SRI LANKA INSTITUE OF INFORMATION TECHNOLOGY

ID No:	**IT20237486**
Name:	**Gunarathne H.M.Y.B**
Batch:	**DS weekday**
Assignment:	**01**

Table of Contents

Contents

01). Data set selection

Data set	:	Supermarket sales
Source	:	Kaggle
Link to the source:		https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales

The dataset contains the supermarket sales information. The data set consists of five files: three csv file, one excel file and one text files (Necessary modifications has been done in order to meet the requirements).

ER Diagram

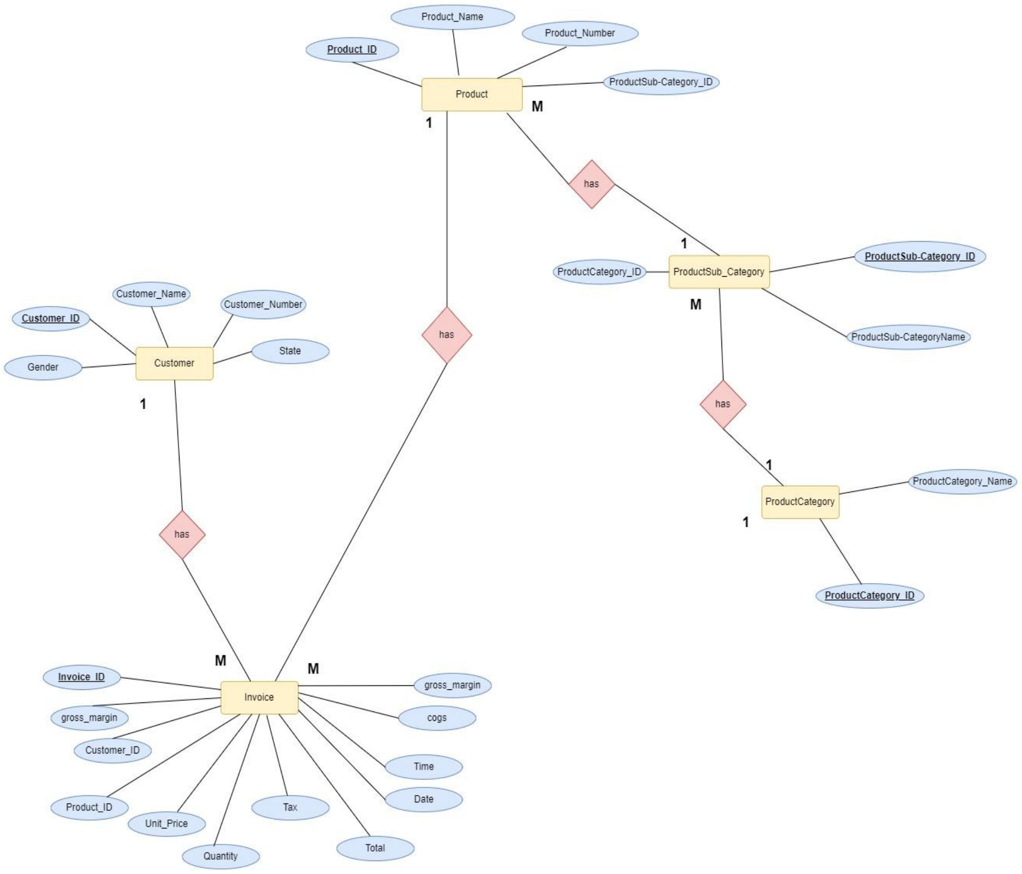


Figure 1.0-ER diagram

02).Preparation of Data Sources

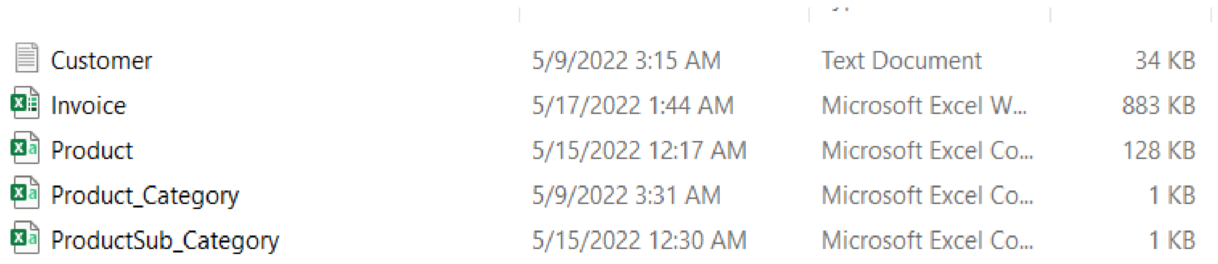
The dataset was originally in the form of one excel file. Data in the file has been separated into 5 different files of type Excel,CSV and text.

Table	File type
Product	Csv file(.csv)
Product Category	Csv file(.csv)
Product Sub Category	Csv file(.csv)
Invoice	Excel file(.xlsx)
Customer	Text file(.txt)

Figure 1.1-sample source table creation

Similarly other tables has also been created and then the tables has been exported in relevant file types.

Final set of Sources:



Customer	5/9/2022 3:15 AM	Text Document	34 KB
Invoice	5/17/2022 1:44 AM	Microsoft Excel W...	883 KB
Product	5/15/2022 12:17 AM	Microsoft Excel Co...	128 KB
Product_Category	5/9/2022 3:31 AM	Microsoft Excel Co...	1 KB
ProductSub_Category	5/15/2022 12:30 AM	Microsoft Excel Co...	1 KB

03). Solution architecture

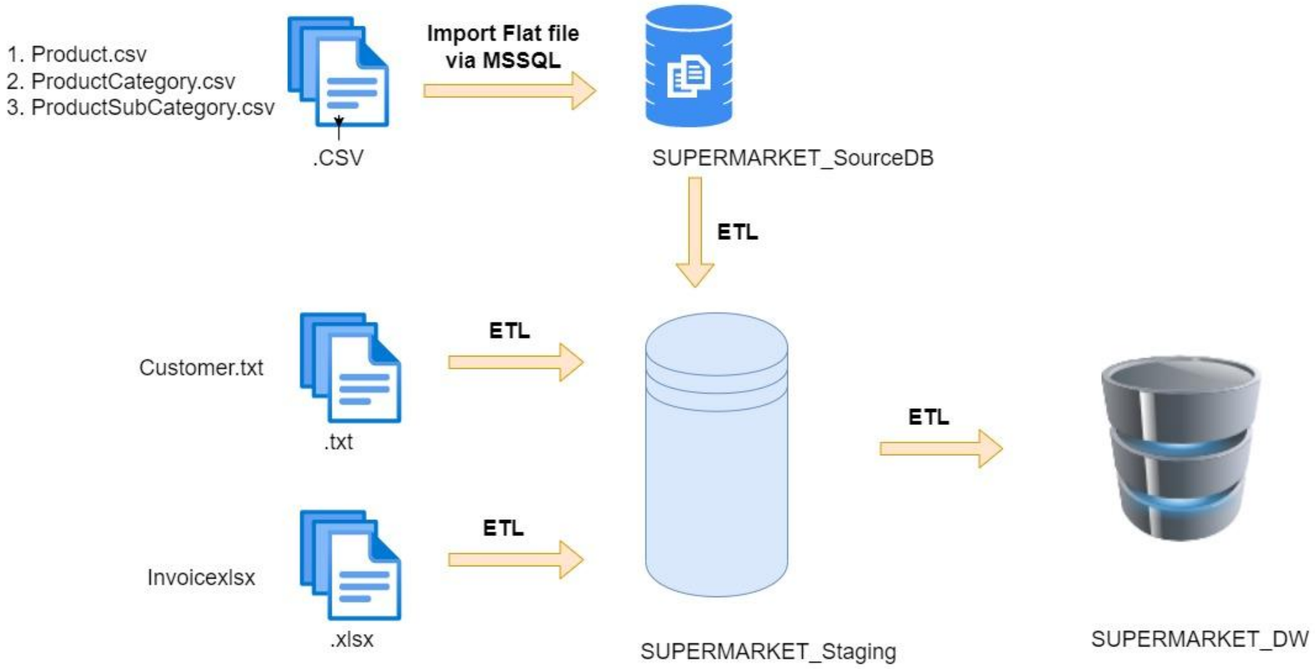
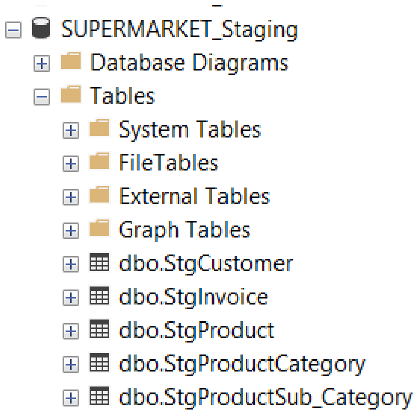


Figure 2.0-Solution architecture

As can be seen in the figure 4 different resource types has been used to extract data to staging. Staging layer has been used to have all the tables in a single location as in the below figure.



```
⊟ SUPERMARKET_Staging
  ⊞ Database Diagrams
  ⊟ Tables
    ⊞ System Tables
    ⊞ FileTables
    ⊞ External Tables
    ⊞ Graph Tables
    ⊞ dbo.StgCustomer
    ⊞ dbo.StgInvoice
    ⊞ dbo.StgProduct
    ⊞ dbo.StgProductCategory
    ⊞ dbo.StgProductSub_Category
```

Figure 3.0-Staging

The tables at the staging are then profiled and after performing a rich set of ETL tasks, data is loaded to the data warehouse where from that several reporting tools and analyzing tools can use data for reporting mining and analyzing.

04). Data warehouse design & development

The data warehouse is designed as a snowflake schema with one fact table and five-dimension table.

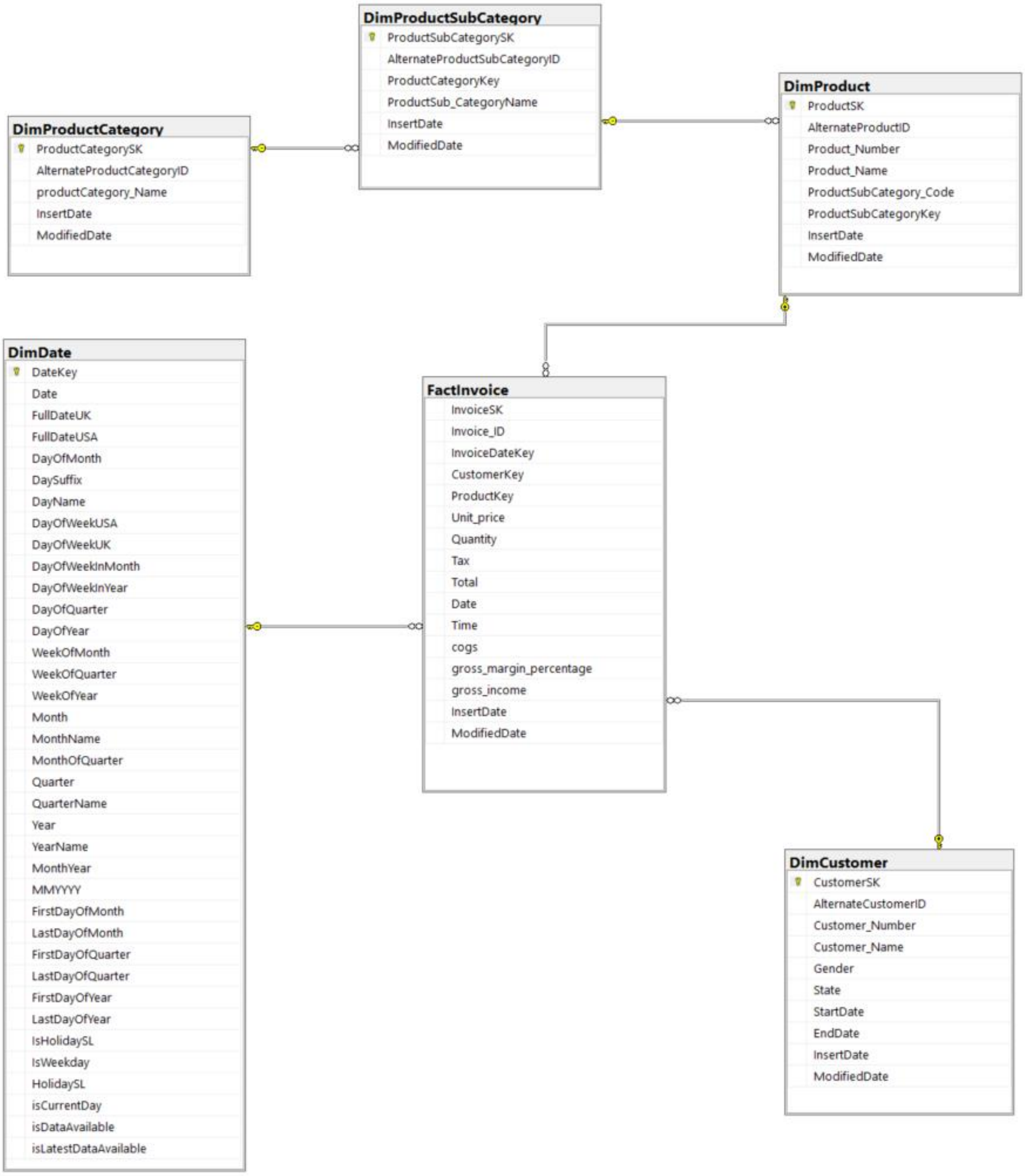


Figure 4.0-Datawarehouse design

05). ETL Development

As the first step data has been extracted from sources to staging area. Data flow task has been used for every extraction.

Data Extraction

5.1 Product Subcategory Data from Source to Staging

5.1.1 Data Flow

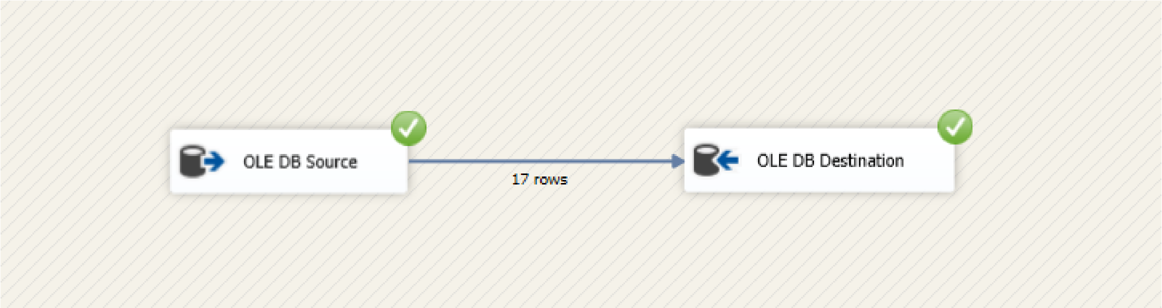
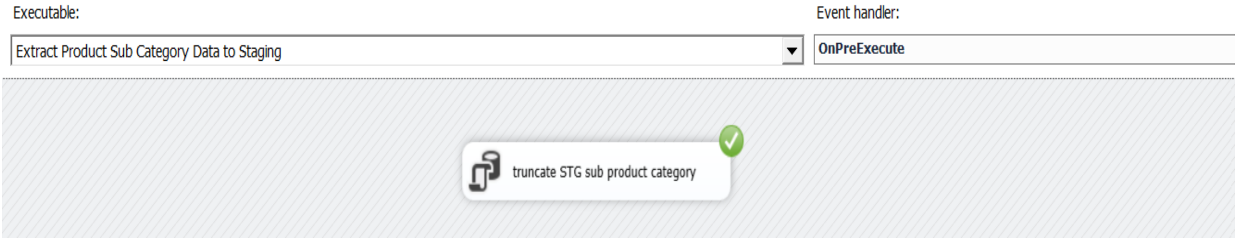


Figure 5.1.1-Product Subcategory data flow

5.1.2 Event handler

Before executing ‘Extract Product Subcategory to Staging’ existing data in the staging layer has been truncated.



Executable: Extract Product Sub Category Data to Staging

Event handler: OnPreExecute

Figure 5.1.2.-Product Subcategory Event handler

5.2 Invoice Data from Source to Staging

5.2.1 Data Flow

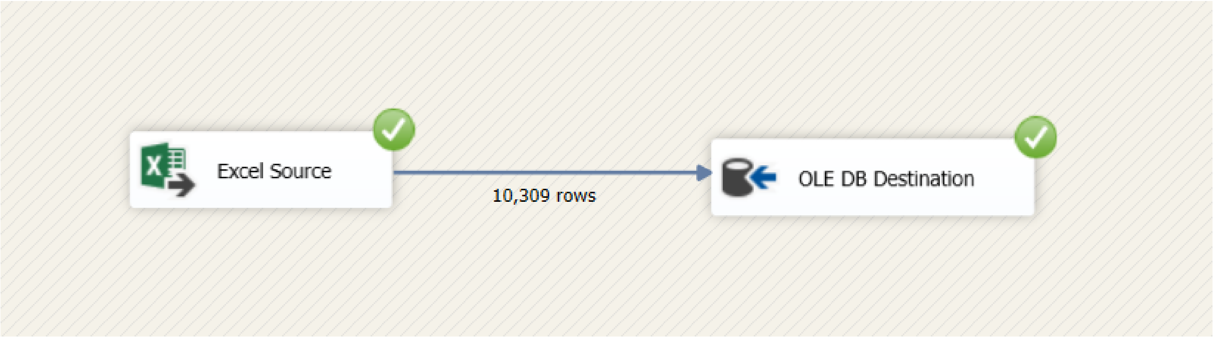
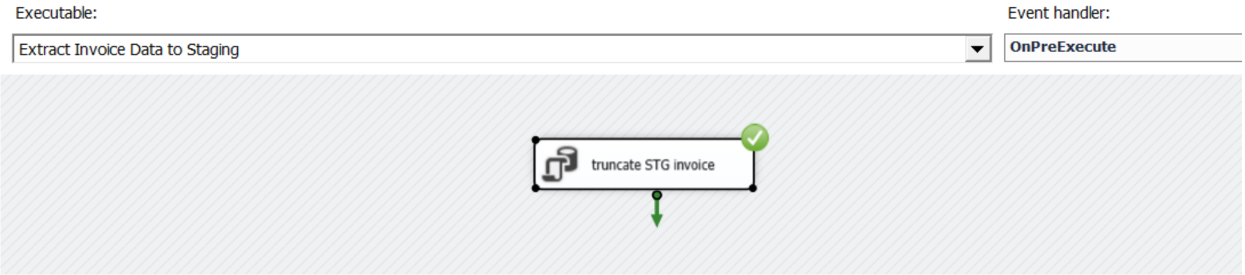


Figure 5.2.1-Invoice data flow

5.2.2 Event handler

Before executing 'Extract Invoice to Staging' existing data in the staging layer has been truncated.


Executable: Extract Invoice Data to Staging

Event handler: OnPreExecute

Figure 5.2.2.-Invoice Event handler

5.3 Customer Data from Source to Staging

5.3.1 Data Flow

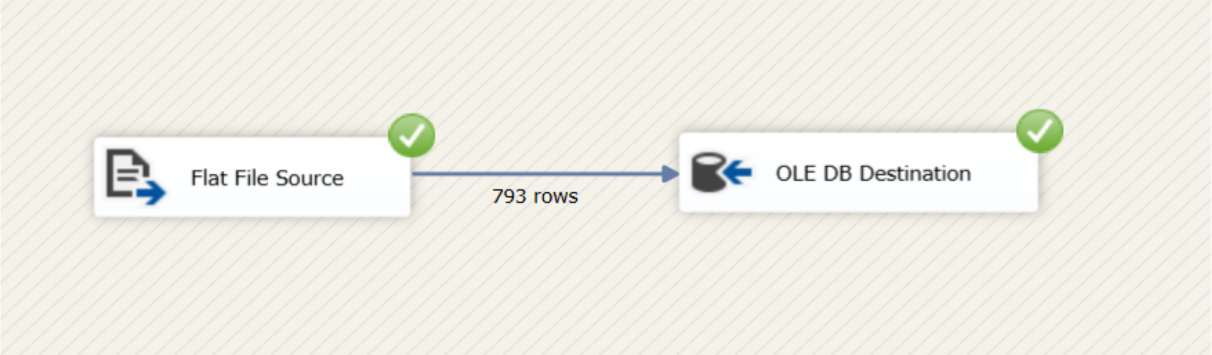
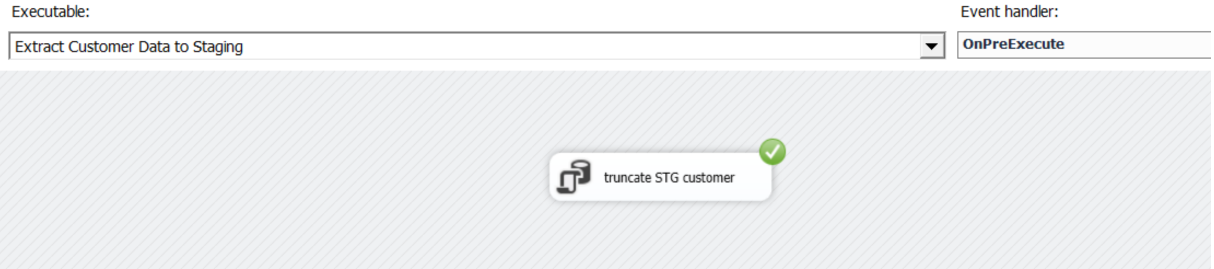


Figure 5.3.-Customer data flow

5.3.2 Event handler

Before executing ‘Extract Customer to Staging’ existing data in the staging layer has been truncated.


Executable: Extract Customer Data to Staging

Event handler: OnPreExecute

Figure 5.3.1.-Customer Event handler

Overall control flow

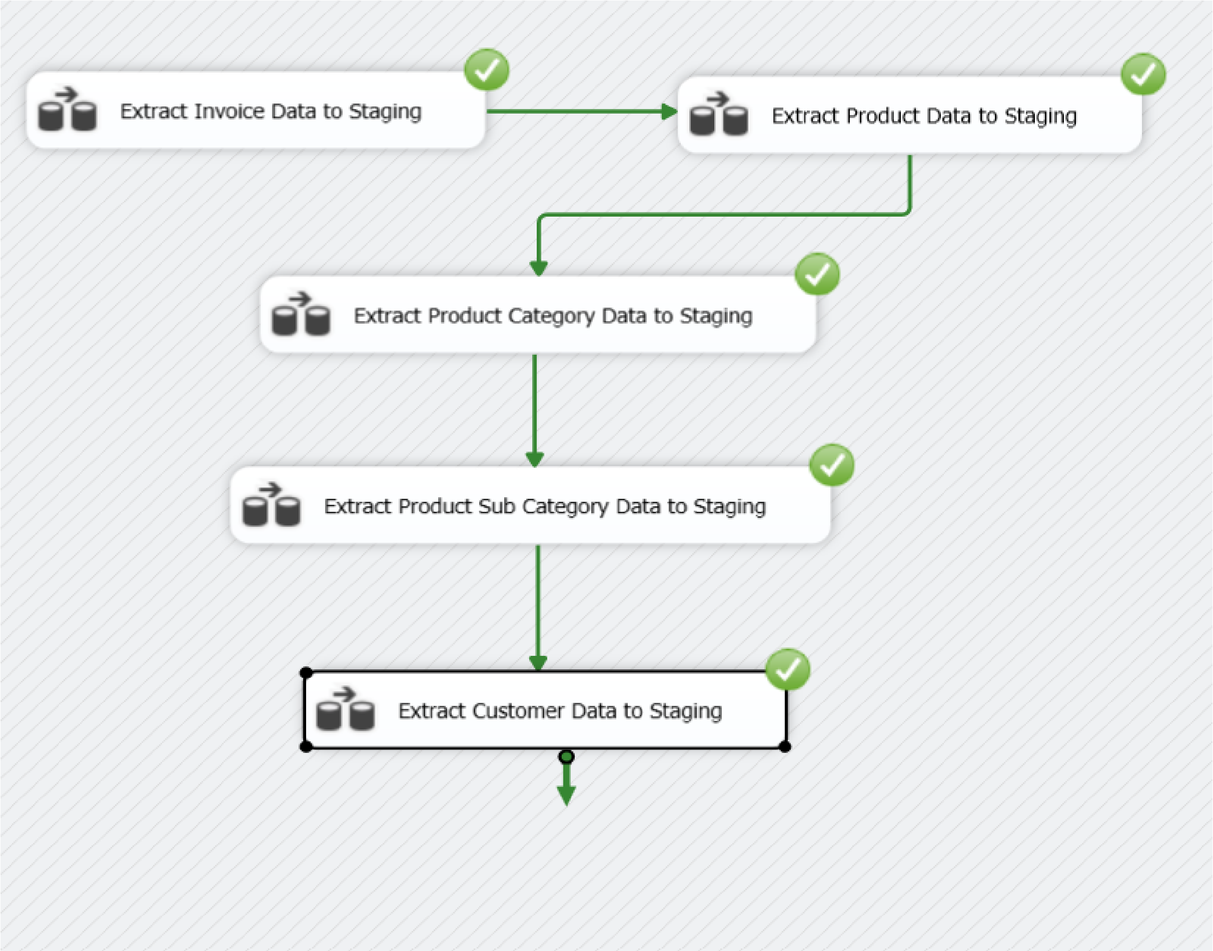

Extract Invoice Data to Staging
Extract Product Data to Staging
Extract Product Category Data to Staging
Extract Product Sub Category Data to Staging
Extract Customer Data to Staging

Figure 5.8.-source to staging control flow

Data Profiling

Before Loading staging tables to the data warehouse data must be enriched to obtain the most suitable data for analyzing. Data profiling has been done in order to identify what need to be corrected in ETL process in order to meet this requirement.

Each and every table at staging is profiled and stored in a specific file location.

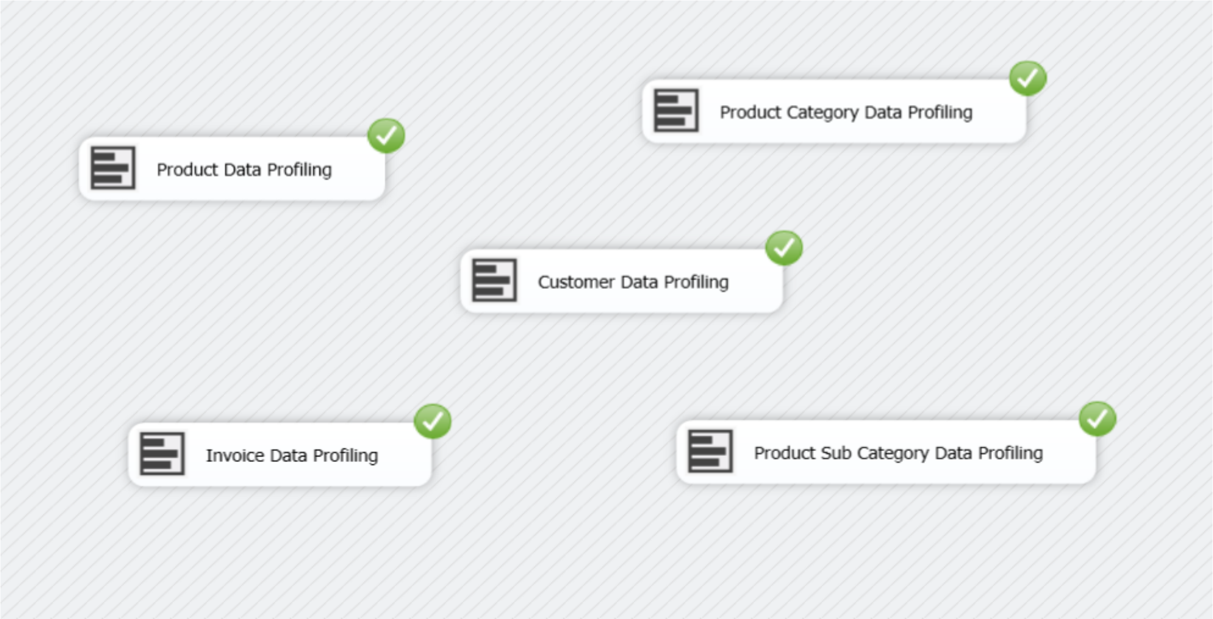
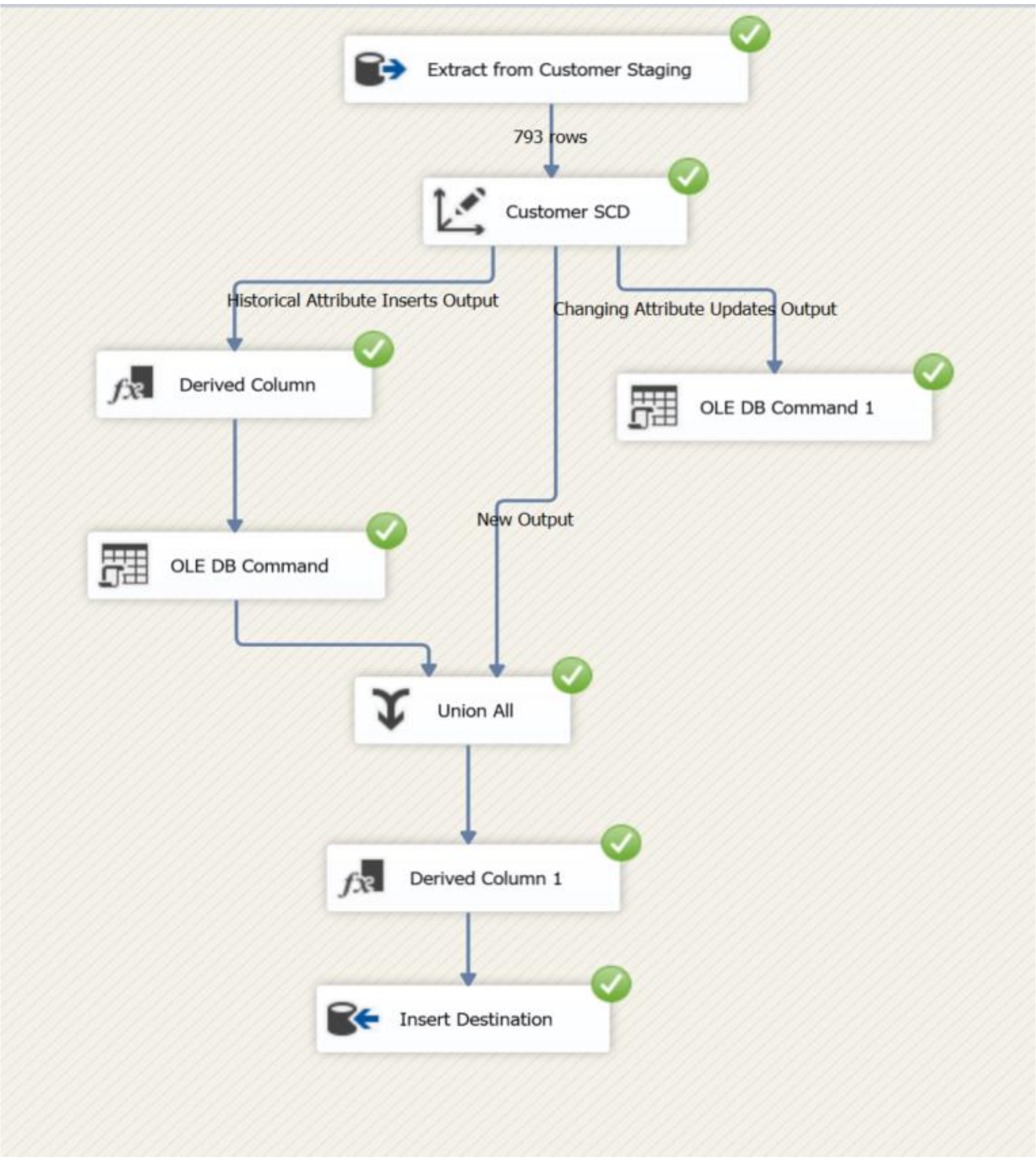


Figure 5.9.1-profiling diagram

Data Transforming and loading

b) Load Slowly changing Dimensions

5.11 Customer Data from Staging to Data Warehouse



5.12 Product Subcategory Data from Staging to Datawarehouse

Product Subcategory data has been loaded to DimProductSubcategory

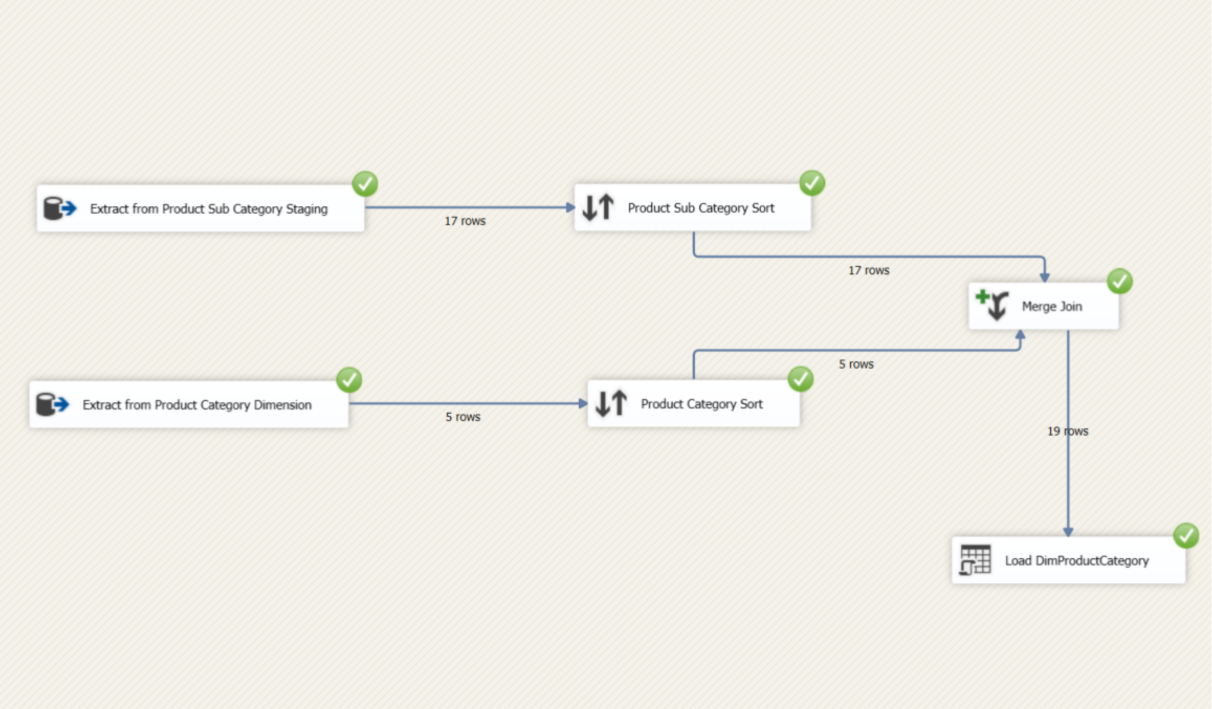
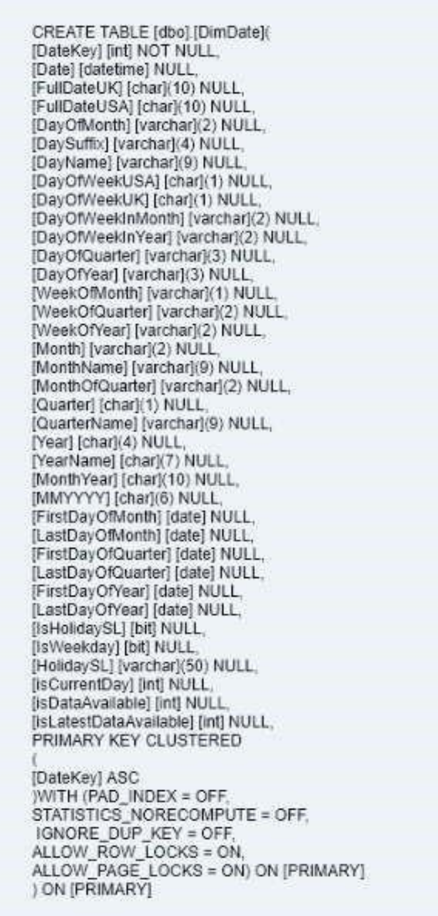

Extract from Product Sub Category Staging
17 rows
Product Sub Category Sort
17 rows
Merge Join
5 rows
Extract from Product Category Dimension
5 rows
Product Category Sort
19 rows
Load DimProductCategory

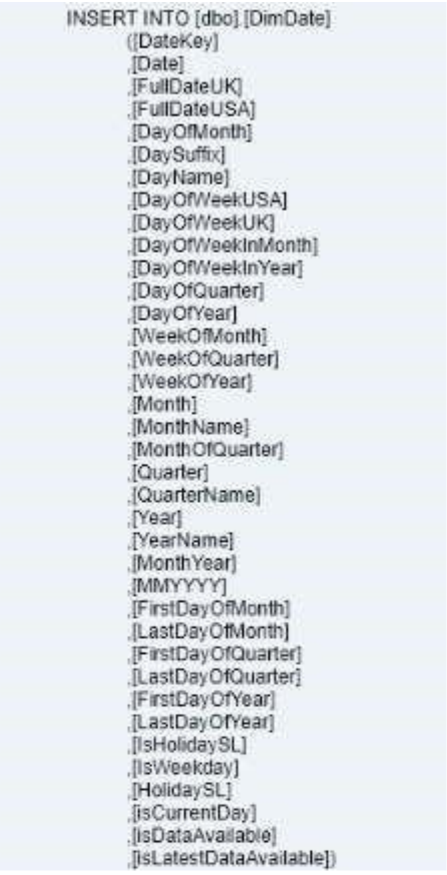
Figure 5.12.1-load to DimProductSubcategory

5.15 Creation of Date Dimension

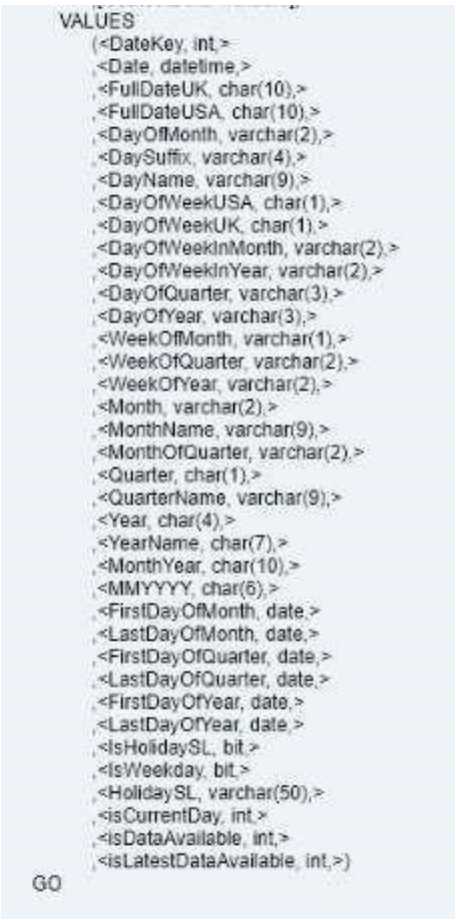


```
CREATE TABLE [dbo].[DimDate](
[DateKey] [int] NOT NULL,
[Date] [datetime] NULL,
[FullDateUK] [char](10) NULL,
[FullDateUSA] [char](10) NULL,
[DayOfMonth] [varchar](2) NULL,
[DaySuffix] [varchar](4) NULL,
[DayName] [varchar](9) NULL,
[DayOfWeekUSA] [char](1) NULL,
[DayOfWeekUK] [char](1) NULL,
[DayOfWeekInMonth] [varchar](2) NULL,
[DayOfWeekInYear] [varchar](2) NULL,
[DayOfQuarter] [varchar](3) NULL,
[DayOfYear] [varchar](3) NULL,
[WeekOfMonth] [varchar](1) NULL,
[WeekOfQuarter] [varchar](2) NULL,
[WeekOfYear] [varchar](2) NULL,
[Month] [varchar](2) NULL,
[MonthName] [varchar](9) NULL,
[MonthOfQuarter] [varchar](2) NULL,
[Quarter] [char](1) NULL,
[QuarterName] [varchar](9) NULL,
[Year] [char](4) NULL,
[YearName] [char](7) NULL,
[MonthYear] [char](10) NULL,
[MMYYYY] [char](6) NULL,
[FirstDayOfMonth] [date] NULL,
[LastDayOfMonth] [date] NULL,
[FirstDayOfQuarter] [date] NULL,
[LastDayOfQuarter] [date] NULL,
[FirstDayOfYear] [date] NULL,
[LastDayOfYear] [date] NULL,
[IsHolidaySL] [bit] NULL,
[IsWeekday] [bit] NULL,
[HolidaySL] [varchar](50) NULL,
[isCurrentDay] [int] NULL,
[isDataAvailable] [int] NULL,
[isLatestDataAvailable] [int] NULL,
PRIMARY KEY CLUSTERED
(
[DateKey] ASC
)WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF,
 IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Figure 5.14.1-Load to DimDate



```
INSERT INTO [dbo].[DimDate]
        ([DateKey]
        ,[Date]
        ,[FullDateUK]
        ,[FullDateUSA]
        ,[DayOfMonth]
        ,[DaySuffix]
        ,[DayName]
        ,[DayOfWeekUSA]
        ,[DayOfWeekUK]
        ,[DayOfWeekInMonth]
        ,[DayOfWeekInYear]
        ,[DayOfQuarter]
        ,[DayOfYear]
        ,[WeekOfMonth]
        ,[WeekOfQuarter]
        ,[WeekOfYear]
        ,[Month]
        ,[MonthName]
        ,[MonthOfQuarter]
        ,[Quarter]
        ,[QuarterName]
        ,[Year]
        ,[YearName]
        ,[MonthYear]
        ,[MMYYYY]
        ,[FirstDayOfMonth]
        ,[LastDayOfMonth]
        ,[FirstDayOfQuarter]
        ,[LastDayOfQuarter]
        ,[FirstDayOfYear]
        ,[LastDayOfYear]
        ,[IsHolidaySL]
        ,[IsWeekday]
        ,[HolidaySL]
        ,[isCurrentDay]
        ,[isDataAvailable]
        ,[isLatestDataAvailable])
```



```
  VALUES
        (<DateKey, int,>
        ,<Date, datetime,>
        ,<FullDateUK, char(10),>
        ,<FullDateUSA, char(10),>
        ,<DayOfMonth, varchar(2),>
        ,<DaySuffix, varchar(4),>
        ,<DayName, varchar(9),>
        ,<DayOfWeekUSA, char(1),>
        ,<DayOfWeekUK, char(1),>
        ,<DayOfWeekInMonth, varchar(2),>
        ,<DayOfWeekInYear, varchar(2),>
        ,<DayOfQuarter, varchar(3),>
        ,<DayOfYear, varchar(3),>
        ,<WeekOfMonth, varchar(1),>
        ,<WeekOfQuarter, varchar(2),>
        ,<WeekOfYear, varchar(2),>
        ,<Month, varchar(2),>
        ,<MonthName, varchar(9),>
        ,<MonthOfQuarter, varchar(2),>
        ,<Quarter, char(1),>
        ,<QuarterName, varchar(9),>
        ,<Year, char(4),>
        ,<YearName, char(7),>
        ,<MonthYear, char(10),>
        ,<MMYYYY, char(6),>
        ,<FirstDayOfMonth, date,>
        ,<LastDayOfMonth, date,>
        ,<FirstDayOfQuarter, date,>
        ,<LastDayOfQuarter, date,>
        ,<FirstDayOfYear, date,>
        ,<LastDayOfYear, date,>
        ,<IsHolidaySL, bit,>
        ,<IsWeekday, bit,>
        ,<HolidaySL, varchar(50),>
        ,<isCurrentDay, int,>
        ,<isDataAvailable, int,>
        ,<isLatestDataAvailable, int,>)
GO
```

Query used to create and load data to date dimension is listed above. Date dimension is assumed to tally with reservation status date in booking table

c)Load Fact Table

TransactionStaging table and some of the coulmns in BookingStaging table is merged in order to make the fact table. RoomStaging is loaded and merged to obtain RoomId. All required surrogate keys has been loaded to data warehouse after a lookup through alternate keys in dimension tables.

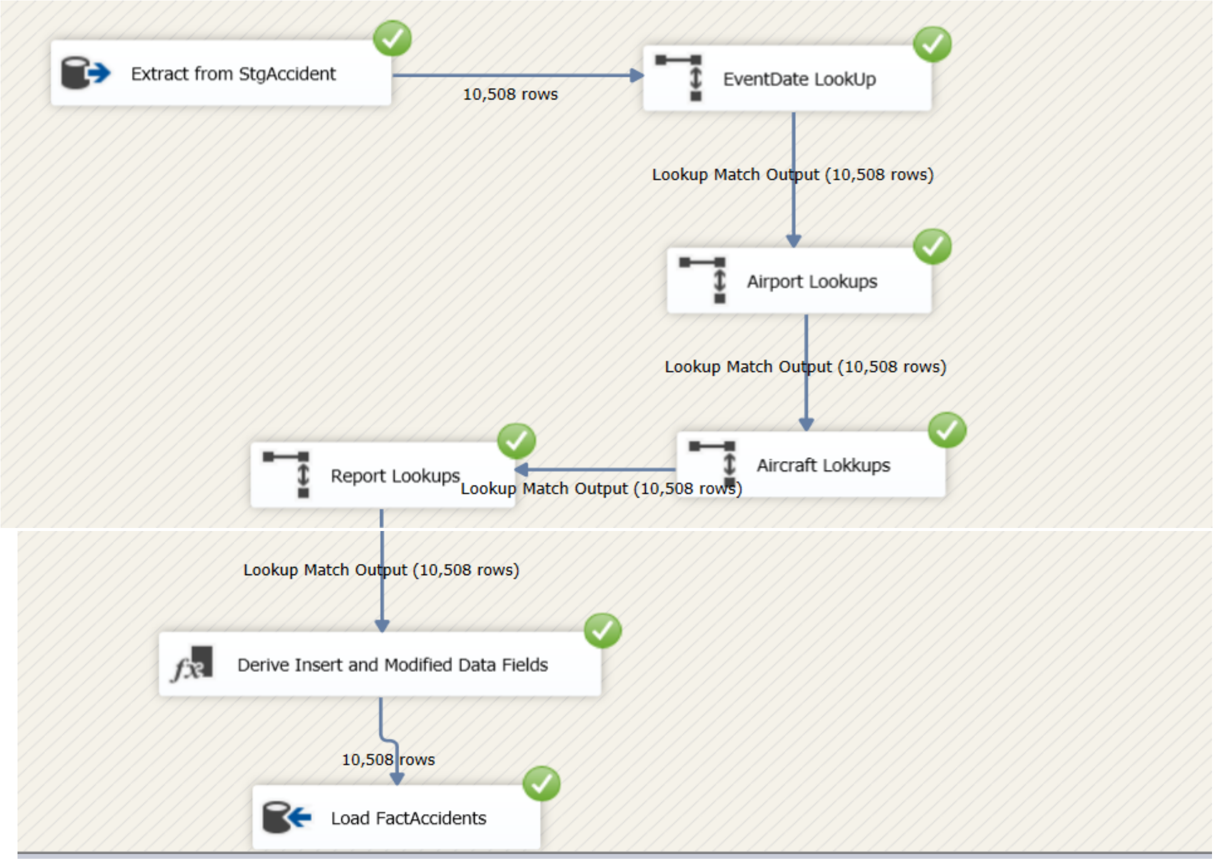

Extract from StgAccident
10,508 rows
EventDate LookUp
Lookup Match Output (10,508 rows)
Airport Lookups
Lookup Match Output (10,508 rows)
Aircraft Lokkups
Lookup Match Output (10,508 rows)
Report Lookups
Lookup Match Output (10,508 rows)
Derive Insert and Modified Data Fields
10,508 rows
Load FactAccidents

Figure 5.16.1-FactAccident ETL

Overall ETL Transformation

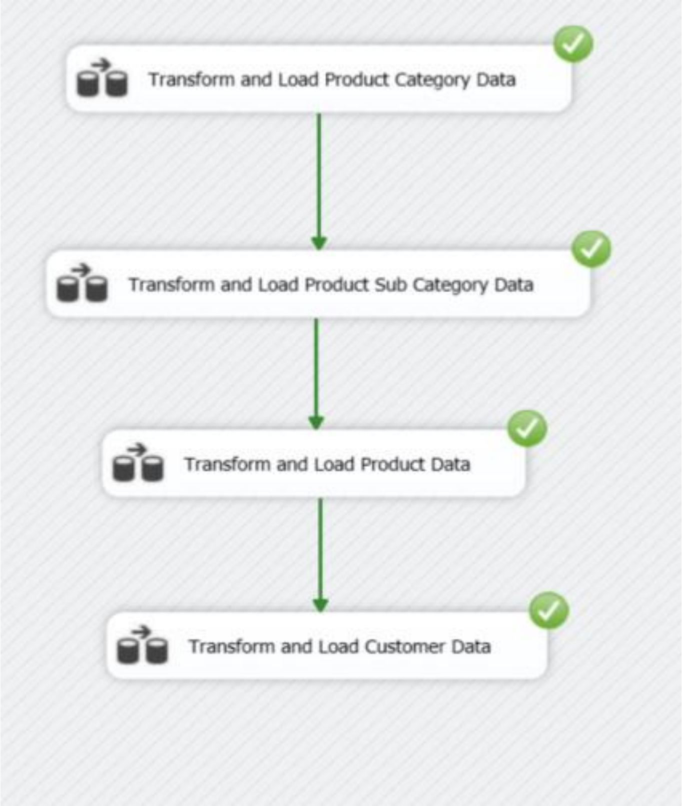

Transform and Load Product Category Data
Transform and Load Product Sub Category Data
Transform and Load Product Data
Transform and Load Customer Data

Figure 5.17.1-overall ETL to data warehouse

Thank you!