

# **IT3021 – Data Warehousing & Business Intelligence**

## **Assignment 01**

Submitted By:  
Botheju W.M.N  
IT20139230

# Step 1: Data set selection

I have selected a dataset which tracks the fictional telco company's customer churn based on a variety of possible factors including customer details, dependents, monthly charges, and many with information about the types of services each customer has.

Data Set: Telco customer churn (11.1.3+) -

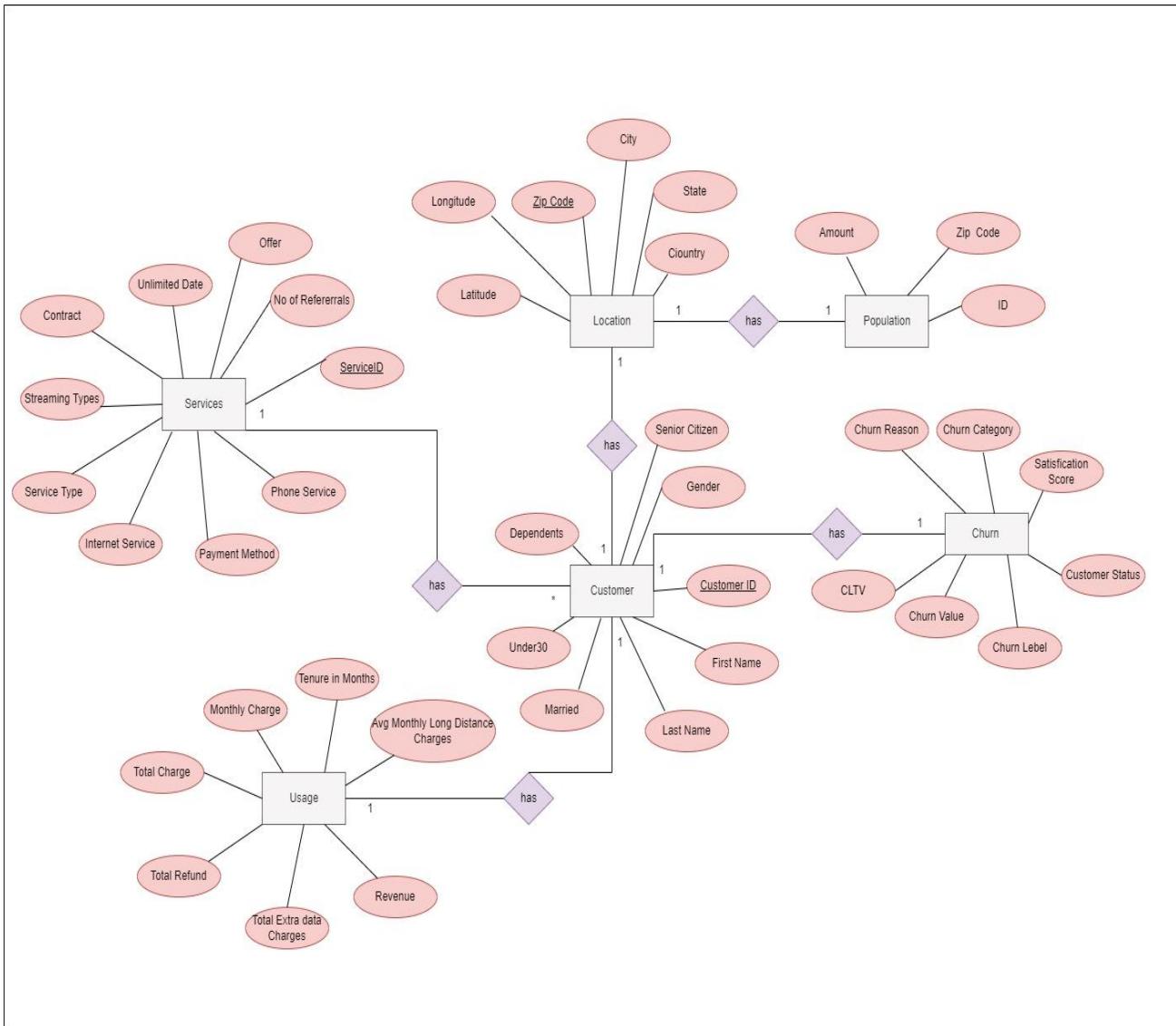
<https://www.kaggle.com/datasets/ylchang/telco-customer-churn-1113>

There were five files which including customer details, customer churn status, customer location, population details of the area which pertinent to the zip code and customer service details. Modifications and extendedness were done in order to come up with a proper dataset.

The screenshot shows the Kaggle dataset page for 'Telco customer churn (11.1.3+)'. The main content includes:

- Telco customer churn (11.1.3+)**: The title of the dataset.
- Data, Code (4), Discussion (0), Metadata**: Navigation links.
- 24**: A button with the number 24.
- New Notebook**: A button to start a new notebook.
- Download (4 MB)**: A button to download the dataset.
- Data Explorer**: A section showing the structure of the dataset.
  - Version 1 (3.93 MB)**
  - Telco\_customer\_churn.xlsx (1.37 MB)**: The main file, showing Total Rows (7043) and Total Columns (33).
    - Table**: Telco\_Churn
  - Sub-files**: Telco\_Churn, Telco\_churn\_S..., Telco\_churn\_J..., Telco\_churn\_I..., Telco\_churn\_C..., Telco\_churn\_S...
- Summary**: A section providing an overview of the dataset.
  - 6 files**
  - 95 columns**

## ER Diagram:



## Step 2: Preparation of data sources

Initial dataset source had five files which include customer details with 9 columns, location details with 3 columns, population details with 3 columns, customer service details with 30 columns and customer churn status details with 11 columns.

Modifications were done for separating service details as service details and usage details in order to having meaningful dataset. And also in order to have customer names I have used separated data source and removed some unnecessary details from the sources.

As a final data sources I have come up with Customer Details, Customer Names, Customer Service Details, Location Details, Population Details, Churn Status, Customer usage Details. And I have converted initial data sources into different files in order to having different data source types. Some of files extract and loaded into database which I have created as IT20139230DataSource in order to have a database source.

CustomerID	Gender	Age	Under30	SeniorCitizen	Married	Dependents	NumberOfDependents	ServiceID
8779-QRDMV	Male 78	No	Yes	No 0	S0001			
7495-OKFY	Female	74	No	Yes	Yes 1	S0002		
1658-BYGOY	Male 71	No	Yes	No	Yes 3	S0003		
4598-XLKNJ	Female	78	No	Yes	Yes 1	S0004		
4846-WHAFZ	Female	80	No	Yes	Yes 1	S0005		
4412-YLTKF	Female	72	No	Yes	No 1	S0006		
0390-DCFQD	Female	76	No	Yes	Yes 2	S0007		
3445-HXXGF	Male 66	No	Yes	Yes	No 0	S0008		
2656-FMOKZ	Female	70	No	Yes	No 2	S0009		
2070-FNEXE	Female	77	No	Yes	No 2	S0010		
0094-OIFMO	Female	78	No	Yes	No 1	S0011		
9947-OTFQU	Male 65	No	Yes	No	No 0	S0012		
9514-JDSKI	Male 77	No	Yes	Yes	No 0	S0013		
7273-TEFQD	Male 67	No	Yes	No	No 0	S0014		
3606-TWKGI	Male 68	No	Yes	No	No 0	S0015		
4385-GZQXV	Female	68	No	Yes	No 0	S0016		
3488-PGMQJ	Male 66	No	Yes	No	No 0	S0017		
7534-BFESC	Male 69	No	Yes	No	No 0	S0018		
8098-LLAZX	Female	74	No	Yes	No 0	S0019		
0265-EDXBD	Male 76	No	Yes	Yes	No 0	S0020		
2840-XANRC	Male 77	No	Yes	Yes	No 0	S0021		
5020-ZSTTY	Female	70	No	Yes	No 0	S0022		
5804-LEPIM	Female	74	No	Yes	No 0	S0023		
0623-IIHUG	Female	77	No	Yes	No 0	S0024		
9057-MSWCO	Male 71	No	Yes	Yes	No 0	S0025		
4895-TMNIR	Male 79	No	Yes	Yes	No 0	S0026		
0533-BNWKF	Female	66	No	Yes	Yes 0	S0027		
0334-GDDSO	Male 77	No	Yes	No	No 0	S0028		
5564-NEMQO	Female	80	No	Yes	No 0	S0029		
6235-VDHOM	Female	77	No	Yes	No 0	S0030		
5609-CEBID	Female	77	No	Yes	No 0	S0031		

CustomerDetails.txt

CustomerNames.csv - Saved

	A	B	C	D	E	F	G	H
1	CustomerID	FirstName	LastName					
2	8779-QRDMV	abad	jonathan r					
3	7495-OOKFY	abellas-bauzo	jonathan a					
4	1658-BYGOY	abreu	pablo					
5	4598-XLKNJ	abreu sosa	basilio l					
6	4846-WHAFZ	abreu-bolano	german					
7	4412-YLTKF	abreu-perez	javier					
8	0390-DCFDQ	abrew	luis a					
9	3445-HXXGF	acevedo	angel					
10	2656-FMOKZ	acevedo	antonio					
11	2070-FNEXE	acevedo	brandy o					
12	0094-OIFMO	acevedo	carlos r					
13	9947-OTFQU	acevedo	diego					
14	9514-JDSKI	acevedo	harold					
15	7273-TEFQD	acevedo	jose					
16	3606-TWKGI	acevedo	michael					
17	4385-GZQXV	aceves	christopher g					
18	3488-PGMQJ	acevevo	felipe					
19	7534-BFESC	acosta	abraham r					
20	8098-LLAZK	acosta	alberto					
21	0265-EDXBD	acosta	brian n					
22	2840-XANRC	acosta	carlos e					
23	5020-ZSTTY	acosta	francisco s					
24	5804-LEPIM	acosta	joe					
25	0623-IIHUG	acosta	julio					
26	9057-MSWCO	acosta	luis a					
27	4895-TMWIR	acosta	moises b sr					
28	0533-BNWKF	acosta	orlando g					
29	0324-GDN5O	acosta	robert					

CustomerNames.csv

CustomerServicesDetails.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	ServiceID	CustomerID	Number	Offer	Phone	Serv	Multiple	Line	Internet	Secu	Online	Back	Device	Pro	Premium	T	Streaming	Streaming	Streaming	Unlimited	Contract	Paperless	PaymentMethod
2	S0001	8779-QRD	0	None	No	No	Yes	DSL	No	No	Yes	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal				
3	S0002	7495-OKF	1	Offer E	Yes	Yes	Yes	Fiber Optic	No	Yes	No	No	No	No	No	No	Yes	Month-to-Yes	Credit Card				
4	S0003	1658-BYG	0	Offer D	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal				
5	S0004	4598-XLKNJ	1	Offer C	Yes	No	Yes	Fiber Optic	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Month-to-Yes	Bank Withdrawal				
6	S0005	4846-WHAFZ	1	Offer C	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal				
7	S0006	4412-YLT	0	Offer C	Yes	Yes	Yes	Fiber Optic	No	No	Yes	No	No	No	No	No	No	Month-to-Yes	Bank Withdrawal				
8	S0007	0390-DCFDQ	1	Offer E	Yes	No	Yes	Fiber Optic	No	No	No	No	No	No	No	No	Yes	Month-to-Yes	Mailed Check				
9	S0008	3445-HXXG	6	Offer B	No	No	Yes	DSL	No	Yes	Yes	No	No	Yes	No	Yes	Month-to-Yes	Bank Withdrawal					
10	S0009	2656-FMOKZ	0	Offer D	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	Yes	Month-to-Yes	Mailed Check					
11	S0010	2070-FNEXE	0	Offer E	Yes	No	Yes	Fiber Optic	Yes	No	No	No	No	No	No	No	Month-to-No	Bank Withdrawal					
12	S0011	0094-OIFM	0	Offer D	Yes	No	Yes	Fiber Optic	No	Yes	No	Yes	Yes	No	No	No	Month-to-Yes	Bank Withdrawal					
13	S0012	9947-OTFQ	0	Offer D	Yes	No	Yes	Fiber Optic	No	Yes	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					
14	S0013	9514-JDSK	1	Offer E	No	No	Yes	Fiber Optic	No	Yes	No	No	No	No	No	Yes	Month-to-No	Bank Withdrawal					
15	S0014	7273-TEFC	0	Offer E	No	No	Yes	DSL	No	No	Yes	No	No	Yes	Yes	Yes	Month-to-Yes	Bank Withdrawal					
16	S0015	3606-TWK	0	Offer D	Yes	Yes	Yes	Cable	No	Yes	Yes	No	Yes	Yes	No	No	Month-to-Yes	Bank Withdrawal					
17	S0016	4385-GZQ	0	Offer D	Yes	No	Yes	Fiber Optic	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-Yes	Bank Withdrawal					
18	S0017	3488-PGM	0	None	Yes	No	Yes	Fiber Optic	No	Yes	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					
19	S0018	7534-BFES	0	Offer C	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					
20	S0019	8098-LLAZK	0	Offer E	Yes	Yes	Yes	Fiber Optic	No	No	No	Yes	Yes	No	No	Yes	Month-to-Yes	Bank Withdrawal					
21	S0020	0265-EDX	2	Offer C	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					
22	S0021	2840-XANI	1	Offer C	Yes	No	Yes	Cable	No	No	Yes	No	Yes	Yes	Yes	Yes	Month-to-Yes	Bank Withdrawal					
23	S0022	5020-ZSTT	0	Offer B	Yes	Yes	Yes	DSL	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	One Year	No	Bank Withdrawal				
24	S0023	5804-LEPII	0	Offer E	Yes	No	Yes	Fiber Optic	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					
25	S0024	0623-IIHUG	0	None	No	No	Yes	Cable	Yes	No	No	Yes	No	No	No	No	Month-to-Yes	Bank Withdrawal					
26	S0025	9057-MSWCO	0	None	No	No	Yes	Cable	Yes	No	No	No	No	No	No	No	Month-to-No	Credit Card					
27	S0026	4895-TMW	1	None	Yes	No	No	None	No	No	No	No	No	No	No	No	Month-to-No	Mailed Check					
28	S0027	0533-BNW	2	Offer B	Yes	Yes	Yes	Fiber Optic	No	Yes	No	No	No	No	No	Month-to-Yes	Bank Withdrawal						
29	S0028	0324-GDN	0	Offer E	Yes	Yes	Yes	Fiber Optic	No	No	No	No	No	No	No	Yes	Month-to-Yes	Bank Withdrawal					

CustomerServiceDetails.csv

Location.csv - Excel

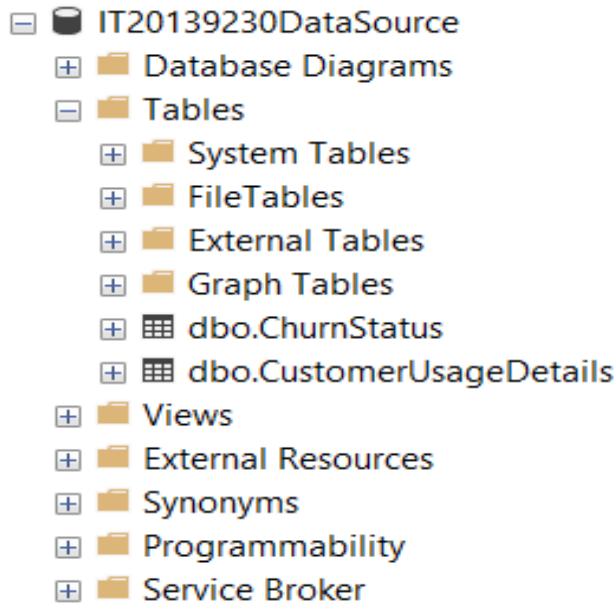
	A	B	C	D	E	F	G	H	I	J	K
1	Customer ID	Country	State	City	Zip Code	Latitude	Longitude				
2	8779-QRDMV	United States	California	Los Angeles	90022	34.02381	-118.157				
3	7495-OOKFY	United States	California	Los Angeles	90063	34.04427	-118.185				
4	1658-BYGOY	United States	California	Los Angeles	90065	34.10883	-118.23				
5	4598-XLKNJ	United States	California	Inglewood	90303	33.93629	-118.333				
6	4846-WHAFZ	United States	California	Whittier	90602	33.97212	-118.02				
7	4412-YLTKF	United States	California	Pico Rivera	90660	33.98952	-118.089				
8	0390-DCFDQ	United States	California	Los Alamitos	90720	33.79499	-118.066				
9	3445-HXXGF	United States	California	Sierra Madre	91024	34.16869	-118.058				
10	2656-FMOKZ	United States	California	Pasadena	91106	34.1394	-118.129				
11	2070-FNEXE	United States	California	Pasadena	91107	34.15901	-118.087				
12	0094-OIFMO	United States	California	North Hollywood	91605	34.2073	-118.4				
13	9947-OTFQU	United States	California	Covina	91722	34.09735	-117.907				
14	9514-IDSKJ	United States	California	El Monte	91732	34.07449	-118.015				
15	7273-TEFQD	United States	California	La Puente	91746	34.03898	-117.991				
16	3606-TWKGI	United States	California	Rowland Heights	91748	33.97675	-117.897				
17	4385-GZQXV	United States	California	Ontario	91764	34.07409	-117.606				
18	3488-PGMQJ	United States	California	Alpine	91901	32.82718	-116.704				
19	7534-BFESC	United States	California	San Diego	92122	32.85723	-117.21				
20	8098-LLAZK	United States	California	San Diego	92122	32.85723	-117.21				
21	0265-EDXBD	United States	California	San Diego	92122	32.85723	-117.21				
22	2840-XANRC	United States	California	San Diego	92122	32.85723	-117.21				
23	5020-ZSTTY	United States	California	San Diego	92122	32.85723	-117.21				
24	5804-LEPIM	United States	California	San Diego	92122	32.85723	-117.21				
25	0623-IIHUG	United States	California	San Diego	92122	32.85723	-117.21				
26	9057-MSWCO	United States	California	San Diego	92122	32.85723	-117.21				
27	4895-TMWIR	United States	California	San Diego	92122	32.85723	-117.21				
28	0533-BNWKF	United States	California	San Diego	92122	32.85723	-117.21				
29	0334-GDPNSO	United States	California	San Diego	92122	32.85723	-117.21				

Location.csv

Population.txt - Notepad

ID	Zip Code	Population
1	90001	54492
2	90002	44586
3	90003	58198
4	90004	67852
5	90005	43019
6	90006	62784
7	90007	45025
8	90008	30852
9	90010	1957
10	90011	101215
11	90012	30596
12	90013	9732
13	90014	3524
14	90015	15140
15	90016	46984
16	90017	20692
17	90018	47143
18	90019	67520
19	90020	42394
20	90021	3012
21	90022	68701
22	90023	47487
23	90024	44150
24	90025	41175
25	90026	73686
26	90027	48727
27	90028	30568
28	90029	41713

Population.txt



IT20139230DataSource

- CustomerDetails**- Details about the Customer
- CustomerNames**- Details of the first and last names of the customer
- CustomerServiceDetails**- Details of the services for the customer
- Location**- Details of the customer location
- Population**- Population details for the relevant zip Codes
- ChurnStatus**- Churn status of the customer
- CustomerUsageDetails**- Service usage details of the customer

## Step 3: Solution architecture

Using different processes, architectures, and technologies we can manage data from various sources and convert them to business insights to make decisions, analysis data and report building.

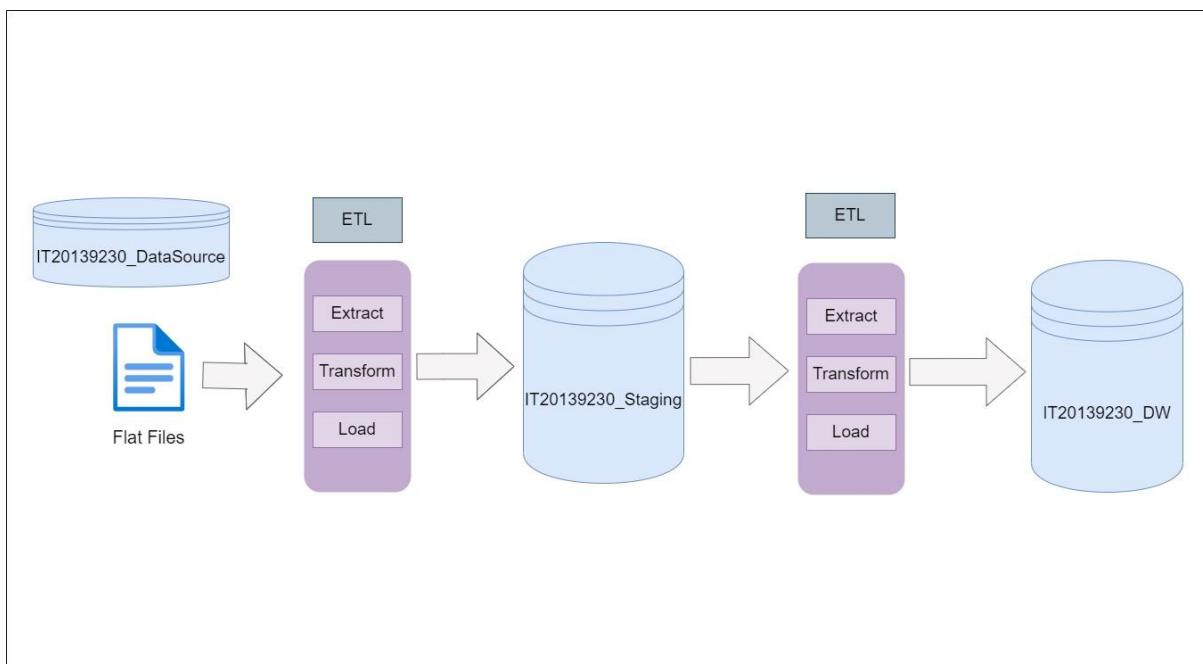
In this solution, there are 3 main components up to the ETL process in the process of Datawarehouse design.

- Source Files
- Staging DB
- Datawarehouse

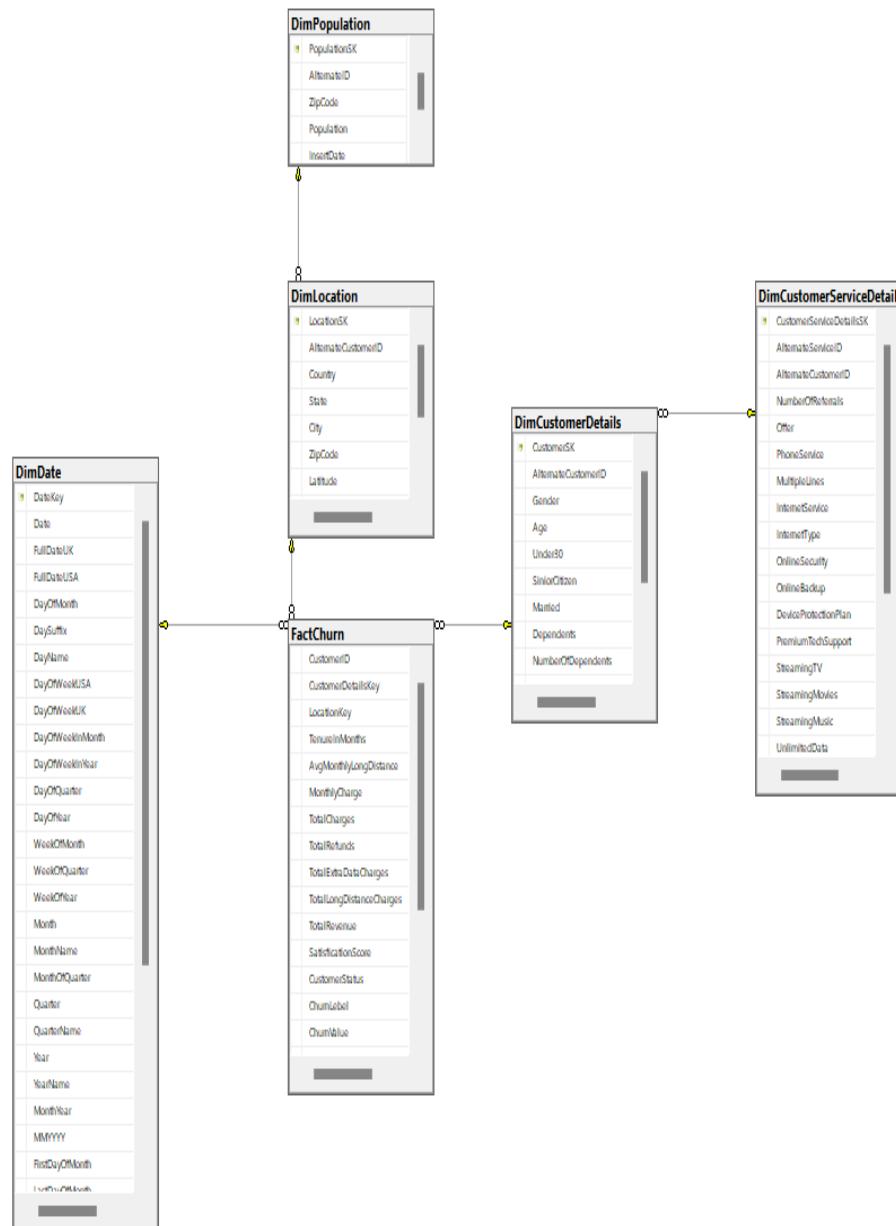
**Source Files :** There are 7 source tables in three different format as the data sources.

**Staging DB :** Extracted all source details and change the data types of the attributes that are in source file to relevant data type and loaded that details into the staging database.

**Datawarehouse :** After the staging, the information in staging database will become the sources to the transformation process. In this process data will transformed & loaded in to the tables in Datawarehouse database.



## Step 4: Data warehouse design & development



## Assumptions

- Snowflake schema is used to design the Datawarehouse design. There is one fact table and 4 dimensions apart from the data dimension.
- DimCustomerServiceDetails were considered as a slowly changing dimension.

Before creating the FactChurn table & other dimensions, started by creating the Date dimension using 'DateMaster.sql' file code dimension using 'DateMaster.sql' file code.

- ❑ **DimCustomerDetails**- Extracting data from CustomerDetails and CustomerNames from the staging and loding them.
- ❑ **DimCustomerServiceDetails**- Extracting data from CustomerServiceDetails from the staging and loding. Ans also this is a slowly changing dimention.
- ❑ **DimLocation**- Extarcrt data from Location from the staging and loading.
- ❑ **DimPopulation**- Extract data from Population from then staging.
- ❑ **FactChurn**- Extract data from CustomerUsageDetails and CustomerChurnStatus

DateMaster.sql - D...N27\Mereesha (53) IT20139230\_DimSql...27\Mereesha (63) IT20139230\_DimSql...27\Mereesha (63)

```

BEGIN TRY
    DROP TABLE [dbo].[DimDate]
END TRY
BEGIN CATCH
    /*No Action*/
END CATCH
/*************************************************************************************************/
CREATE TABLE [dbo].[DimDate]
(
    [DateKey] INT primary key,
    [Date] DATETIME,
    [FullDateUK] CHAR(10), -- Date in dd-MM-yyyy format
    [FullDateUSA] CHAR(10),-- Date in MM-dd-yyyy format
    [DayOfMonth] VARCHAR(2), -- Field will hold day number of Month
    [DaySuffix] VARCHAR(4), -- Apply suffix as 1st, 2nd ,3rd etc
    [DayName] VARCHAR(9), -- Contains name of the day, Sunday, Monday
    [DayOfWeekUSA] CHAR(1),-- First Day Sunday=1 and Saturday=7
    [DayOfWeekUK] CHAR(1),-- First Day Monday=1 and Sunday=7
    [DayOfWeekInMonth] VARCHAR(2), --1st Monday or 2nd Monday in Month
    [DayOfWeekInYear] VARCHAR(2),
    [DayOfQuarter] VARCHAR(3),
    [DayOfYear] VARCHAR(3),
    [WeekofMonth] VARCHAR(1),-- Week Number of Month
    [WeekofQuarter] VARCHAR(2), --Week Number of the Quarter
    [WeekofYear] VARCHAR(2),--Week Number of the Year
    [Month] VARCHAR(2), --Number of the Month 1 to 12
    [MonthName] VARCHAR(9),--January, February etc
    [MonthOfQuarter] VARCHAR(2),-- Month Number belongs to Quarter
    [Quarter] CHAR(1),
    [QuarterName] VARCHAR(9),--First,Second..
    [Year] CHAR(4),-- Year value of Date stored in Row
    [YearName] CHAR(7),--CY 2012,CY 2013
    [MonthYear] CHAR(10), --Jan-2013,Feb-2013
    [MMYY] CHAR(6),
    [FirstDayOfMonth] DATE,
    [LastDayOfMonth] DATE,
    [FirstDayOfQuarter] DATE,
    [LastDayOfQuarter] DATE,
    [FirstDayOfYear] DATE,

```

Column Name Data Type Allow Nulls

Column Name	Data Type	Allow Nulls
DateKey	int	<input checked="" type="checkbox"/>
Date	datetime	<input checked="" type="checkbox"/>
FullDateUK	char(10)	<input checked="" type="checkbox"/>
FullDateUSA	char(10)	<input checked="" type="checkbox"/>
DayOfMonth	varchar(2)	<input checked="" type="checkbox"/>
DaySuffix	varchar(4)	<input checked="" type="checkbox"/>
DayName	varchar(9)	<input checked="" type="checkbox"/>
DayOfWeekUSA	char(1)	<input checked="" type="checkbox"/>
DayOfWeekUK	char(1)	<input checked="" type="checkbox"/>
DayOfWeekInMonth	varchar(2)	<input checked="" type="checkbox"/>
DayOfWeekInYear	varchar(2)	<input checked="" type="checkbox"/>
DayOfQuarter	varchar(3)	<input checked="" type="checkbox"/>
DayOfYear	varchar(3)	<input checked="" type="checkbox"/>
WeekOfMonth	varchar(1)	<input checked="" type="checkbox"/>
WeekOfQuarter	varchar(2)	<input checked="" type="checkbox"/>
WeekOfYear	varchar(2)	<input checked="" type="checkbox"/>
Month	varchar(2)	<input checked="" type="checkbox"/>
MonthName	varchar(9)	<input checked="" type="checkbox"/>
MonthOfQuarter	varchar(2)	<input checked="" type="checkbox"/>

Column Properties

(General)

- (Name) DateKey
- Allow Nulls No
- Data Type int
- Default Value or Binding

(General)

## DimDate

IT20139230\_DimSql...27\Mereesha (63) IT20139230\_DimSql...27\Mereesha (62) IT20139230\_DimSql...27\Mereesha (63)

```

drop table if exists DimPopulation;
create table DimPopulation(
    PopulationSK int identity(1,1) primary key,
    AlternateID nvarchar(15),
    ZipCode nvarchar(50),
    Population nvarchar(50),
    InsertDate datetime,
    ModifiedDate datetime
)

```

Column Name Data Type Allow Nulls

Column Name	Data Type	Allow Nulls
PopulationSK	int	<input checked="" type="checkbox"/>
AlternateID	nvarchar(15)	<input checked="" type="checkbox"/>
ZipCode	nvarchar(50)	<input checked="" type="checkbox"/>
Population	nvarchar(50)	<input checked="" type="checkbox"/>
InsertDate	datetime	<input checked="" type="checkbox"/>
ModifiedDate	datetime	<input checked="" type="checkbox"/>

Column Properties

(General)

- (Name) PopulationSK
- Allow Nulls No
- Data Type int
- Default Value or Binding

Table Designer

(General)

## DimPopulation

```

drop table if exists DimLocation;
create table DimLocation(
    LocationSK int identity(1,1) primary key,
    AlternateCustomerID nvarchar(50),
    Country nvarchar(50),
    State nvarchar(50),
    City nvarchar(50),
    ZipCode nvarchar(50),
    Latitude nvarchar(50),
    Longitude nvarchar(50),
    Populationkey int foreign key references DimPopulation(PopulationSK),
    InsertDate datetime,
    ModifiedDate datetime
)

```

Column Name	Data Type	Allow Nulls
LocationSK	int	<input type="checkbox"/>
AlternateCustomerID	nvarchar(50)	<input checked="" type="checkbox"/>
Country	nvarchar(50)	<input checked="" type="checkbox"/>
State	nvarchar(50)	<input checked="" type="checkbox"/>
City	nvarchar(50)	<input checked="" type="checkbox"/>
ZipCode	nvarchar(50)	<input checked="" type="checkbox"/>
Latitude	nvarchar(50)	<input checked="" type="checkbox"/>
Longitude	nvarchar(50)	<input checked="" type="checkbox"/>
Populationkey	int	<input checked="" type="checkbox"/>
InsertDate	datetime	<input checked="" type="checkbox"/>
ModifiedDate	datetime	<input checked="" type="checkbox"/>

### DimLocation

```

20139230_DimSql...27\Mereesha (63) IT20139230_DimSql...27\Mereesha (62) ✎ ×
---***Slowly changing dimension***---
drop table if exists DimCustomerServiceDetails;
create table DimCustomerServiceDetails(
    CustomerServiceDetailsSK int identity(1,1) primary key,
    AlternateServiceID nvarchar(15),
    AlternateCustomerID nvarchar(15),
    NumberOfReferrals numeric(18,0),
    Offer nvarchar(10),
    PhoneService nvarchar(5),
    MultipleLines nvarchar(5),
    InternetService nvarchar(5),
    InternetType nvarchar(20),
    OnlineSecurity nvarchar(5),
    OnlineBackup nvarchar(5),
    DeviceProtectionPlan nvarchar(5),
    PremiumTechSupport nvarchar(5),
    StreamingTV nvarchar(5),
    StreamingMovies nvarchar(5),
    StreamingMusic nvarchar(5),
    UnlimitedData nvarchar(5),
    Contract nvarchar(50),
    PaperlessBilling nvarchar(5),
    PaymentMethod nvarchar(30),
    StartDate datetime,
    EndDate datetime,
    InsertDate datetime,
    ModifiedDate datetime
)

```

Column Name	Data Type	Allow Nulls
CustomerServiceDetailsSK	int	<input type="checkbox"/>
AlternateServiceID	nvarchar(15)	<input checked="" type="checkbox"/>
AlternateCustomerID	nvarchar(15)	<input checked="" type="checkbox"/>
NumberOfReferrals	numeric(18,0)	<input checked="" type="checkbox"/>
Offer	nvarchar(10)	<input checked="" type="checkbox"/>
PhoneService	nvarchar(5)	<input checked="" type="checkbox"/>
MultipleLines	nvarchar(5)	<input checked="" type="checkbox"/>
InternetService	nvarchar(5)	<input checked="" type="checkbox"/>
InternetType	nvarchar(20)	<input checked="" type="checkbox"/>
OnlineSecurity	nvarchar(5)	<input checked="" type="checkbox"/>
OnlineBackup	nvarchar(5)	<input checked="" type="checkbox"/>
DeviceProtectionPlan	nvarchar(5)	<input checked="" type="checkbox"/>
PremiumTechSupport	nvarchar(5)	<input checked="" type="checkbox"/>
StreamingTV	nvarchar(5)	<input checked="" type="checkbox"/>
StreamingMovies	nvarchar(5)	<input checked="" type="checkbox"/>
StreamingMusic	nvarchar(5)	<input checked="" type="checkbox"/>
UnlimitedData	nvarchar(5)	<input checked="" type="checkbox"/>
Contract	nvarchar(50)	<input checked="" type="checkbox"/>

Column Properties

(General)

- (Name) CustomerServiceDetailsSK
- Allow Nulls No
- Data Type int
- Default Value or Binding

Table Designer

(General)

### DimCustomerServiceDetails

```

drop table if exists DimCustomerDetails;
create table DimCustomerDetails(
    CustomerSK int identity(1,1) primary key,
    AlternateCustomerID nvarchar(15),
    Gender nvarchar(10),
    Age numeric(18,0),
    Under30 nvarchar(5),
    SeniorCitizen nvarchar(5),
    Married nvarchar(5),
    Dependents nvarchar(5),
    NumberOfDependents numeric(18,0),
    ServiceKey int foreign key references DimCustomerServiceDetails(CustomerServiceDetailsSK),
    FirstName nvarchar(30),
    LastName nvarchar(30),
    InsertDate datetime,
    ModifiedDate datetime
)

drop table if exists DimPopulation;
create table DimPopulation(

```

DimCustomerDetails

Column Name	Data Type	Allow Nulls
CustomerSK	int	<input checked="" type="checkbox"/>
AlternateCustomerID	nvarchar(15)	<input checked="" type="checkbox"/>
Gender	nvarchar(10)	<input checked="" type="checkbox"/>
Age	numeric(18, 0)	<input checked="" type="checkbox"/>
Under30	nvarchar(5)	<input checked="" type="checkbox"/>
SeniorCitizen	nvarchar(5)	<input checked="" type="checkbox"/>
Married	nvarchar(5)	<input checked="" type="checkbox"/>
Dependents	nvarchar(5)	<input checked="" type="checkbox"/>
NumberOfDependents	numeric(18, 0)	<input checked="" type="checkbox"/>
ServiceKey	int	<input checked="" type="checkbox"/>
FirstName	nvarchar(30)	<input checked="" type="checkbox"/>
LastName	nvarchar(30)	<input checked="" type="checkbox"/>
InsertDate	datetime	<input checked="" type="checkbox"/>
ModifiedDate	datetime	<input checked="" type="checkbox"/>

Column Properties

(General)	CustomerSK
(Name)	No
Allow Nulls	<input checked="" type="checkbox"/>
Data Type	int
Default Value or Binding	
Table Designer	
(General)	

### DimCustomerDetails

```

IT20139230_DimSql...ql - not connected IT20139230_DimSql...ql - not connected DESKTOP-IKSVN27\...merServiceDetails
drop table if exists FactChurn;
create table FactChurn(
    CustomerID nvarchar(15),
    CustomerDetailsKey int foreign key references DimCustomerDetails(CustomerSK),
    LocationKey int foreign key references DimLocation(LocationSK),
    TenureInMonths numeric(18,0),
    AvgMonthlyLongDistance real,
    MonthlyCharge real,
    TotalCharges real,
    TotalRefunds real,
    TotalExtraDataCharges real,
    TotalLongDistanceCharges real,
    TotalRevenue real,
    SatisfactionScore numeric(18,0),
    CustomerStatus nvarchar(15),
    ChurnLabel nvarchar(5),
    ChurnValue numeric(18,0),
    ChurnScore numeric(18,0),
    CLTV numeric(18,0),
    ChurnCategory nvarchar(30),
    ChurnReason nvarchar(60),
    ChurnedDateKey int foreign key references DimDate(DateKey),
    InsertDate datetime,
    ModifiedDate datetime
)

```

FactChurn

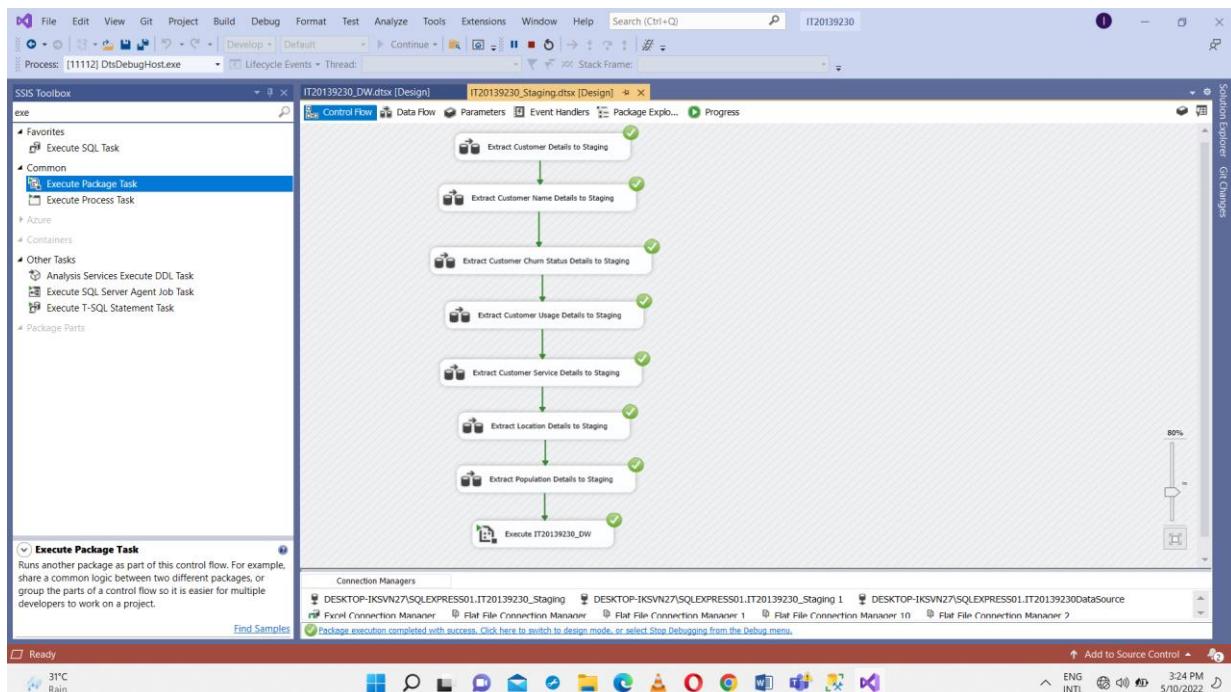
Column Name	Data Type	Allow Nulls
CustomerID	nvarchar(15)	<input checked="" type="checkbox"/>
CustomerDetailsKey	int	<input checked="" type="checkbox"/>
LocationKey	int	<input checked="" type="checkbox"/>
TenureInMonths	numeric(18, 0)	<input checked="" type="checkbox"/>
AvgMonthlyLongDistance	real	<input checked="" type="checkbox"/>
MonthlyCharge	real	<input checked="" type="checkbox"/>
TotalCharges	real	<input checked="" type="checkbox"/>
TotalRefunds	real	<input checked="" type="checkbox"/>
TotalExtraDataCharges	real	<input checked="" type="checkbox"/>
TotalLongDistanceCharges	real	<input checked="" type="checkbox"/>
TotalRevenue	real	<input checked="" type="checkbox"/>
SatisfactionScore	numeric(18, 0)	<input checked="" type="checkbox"/>
CustomerStatus	nvarchar(15)	<input checked="" type="checkbox"/>
ChurnLabel	nvarchar(5)	<input checked="" type="checkbox"/>
ChurnValue	numeric(18, 0)	<input checked="" type="checkbox"/>
ChurnScore	numeric(18, 0)	<input checked="" type="checkbox"/>
CLTV	numeric(18, 0)	<input checked="" type="checkbox"/>
ChurnCategory	nvarchar(30)	<input checked="" type="checkbox"/>

Column Properties

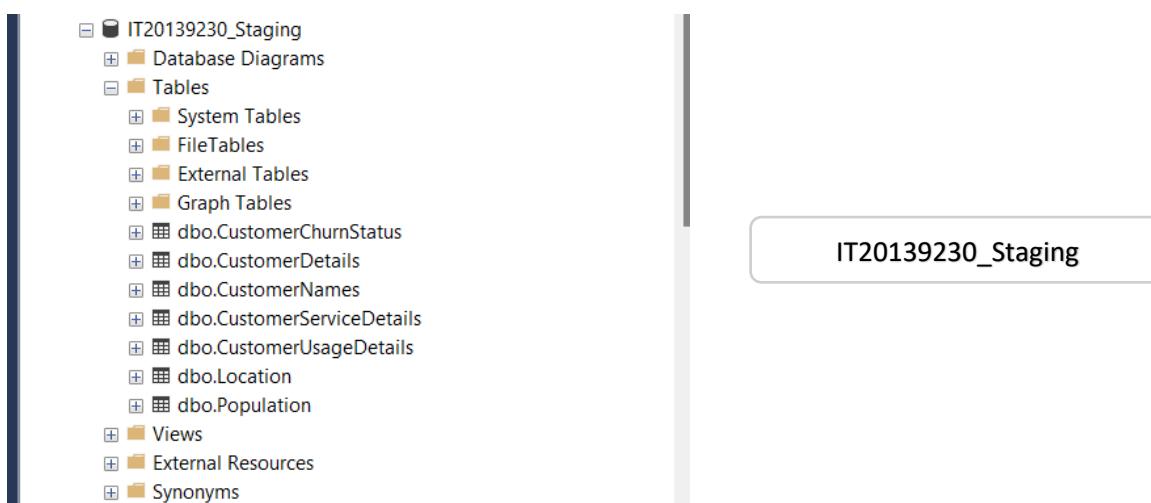
(General)	CustomerID
(Name)	Yes
Allow Nulls	<input checked="" type="checkbox"/>
Data Type	nvarchar
Default Value or Binding	
Length	15
(General)	

### FactChurn

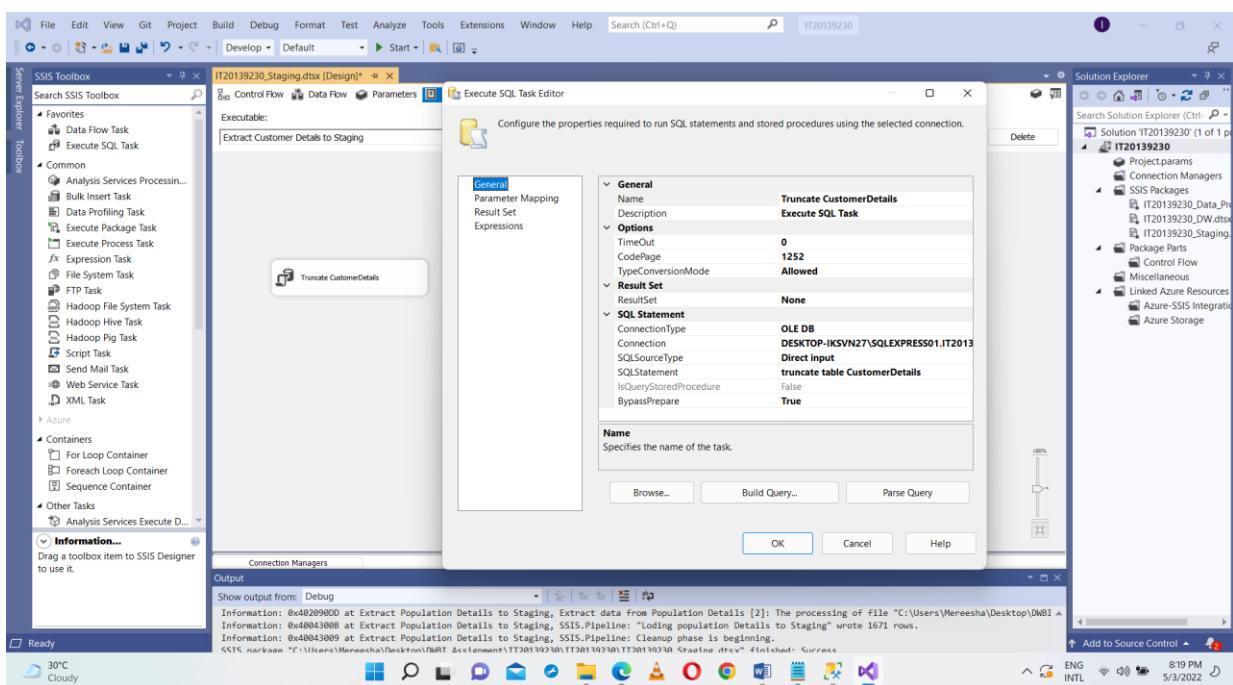
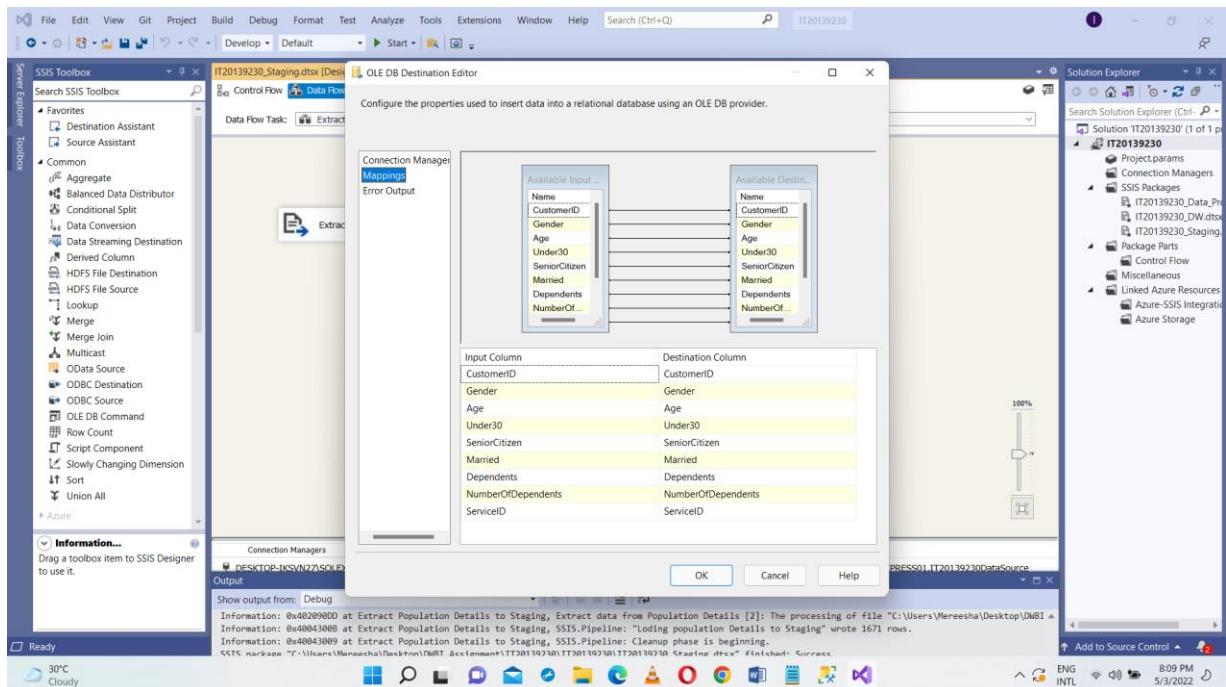
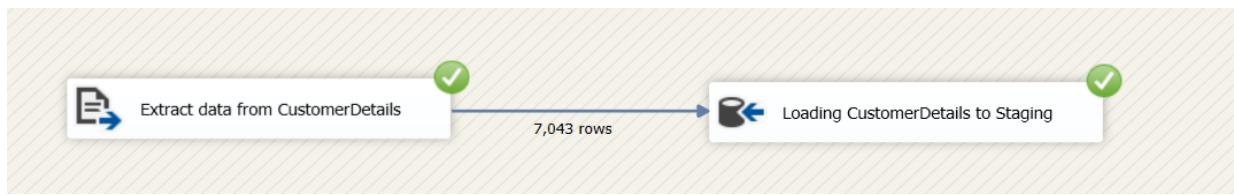
# Step 5: ETL development



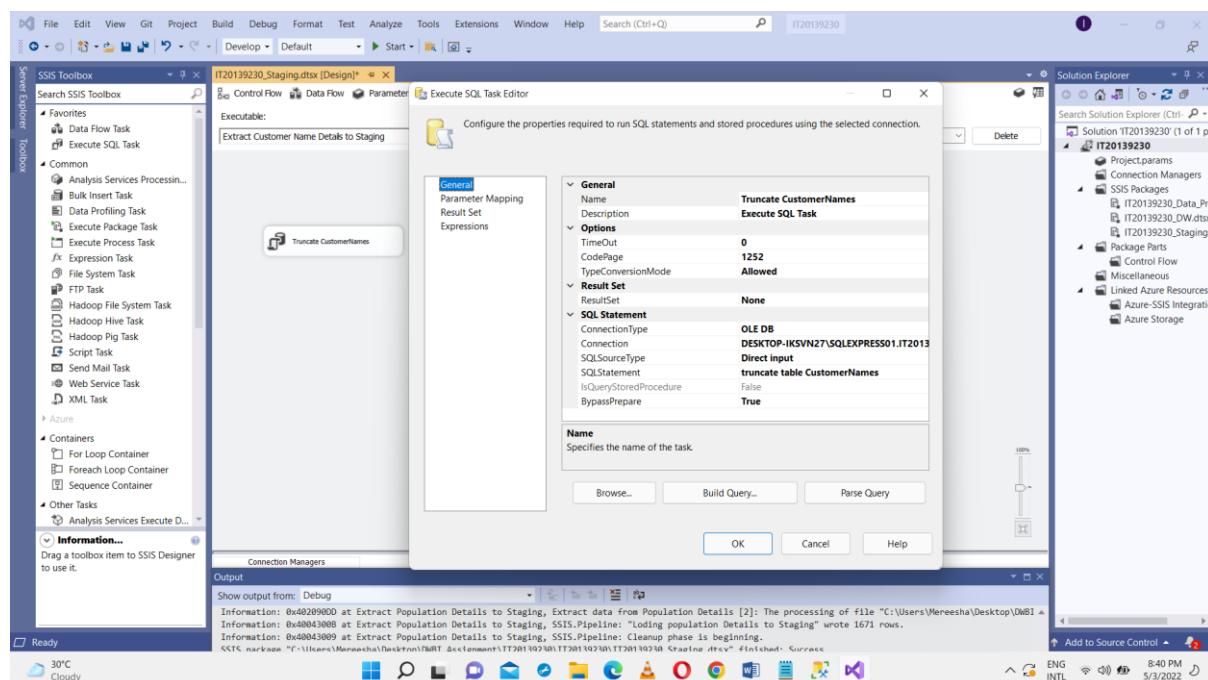
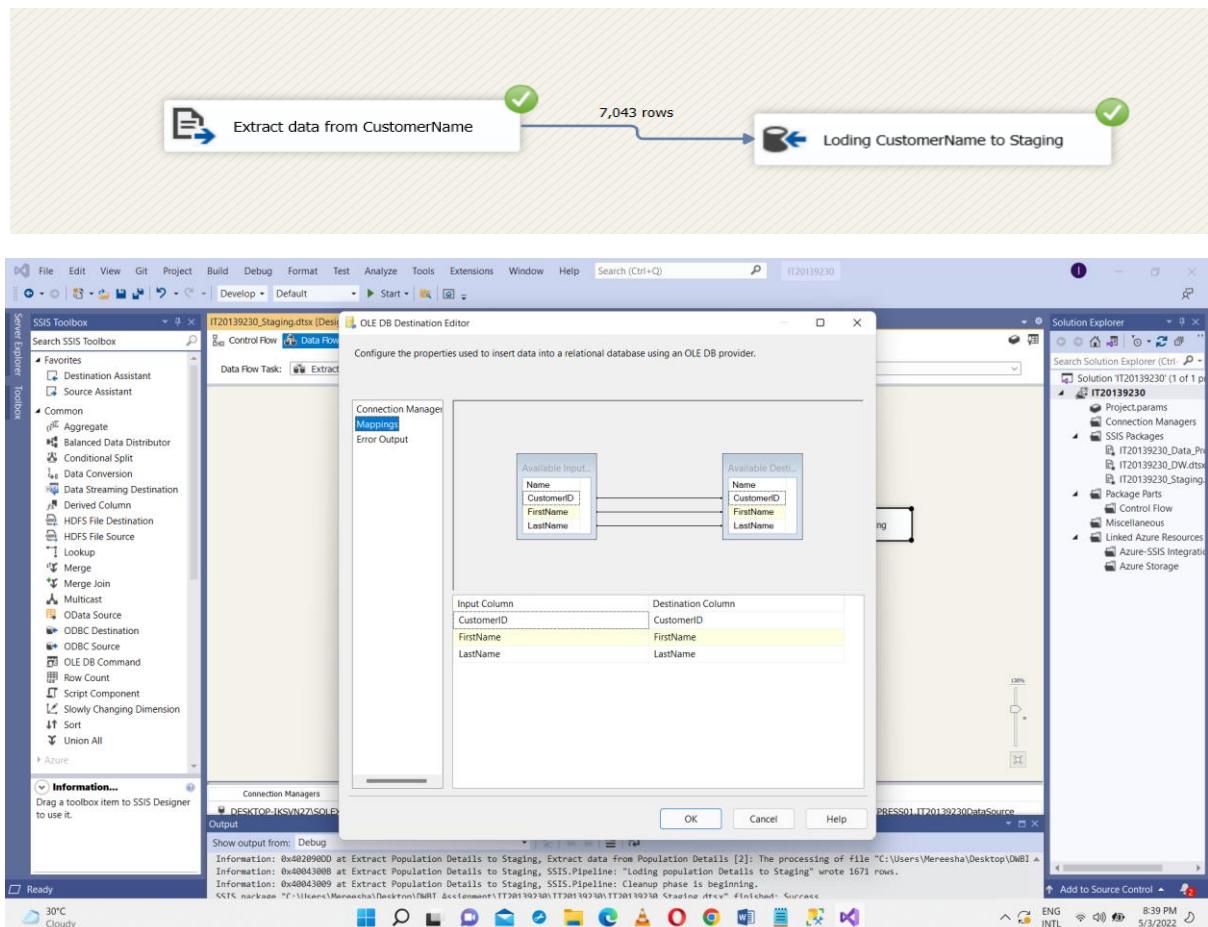
First extracted all the data from tables which are in the IT20139230\_DataSource, and flat files into separated staging DB called IT20139230\_Staging as shown in the below using SQL Server Integration Service Software.



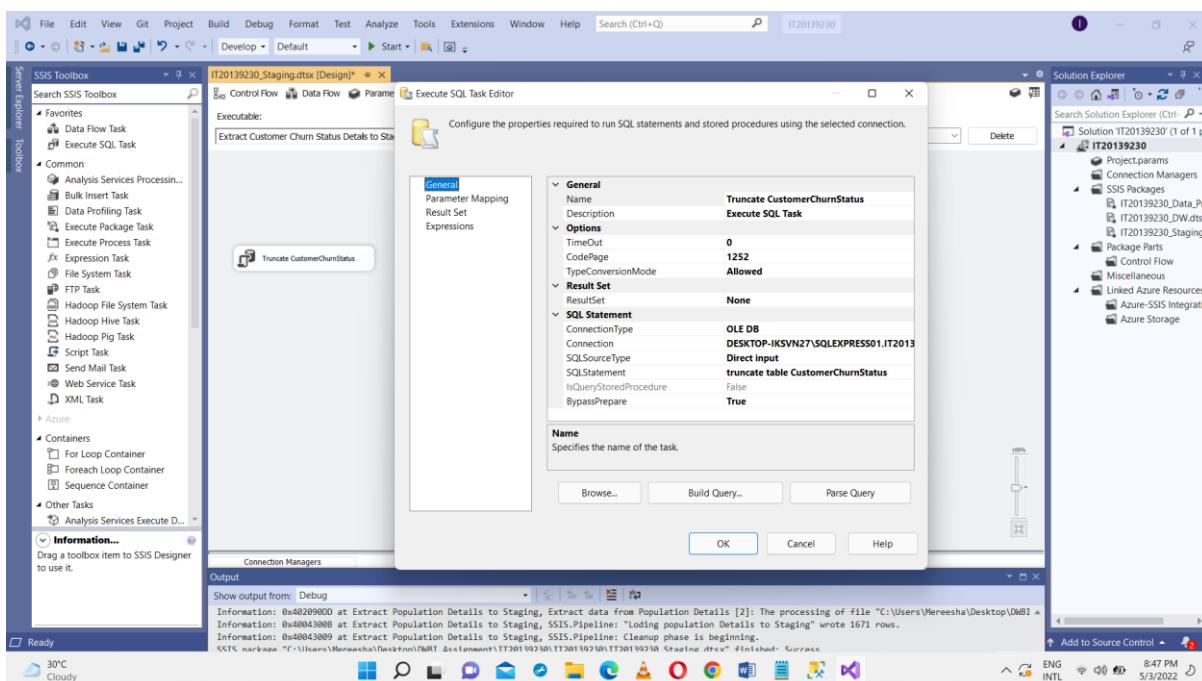
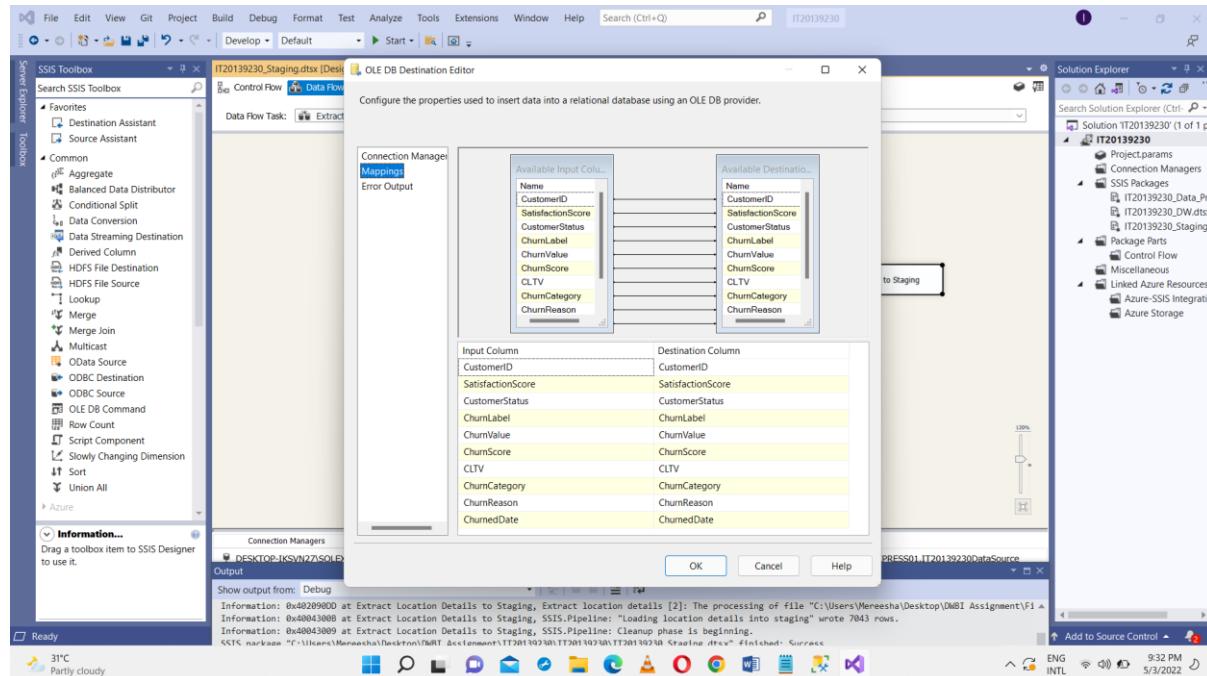
Extract Customer Details from CustomerDetails.txt file and load into staging CustomerDetails table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.CustomerDetails in IT20139230\_Staging Database



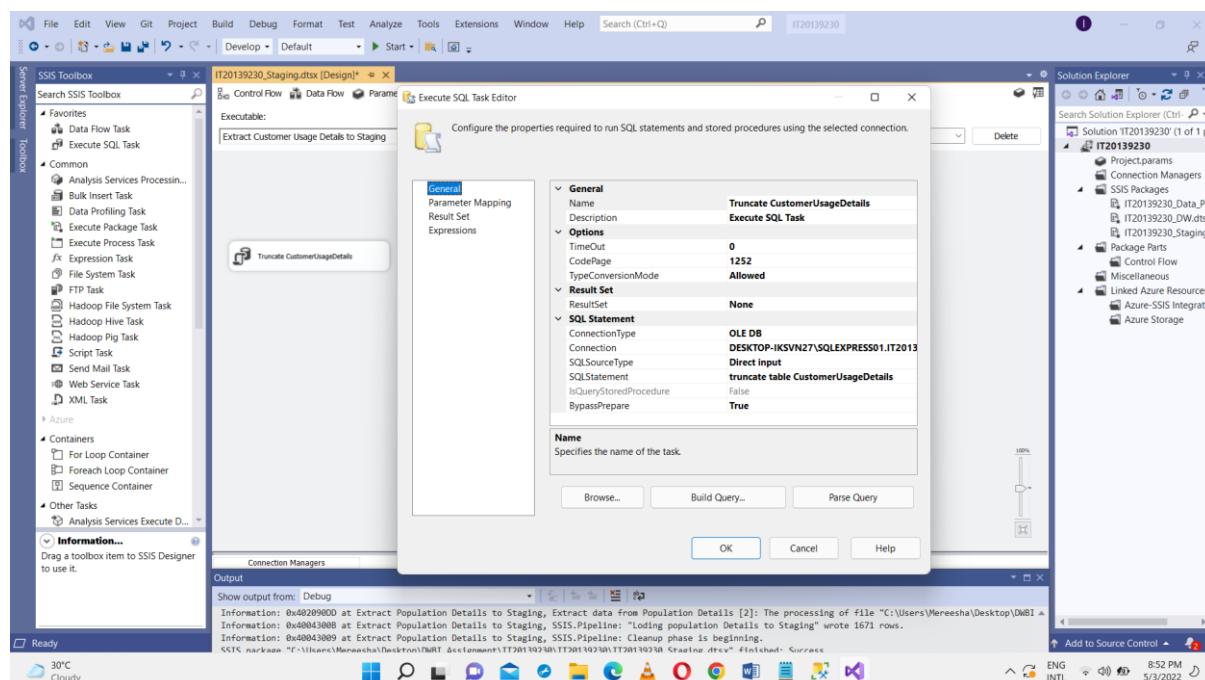
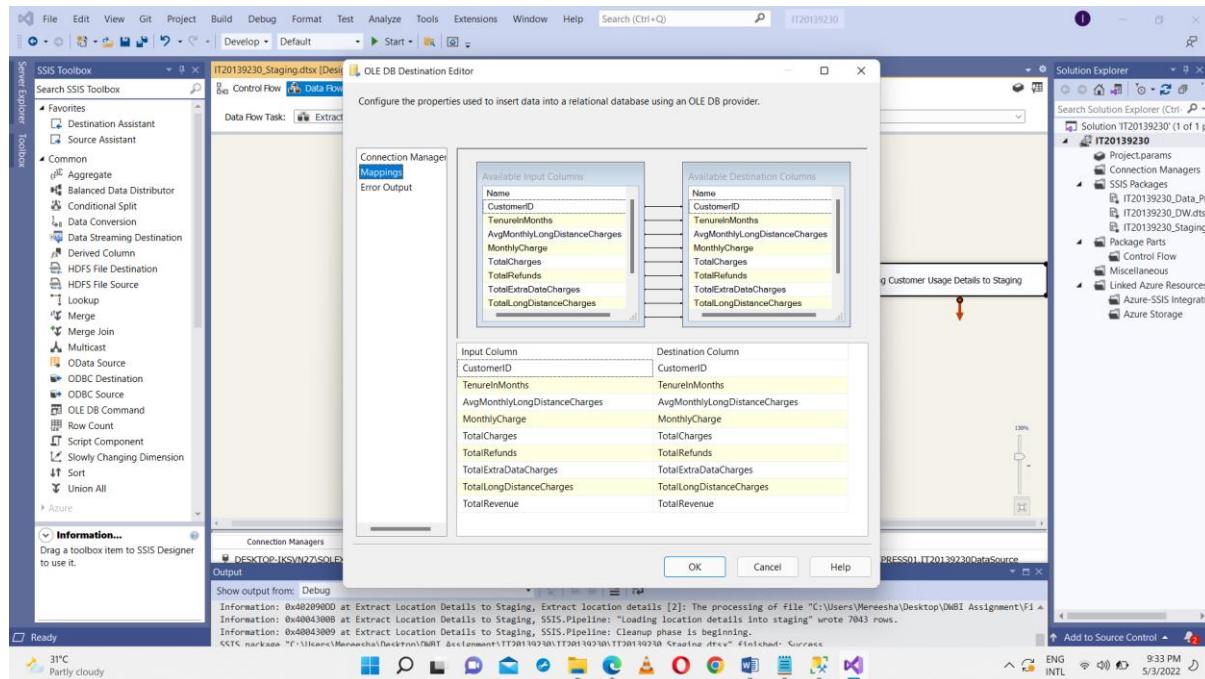
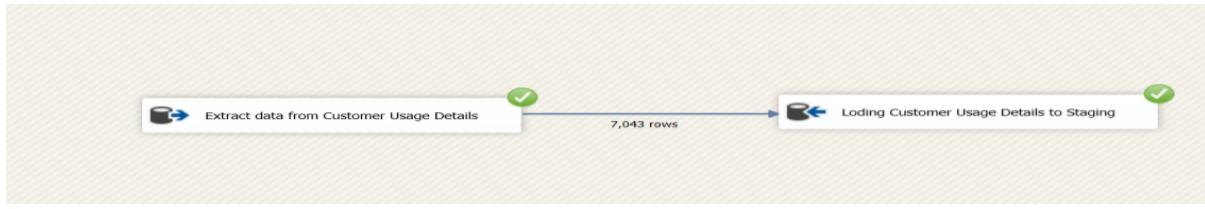
Extract Customer Name Details from CustomerNames.csv file and load into staging CustomerNames table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.CustomerNames in IT20139230\_Staging Database.



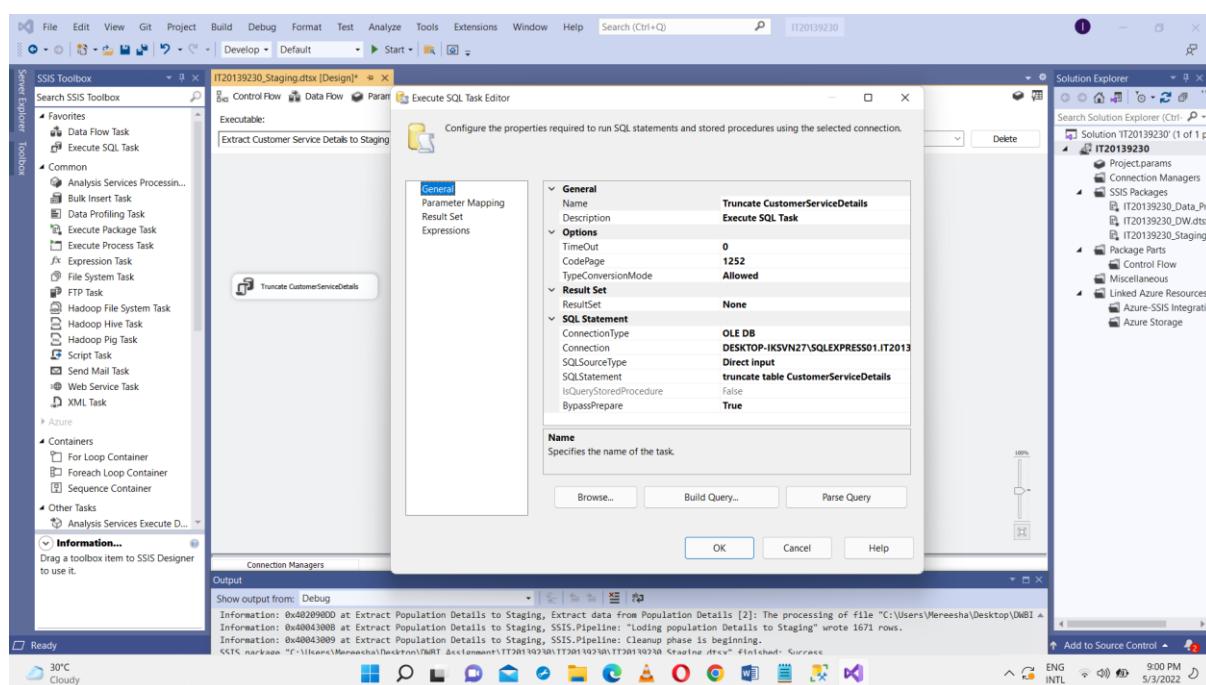
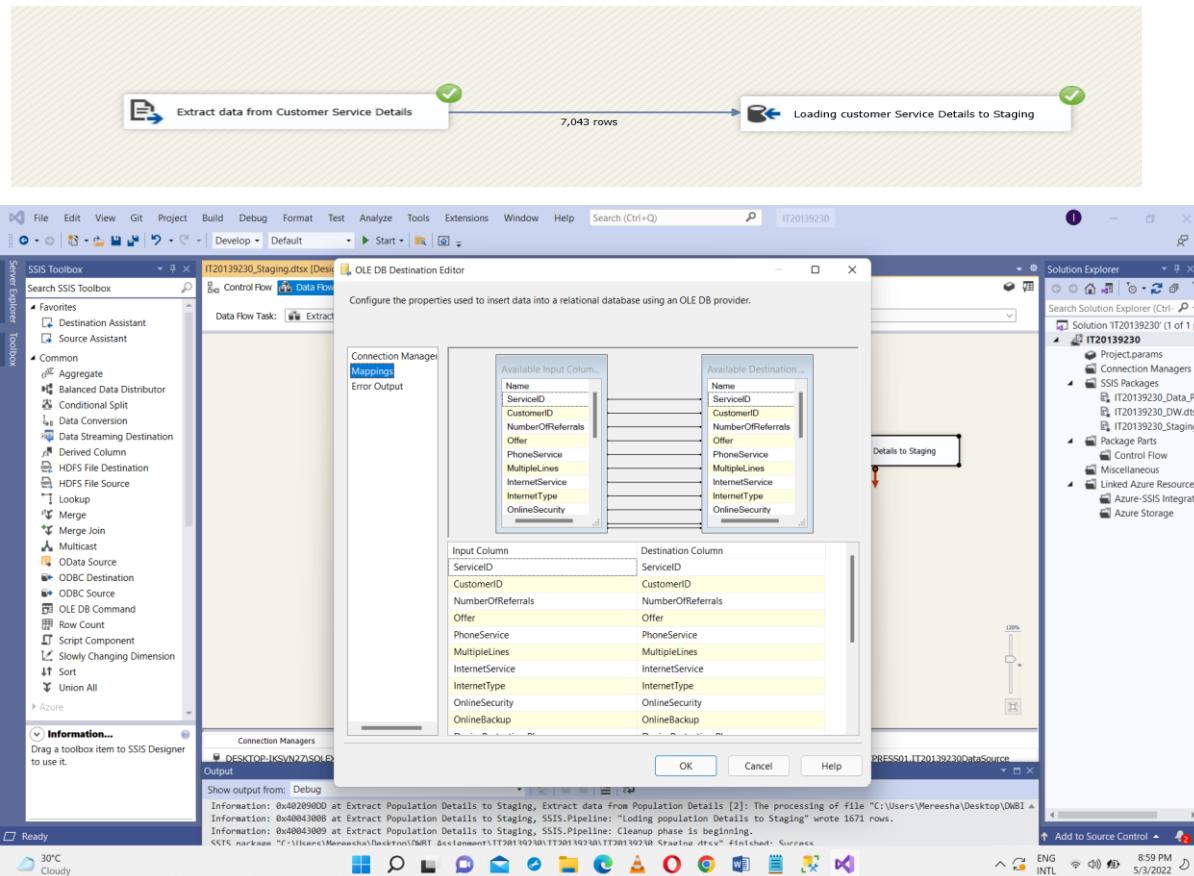
Extract Customer Churn Details from IT20139230\_DataSource database and load into staging CustomerChurnStatus table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.CustomerChurnStatus in IT20139230\_Staging Database.



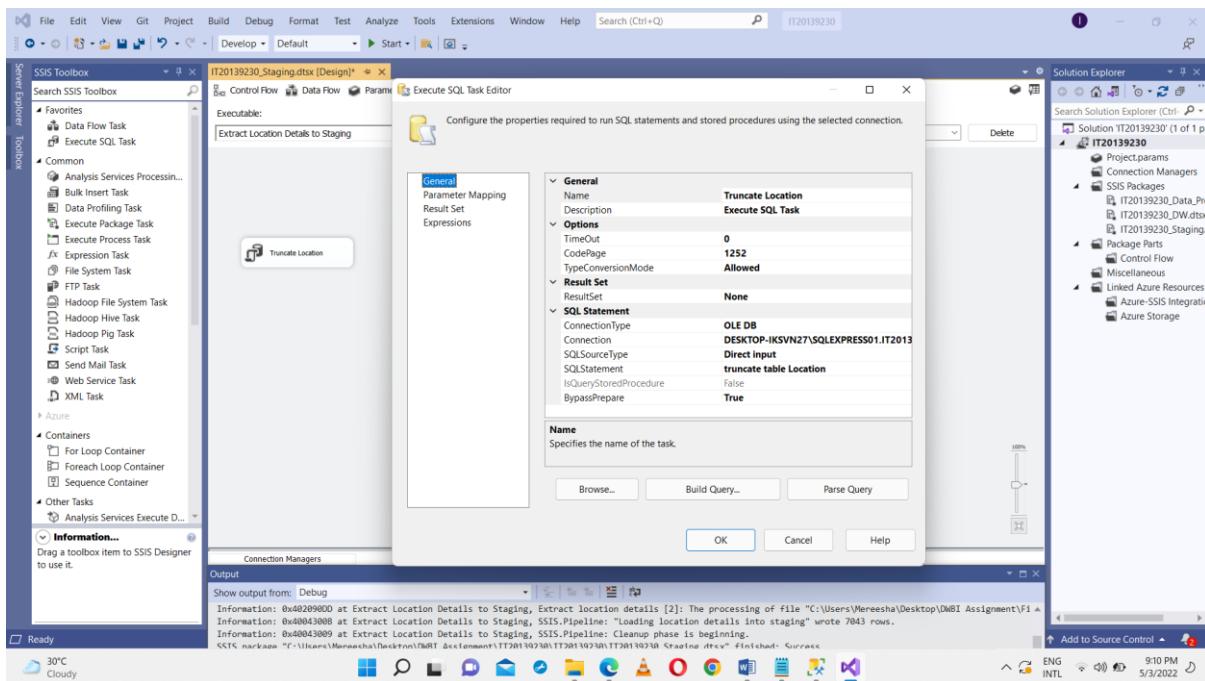
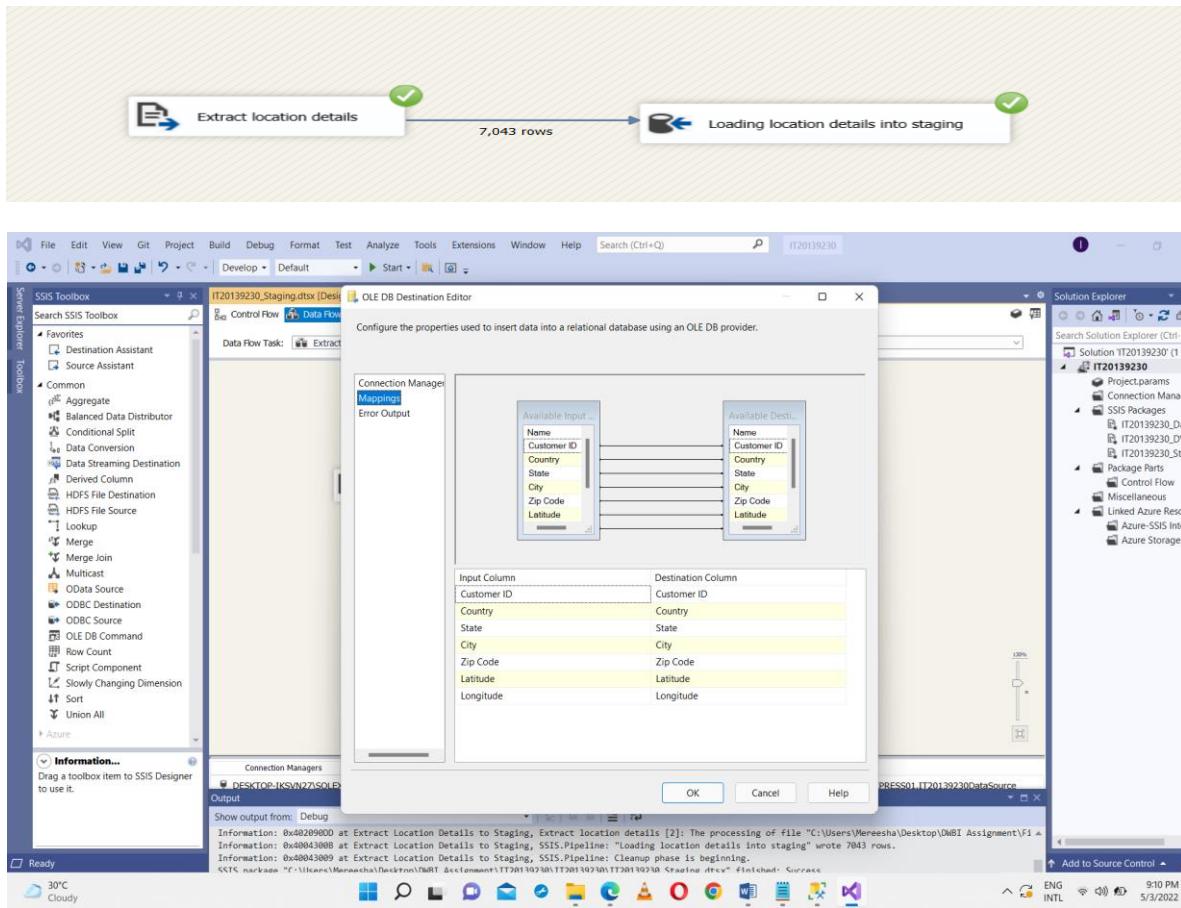
Extract Customer Usage Details from IT20139230\_DataSource database and load into staging CustomerUsageDetails table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.CustomerUsageDetails in IT20139230\_Staging Database.



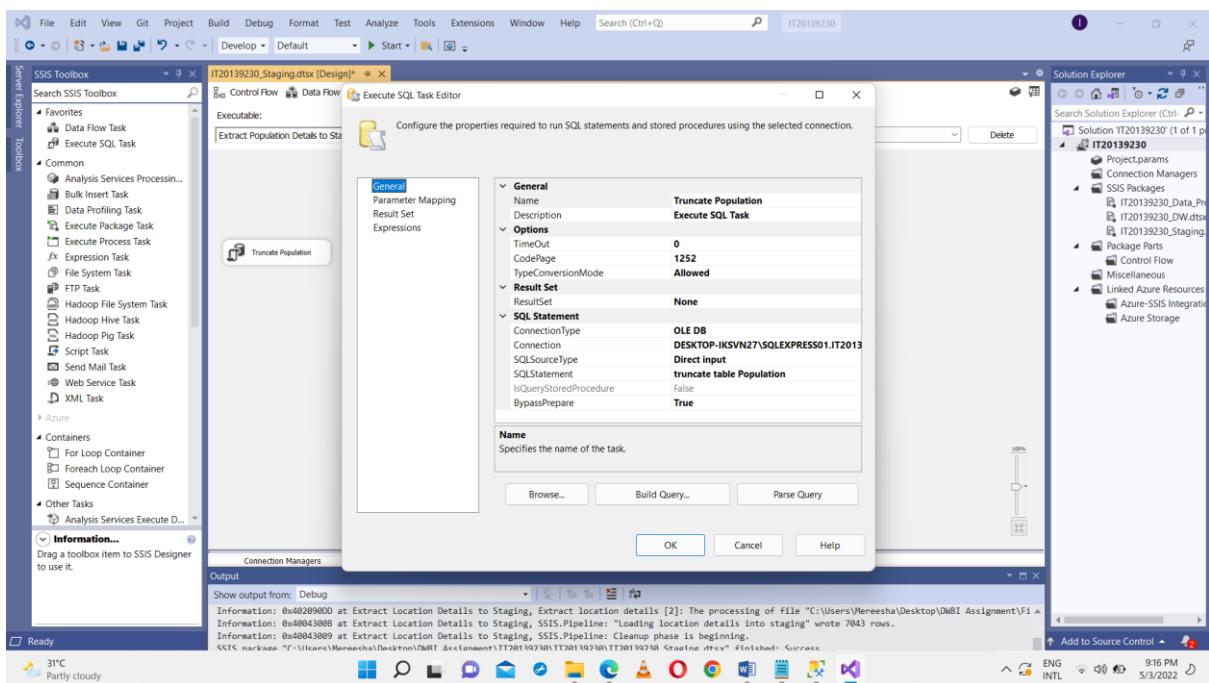
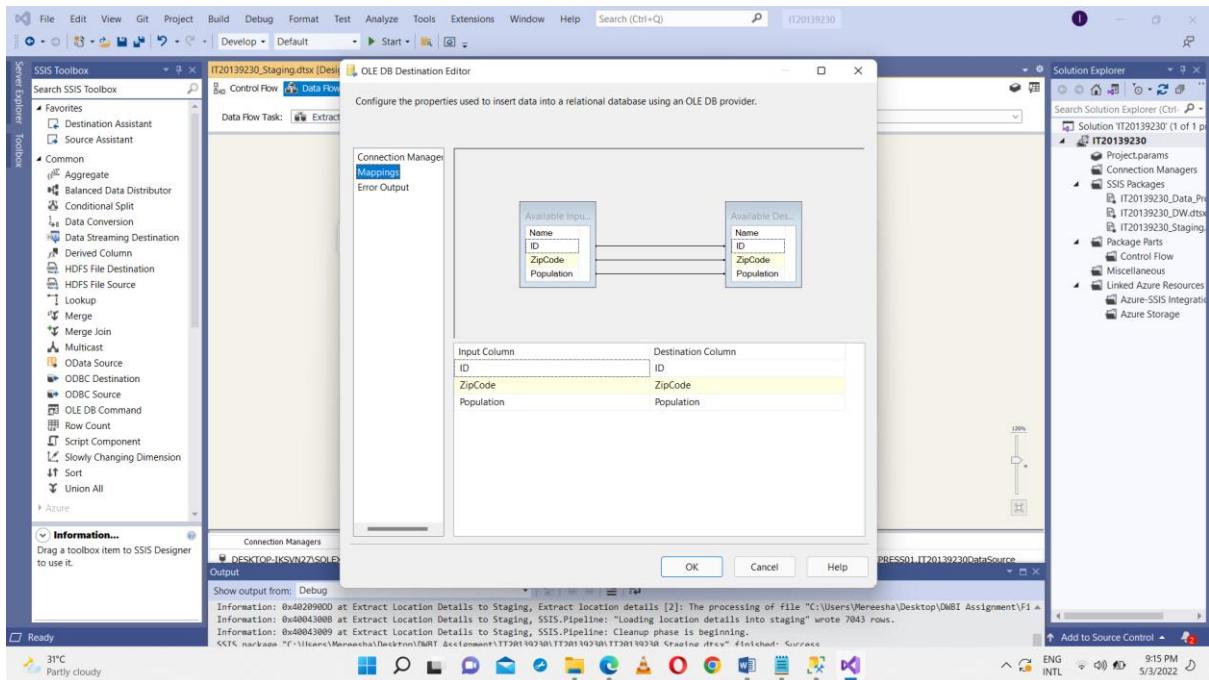
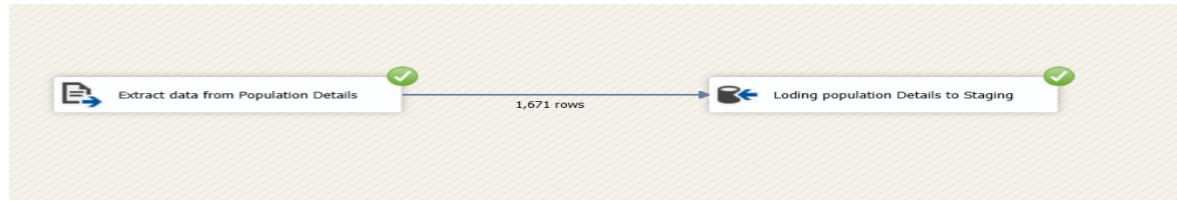
Extract Customer Service Details from CustomerServiceDetails.csv database and load into staging CustomerServiceDetails table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.CustomerServiceDetails in IT20139230\_Staging Database.



Extract Location Details from Location.csv and load into staging Location table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo.Location in IT20139230\_Staging Database.



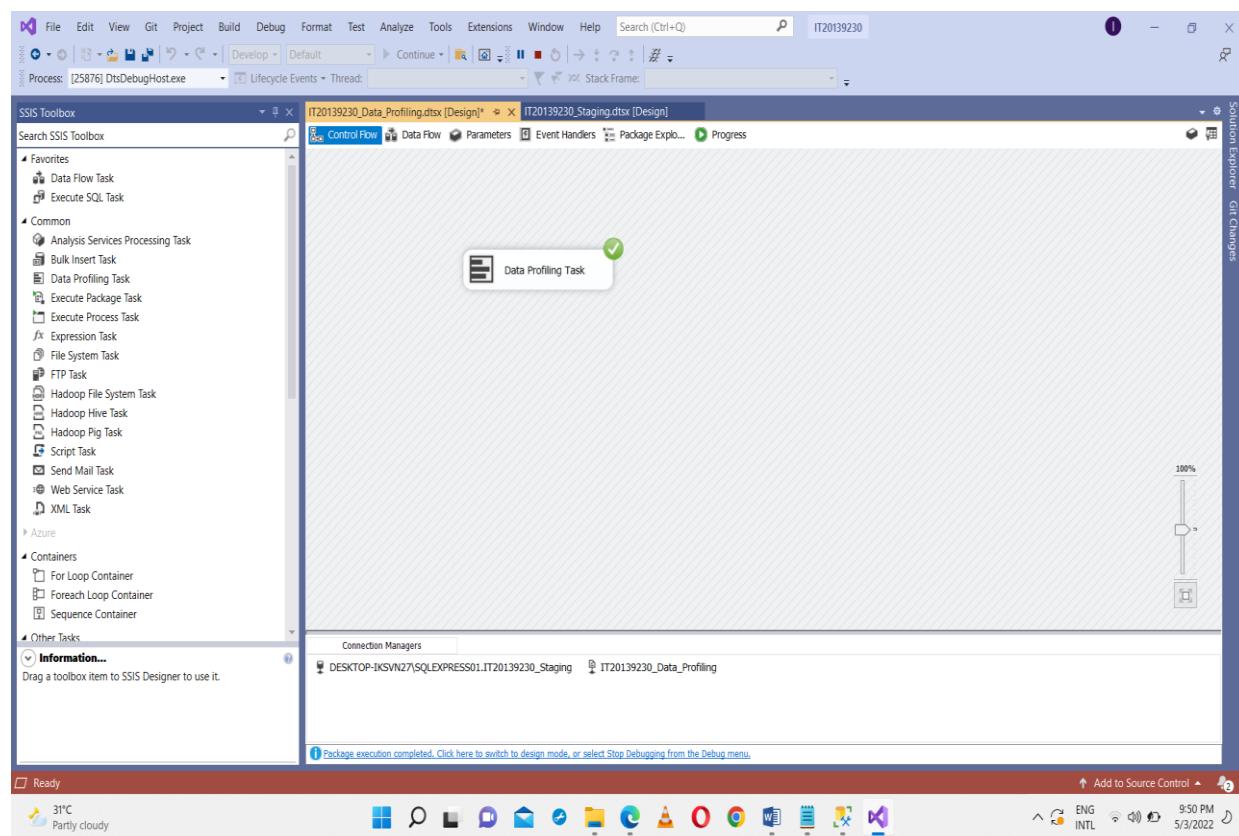
Extract Location Details from Population.txt and load into staging Population table. Used Execute SQL Task SSIS tool Truncate table for SQL command as truncate table dbo. Population in IT20139230\_Staging Database.



## Data Profiling

Staging table data was used to analyzed and determined what types of transformation was needed to perform on the data.

- Right click on SSIS Packages and New SSIS Package selected.
- In the Control Flow of IT20139230\_Data\_Profiling.dtsx, drag and drop the Data Profiling Task and double click to open the configuration.
- Clicked on Quick Profile button to open Single Table Quick Profile Form.
- Clicked on New button and create the connection to IT20139230\_Staging
- Once the green tick appeared, double click the Data Profiling Task and Click on the Open Profile Viewer to view the analyzed data



Data Profile Viewer -

Open Refresh

Profiles (Table View)

Column Value Distribution Profiles - [dbo].[CustomerChurnStatus]

Column	Number Of Distinct Values
ChurnCategory	5
ChurnedDate	179
ChurnLabel	2
ChurnReason	20
CustomerID	7043

Frequent Value Distribution (0.1000 %) - ChurnCategory

Value	Count	Percentage
Competitor	841	11.9409 %
Price	211	2.9959 %
Dissatisfaction	303	4.3021 %
Other	200	2.8397 %
Altitude	314	4.4583 %

Successfully loaded data profile from ...

Message

31°C Partly cloudy

ENG INTL 9:51 PM 5/3/2022

Data Profile Viewer -

Open Refresh

Profiles (Table View)

Column Statistics Profiles - [dbo].[CustomerUsageDetails]

Column	Minimum	Maximum	Mean	Standard Deviation
AvgMonthlyLongDist...	0	49.99	22.9589535712	15.4470166096381
MonthlyCharge	18.25	118.75	64.7616924812	30.0879108835307
TenureInMonths	1	72	32.3867670262	24.5403186479282
TotalCharges	18.8	9584.8	2280.38126420	2266.05957283357
TotalExtraDataCharges	0	150	6.86071276444	25.1031956719383
TotalLongDistanceCh...	0	3564.72	749.0992613970	846.59994551224
TotalRefunds	0	49.79	1.96218230659	7.90205333711093
TotalRevenue	21.36	11379.34	3034.37905502	2865.0012614022

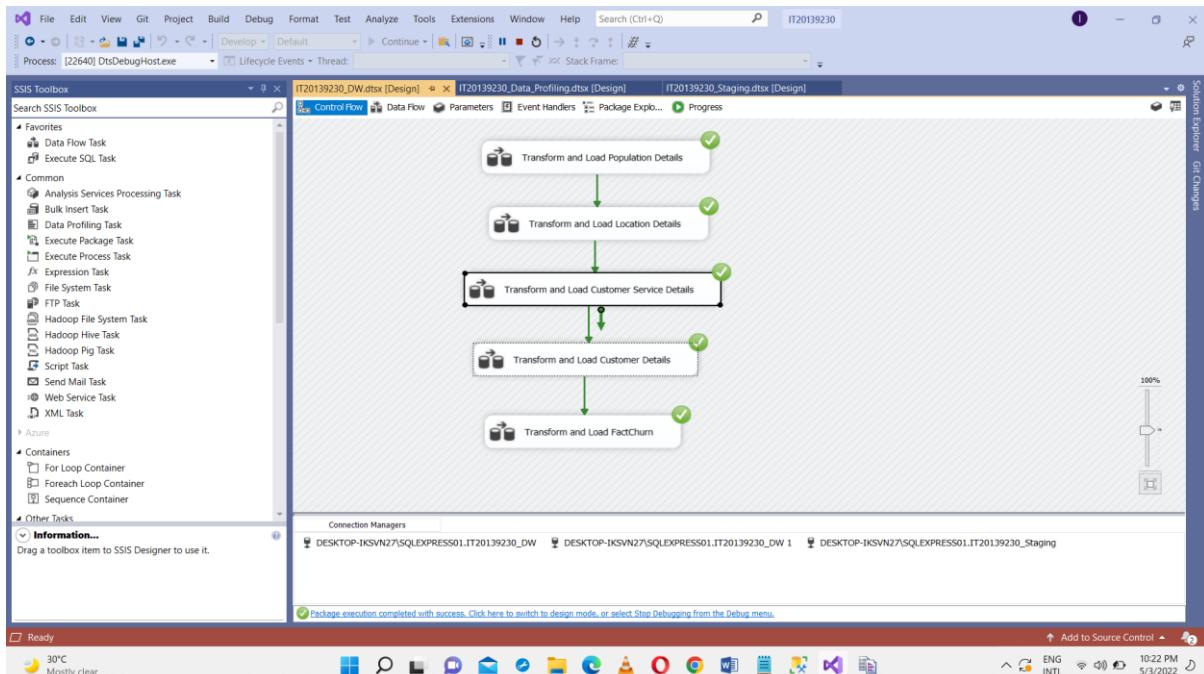
Successfully loaded data profile from ...

Message

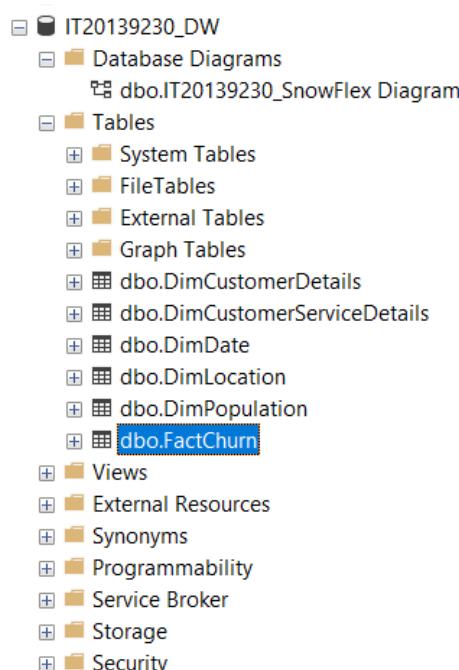
31°C Partly cloudy

ENG INTL 9:53 PM 5/3/2022

## Data Transformation



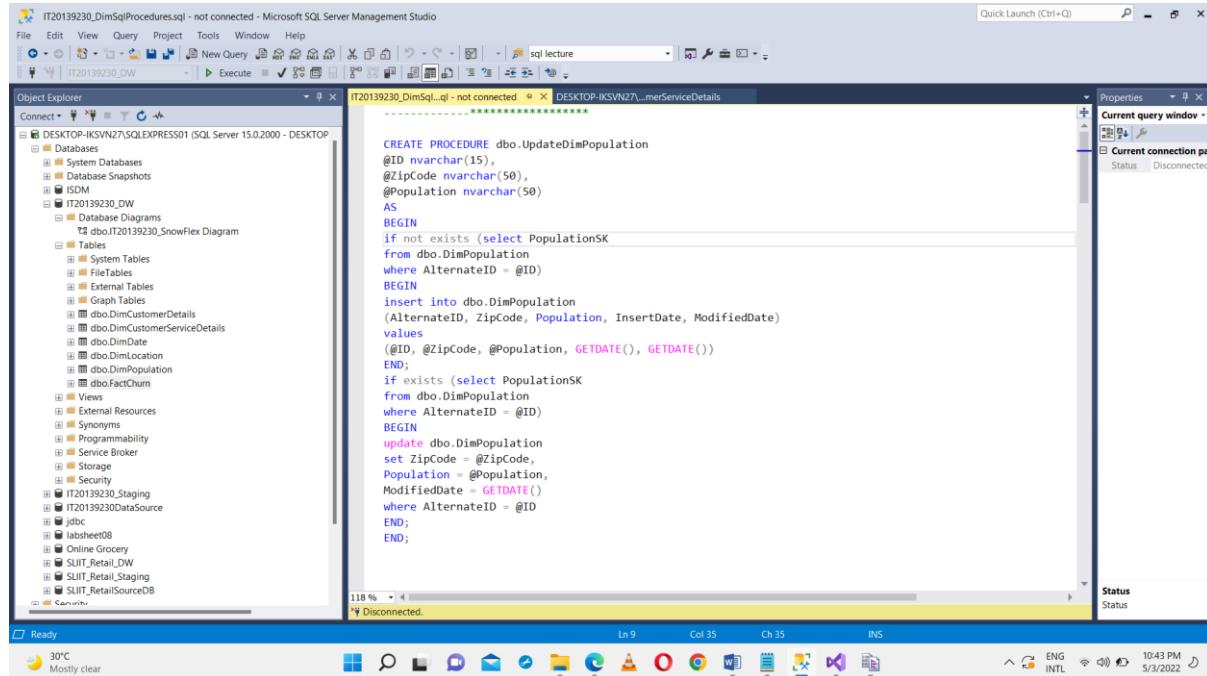
Finally extracted all the data from tables which are in the IT20139230\_Staging into DB named IT20139230\_DW database as shown in the below using SQL Server Integration Service Software.



IT20139230\_DW

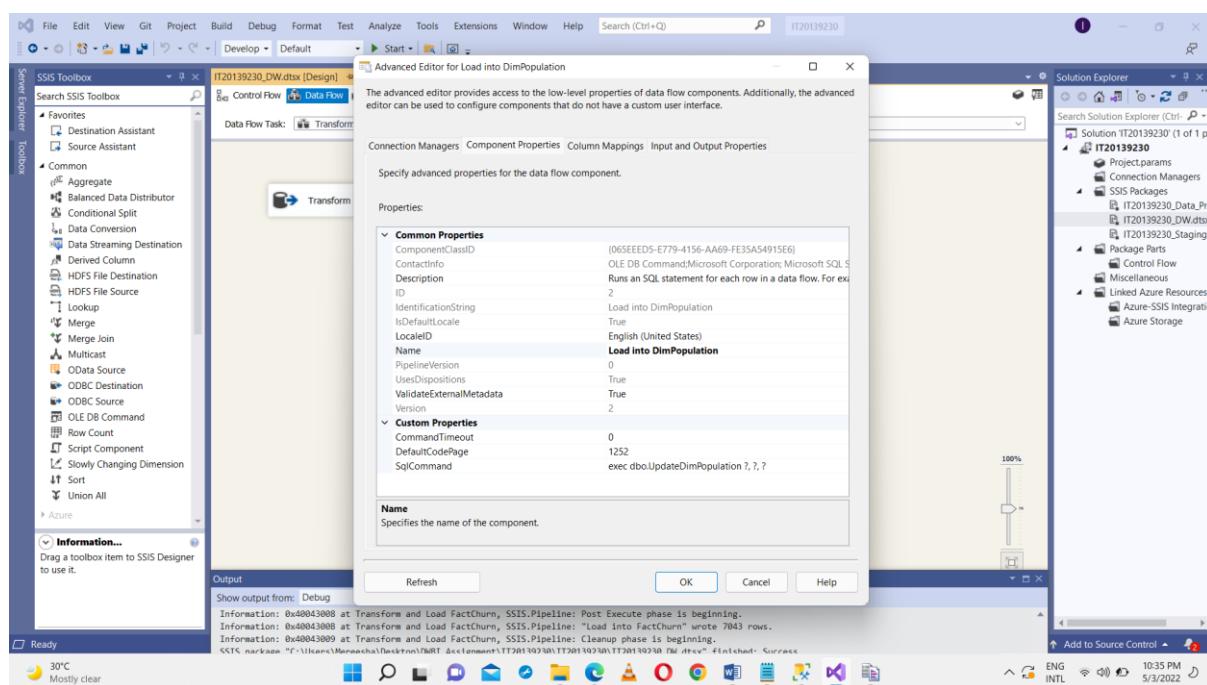
## DimPopulation

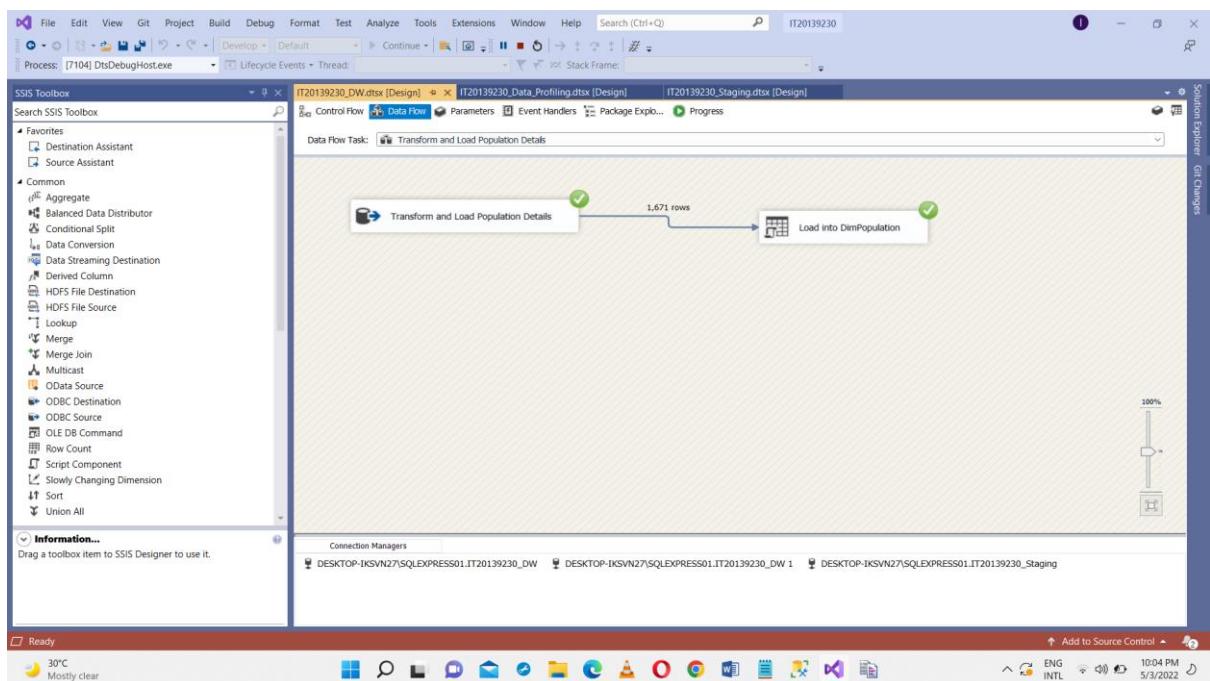
First created a Procedure called UpdateDimPopulation and executed in the IT20139230\_DW database



```
CREATE PROCEDURE dbo.UpdateDimPopulation
@ID nvarchar(15),
@ZipCode nvarchar(50),
@Population nvarchar(50)
AS
BEGIN
if not exists (select PopulationSK
from dbo.DimPopulation
where AlternateID = @ID)
BEGIN
insert into dbo.DimPopulation
(AlternateID, ZipCode, Population, InsertDate, ModifiedDate)
values
(@ID, @ZipCode, @Population, GETDATE(), GETDATE())
END;
if exists (select PopulationSK
from dbo.DimPopulation
where AlternateID = @ID)
BEGIN
update dbo.DimPopulation
set ZipCode = @ZipCode,
Population = @Population,
ModifiedDate = GETDATE()
where AlternateID = @ID
END;
END;
```

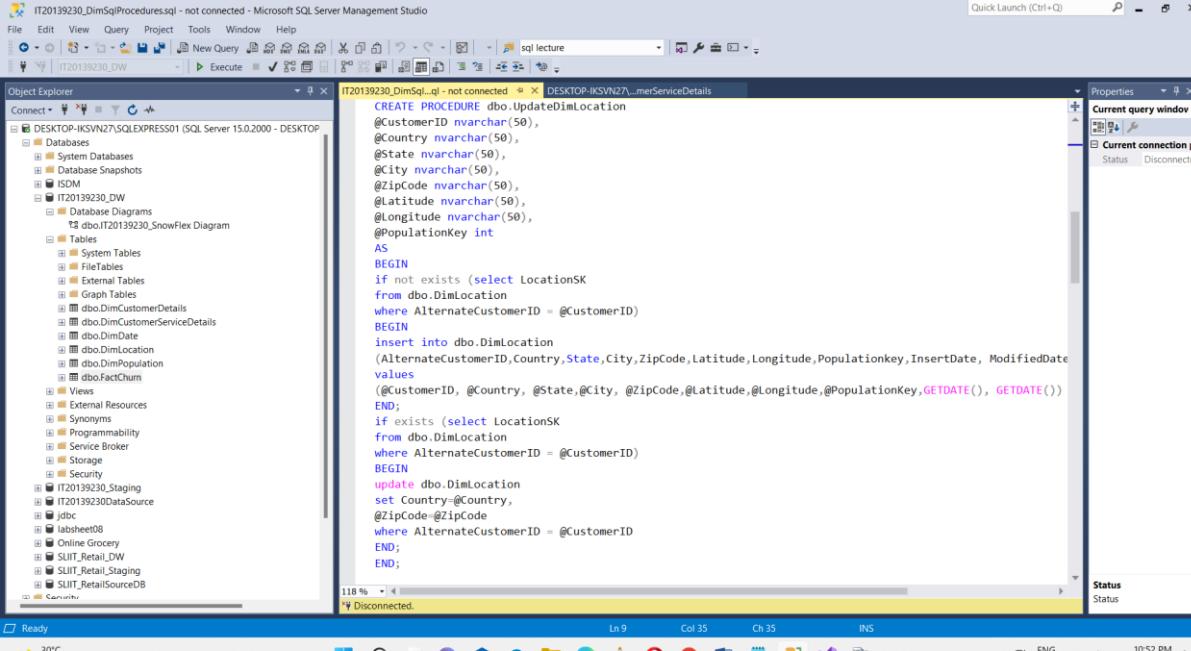
OLE DB Command SSIS tool used to execute, UpdateDimPopulation procedure, it used to insert data from Population details which is in IT20139230\_Staging to DimPopulation table which is in IT20139230\_DW. Above Stored Procedure ensure no duplicates are entered into the data warehouse table 'DimPopulation'. If there is an existing Population record, it will be updated with the latest record coming from staging table. Else, if it is a new record, it will be inserted into DimPopulation.





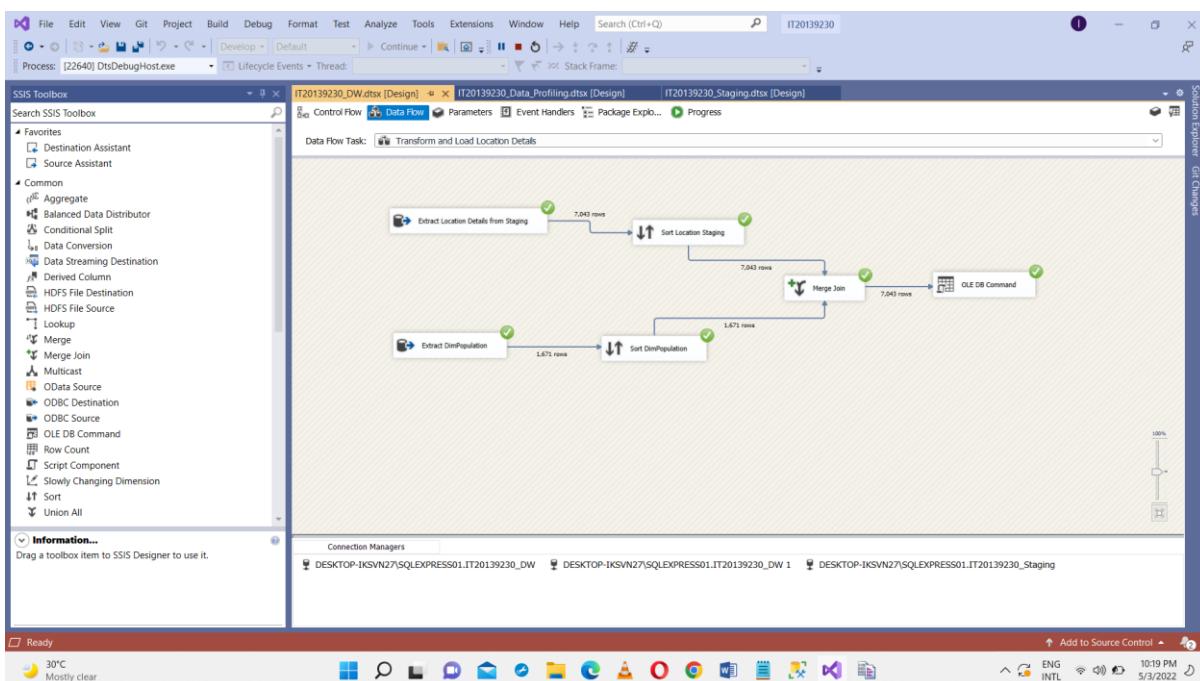
## DimLocation

Created a Procedure called UpdateDimLocation and executed in the IT20139230\_DW database

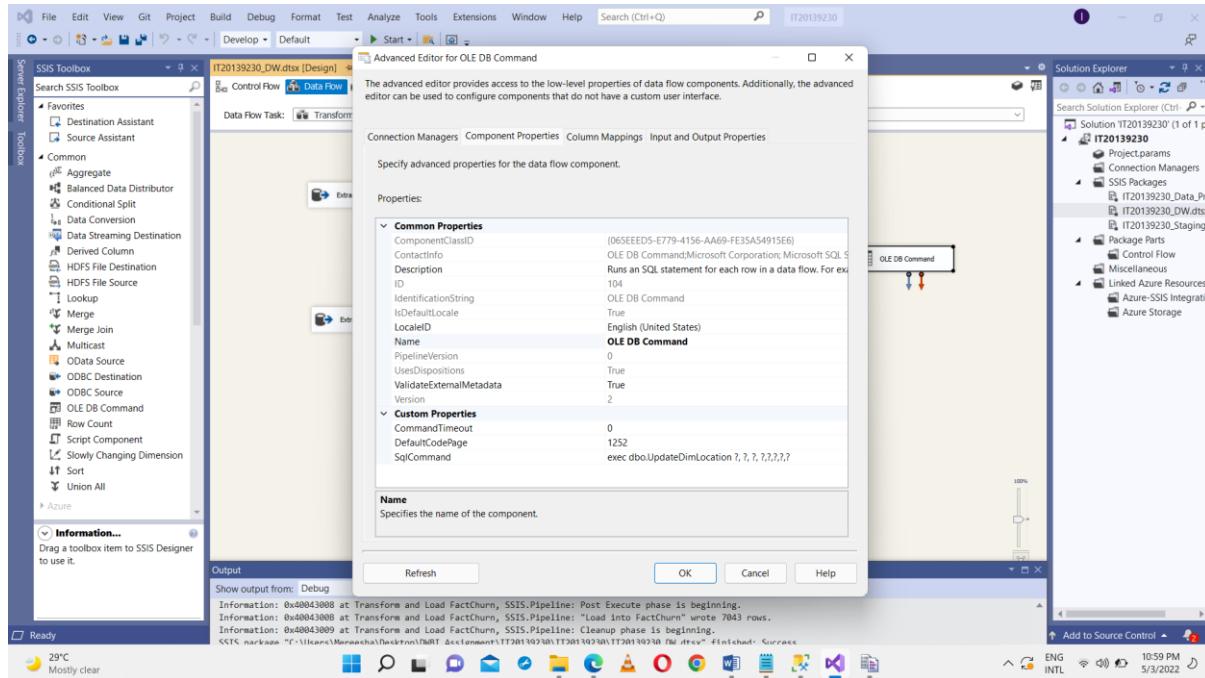


```
CREATE PROCEDURE [dbo].[UpdateDimLocation]
@CustomerID nvarchar(50),
@Country nvarchar(50),
@State nvarchar(50),
@City nvarchar(50),
@ZipCode nvarchar(50),
@Latitude nvarchar(50),
@Longitude nvarchar(50),
@PopulationKey int
AS
BEGIN
if not exists (select LocationSK
from dbo.DimLocation
where AlternateCustomerID = @CustomerID)
BEGIN
insert into dbo.DimLocation
(AlternateCustomerID,Country,State,City,ZipCode,Latitude,Longitude,Populationkey,InsertDate, ModifiedDate
values
(@CustomerID, @Country, @State,@City, @ZipCode,@Latitude,@Longitude,@PopulationKey,GETDATE(), GETDATE())
END;
if exists (select LocationSK
from dbo.DimLocation
where AlternateCustomerID = @CustomerID)
BEGIN
update dbo.DimLocation
set Country=@Country,
@ZipCode
where AlternateCustomerID = @CustomerID
END;
END;
```

Used sort to sort the details of the location details which are coming from staging and sorted DimPopulationSK which is coming from DimPopulation and merge all the details in order to come up with foreign keys.



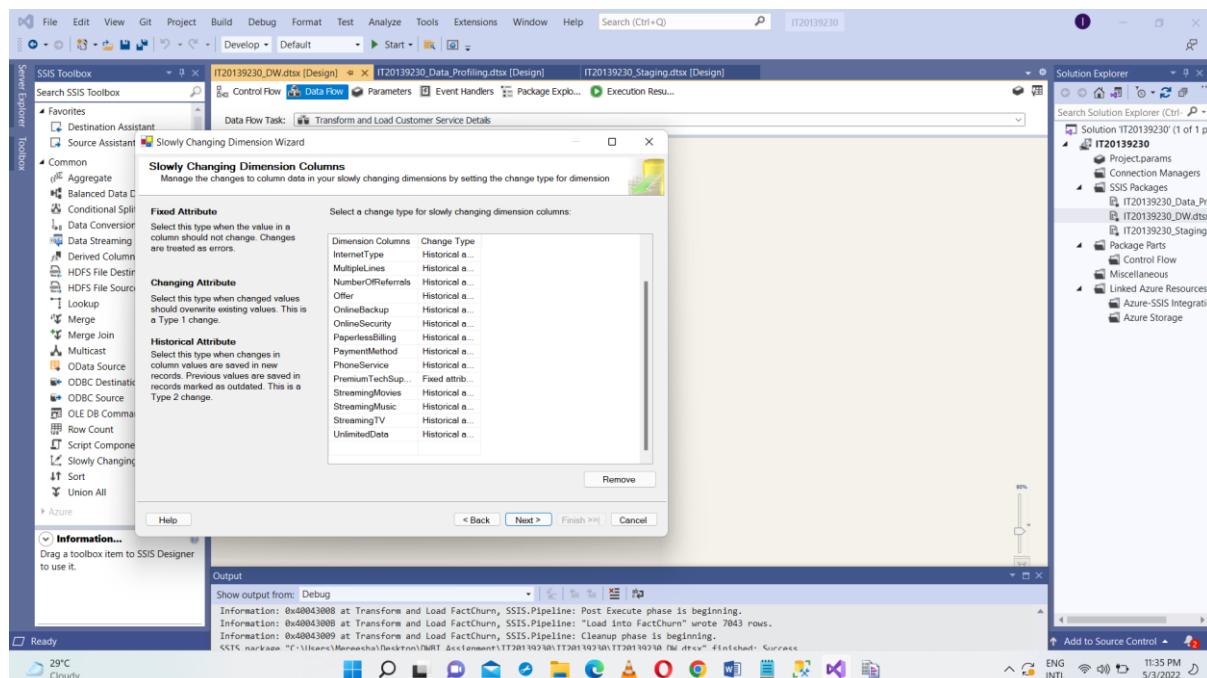
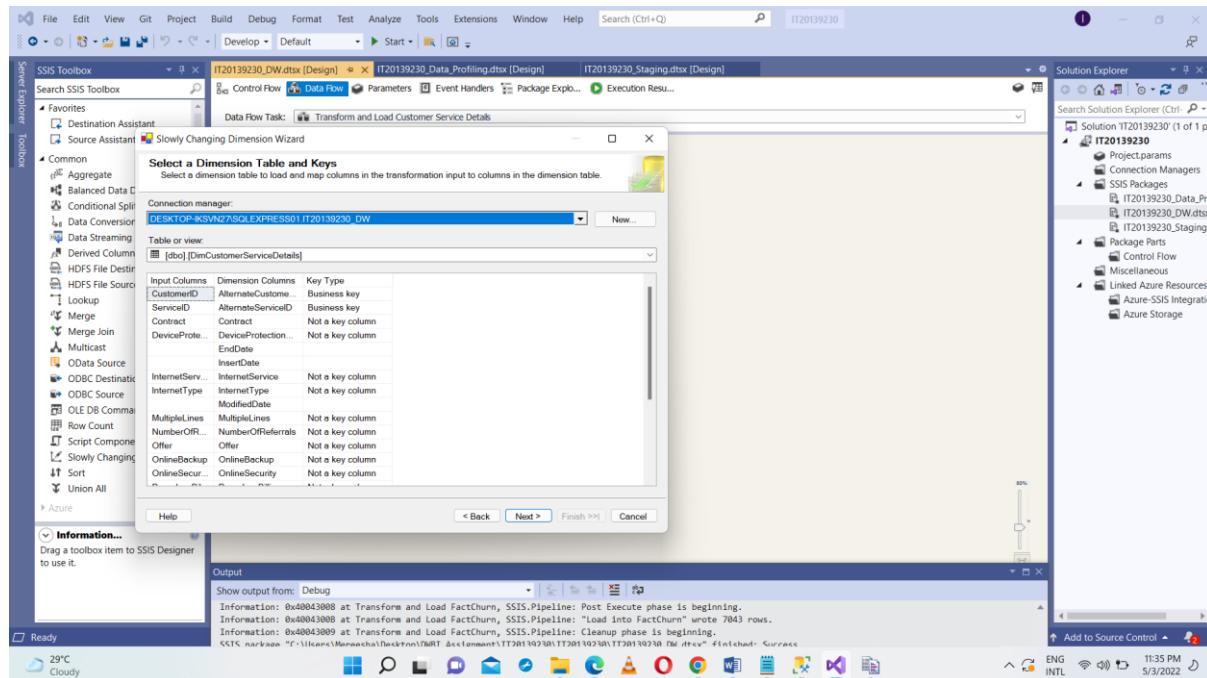
OLE DB Command SSIS tool used to execute, UpdateDimPopulation procedure, it used to insert data from Location details which is in IT20129230\_Staging to DimLocation table which is in IT20139230\_DW. Above Stored Procedure ensure no duplicates are entered into the data warehouse table 'DimLocation'. If there is an existing Population record, it will be updated with the latest record coming from staging table. Else, if it is a new record, it will be inserted into DimLocation.

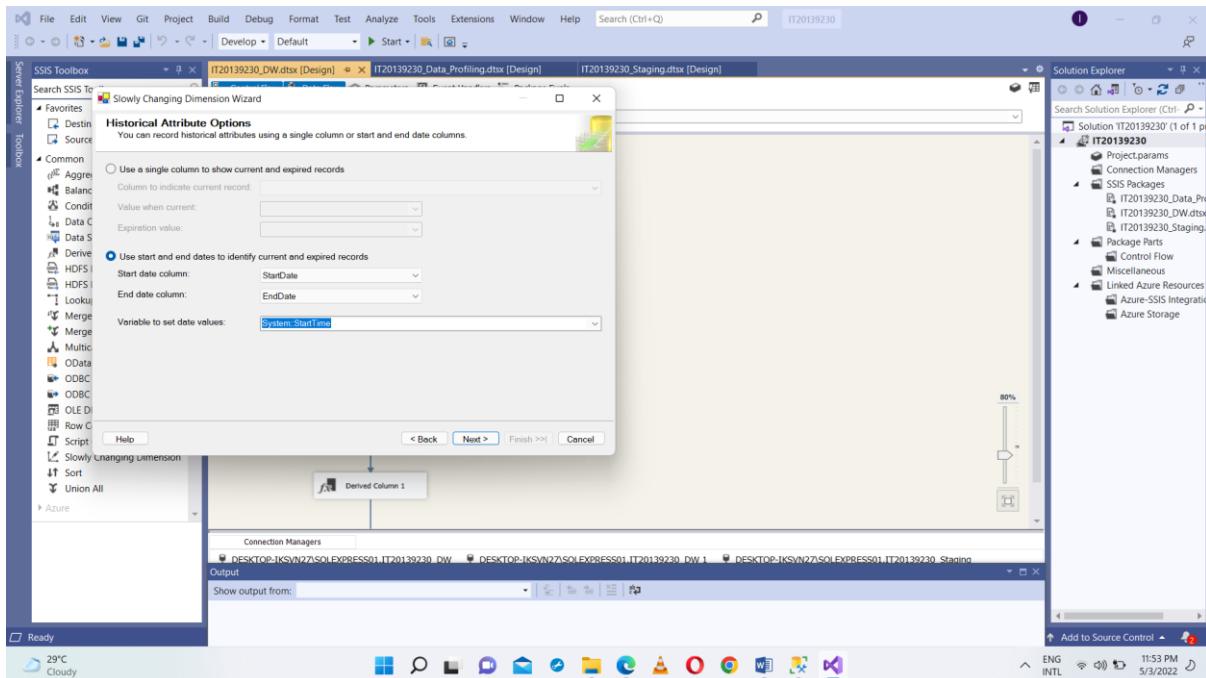


## DimCustomerServiceDetails

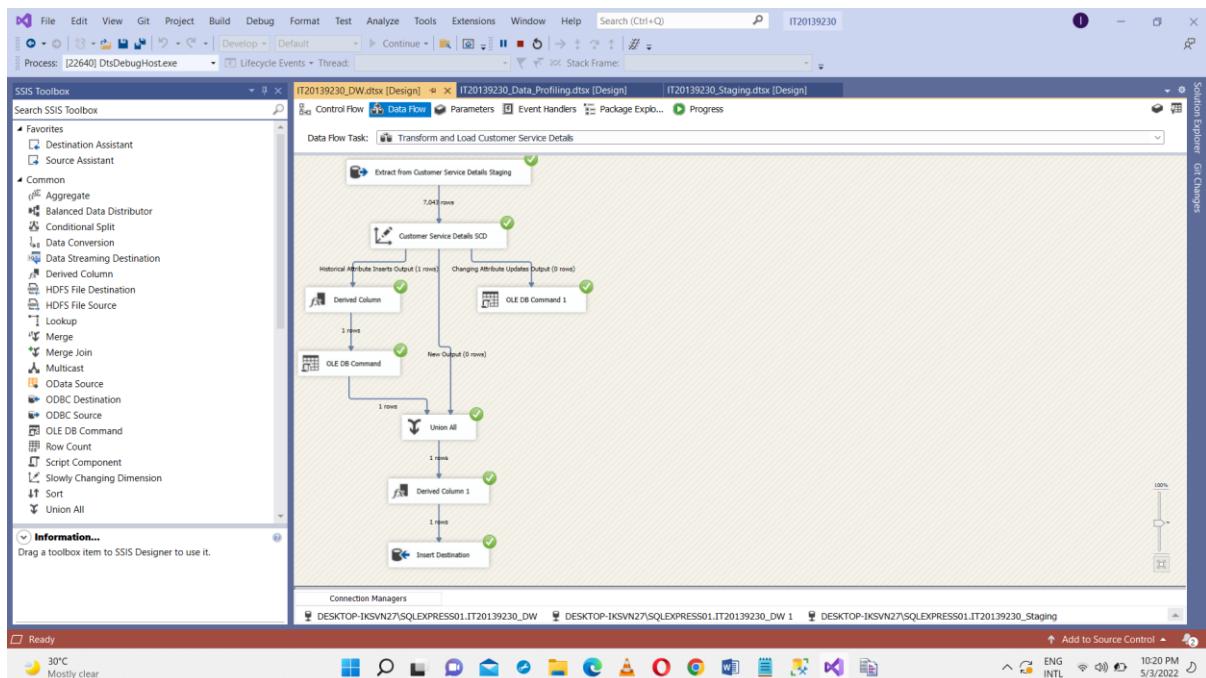
Extract the customer service details from staging and consider DimCustomerServiceDetails as a slowly dimension

In the SCD Configuration Wizard I set the configurations as below



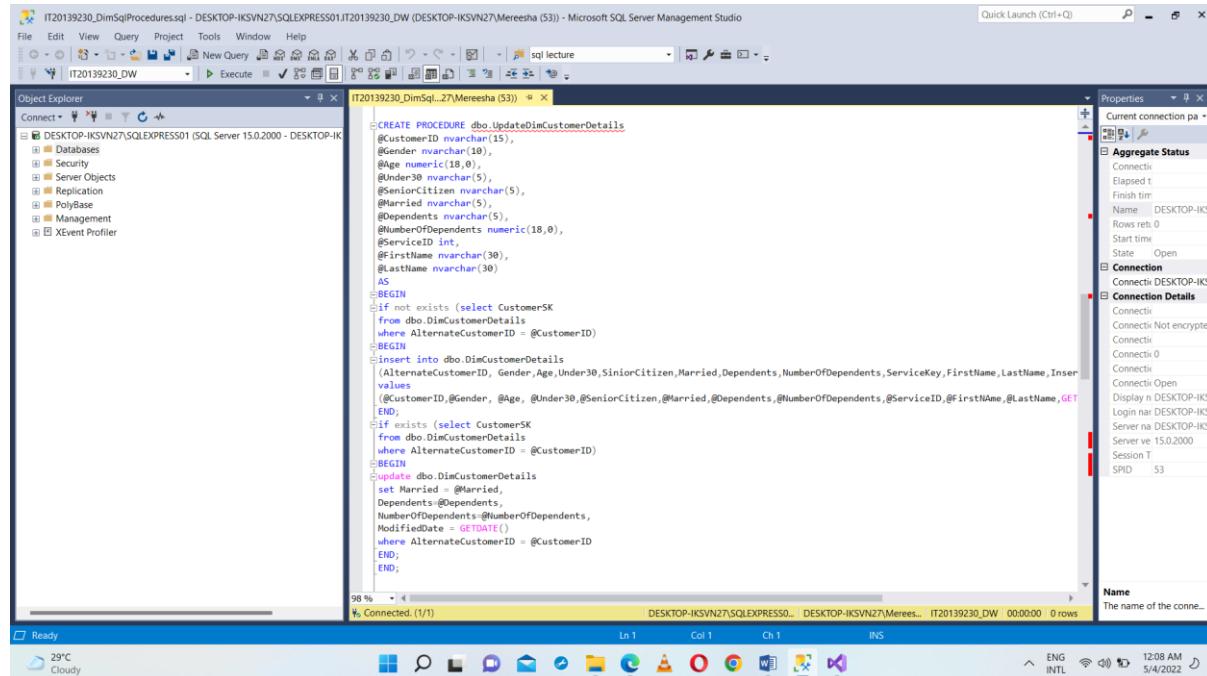


Once All Configurations done properly, it will automatically create the Slowly Changing Dimension as shown below.



## DimCustomerDetails

Created a Procedure called UpdateCustomerDetails and executed in the IT20139230\_DW database.

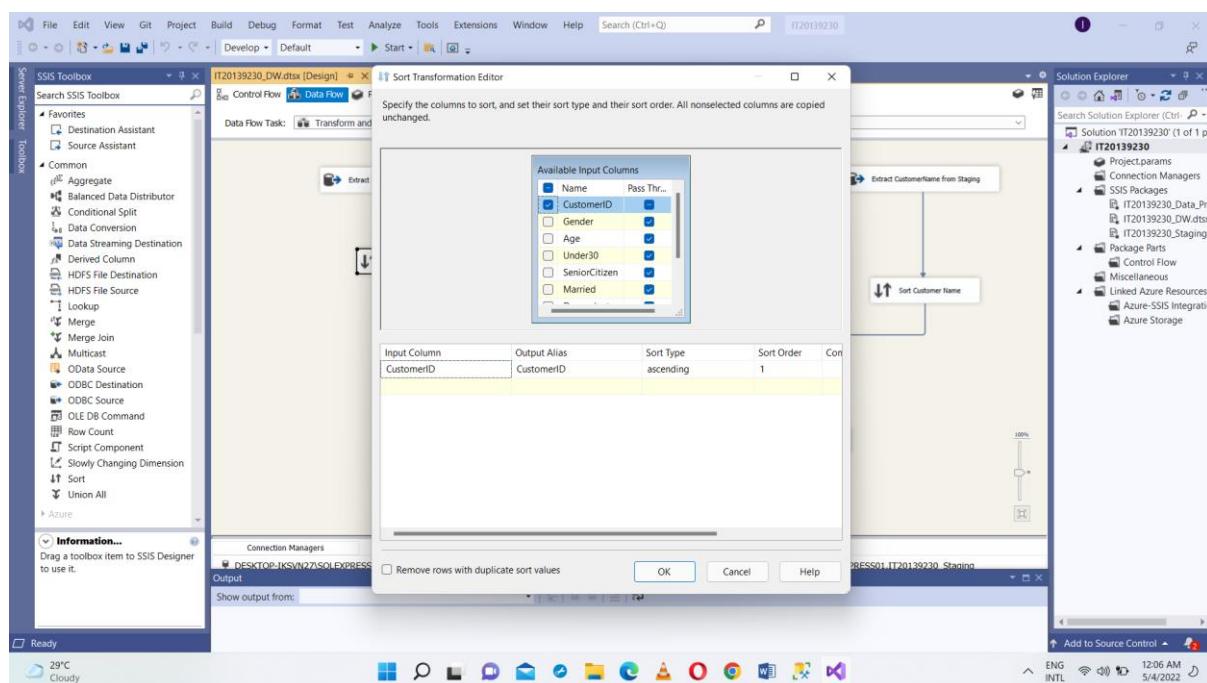


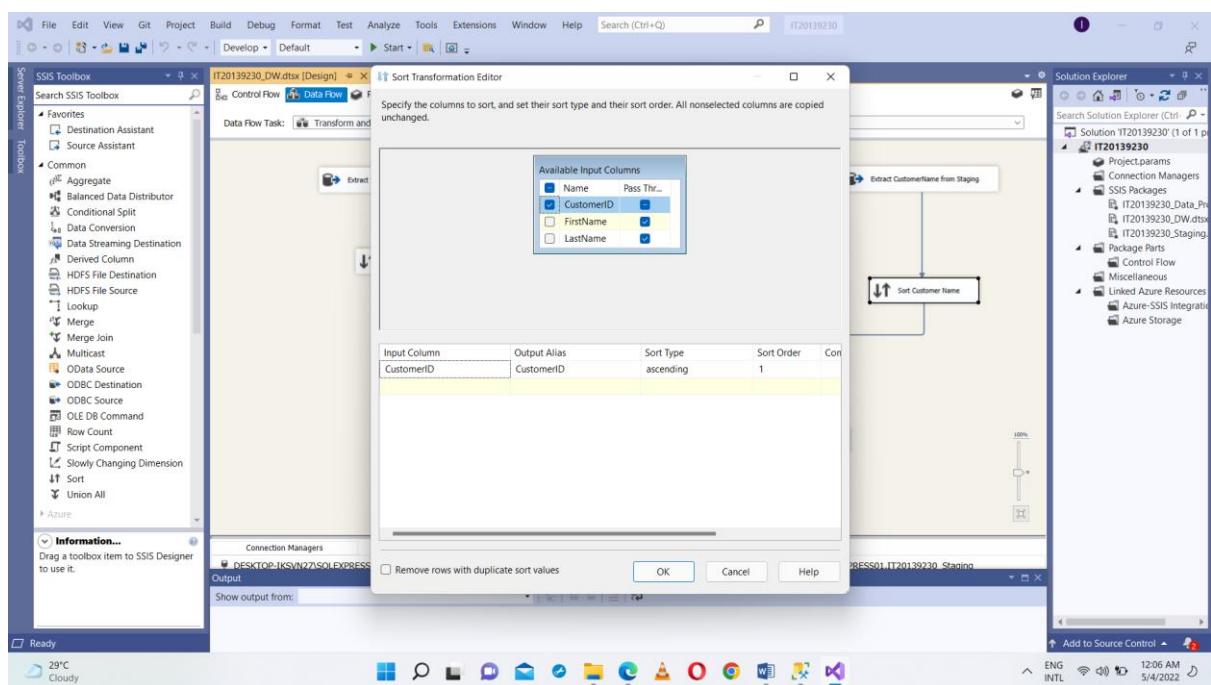
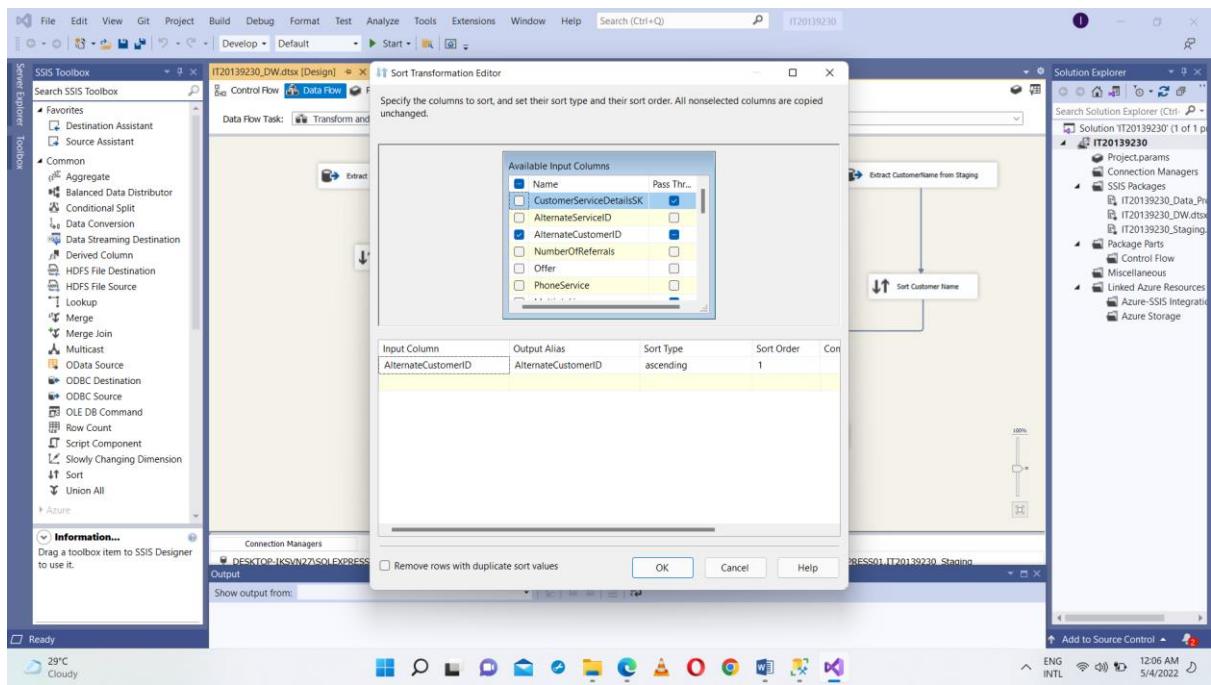
```

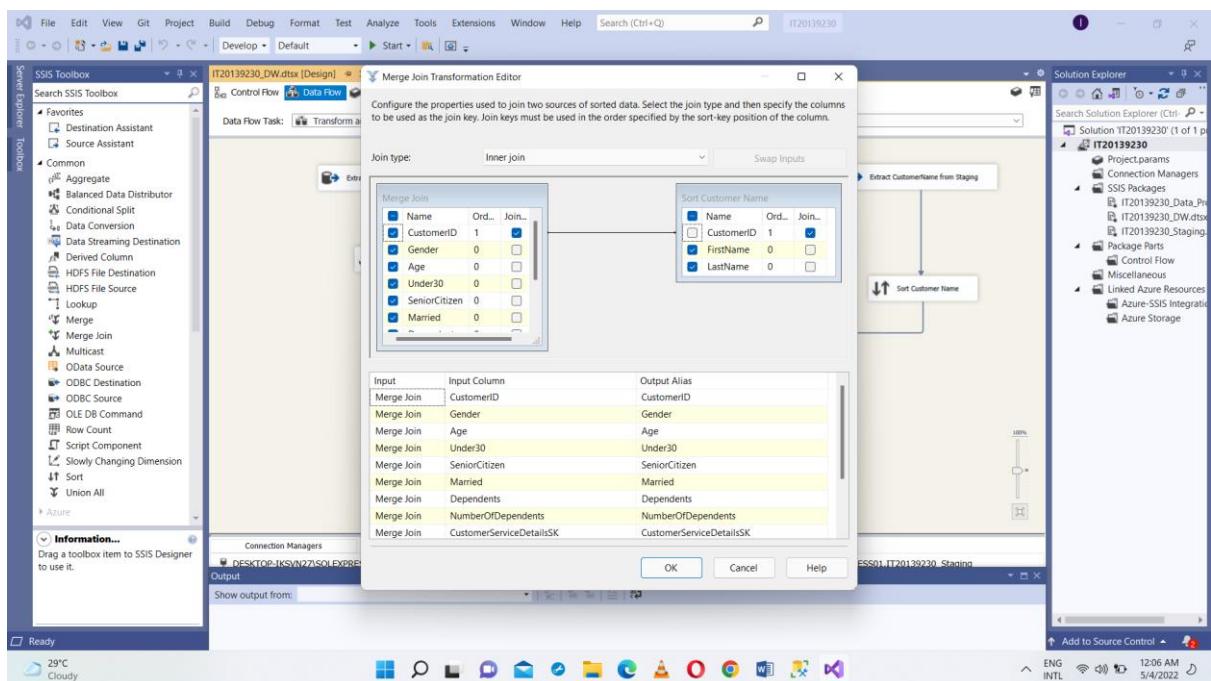
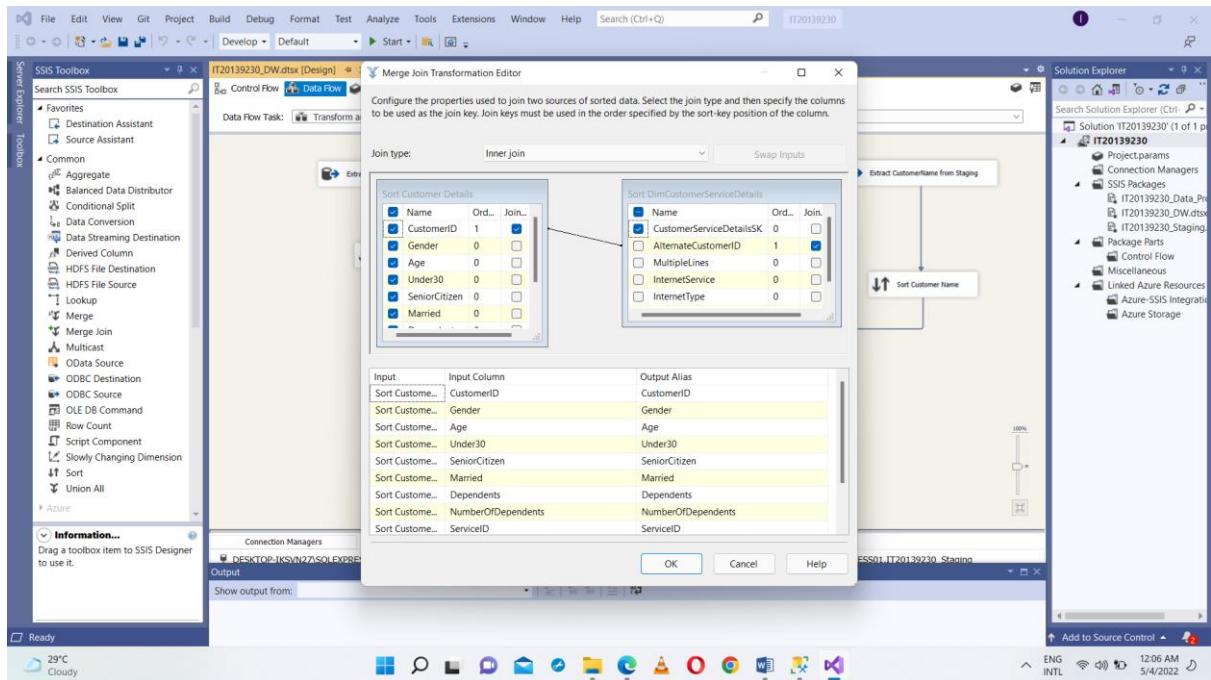
CREATE PROCEDURE [dbo].[UpdateCustomerDetails]
    @CustomerID nvarchar(15),
    @Gender nvarchar(10),
    @Age numeric(18,0),
    @Under30 nvarchar(5),
    @SeniorCitizen nvarchar(5),
    @Married nvarchar(5),
    @Dependents nvarchar(5),
    @NumberOfDependents numeric(18,0),
    @ServiceID int,
    @FirstName nvarchar(30),
    @LastName nvarchar(30)
AS
BEGIN
    IF NOT EXISTS (SELECT CustomerSK
                   FROM dbo.DimCustomerDetails
                   WHERE AlternateCustomerID = @CustomerID)
    BEGIN
        INSERT INTO dbo.DimCustomerDetails
        (AlternateCustomerID, Gender, Age, Under30, SeniorCitizen, Married, Dependents, NumberOfDependents, ServiceKey, FirstName, LastName, InsertDate)
        VALUES
        (@CustomerID, @Gender, @Age, @Under30, @SeniorCitizen, @Married, @Dependents, @NumberOfDependents, @ServiceID, @FirstName, @LastName, GETDATE())
    END;
    ELSE IF EXISTS (SELECT CustomerSK
                    FROM dbo.DimCustomerDetails
                    WHERE AlternateCustomerID = @CustomerID)
    BEGIN
        UPDATE dbo.DimCustomerDetails
        SET Married = @Married,
            Dependents = @Dependents,
            NumberOfDependents = @NumberOfDependents,
            ModifiedDate = GETDATE()
        WHERE AlternateCustomerID = @CustomerID
    END;
END;
END;

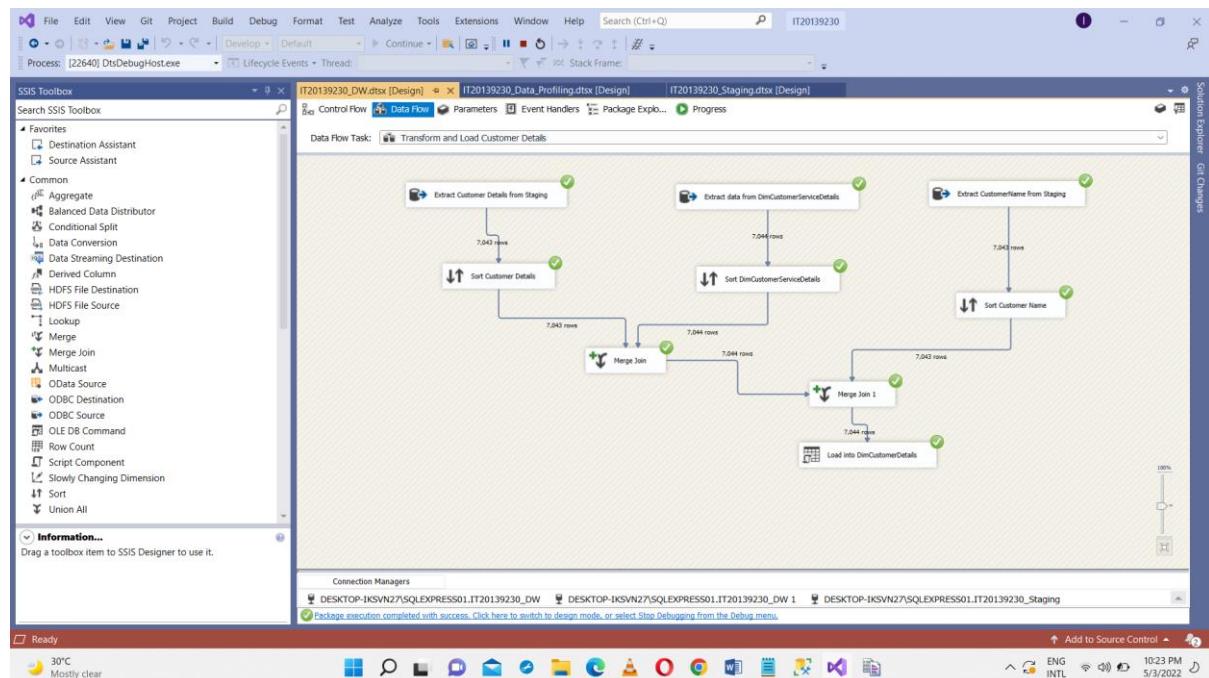
```

Extract the customer details from staging and using Sort, sorted the details of the customer and also DimCustomerDetails references to CustomerServiceDetailsSk. In order to come up with CustomerServiceDetailsSk extract customer service details from DimCustomerServiceDetails and sorted it SK. There after merge those sorted details. Customer names were in different staging table in order to have those details extract CustomerName table from the staging and sorted the relevant details. After all merge those details and load into DimCustomerDetails.

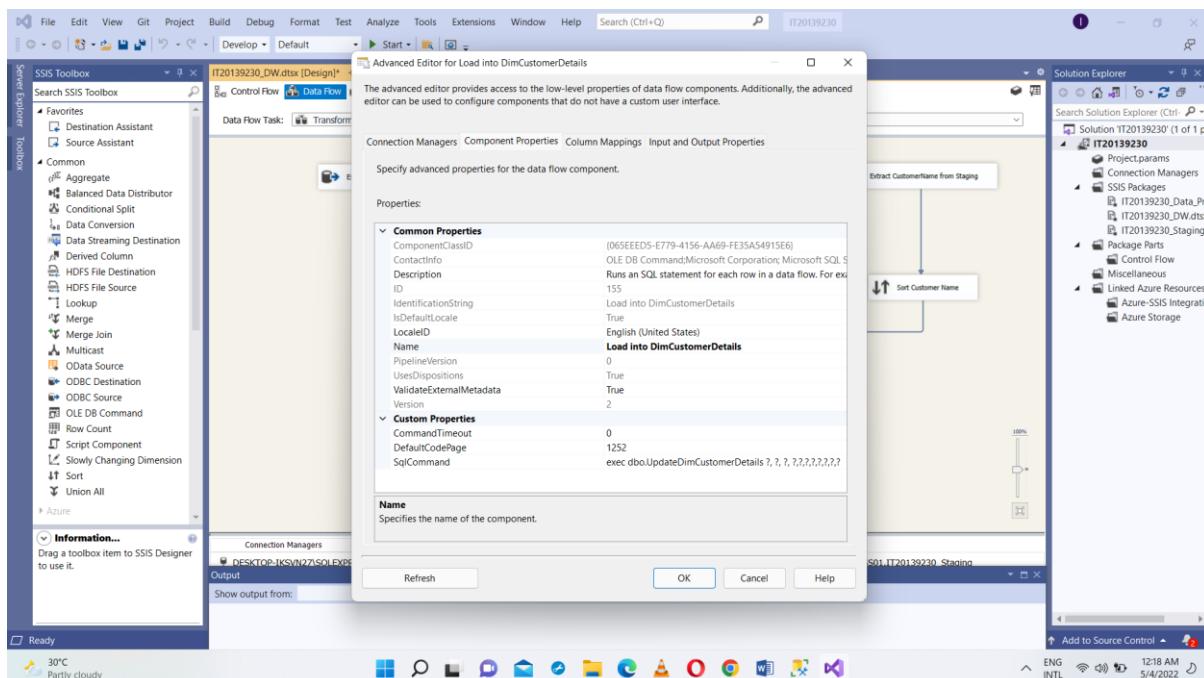






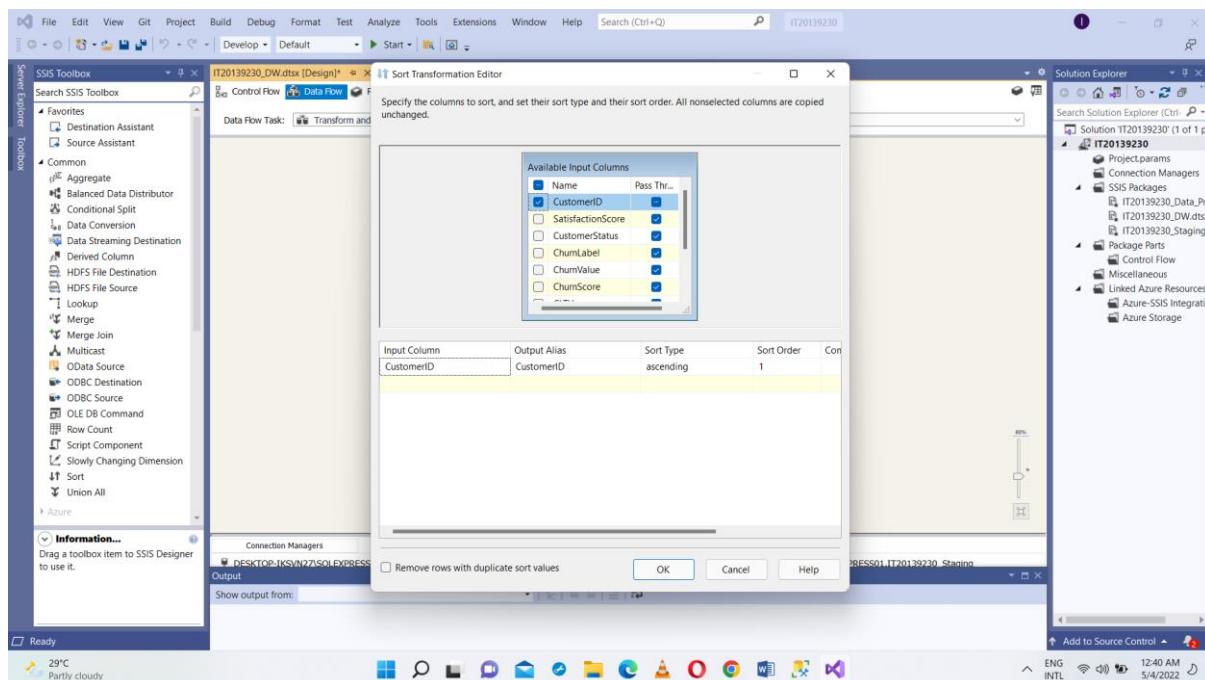
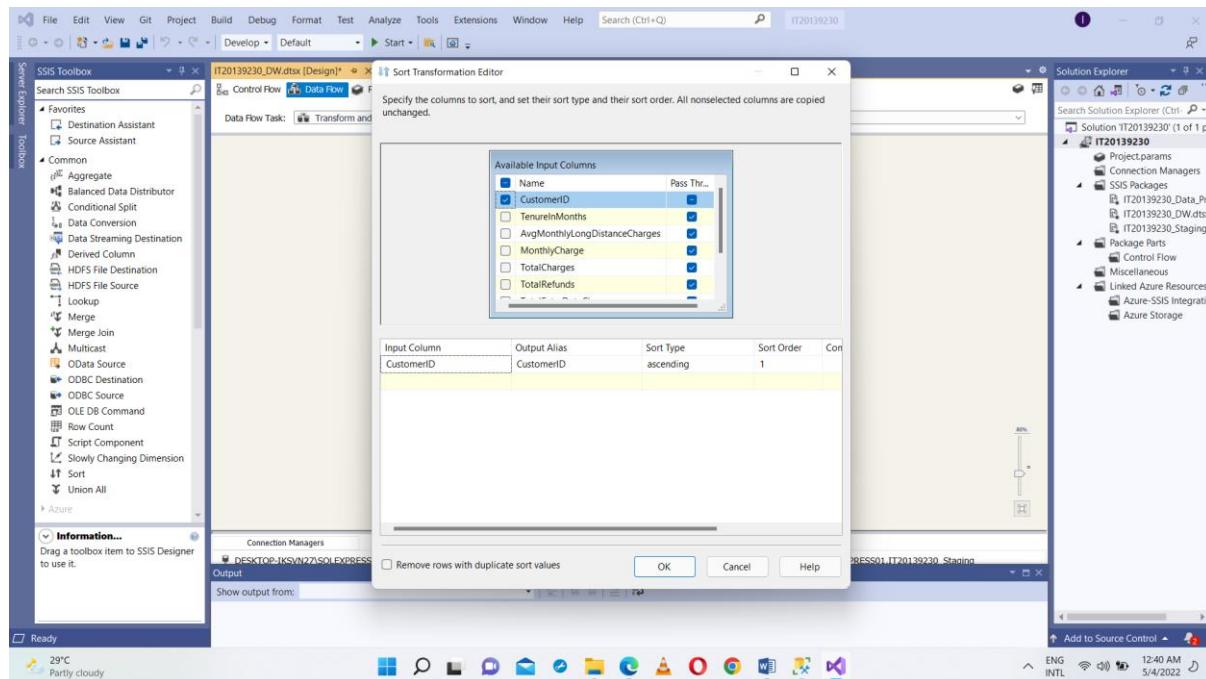


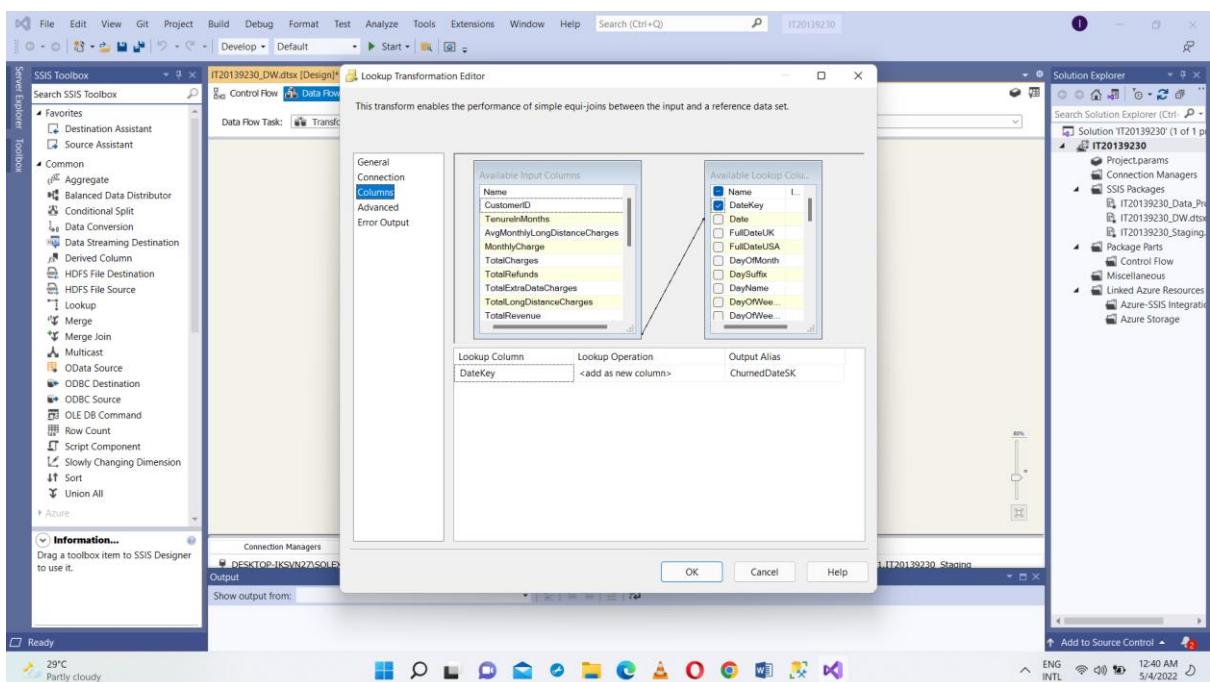
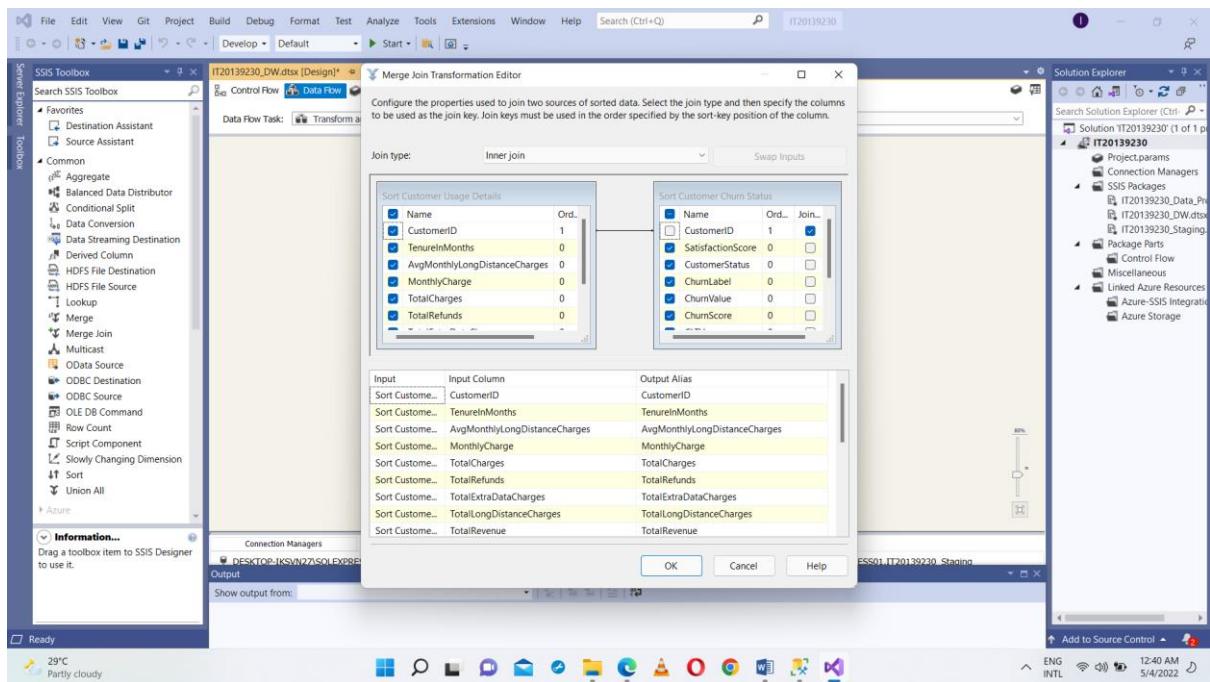
OLE DB Command SSIS tool used to execute, UpdateDimCustomerDetails procedure, it used to insert data from Population details which is in IT20129230\_Staging to DimCustomerDetails table which is in IT20139230\_DW. Stored Procedure ensure no duplicates are entered into the data warehouse table ‘DimCustomerDetails’. If there is an existing Population record, it will be updated with the latest record coming from staging table. Else, if it is a new record, it will be inserted into DimCustomerDetails.

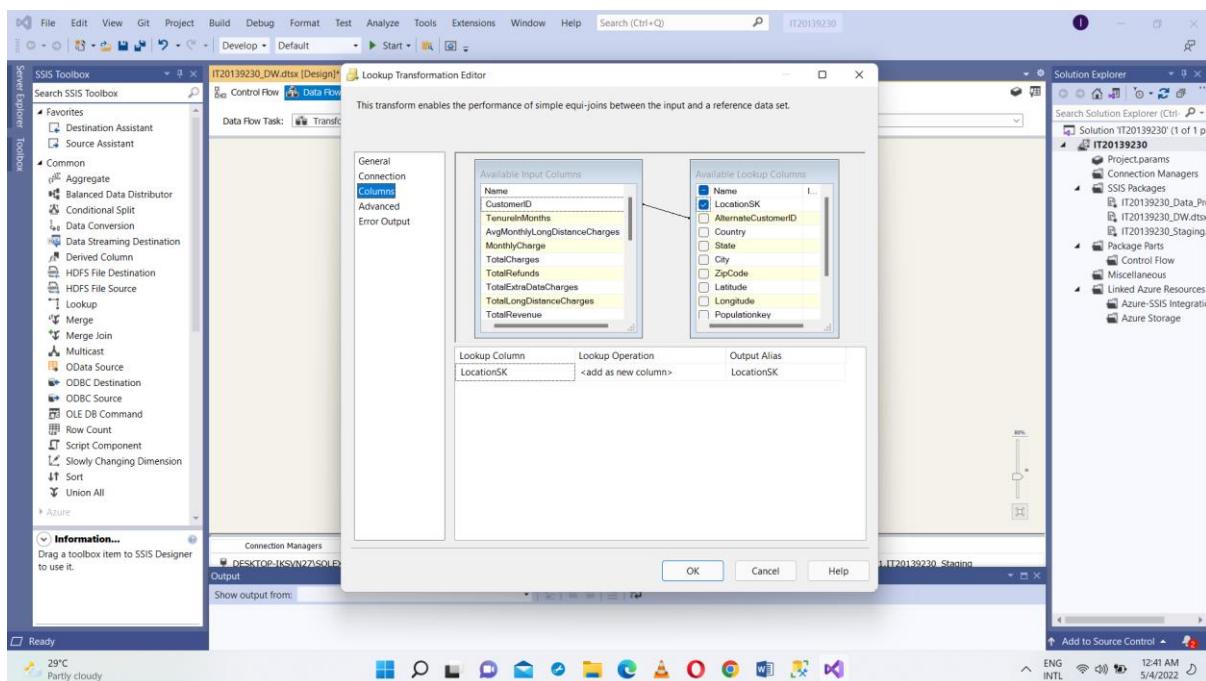
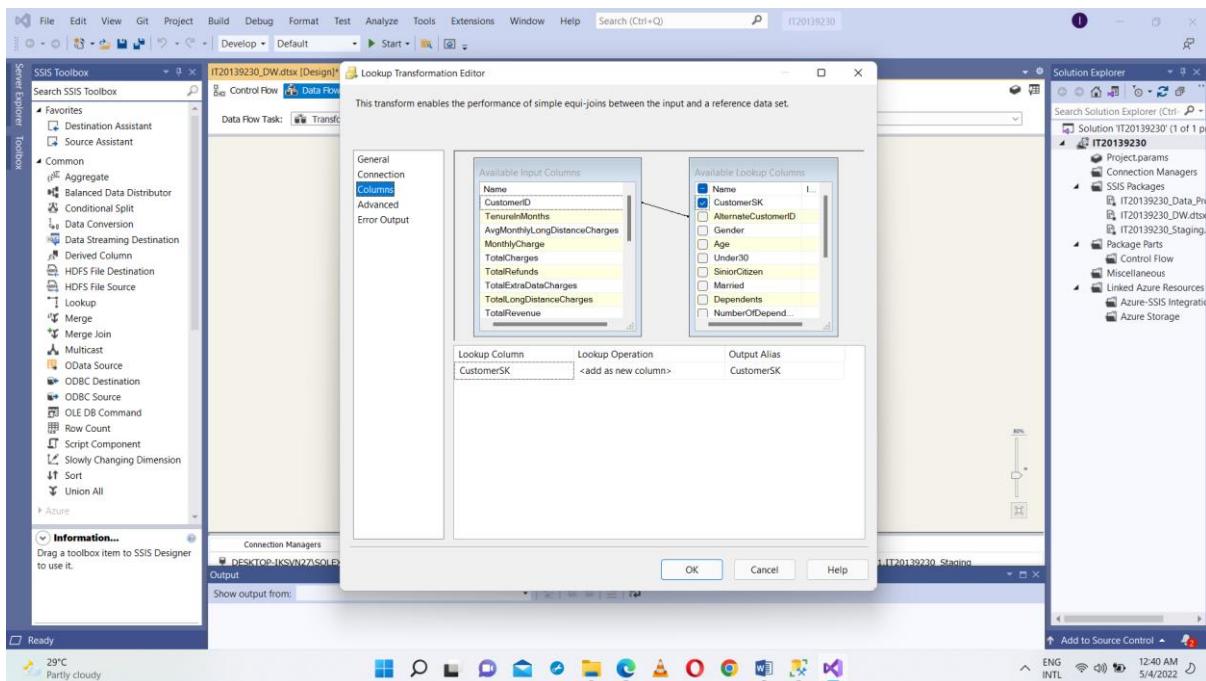


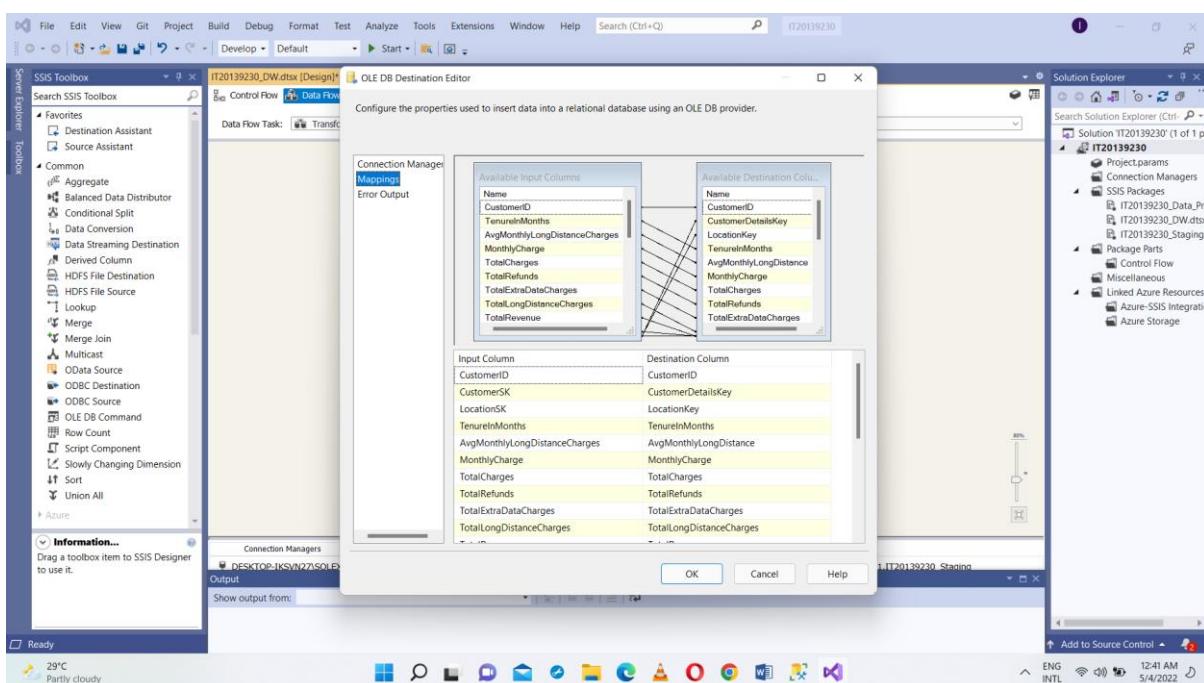
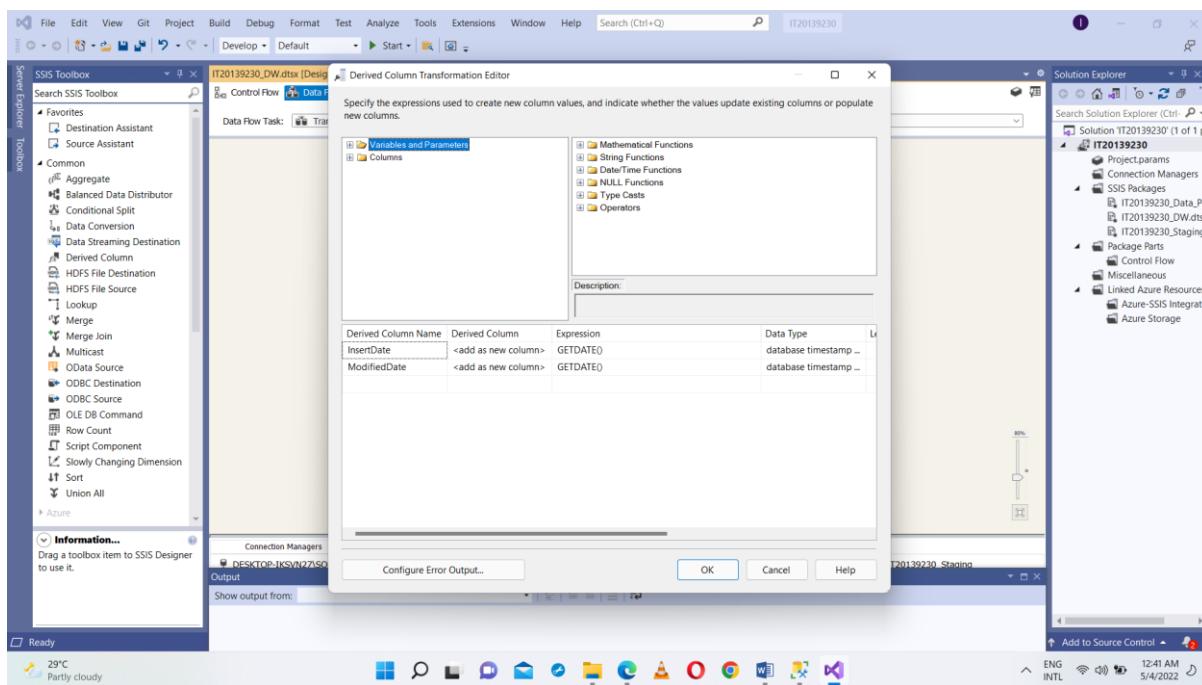
## FactChurn

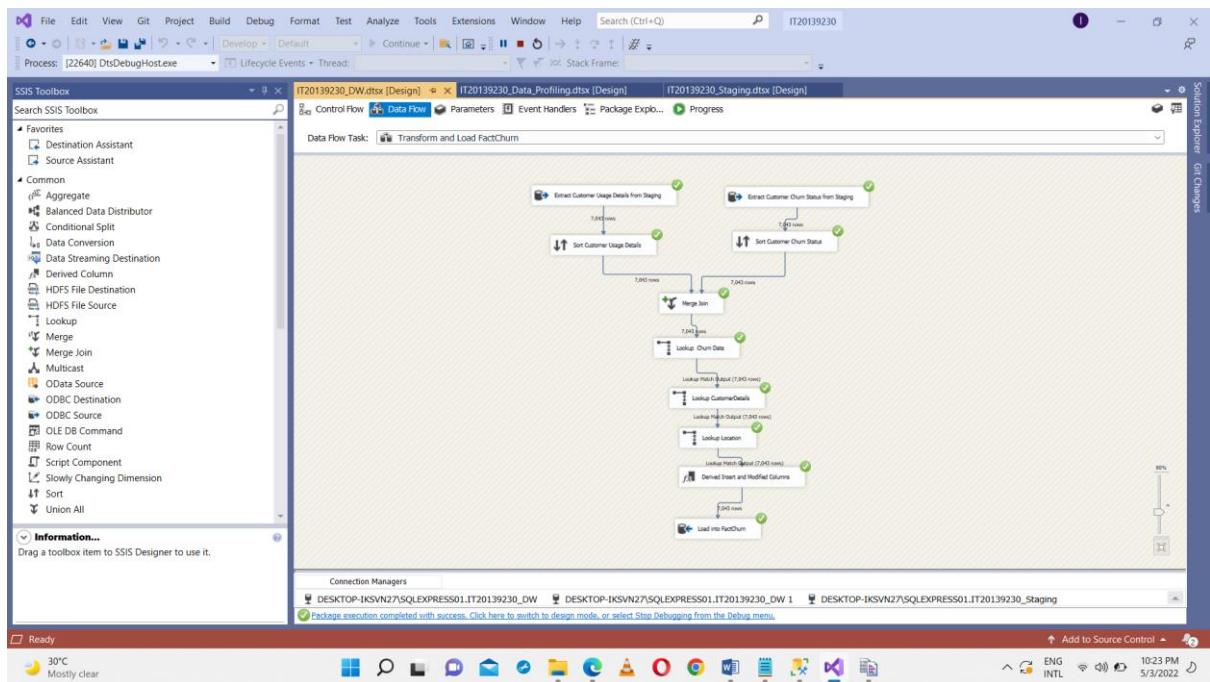
Extract the customer usage details and customer churn status details from staging then sort the customer usage details and customer churn status details using Sort. Using Merge join join those details. There after use Lookup for Churn Date in order to having a references to DimDate dimention. Again used Lookup for DimCustomerDetails to having it Sk as a references in FactChurn table. Samething was done to comeup with LocationSK as a references. There after derived 2 columns as Insert and Modified date. After all loading transformed data into FactChurn.











## Step 6:ETL development – Accumulating fact tables

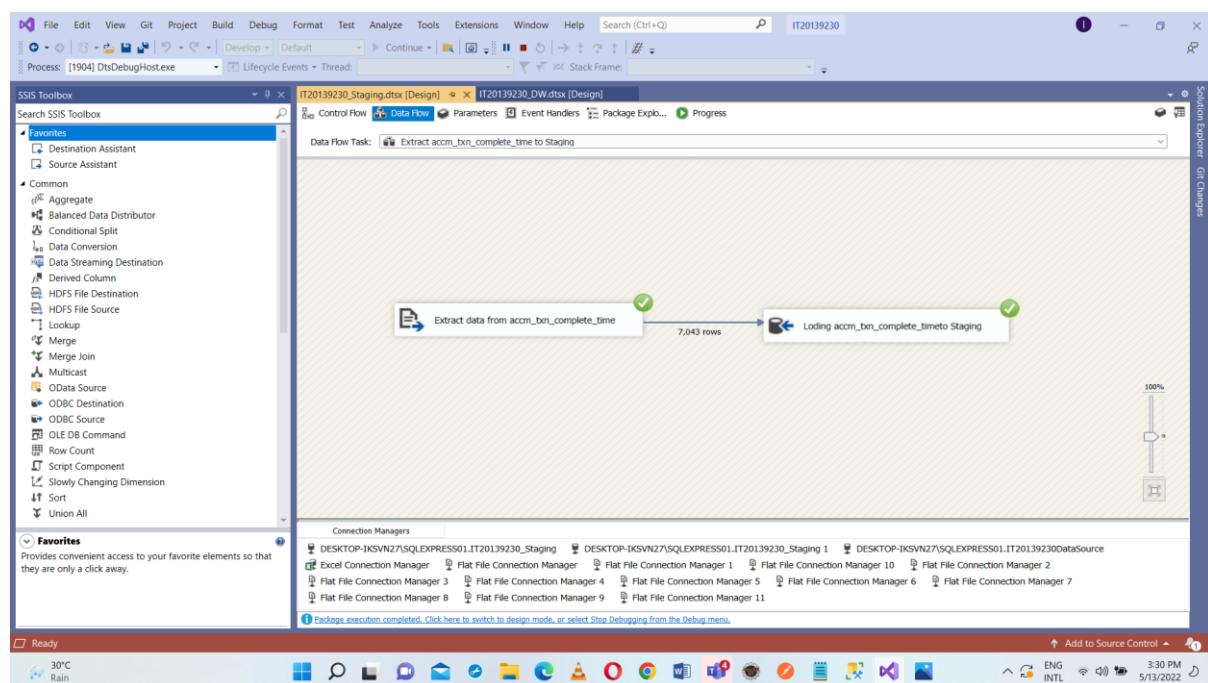
I have extended my FactChurn table with adding accm\_txn\_create\_time, accm\_txn\_complete\_time and txn\_process\_time\_hours new columns.

Setted accm\_txn\_create\_time to be equal to the current system date when load data to FactChurn table.

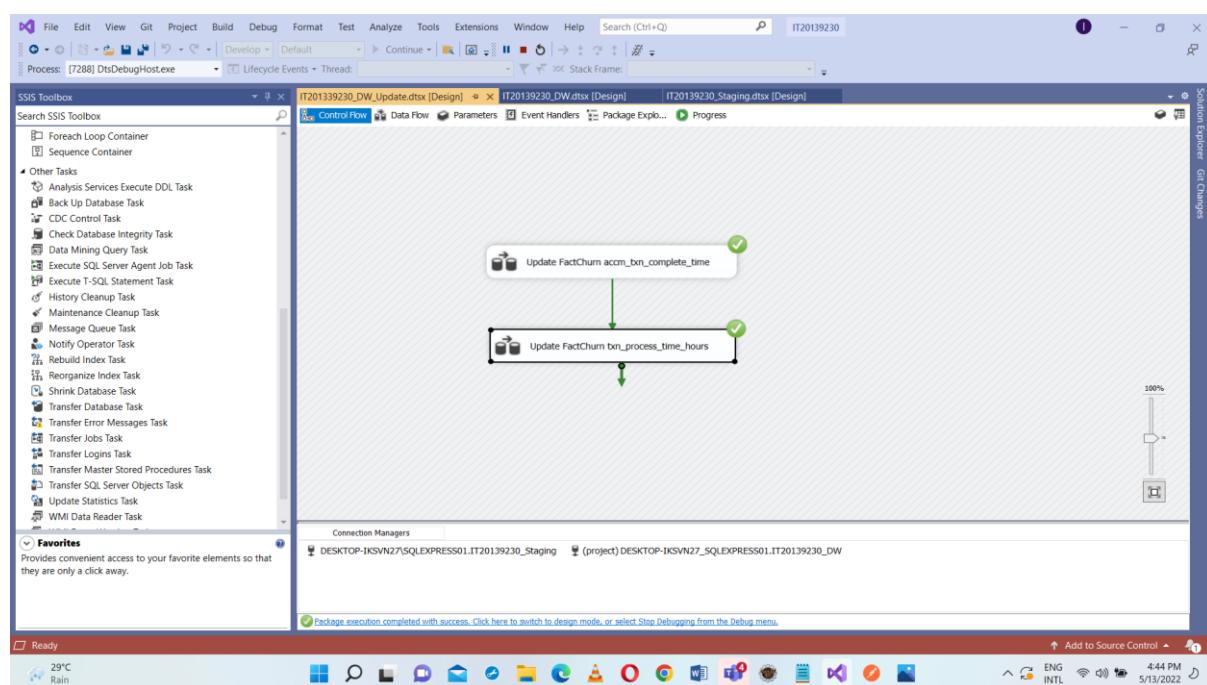
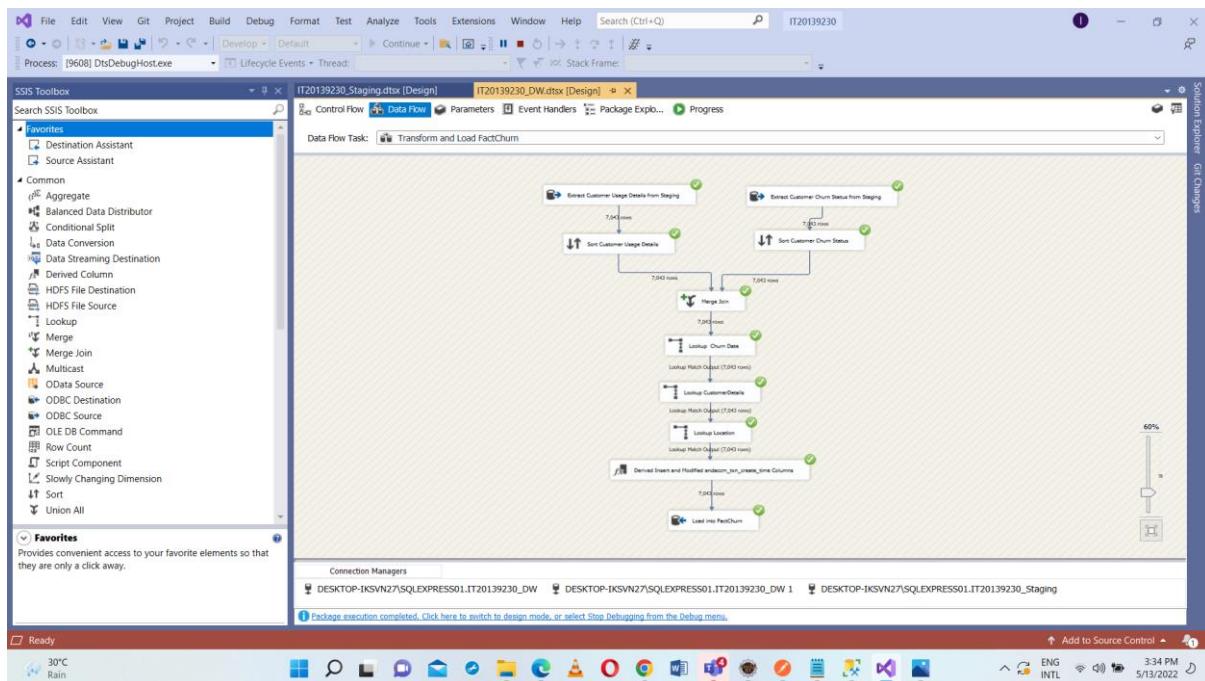
Prepared a separate csv file which contains CustomerID and accm\_txn\_complete\_time and put it into the IT20139230\_Staging Database.

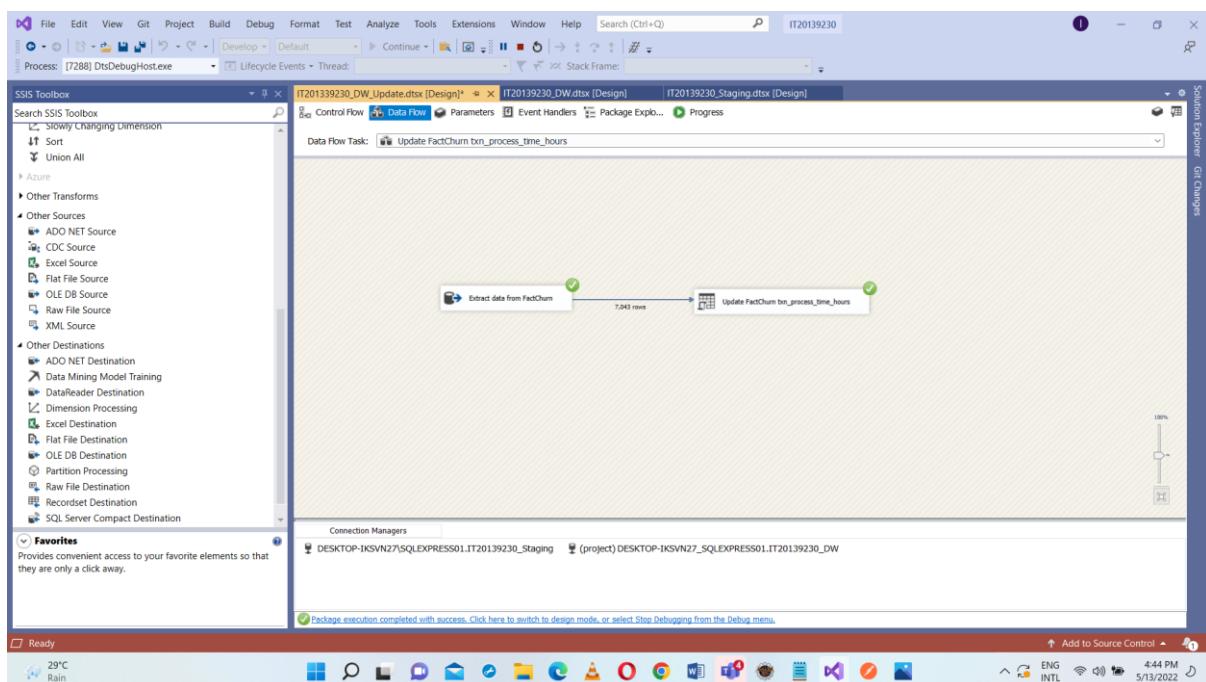
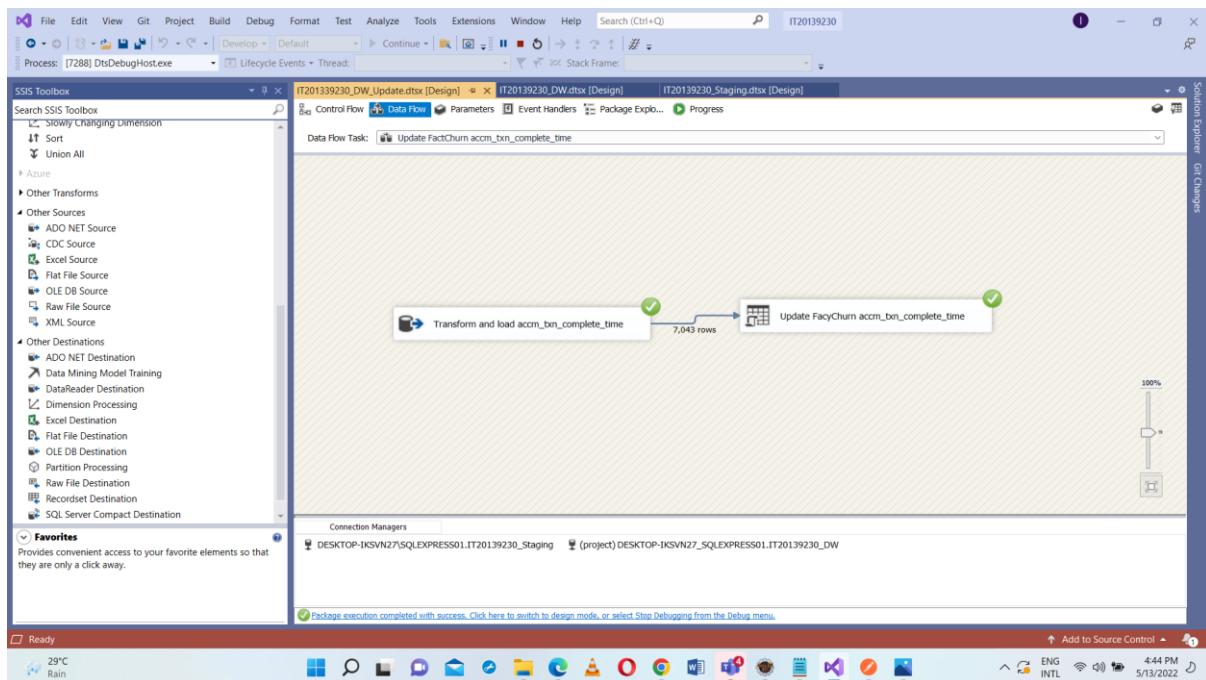
Designed a separate ETL ssis package which names as IT20139230\_DW\_Update which reads data from accm\_txn\_complete\_time which is in IT20139230\_Staging Database and update the FactChurn accm\_txn\_complete\_time in FactChurn table which is in IT20139230\_DW.

Updated txn\_process\_time\_hours as well.



Configurations were done to add new derived columns to accm\_txn\_create\_time to get current system date when load data to FactChurn table.





**References:**

<https://courseweb.sliit.lk/course/view.php?id=4696>

<https://www.geeksforgeeks.org/etl-process-in-data-warehouse/>

<https://www.javatpoint.com/etl-process-in-data-warehouse>

[https://www.youtube.com/watch?v=oF\\_2uDb7DvQ](https://www.youtube.com/watch?v=oF_2uDb7DvQ)