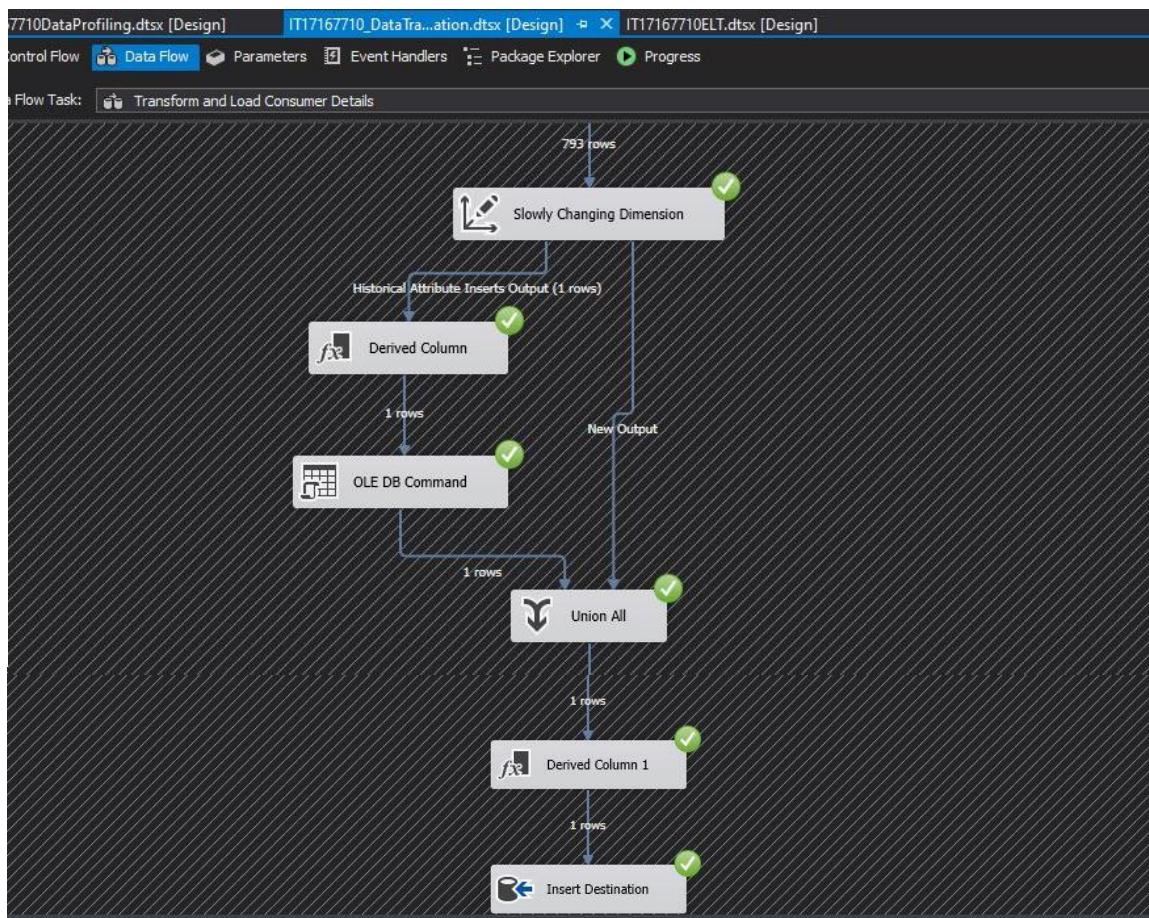
```
USE [IT17167710_DataWarehouse]
GO
UPDATE dbo.DimConsumerDetails
SET City= 'Los Angeles', State='California', Region='West'
WHERE CustomerID=1;
```

82 %

Messages

(1 row affected)

Completion time: 2020-04-29T03:13:38.4324175+05:30



When the value of a chosen attribute changes

```

----- Check Whether Slowly Dimensions Work as Expected -----
USE [IT17167710_DataWarehouse]
GO
select *
from dbo.DimConsumerDetails c
where c.FirstName='Alex' AND c.LastName='Avila';
  
```

Results grid:

	CustomerID	Customer_AlternateID	FirstName	LastName	Segment	City	State	PostalCode	Country	Region	StartDate	EndDate
1	1	AA-10315	Alex	Avila	Consumer	Los Angeles	California	55407	United States	West	2020-04-26 09:38:15.000	2020-04-29 03:16:07.000
2	794	AA-10315	Alex	Avila	Consumer	Minneapolis	Minnesota	55407	United States	Central	2020-04-29 03:16:07.000	NULL

A new record is created with the changed data values and this new record becomes the current record

the current record is closed

Before update the recode

```
----- Dim Commodity Slowly Changing Dimensions -----
USE [IT17167710_DataWarehouse]
GO
select *
from dbo.DimCommodity c
```

82 % ▶ Results Messages

ProductID	AlternateProductID	Product_Name	ProductNumber	ProductSubCategoryID	StartDate	EndDate
1	1	Bush Birmingham Collection Bookcase, Dark Cherry	FUR-BO-10000112	1	2020-04-26 12:07:27.000	NULL
2	28	Sauder Camden County Collection Libraries, Plank...	FUR-BO-10003159	1	2020-04-26 12:07:27.000	NULL
3	29	O'Sullivan Living Dimensions 5-Shelf Bookcases	FUR-BO-10003272	1	2020-04-26 12:07:27.000	NULL
4	30	Global Adaptabilities Bookcase, Cherry/Storm Gray ...	FUR-BO-10003404	1	2020-04-26 12:07:27.000	NULL

Assume Store Owner wants to change that 1st record Product name in to Dark Bookcase

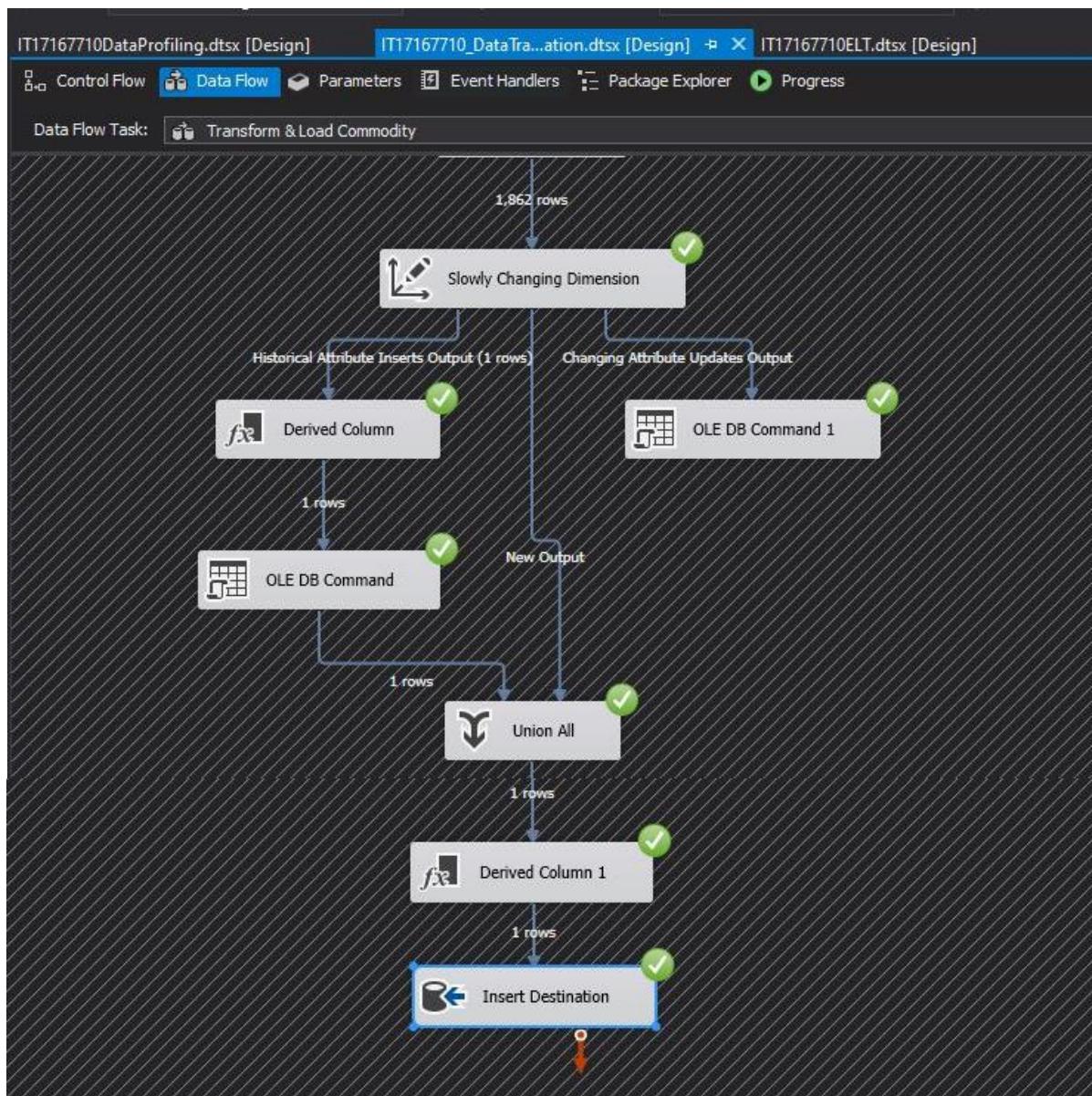
So, I updated the Product name as Dark Bookcase

```
SQLQuery5.sql - D...OG552M\94710 (75)* ➔ X
USE [IT17167710_DataWarehouse]
GO
UPDATE dbo.DimCommodity
SET Product_Name= 'Dark Bookcase'
WHERE ProductID=1;
```

82 % ▶ Messages

(1 row affected)

Completion time: 2020-04-29T03:27:52.5133035+05:30



```

use [IT17167710_DataWarehouse]
go
select *
from dbo.DimCommodity c
where c.ProductNumber='FUR-BO-10000112'
  
```

	ProductID	AlternateProductID	Product_Name	ProductNumber	ProductSubCategoryID	Start Date	End Date
1	1	1	Dark Bookcase	FUR-BO-10000112	1	2020-04-26 12:07:27.000	2020-04-29 03:28:51.000
2	1863	1	Bush Birmingham Collection Bookcase, Dark Cherry	FUR-BO-10000112	1	2020-04-29 03:28:51.000	NULL

A new record is created with the changed data values and this new record becomes the current record

the current record is closed

Summary Report

Cycle No	No of test cases	Passed	Failed	comment
1	27	26	1	This test case failed because Sales Region Representative dimension doesn't have any historical attributes, I only assigned changing attributes.
1	27	27	0	Above mention Defect solved

References

<https://www.guru99.com/ultimate-guide-etl-datawarehouse-testing.html>

<https://www.datagaps.com/concepts/etl-testing/>

<https://www.tutorialgateway.org/ssis-slowly-changing-dimension-type-2/>

<https://www.thinksys.com/qa-testing/entry-exit-criteria/>