



**Data Warehousing & Business Intelligent
(IT)
3rd Year, 1st Semester**

Assignment 1

Test Plan And Test Cases

Submitted to
Sri Lanka Institute of Information Technology

Bachelor of Science Special Honors Degree in Data Science

IT17167710

D.M.J.Prathapa

Weekday Batch

Test plan

Scope	<p>1. <u>Data Completeness Testing</u></p> <p>Test whether the 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> <tr> <th>Task</th><th>Estimate effort</th></tr> <tr> <td>Create the Test plan</td><td>3 man-hours</td></tr> <tr> <td>Create Test Cases</td><td>4 man-hours</td></tr> <tr> <td>Perform Test Execution</td><td>3 man-hours</td></tr> <tr> <td>Test Summary</td><td>2 man-hours</td></tr> <tr> <td>Total</td><td>12 man-hours</td></tr> </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 												
Terms/Acronyms	<ul style="list-style-type: none"> - ETL = (extract, transform, load) - DW = Data warehouse 												

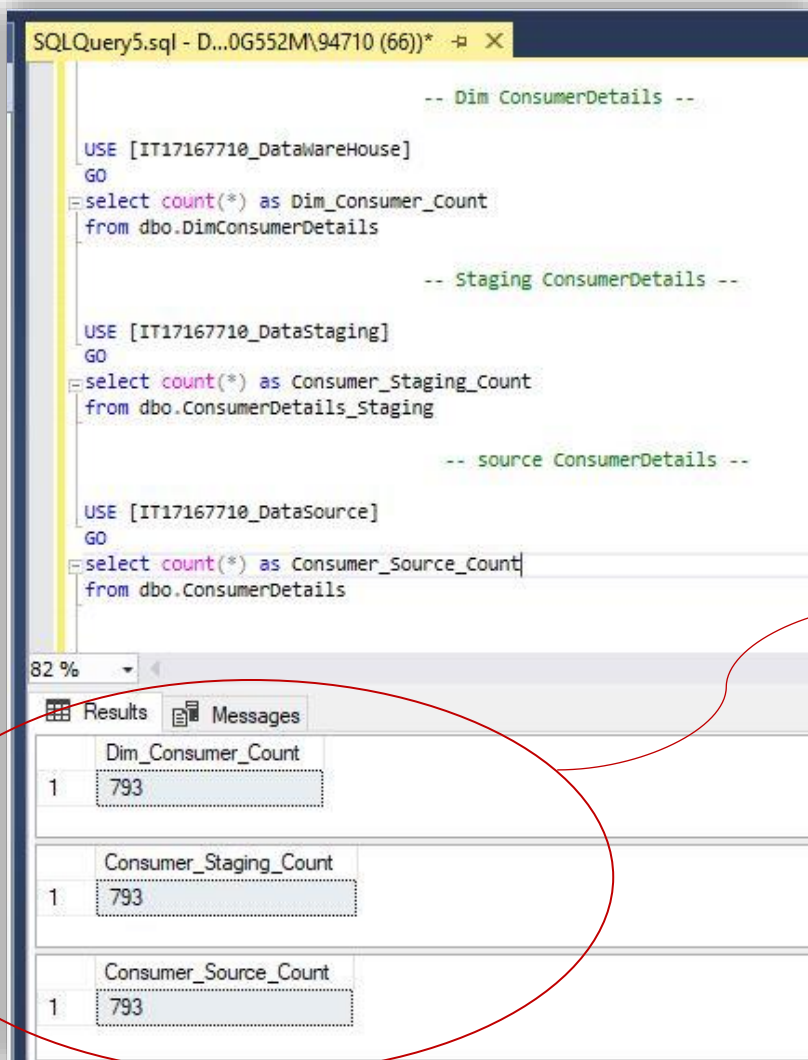
Data Completeness Testing

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 ofSub 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
```



SQLQuery5.sql - D:\...0G552M\94710 (66)*

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

	Dim_Consumer_Count
1	793

	Consumer_Staging_Count
1	793

	Consumer_Source_Count
1	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))*

```

----- 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

```

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
```

The screenshot shows a SQL query window titled 'SQLQuery5.sql - D:\0G552M\94710 (59))' with the following SQL code:

```
-- 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
```

The query was executed successfully, and the results are displayed in the 'Results' pane. The results show three rows, each with a single value of 9994:

	Fact_Table_OrderCount
1	9994

	Staging_OrderCount
1	9994

	Source_OrderCount
1	9994

A status bar at the bottom indicates '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

-- Ssource Sub_Commodity_Catagory --

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

The screenshot shows a SQL query window titled 'SQLQuery5.sql - D:\0G552M\94710 (59))' with the following SQL code:

```
-- 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

-- Ssource Sub_Commodity_Catagory --

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

The query was executed successfully, and the results are displayed in the 'Results' pane. The results show three rows, each with a single value of 17:

	Dim_Sub_Commodity_Catagorye_Count
1	17

	Staging_Sub_Commodity_Catagory_Count
1	17

	Source_Sub_Commodity_Catagory_Count
1	17

A status bar at the bottom indicates 'Query executed successfully.'

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

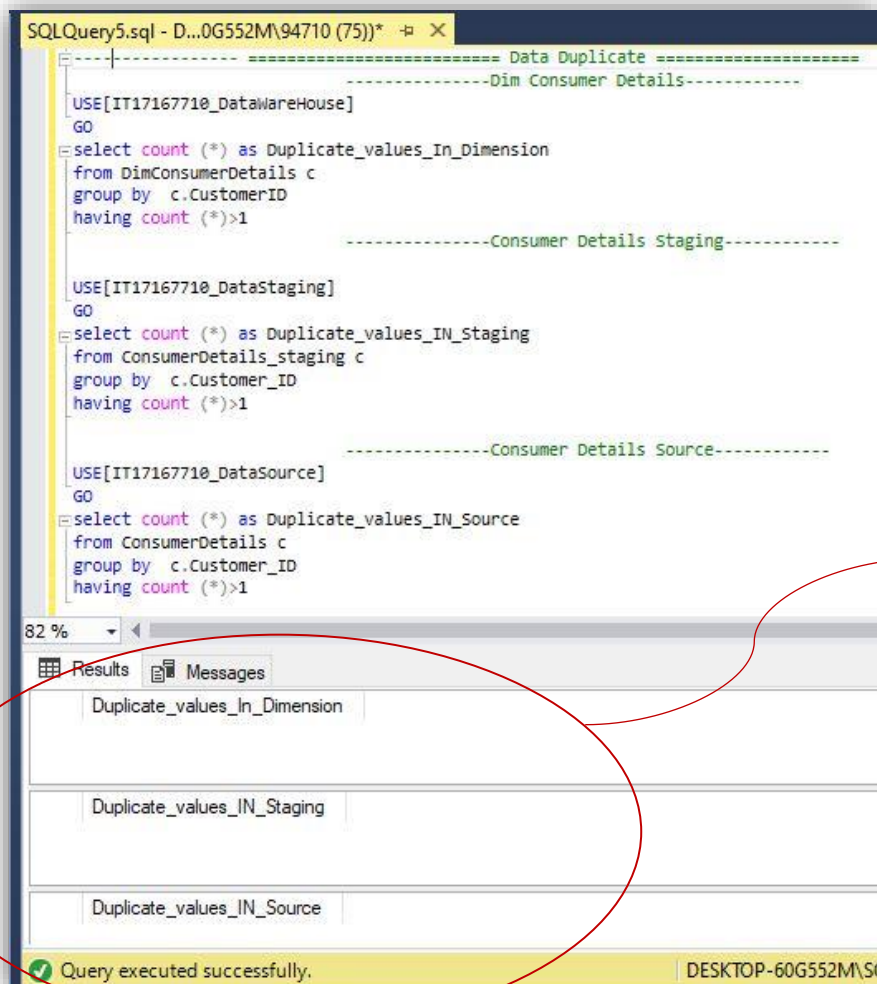
No	Action	Expected Output	Actual Output	SQL Queries and Screen shots	Test Result	Test Comments
1	test whether the duplicate values are available in Consumer source table, staging table and target Dimension table	Duplicate Data count of Consumer Source/Staging And Dim Table Should be 0	Same as Expected 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
```



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

```

82 %

Results Messages

Duplicate_values_In_Commodity_Dimension
Duplicate_values_IN_Commodity_Staging
Duplicate_values_IN_Commodity_Source

Query executed successfully. DESKTOP-60G552M\S

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

- figure 2.3

```

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

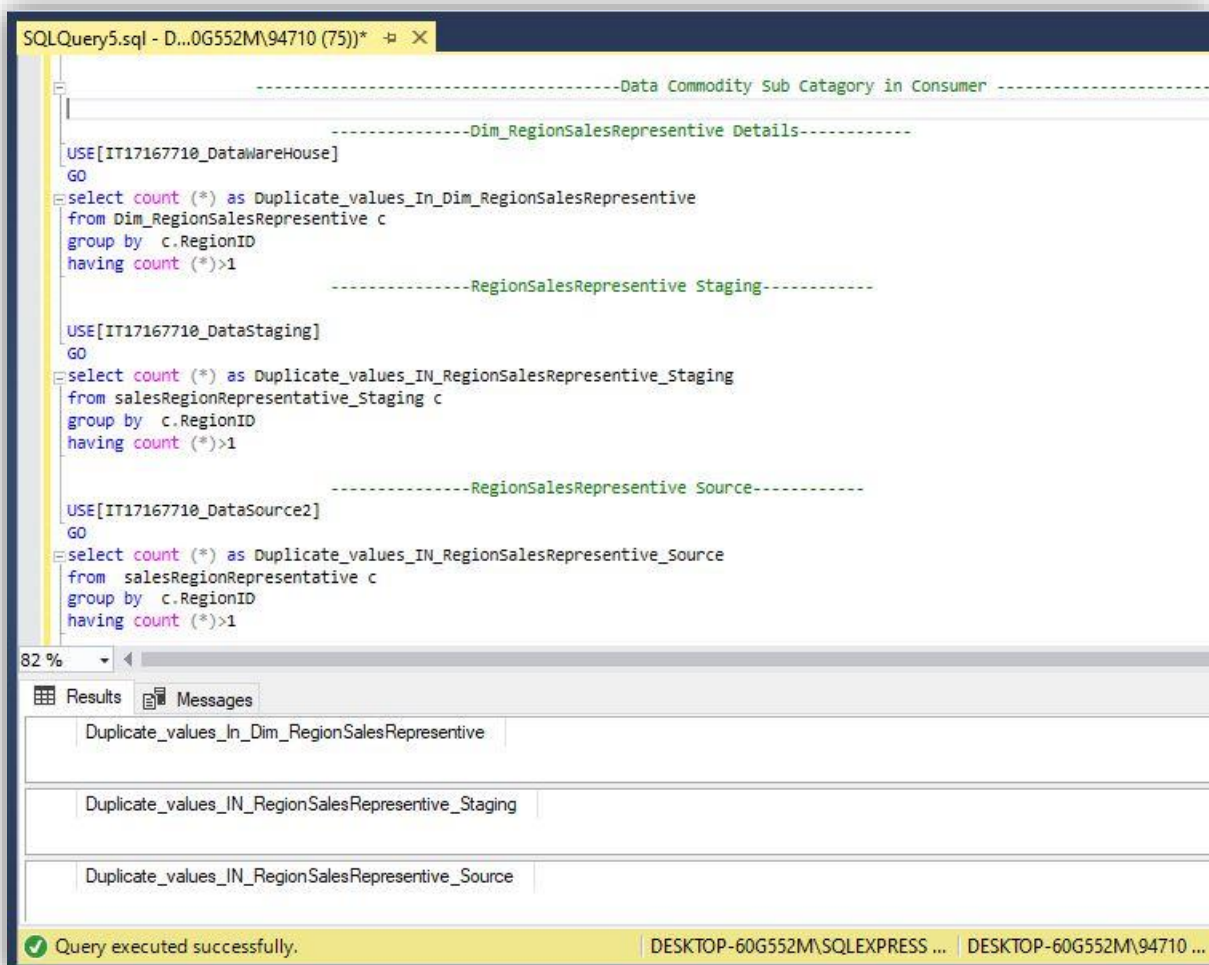
-----RegionSalesRepresentative Staging-----

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

-----RegionSalesRepresentative Source-----

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

```

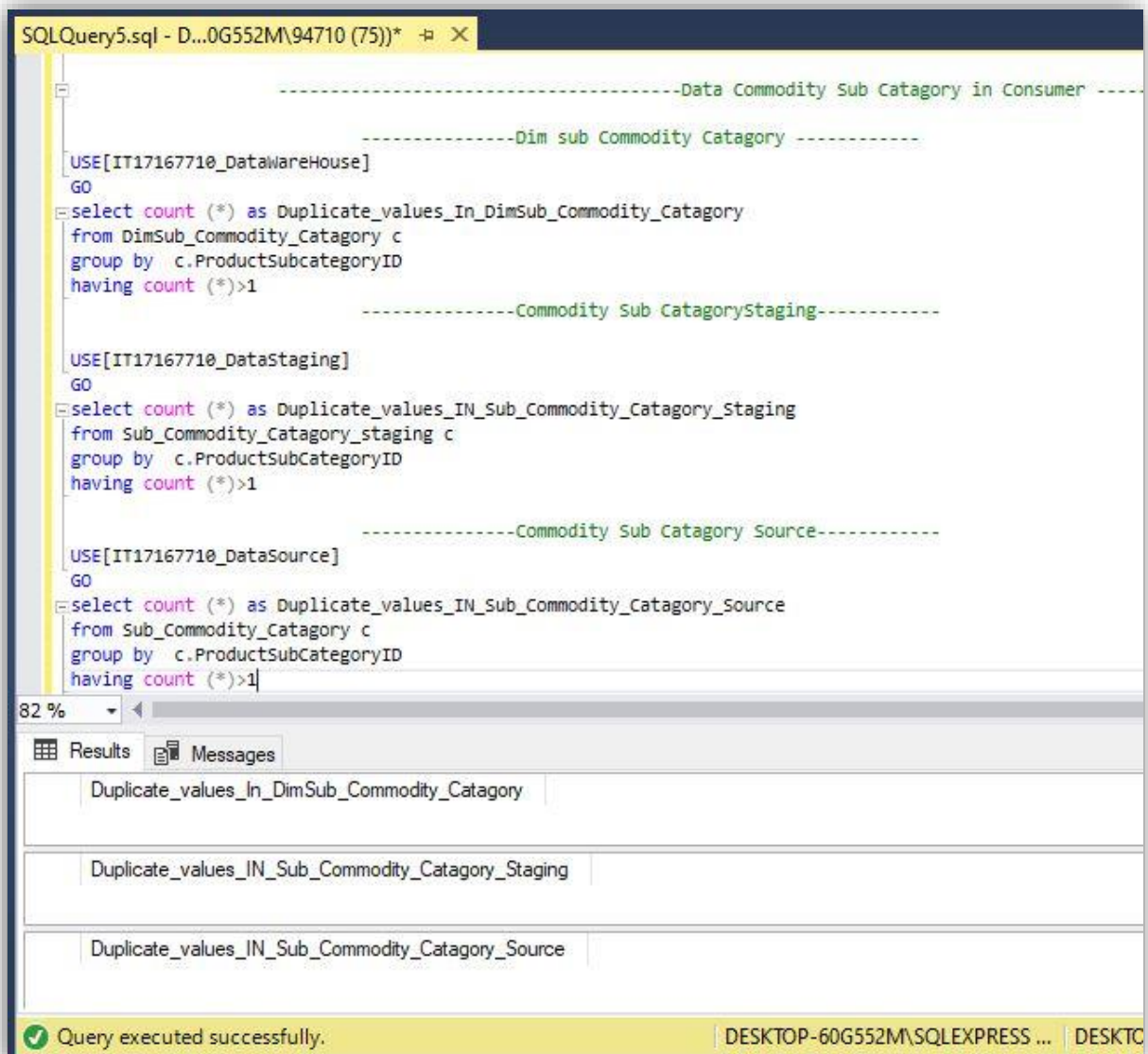


- figure 2.4

```
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
```



- figure 2.5

```

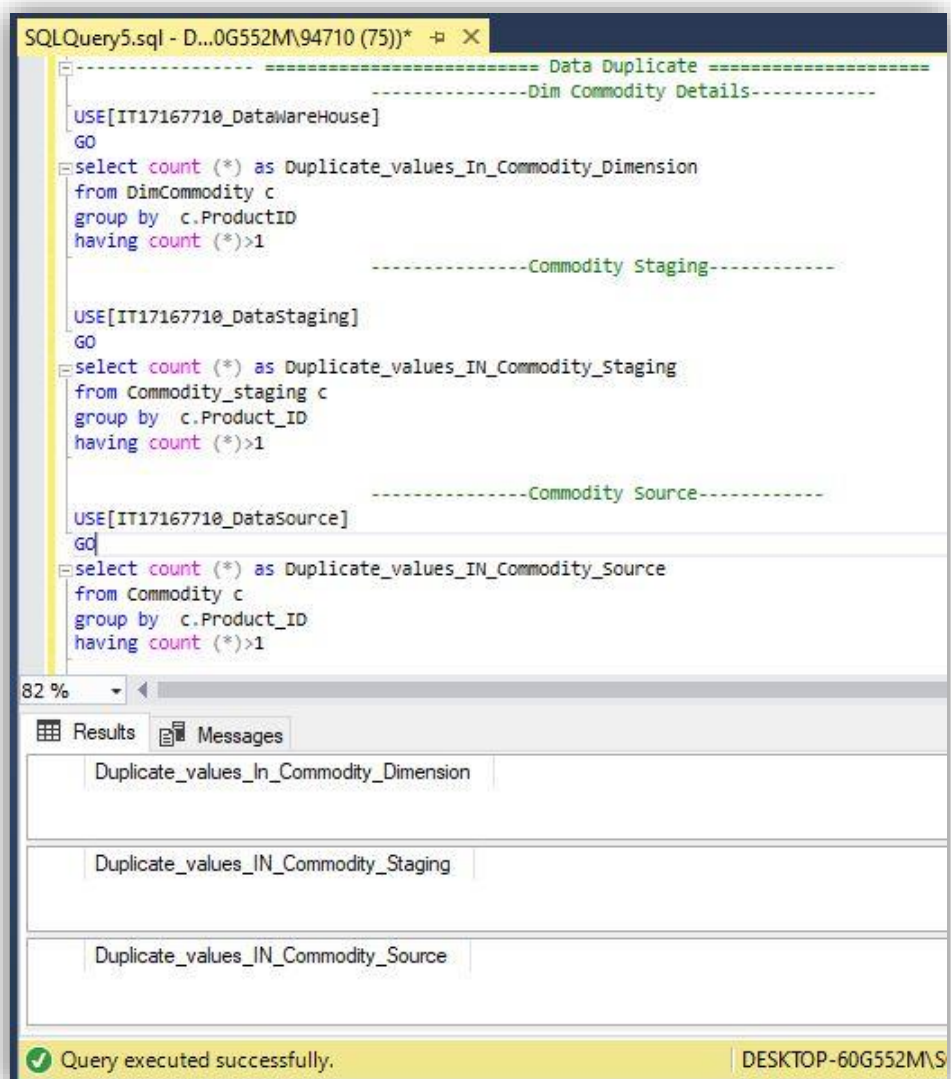
-----Dim Commodity Catagory-----
USE[IT17167710_DataWareHouse]
GO
select count (*) as Duplicate_values_In_Commodity_Catagory_Dimension
from DimCommodity_Catagory c
group by c.ProductCategoryID
having count (*)>1

----- 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

```



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:\OG552M\94710 (59)*

```
----- 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'
```

Results

	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. DESKTOP-60G552M\SQLEXPRESS ... DESKTOP-60G552M\94710 ... IT17167710.D

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))*

```

----- 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, IS_NULLABLE
from INFORMATION_SCHEMA.COLUMNS
where TABLE_NAME='DimCommodity'

```

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

75 %

Results Messages

	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))* X

```

----- 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 %

Results Messages

	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';
```

The screenshot displays three SQL queries executed in SQL Server Enterprise Manager. Each query calculates the data length (in bytes) for the first name, last name, and segment of a consumer record. The results are shown in a table format below each query.

Query 1: Dim Consumer Details (DataWareHouse)

	Fname_length	Lname_length	Segment_length
1	12	12	16

Query 2: Dim Consumer Details (DataStaging)

	Fname_length	Lname_length	Segment_length
1	12	12	16

Query 3: Source Consumer Details (DataSource)

	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))*

```

                                ---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 ProductSubCat
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 ProductSubCat
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 ProductSubCat
From Commodity c
where c.ProductNumber='FUR-BO-10000112' ;

```

82 %

Results Messages

	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

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

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

- 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)* - X

```

---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 ProductSubCat
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 ProductSubCat
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 ProductSubCat
From Commodity c
where c.ProductNumber='FUR-BO-10000112' ;

```

82 %

Results Messages

	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))*

```

---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 %

Results Messages

	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.ProductCategoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCategoryID=1;

```

SQLQuery5.sql - D:\0G552M\94710 (75))* -a 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=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.ProductCategoryID) as CommodityID_length
From Commodity_Catagory c
where c.ProductCategoryID=1;

```

82 %

Results Messages

	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	Ensure that the proper constraints defined on the 'Dim Commodity Catagory' table	All the constraints Should be defined on the 'Dim Commodity Category' 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 3.3.5	Pass	According to the output the 'Dim Commodity Category 'Table constraints are defined as expected
6	Ensure that the proper constraints defined on the 'Dim Sub Commodity Catagory' table	All the constraints Should be defined on the 'Dim Commodity Category' 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 3.3.5	Pass	According to the output the 'Dim Commodity Category 'Table constraints are defined as expected

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 Server Enterprise Manager window titled "SQLQuery5.sql - D...0G552M\94710 (75))". The query editor displays the same SQL query as in figure 3.3.1. Below the query editor, the "Results" tab is active, showing a table with 5 rows and 4 columns: table_name, constraint_name, column_name, and constraint_type. The first row is highlighted, showing "SalesFact" as the table name and "FK_SalesFact_DimCommodity" as the constraint name.

	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_SalesRepresentative	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 displays a SQL query in the query editor, which has been executed. The query selects table constraints and key column usage for the 'DimRegionSalesRepresentative' table. The results pane shows a single row of data.

```
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 = 'DimRegionSalesRepresentative'
    AND a.table_name = b.table_name
    AND a.table_schema = b.table_schema
    AND a.constraint_name = b.constraint_name;
```

	table_name	constraint_name	column_name	constraint_type
1	DimRegionSalesRepresentative	PK_DimRegionSalesRepresentative	RegionID	PRIMARY KEY

- figure 3.3.3

The screenshot displays a SQL query in the query editor, which has been executed. The query selects table constraints and key column usage for the 'DimCommodity' table. The results pane shows two rows of data.

```
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;
```

	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

The screenshot shows a SQL query in the Enterprise Manager interface. The query selects table constraints for the 'DimCommodity_Catagory' table. The results pane shows one result row.

```

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;

```

	table_name	constraint_name	column_name	constraint_type
1	DimCommodity_Catagory	PK__DimCommo__3224ECEE4BA1AFD8	ProductCategoryID	PRIMARY KEY

- figure 3.3.5

The screenshot shows a SQL query in the Enterprise Manager interface. The query selects table constraints for the 'DimSub_Commodity_Catagory' table. The results pane shows two result rows.

```

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;

```

	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</p> <p>the current record Should be closed</p> <p>A new record Should be created with the changed data values and this new record Should become the current record</p>	Same as Expected	Screen shots And SQL queries for the particular section is attached as - figure 4.3	Fail	<p>This test case failed because Sales Region Representative dimension doesn't have any historical attributes, I only assigned changing attributes.</p> <p>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
```

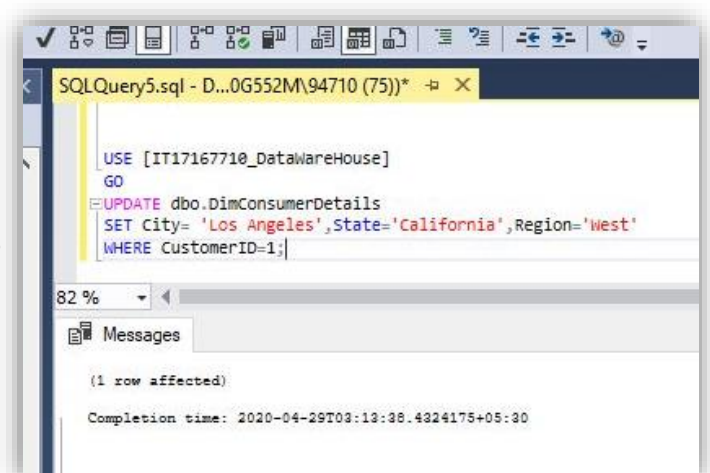
	CustomerID	Customer_AlternateID	FirstName	LastName	Segment	City	State	PostalCode	Country	Region	StartDate	EndDate
1	1	AA-10315	Alex	Avila	Consumer	Minneapolis	Minnesota	55407	United States	Central	2020-04-26 09:38:15.000	NULL
2	2	AA-10375	Allen	Arnold	Consumer	Mesa	Arizona	85204	United States	West	2020-04-26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

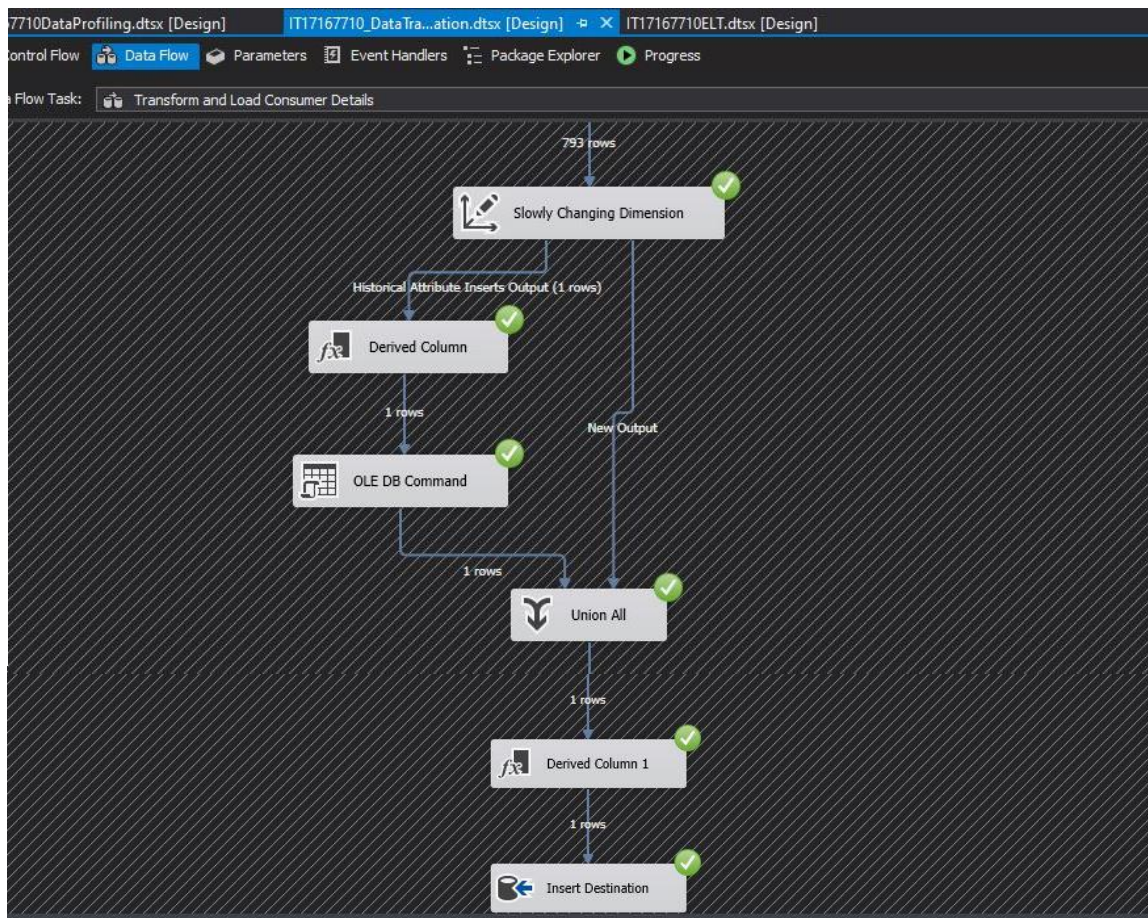
Query executed successfully. DESKTOP-60G552M\SQLEXPRESS ... DESKTOP-60G552M\94710 ... IT17167710_DataWarehouse 00:00:00 793 rows

Assume that Alex Avila 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;
```





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';
  
```

82 %

	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 %

	ProductID	AlternateProductID	Product_Name	ProductNumber	ProductSubCategoryID	StartDate	EndDate
1	1	1	Bush Birmingham Collection Bookcase, Dark Cherry	FUR-BO-10000112	1	2020-04-26 12:07:27.000	NULL
2	2	28	Sauder Camden County Collection Libraries, Plank...	FUR-BO-10003159	1	2020-04-26 12:07:27.000	NULL
3	3	29	O'Sullivan Living Dimensions 5-Shelf Bookcases	FUR-BO-10003272	1	2020-04-26 12:07:27.000	NULL
4	4	30	Global Adaptalites 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...0G552M\94710 (75))*

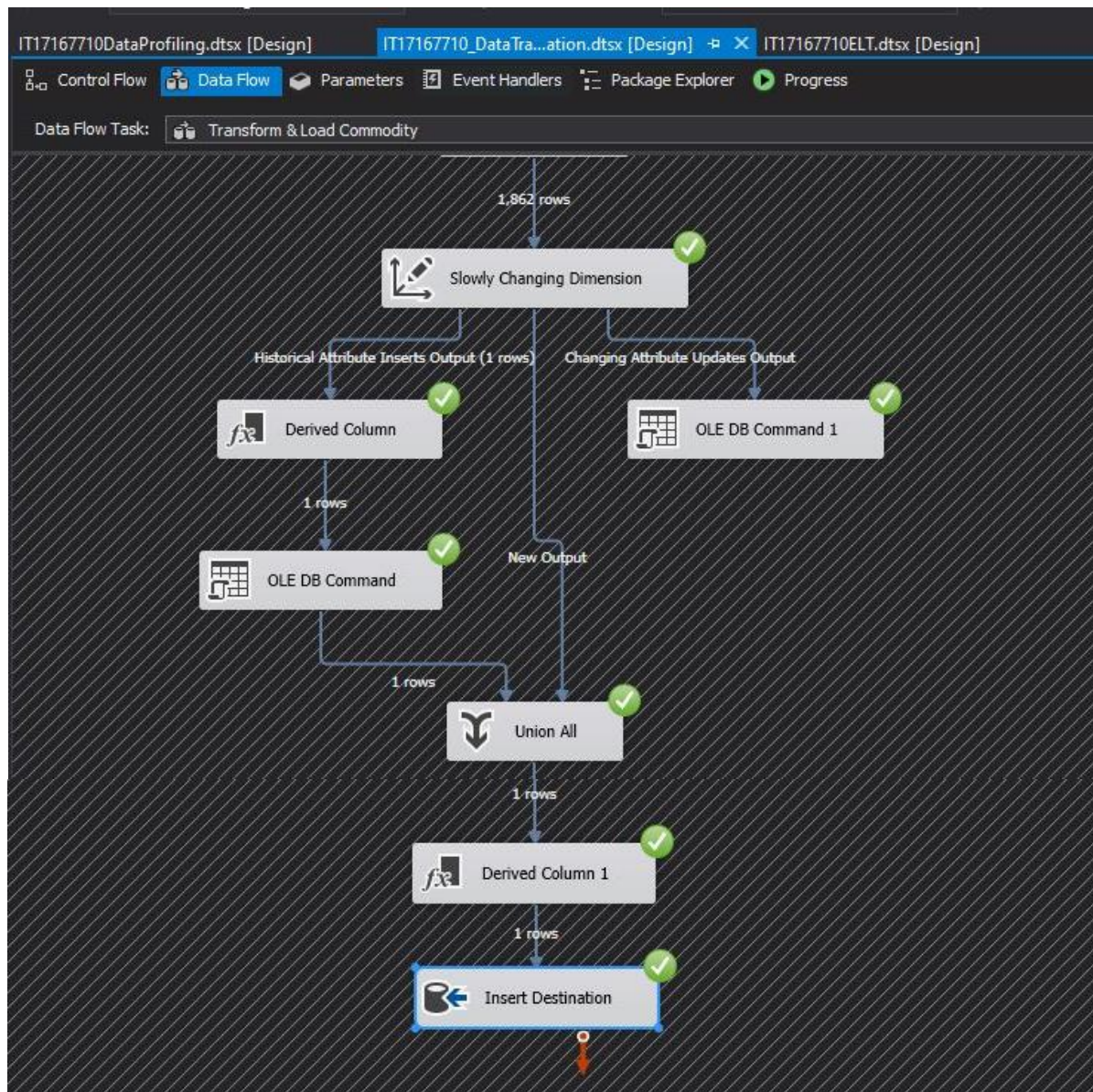
```
USE [IT17167710_DataWarehouse]
GO
UPDATE dbo.DimCommodity
SET Product_Name= 'Dark Bookcase'
WHERE ProductID=1;
```

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	StartDate	EndDate
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