SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

Data Warehousing and Business Intelligence

Assignment: 01

ID No: IT20024918
Name: Mudunkotuwa N.K
Batch: DS weekend

Table of Contents

(01)Data Set selection 3
ER Diagram.......... 4
(02)Preparation of Data Sources.......... 5
(03)Solution Architechture..........8
(04)Data Warehouse design and development 10
(05)ETL Development..........12
Data Extraction..........12
Extract Hotel Data to Staging 12
Extract Owner Data to Staging..........14
Extract Customer Data to Staging..........15
Extract MenuType Data to Staging..........16
Extract CustomerType Data to Staging..........17
Extract RoomType Data to Staging..........18
Extract CustomerAddress Data to Staging..........19
Extract Booking Data to Staging..........20
Overall control flow..........21
Data Profiling..........22
Data Transforming and Loading..........23
Load Owner Data from Staging to Data Warehouse..........24
Load Hotel Data from Staging to Data Warehouse..........25
Load MealType Data from Staging to Data Warehouse..........27
Load RoomType Data from Staging to Data Warehouse..........28
Load CustomerType Data from Staging to Data Warehouse..........29
Load Customer Data from Staging to Data Warehouse..........30
Dim Date creation..........32
Load Fact Table..........35
Overall ETL to data warehouse..........36
(06)ETL Development – Accumulating fact tables..........37

(1)Data Set Selection

- This data set contains the details of hotel booking demand for different hotels by different customers during year 2015 to 2017.
- Each of the hotel has a different owner.
- When the customer places a booking ,according to that they can select a room and a meal they prefer to have.
- Necessary modifications has been done for the data set in order to meet the requirements.

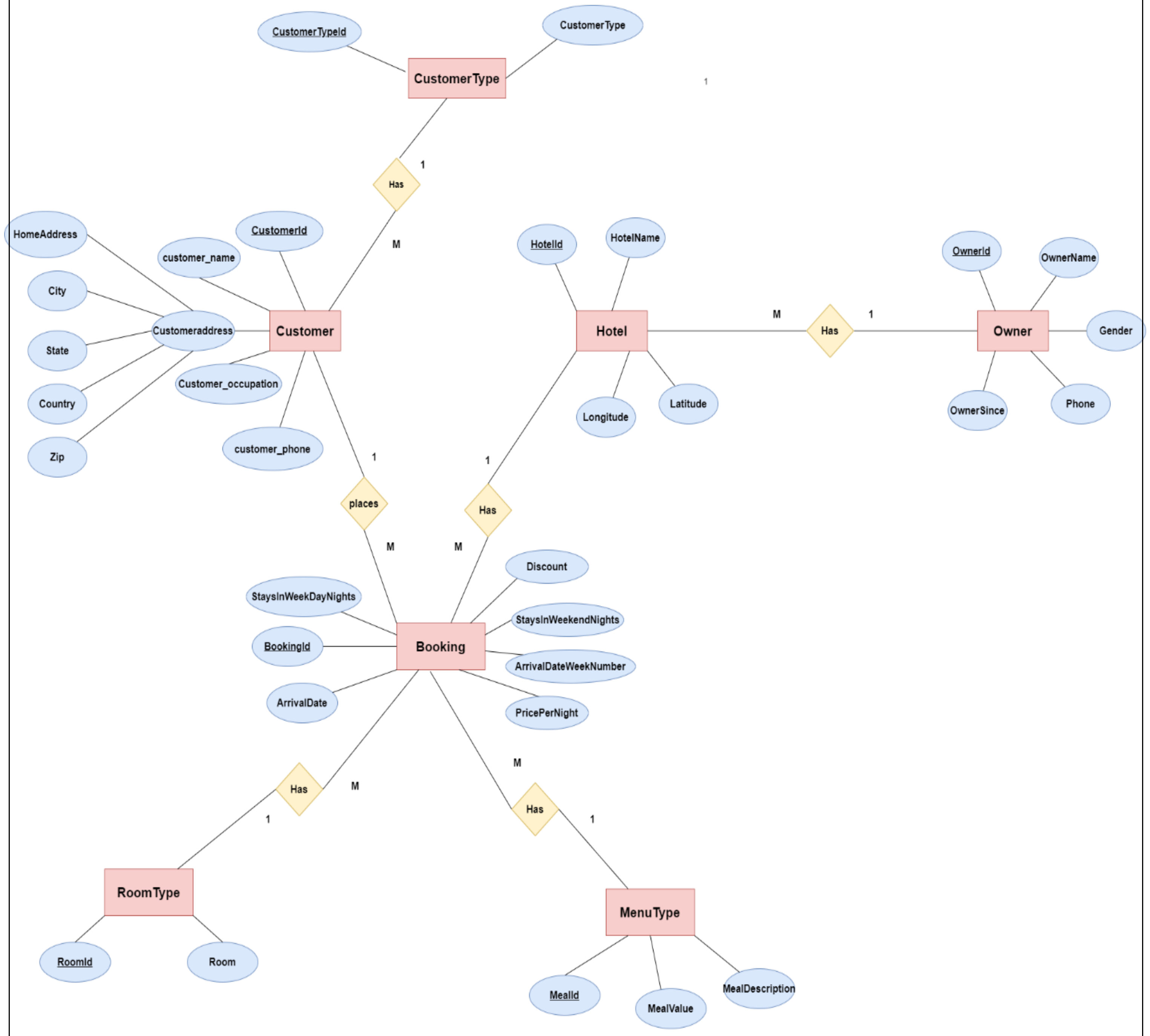
Data set : Cleaned hotel booking demand

Source :Kaggle

Link to the source

https://www.kaggle.com/datasets/rpereiracruz/cleaned-hotel-bookings

ER Diagram


CustomerTypeId
CustomerType
CustomerType
1
Has
1
M
HomeAddress
CustomerId
customer_name
City
Customeraddress
Customer
State
Customer_occupation
Country
customer_phone
Zip
HotelId
HotelName
Hotel
M
Has
1
OwnerId
OwnerName
Owner
Gender
OwnerSince
Phone
Longitude
Latitude
1
places
1
Has
M
M
Discount
StaysInWeekDayNights
StaysInWeekendNights
BookingId
Booking
ArrivalDateWeekNumber
ArrivalDate
PricePerNight
Has
M
1
M
Has
1
RoomType
RoomId
Room
MenuType
MealId
MealValue
MealDescription

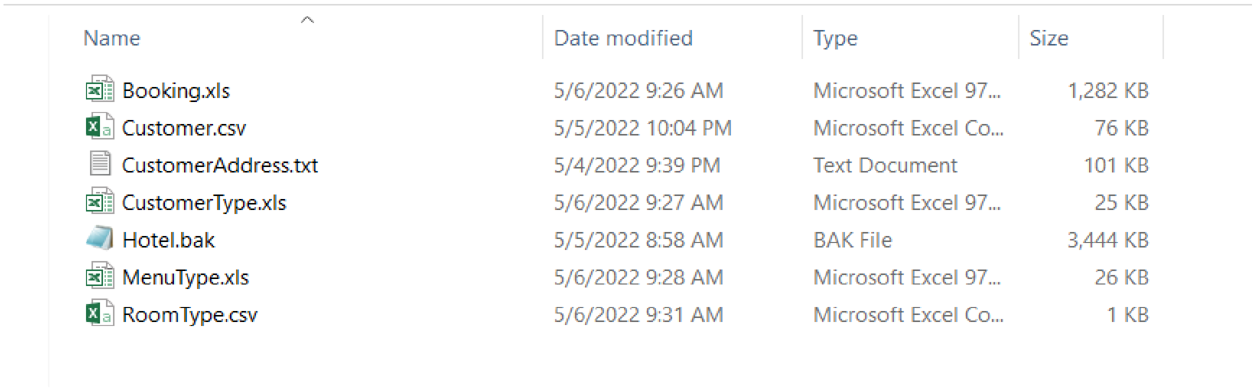
(2)Preparation of Data Sources

The original data set was in one CSV file. Data in the file has been separated into 7 different file types as Excel, CSV, bak and text.

Table	File Type
Booking	Excel file (.xls)
Customer	CSV file (.csv)
Customer Address	Text file (.txt)
CustomerType	Excel file (.xls)
MenuType	Excel file (.xls)
RoomType	CSV file (.csv)
Hotel	Bak file(.bak)

Hotel.bak file is consists with the Hotel.csv and Owner.csv files. Hotel.bak file was imported to the Hotel database.

After the making files using different formats you can see the multiple data sources as below.

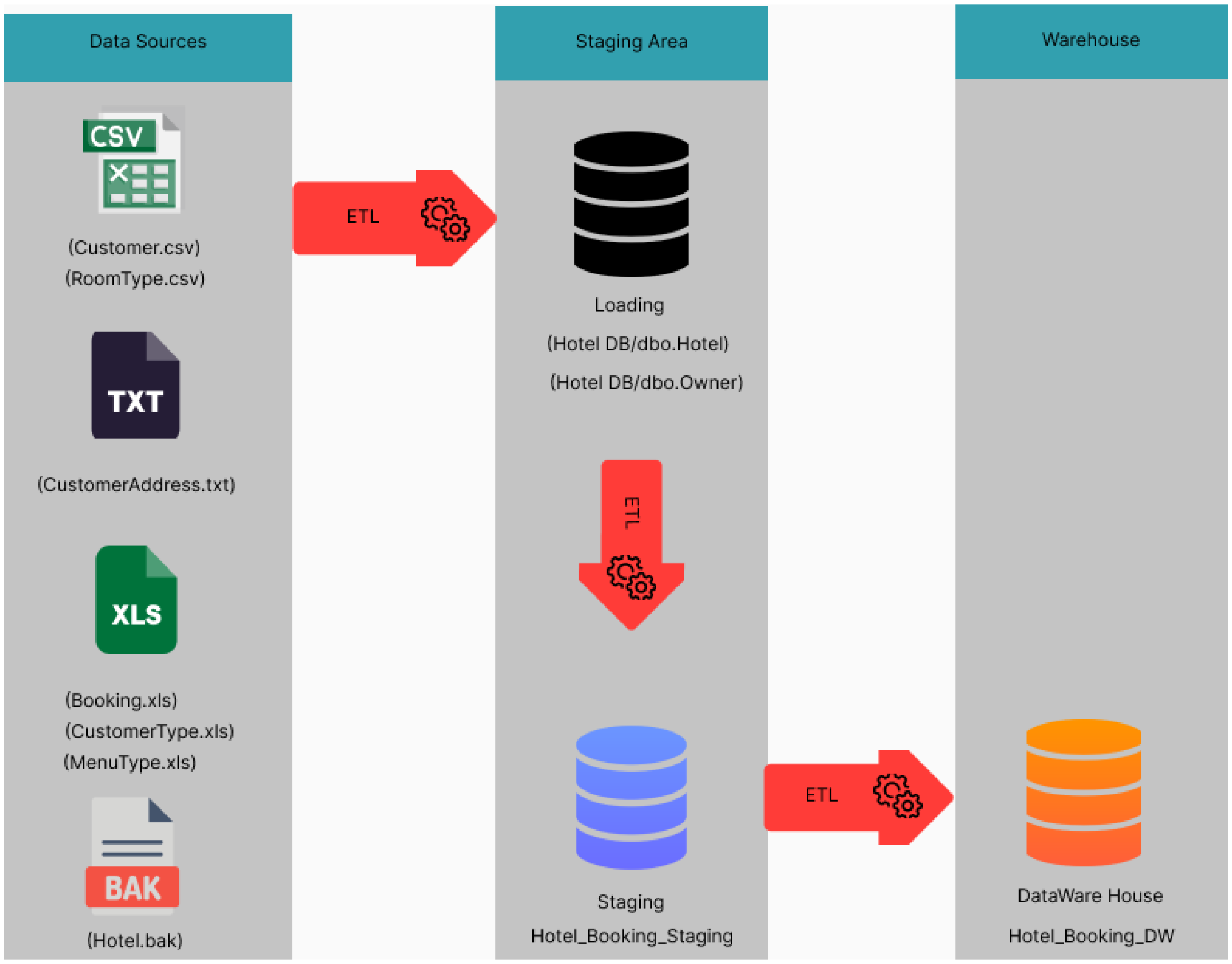


Name	Date modified	Type	Size
Booking.xls	5/6/2022 9:26 AM	Microsoft Excel 97...	1,282 KB
Customer.csv	5/5/2022 10:04 PM	Microsoft Excel Co...	76 KB
CustomerAddress.txt	5/4/2022 9:39 PM	Text Document	101 KB
CustomerType.xls	5/6/2022 9:27 AM	Microsoft Excel 97...	25 KB
Hotel.bak	5/5/2022 8:58 AM	BAK File	3,444 KB
MenuType.xls	5/6/2022 9:28 AM	Microsoft Excel 97...	26 KB
RoomType.csv	5/6/2022 9:31 AM	Microsoft Excel Co...	1 KB

Data Source Type	Source Name	Column Name	Data Type	Description
Excel file	MenuType.xls	MealId	int	Unique ID
		MealValue	nvarchar(255)	There are five type of meals as below.FB,HB,SC,BB, Undefined.
		MealTypeDescription	nvarchar(255)	FB – Full Board HB – Half Board SC - Self Catering BB – Bed and Breakfast Undefined – No meal
	CustomerType.xls	CustomerTypeId	int	Unique ID
		CustomerType	nvarchar(255)	There are 4 type of customers.They are Transient, Contract, Transient-Party, Group
	Booking.xls	BookingId	int	Unique ID
		HotelId	int	Foreign Key
		CustomerId	nvarchar(255)	Foreign Key
		ArrivalDate	datetime	This contains the customer arrival date ,year and month
		ArrivalDateWeekNumber	int	This contains the customer arrival week number
		StaysInWeekendNights	int	No of nights customers going to stay in the week end
		StaysInWeekdayNights	int	No of nights customers going to stay in the weekdays
		MealId	int	Foreign key
		PricePerNight	float	This contains the price that a customer needs to pay per night
		Discount	float	This contains the discount that a customer gets

		RoomTypeId	int	Foreign key
CSV file	Customer.csv	CustomerId	nvarchar(255)	Unique ID
		CustomerName	nvarchar(50)	Name of the customer
		CustomerOccupation	nvarchar(50)	Occupation of the customer
		CustomerPhone	nvarchar(50)	Contact number of the phone
		CustomerTypeId	int	Foreign key
	RoomType.csv	RoomtypeId	int	Unique ID
		RoomTypeValue	nvarchar(50)	From A –I there different types of rooms that a customer can book
Text file	CustomerAddres.txt	CustomerId	nvarchar(255)	Foreign key
		CustomerHomeAddress	nvarchar(50)	Home address of the customer
		CustomerCity	nvarchar(50)	Customer's living city
		CustomerState	nvarchar(50)	Customer's living state
		CustomerCountry	nvarchar(50)	Customer's living Country
		CustomerZipCode	nvarchar(50)	Customer's city ZipCode
Database File	dbo.Hotel	HotelId	int	Unique Id
		HotelName	nvarchar(100)	Name of the hotel
		latitude	float	Hotel location - latitude
		longitude	float	Hotel location - longitude
		OwnerId	int	Foreign key
	dbo.Owner	OwnerId	int	Unique key
		Owner_name	nvarchar(50)	Name of the owner
		Gender	nvarchar(50)	Name of the gender
		Phone	nvarchar(50)	Owner's contact number
		Owner_since	date	

(3)Solution Architechture



The above architecture represents the High-Level BI solution to the warehouse design

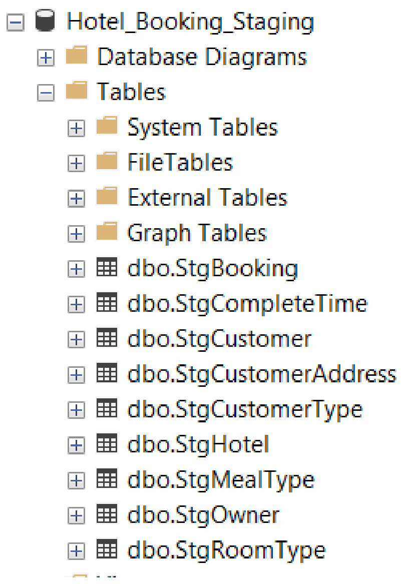
Data Sources

Data Sources In the above diagram, '.txt' component represents Text files, '.xls' component is used to represent Excel files, '.csv' component is used to represent Comma Separated files, and '.bak' component represents Database files.

Staging Area

Loading DB component represents the process of the creating database tables. Owner and Hotel csv files were imported to the database and the relevant database tables were created. These tables were used as the DB source data. Staging DB component represents the creation of staging level tables through the 'Extract' process.

After extracting data into the staging the staging database looks as below.



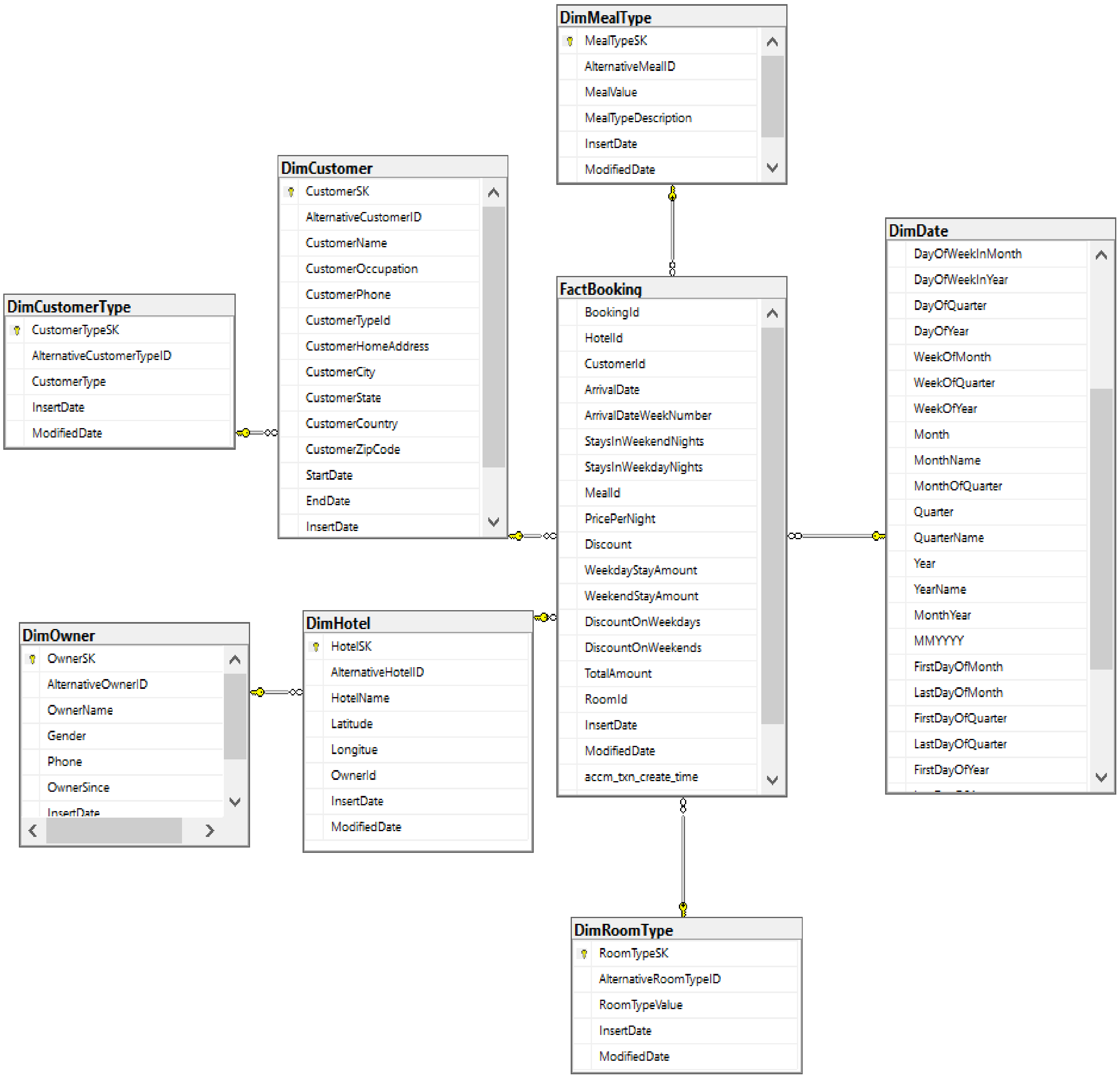
Hotel_Booking_Staging
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.StgBooking
dbo.StgCompleteTime
dbo.StgCustomer
dbo.StgCustomerAddress
dbo.StgCustomerType
dbo.StgHotel
dbo.StgMealType
dbo.StgOwner
dbo.StgRoomType

Data Warehouse

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 analysing tools can use data for reporting mining and analyzing.

(4)Data Warehouse design and development

The following diagram shows how the fact table and dimension tables were combined in a rational manner to implement the data warehouse.



<u>Schema Type</u>
Snowflake schema type was used for this scenario since the tables need to be **normalized.**

<u>Dimension Types</u>

• Hierarchical Dimensions
- Date – all the hierarchies in date
- Customer– Country → State → City → Zip → HomeAddress
- DimCustomerType is taken as a hierarchy of DimCustomer since one customer might use only one CustomerType but single CustomerType is used by several customers.
- DimOwner is taken as a hierarchy of DimHotel since one Hotel might have only one Owner but single Ownercan have many hotels.

• Slowly Changing Dimension
➢ Customer
➢ Following columns were set as changing attributes.
- CustomerCity
- CustomerHomeAddress
- CustomerZipcode
- CustomerState

• Fact Table
➢ Numbers – Stays in weekday nights ,Stays in weekend nights, Price per night,Weekday stay amount,Weekend stay amount,Discount
➢ FKs – Hotel ID, Customer ID, Meal ID,Room ID

<u>Calculations</u>

1. WeekdayStayAmount in [dbo].[FactBooking] table is calculated by, ([NoOfWeekdayNights]*[PricePerNight])

2. WeekendStayAmount in [dbo].[FactBooking] table is calculated by, ([NoOfWeekendNights]*[PricePerNight])

3. DiscountOnweekdays in [dbo].[FactBooking] table is calculated by, ([NoOfWeekdayNights]*[PricePerNight]*[Discount])

4. DiscountOnweekends in [dbo].[FactBooking] table is calculated by, ([NoOfWeekendNights]*[PricePerNight]*[Discount])

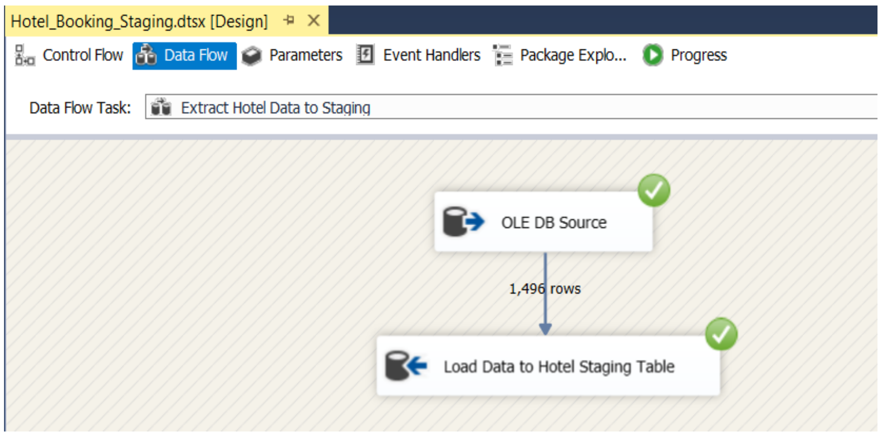
5. TotalAmount in [dbo].[FactBooking] table is calculated by, [WeekdayStayAmount]+[WeekendStayAmount]–[DiscountOnweekdays] – [DiscountOnweekends]

(5)ETL development

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

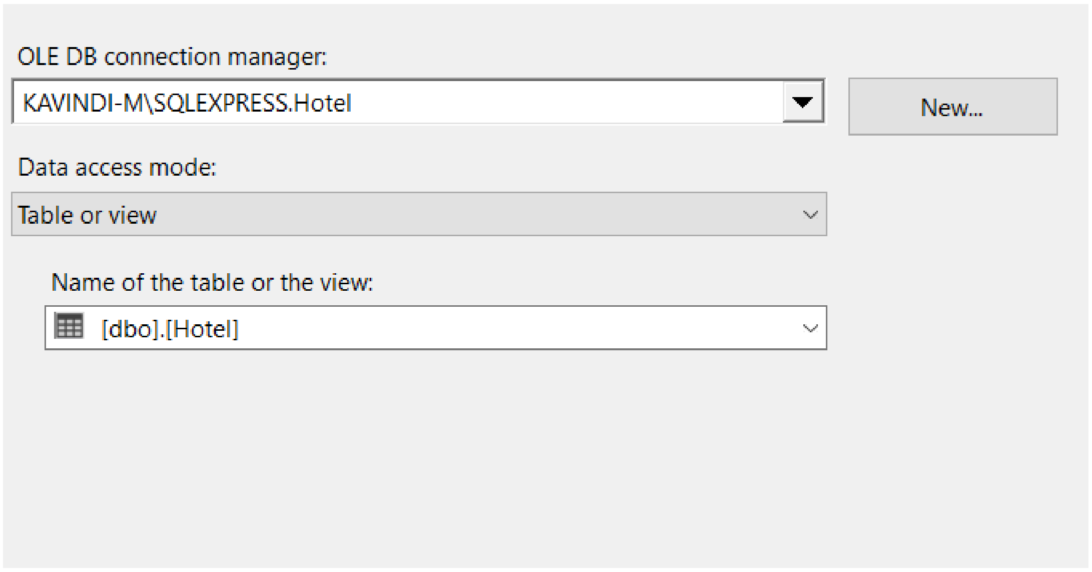
DATA EXTRACTION

Extract Hotel Data to Staging area



Hotel data in Hotel database table has been extracted and loaded to Hotel Staging table

Used OLE DB Source as dbo.Hotel in the Hotel database

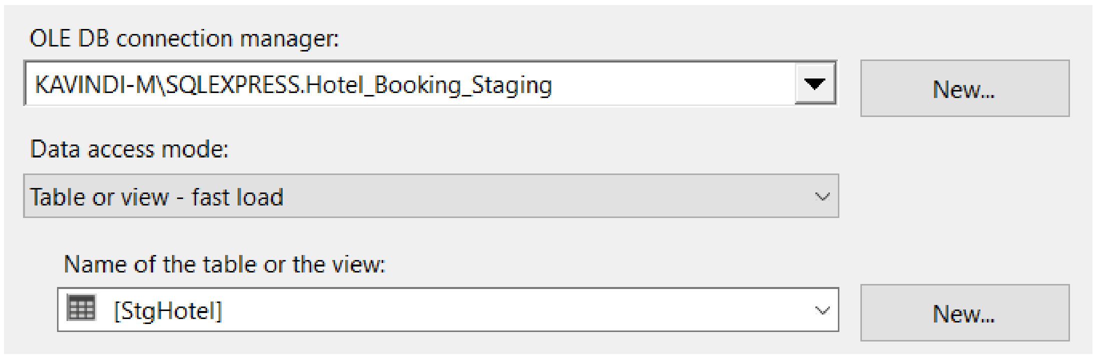


OLE DB connection manager: KAVINDI-M\SQLEXPRESS.Hotel

Data access mode: Table or view

Name of the table or the view: [dbo].[Hotel]

OLE DB Destination for create new table as StgHotel in the Hotel_Booking_Staging database.



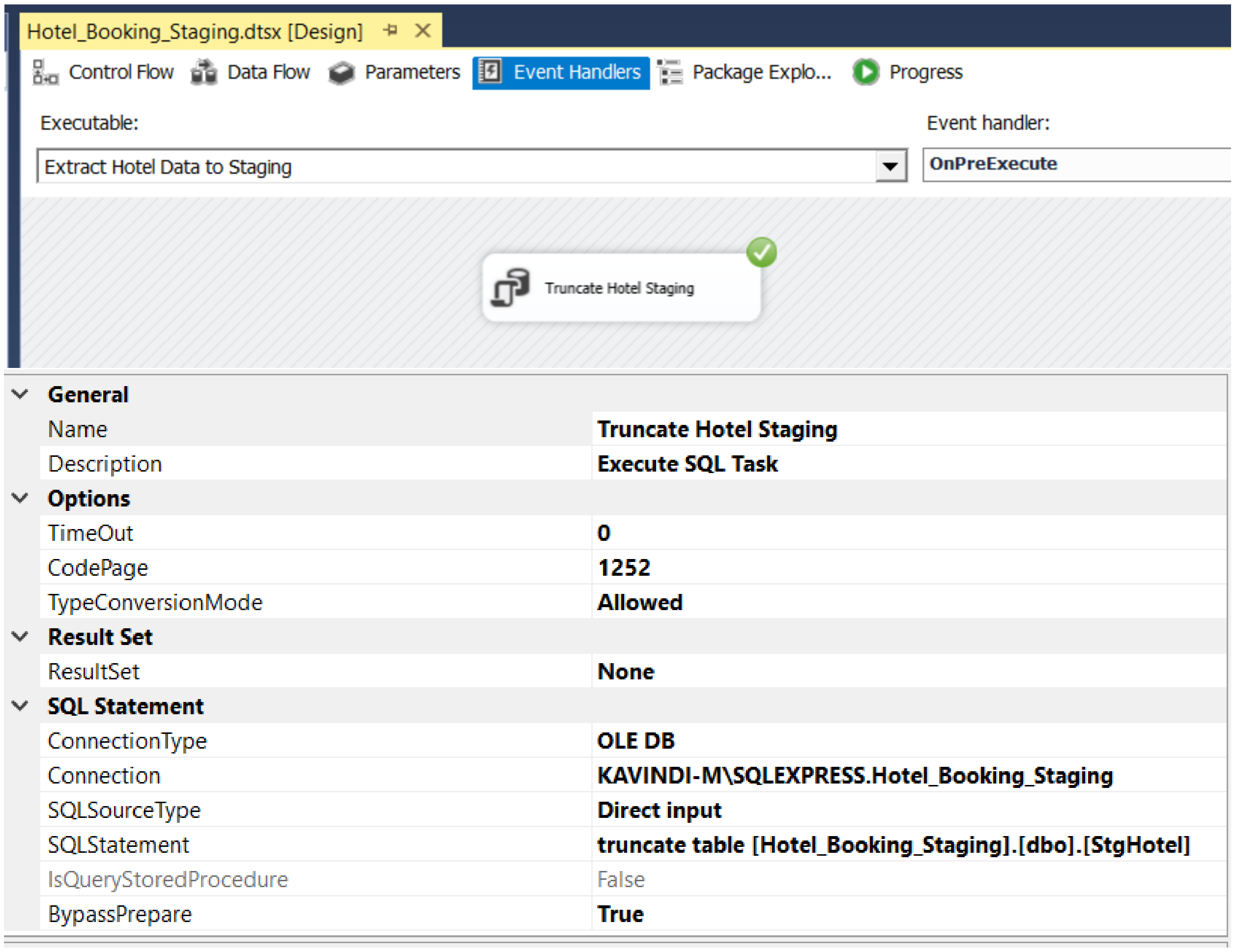
OLE DB connection manager: KAVINDI-M\SQLEXPRESS.Hotel_Booking_Staging

Data access mode: Table or view - fast load

Name of the table or the view: [StgHotel]

Event handler was used to do the truncate

Used Execute SQL Task SSIS tool to Truncate table for SQL command as truncate table dbo.StgHotel

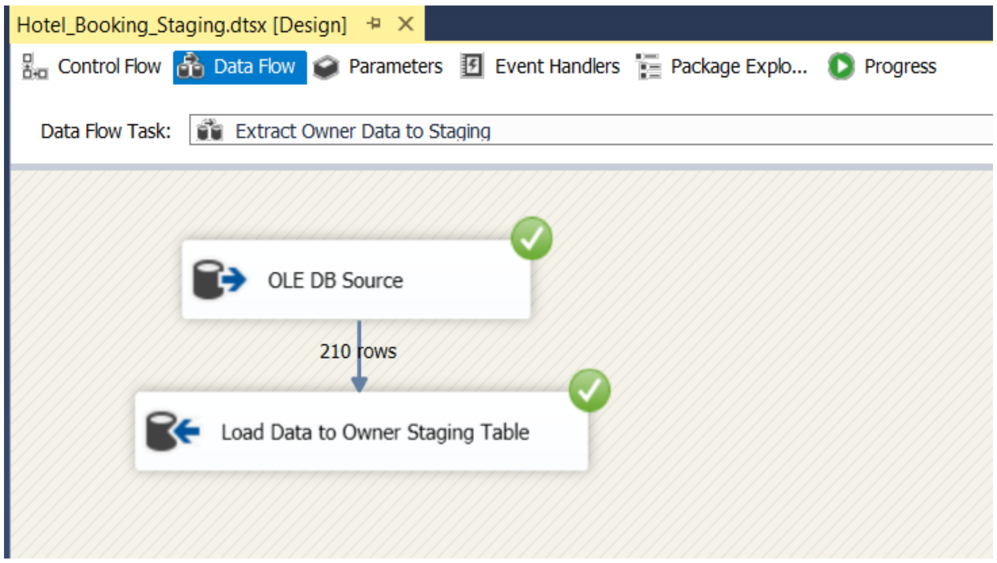


Executable: Extract Hotel Data to Staging

Event handler: OnPreExecute

General	
Name	Truncate Hotel Staging
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	KAVINDI-M\SQLEXPRESS.Hotel_Booking_Staging
SQLSourceType	Direct input
SQLStatement	truncate table [Hotel_Booking_Staging].[dbo].[StgHotel]
IsQueryStoredProcedure	False
BypassPrepare	True

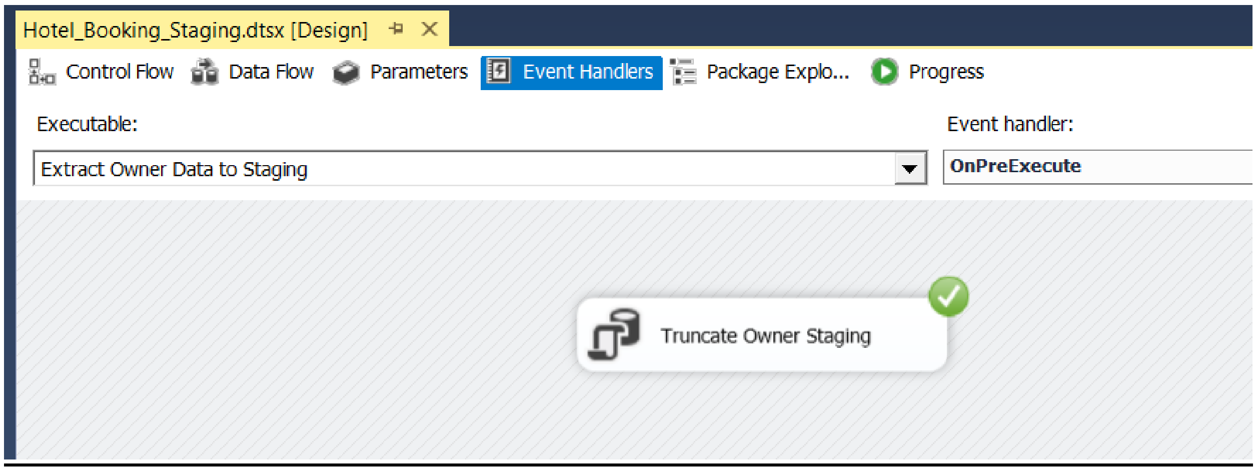
Extract Owner Data to Staging area



Data Flow Task: Extract Owner Data to Staging

Owner data in Hotel database table has been extracted and loaded to Owner Staging table

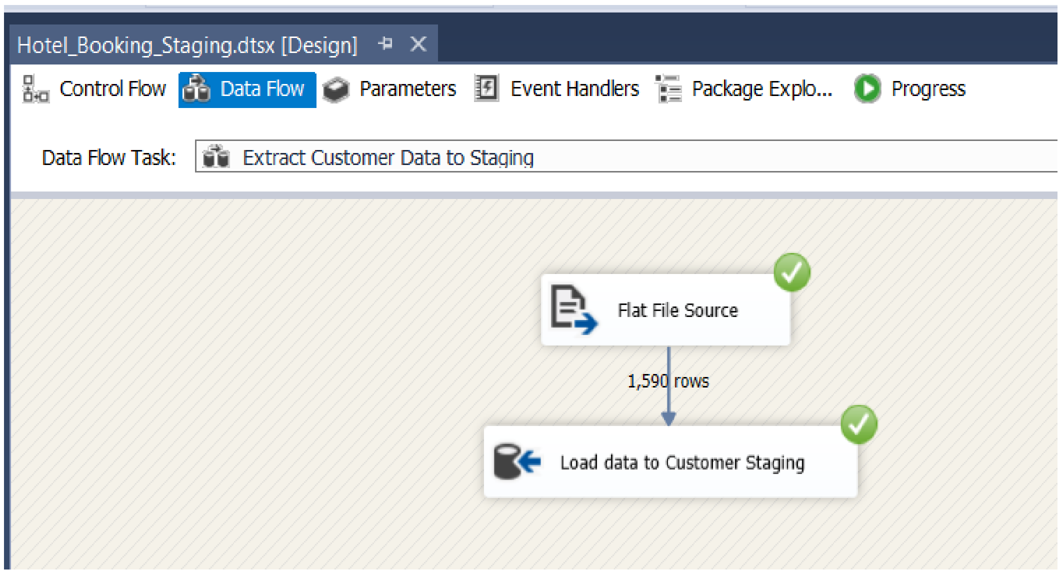
Event handler was used to do the truncate



Executable: Extract Owner Data to Staging

Event handler: OnPreExecute

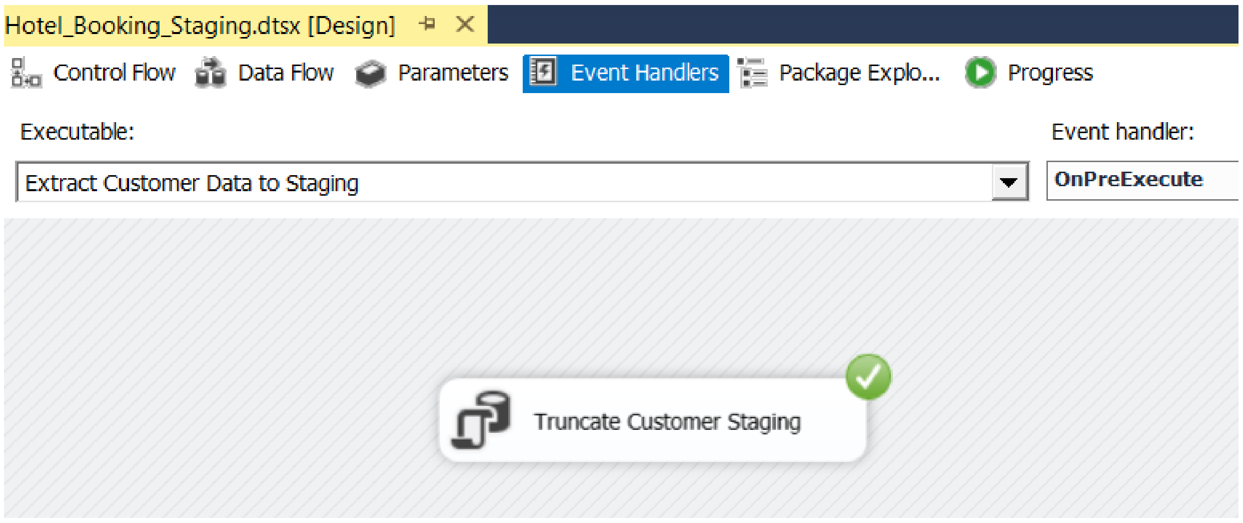
Extract Customer Data to Staging area



Hotel_Booking_Staging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task: Extract Customer Data to Staging
Flat File Source
1,590 rows
Load data to Customer Staging

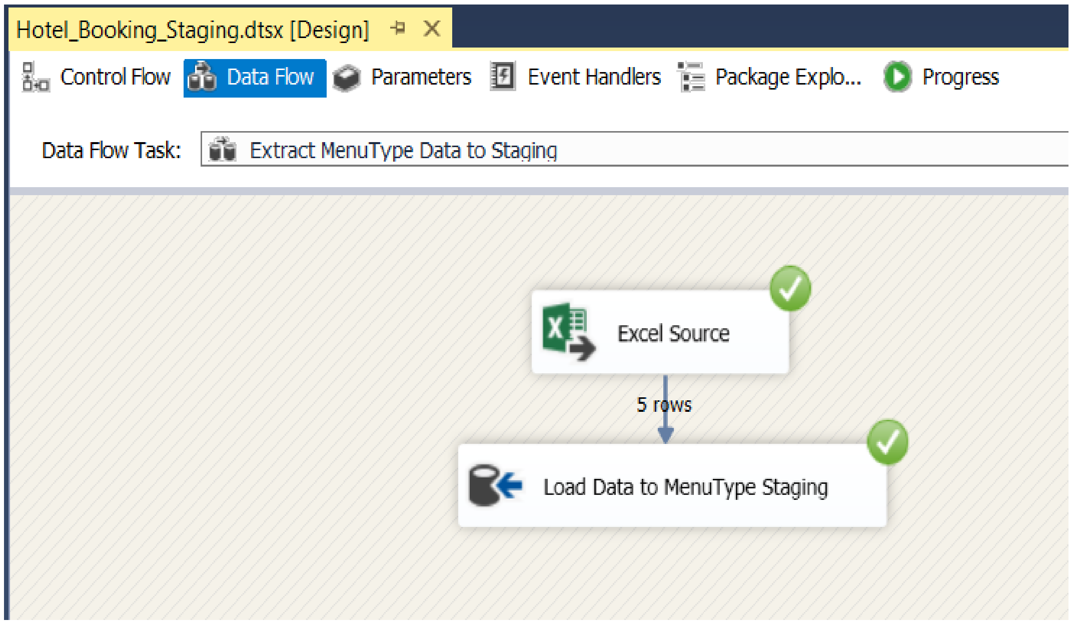
Customer data in Customer.CSV has been extracted and loaded to Customer Staging table

Event handler was used to do the truncate



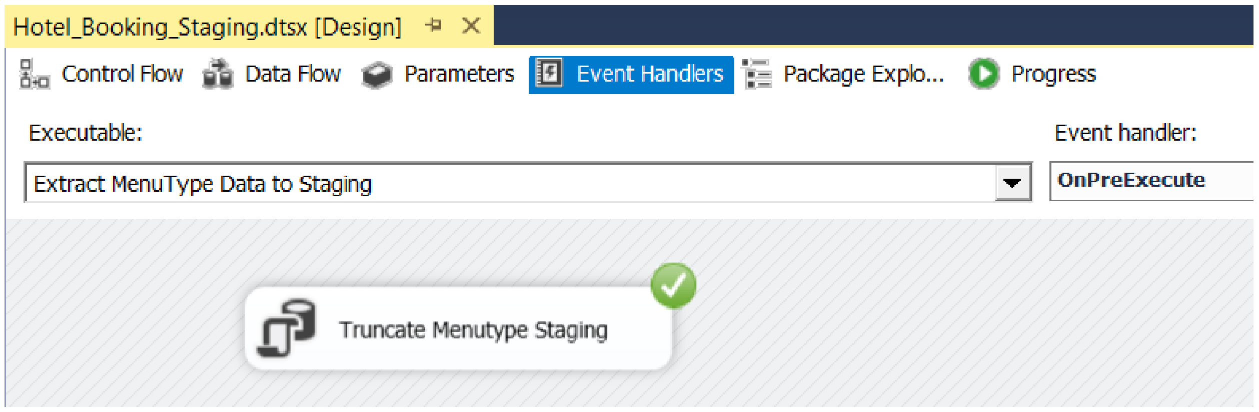
Hotel_Booking_Staging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Executable: Extract Customer Data to Staging
Event handler: OnPreExecute
Truncate Customer Staging

Extract MenuType Data to Staging area

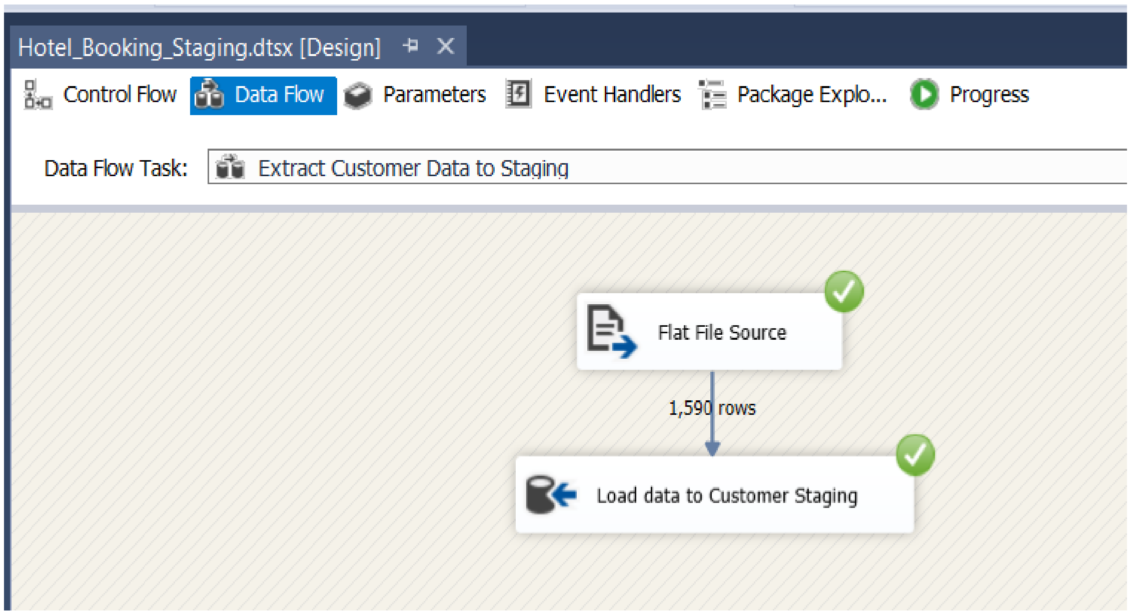


MenuType data in MenuType.xls has been extracted and loaded to MenuType Staging table

Event handler was used to do the truncate

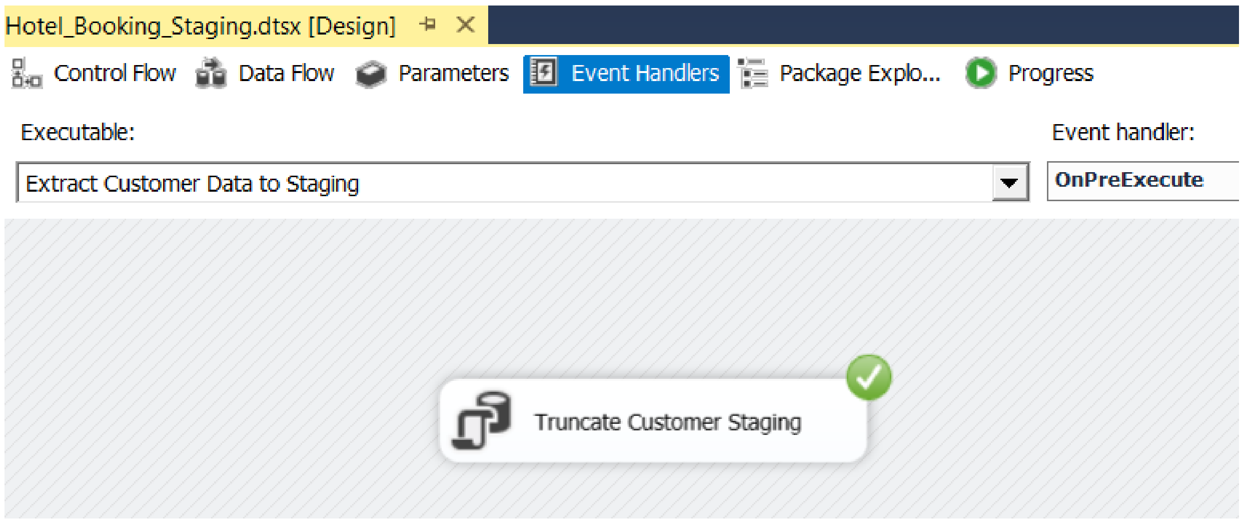


Extract CustomerType Data to Staging area



CustomrType in CustomerType.csv has been extracted and loaded to CustomerType Staging table

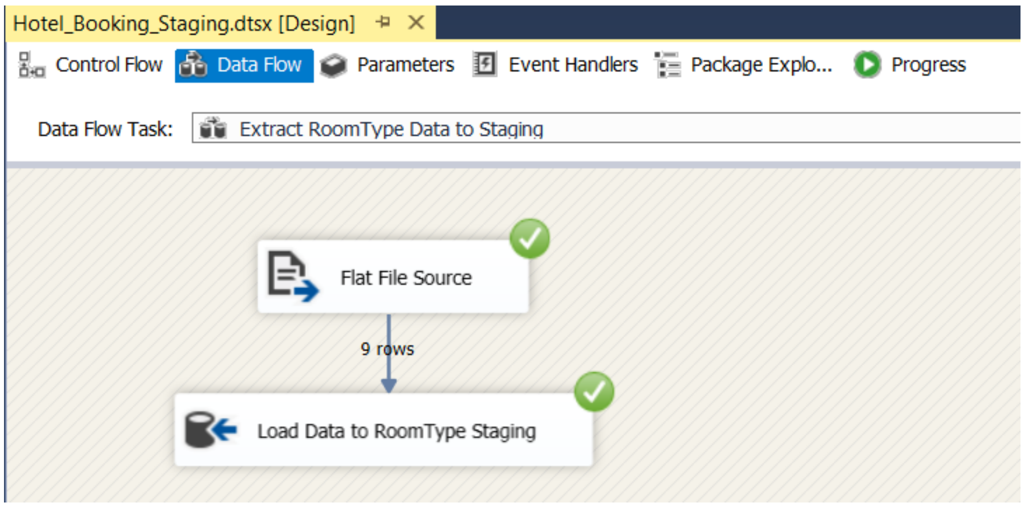
Event handler was used to do the truncate



Executable: Extract Customer Data to Staging

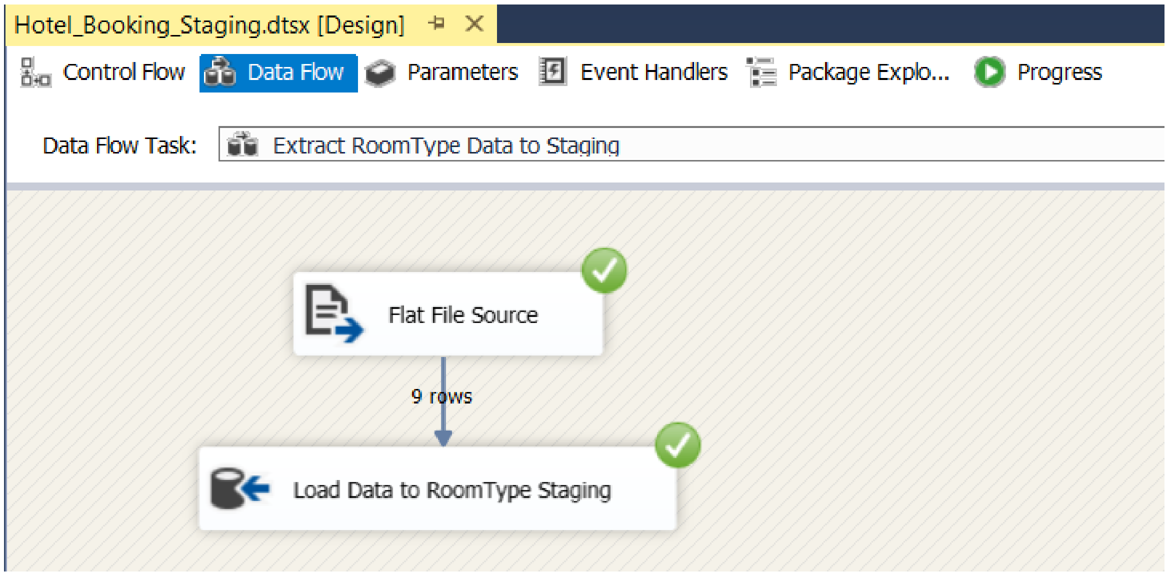
Event handler: OnPreExecute

Extract RoomType Data to Staging area

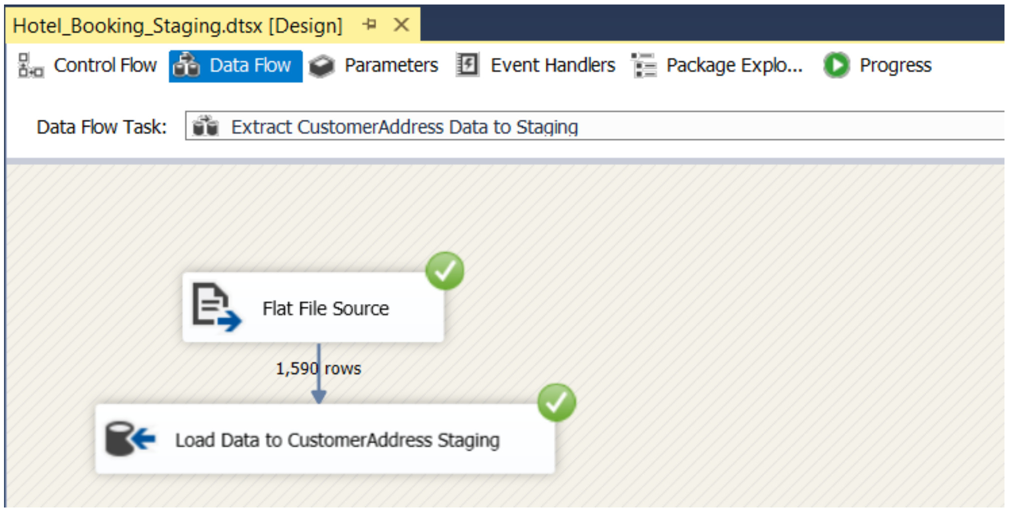


RoomType data in RoomType.csv has been extracted and loaded to RoomType Staging table

Event handler was used to do the truncate

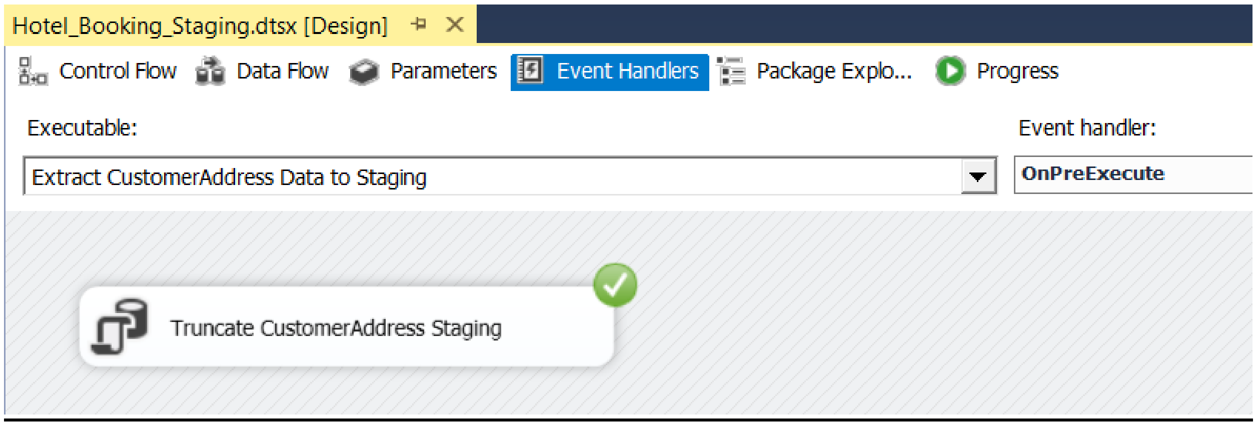


Extract CustomerAddress Data to Staging area


Hotel_Booking_Staging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Extract CustomerAddress Data to Staging
Flat File Source
1,590 rows
Load Data to CustomerAddress Staging

CustomerAddress data in CustomerAddress.txt has been extracted and loaded to CustomerAddress Staging table

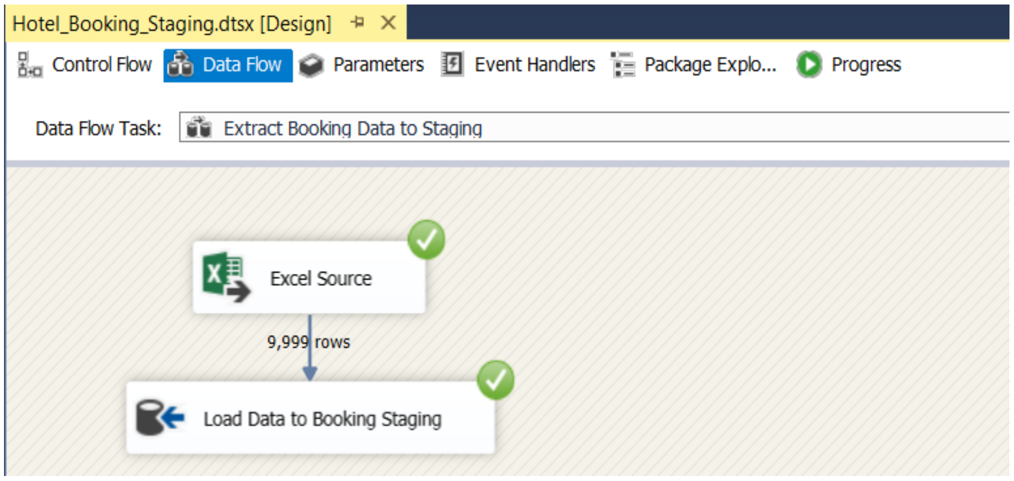
Event handler was used to do the truncate



Executable: Extract CustomerAddress Data to Staging

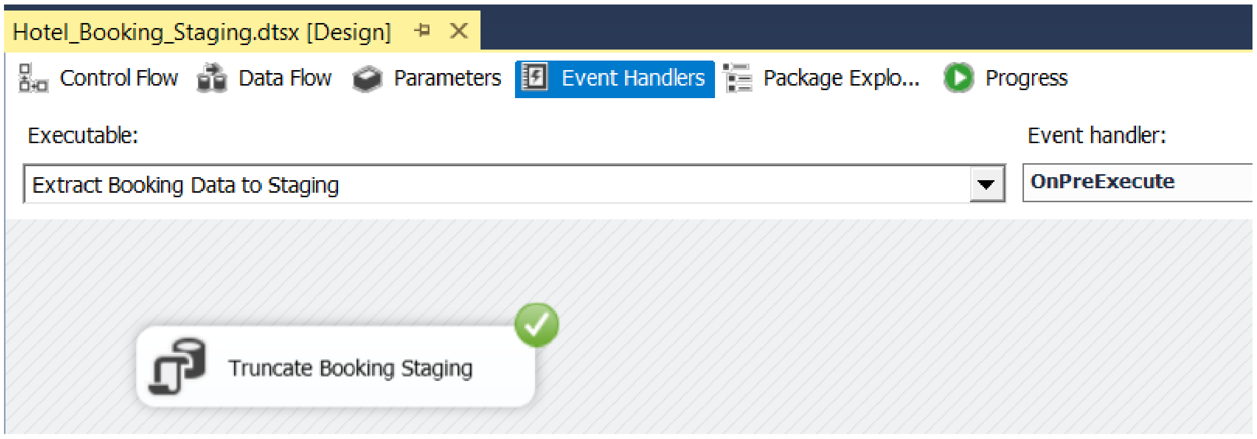
Event handler: OnPreExecute

Extract Booking Data to Staging area

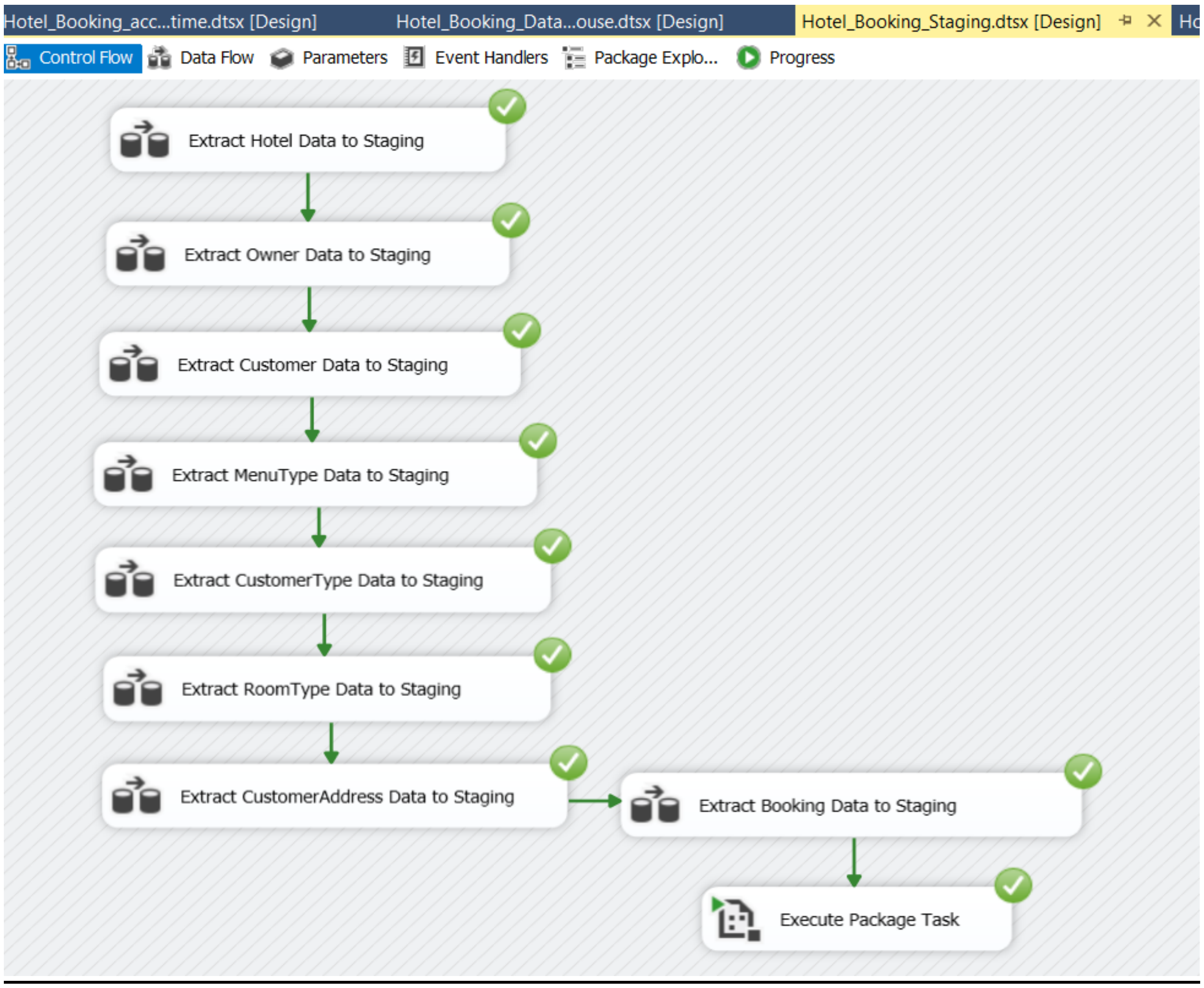


Booking data in Booking.xls has been extracted and loaded to BookingStaging table

Event handler was used to do the truncate

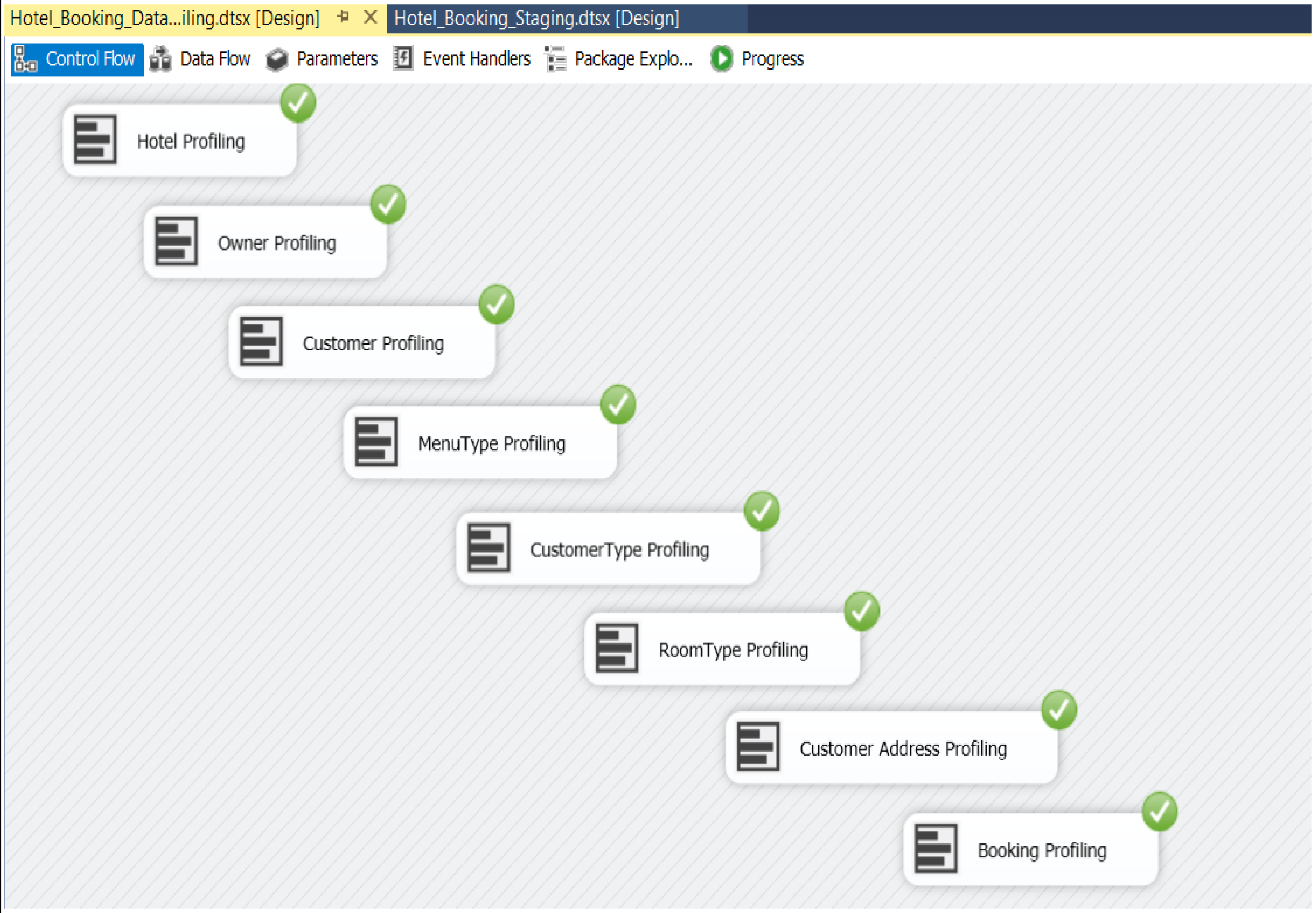


Overall control flow


Hotel_Booking_acc...time.dtsx [Design]
Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Staging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Extract Hotel Data to Staging
Extract Owner Data to Staging
Extract Customer Data to Staging
Extract MenuType Data to Staging
Extract CustomerType Data to Staging
Extract RoomType Data to Staging
Extract CustomerAddress Data to Staging
Extract Booking Data to Staging
Execute Package Task

Data Profiling

Before loading the created staging tables to the data warehouse each and every staging table has been profiled.



Data Transforming and Loading

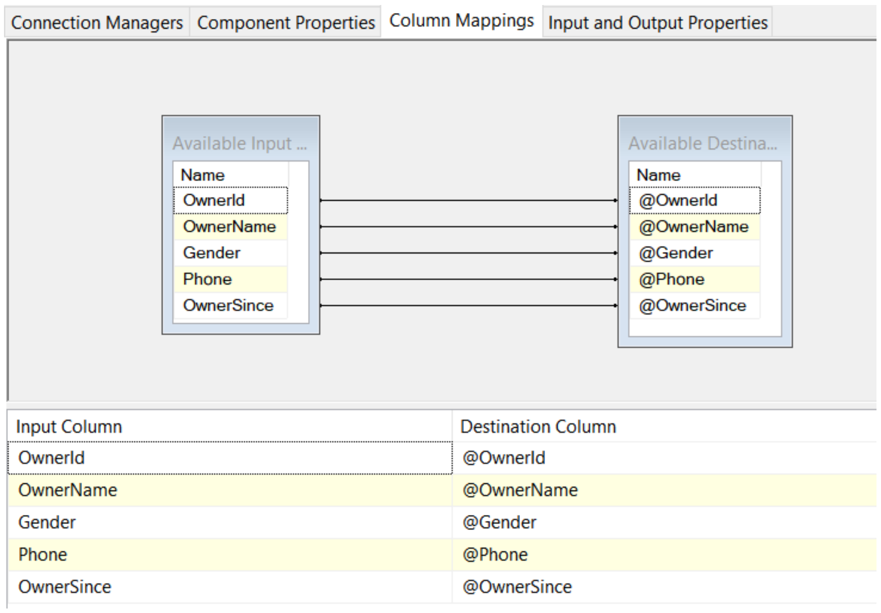
(a)Load Hierarchical Dimensions

This design is consisting with two hierarchical dimensions. They are ,

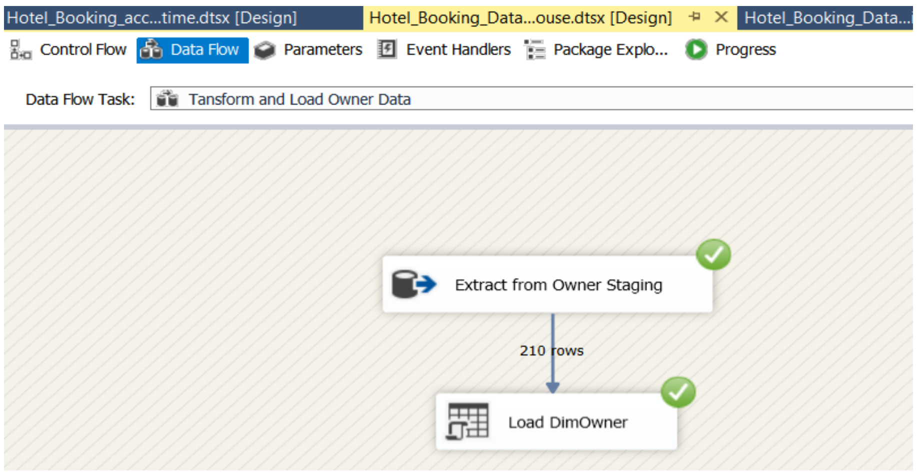
(1)Owner Dimension –Sub category of Hotel Dimension

(2)CutomerType Dimension – Sub category of Customer dimension

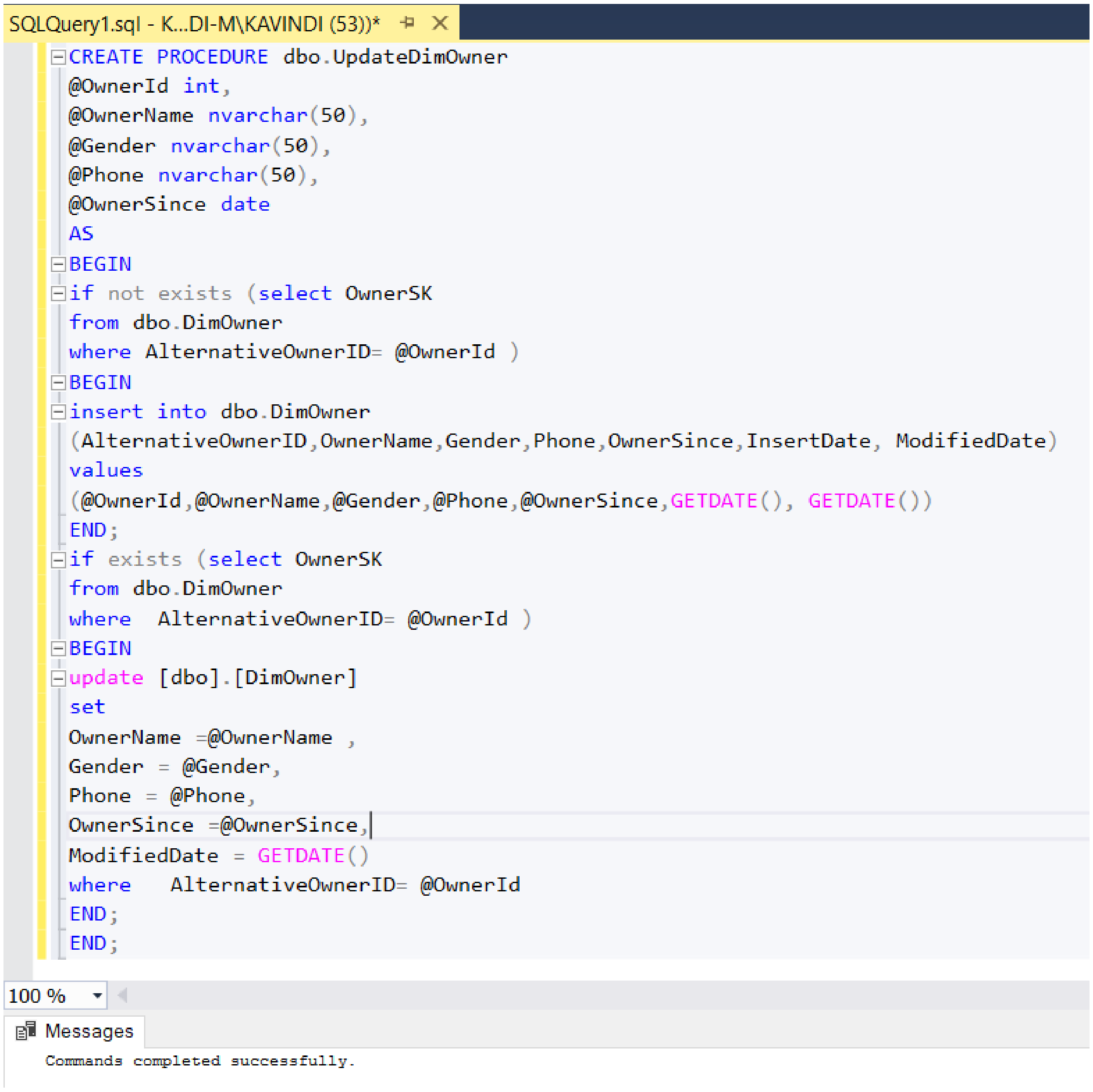
Owner Data from Staging to Data Warehouse



Input Column	Destination Column
OwnerId	@OwnerId
OwnerName	@OwnerName
Gender	@Gender
Phone	@Phone
OwnerSince	@OwnerSince


Hotel_Booking_acc...time.dtsx [Design]
Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Data...
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Transform and Load Owner Data
Extract from Owner Staging
210 rows
Load DimOwner

Owner data has been loaded to DimOwner table in data warehouse.The following procedure is used to load data into the DimOwner table.



SQLQuery1.sql - K...DI-M\KAVINDI (53))*

```
CREATE PROCEDURE dbo.UpdateDimOwner
@OwnerId int,
@OwnerName nvarchar(50),
@Gender nvarchar(50),
@Phone nvarchar(50),
@OwnerSince date
AS
BEGIN
if not exists (select OwnerSK
from dbo.DimOwner
where AlternativeOwnerID= @OwnerId )
BEGIN
insert into dbo.DimOwner
(AlternativeOwnerID,OwnerName,Gender,Phone,OwnerSince,InsertDate, ModifiedDate)
values
(@OwnerId,@OwnerName,@Gender,@Phone,@OwnerSince,GETDATE(), GETDATE())
END;
if exists (select OwnerSK
from dbo.DimOwner
where  AlternativeOwnerID= @OwnerId )
BEGIN
update [dbo].[DimOwner]
set
OwnerName =@OwnerName ,
Gender = @Gender,
Phone = @Phone,
OwnerSince =@OwnerSince,
ModifiedDate = GETDATE()
where   AlternativeOwnerID= @OwnerId
END;
END;
```

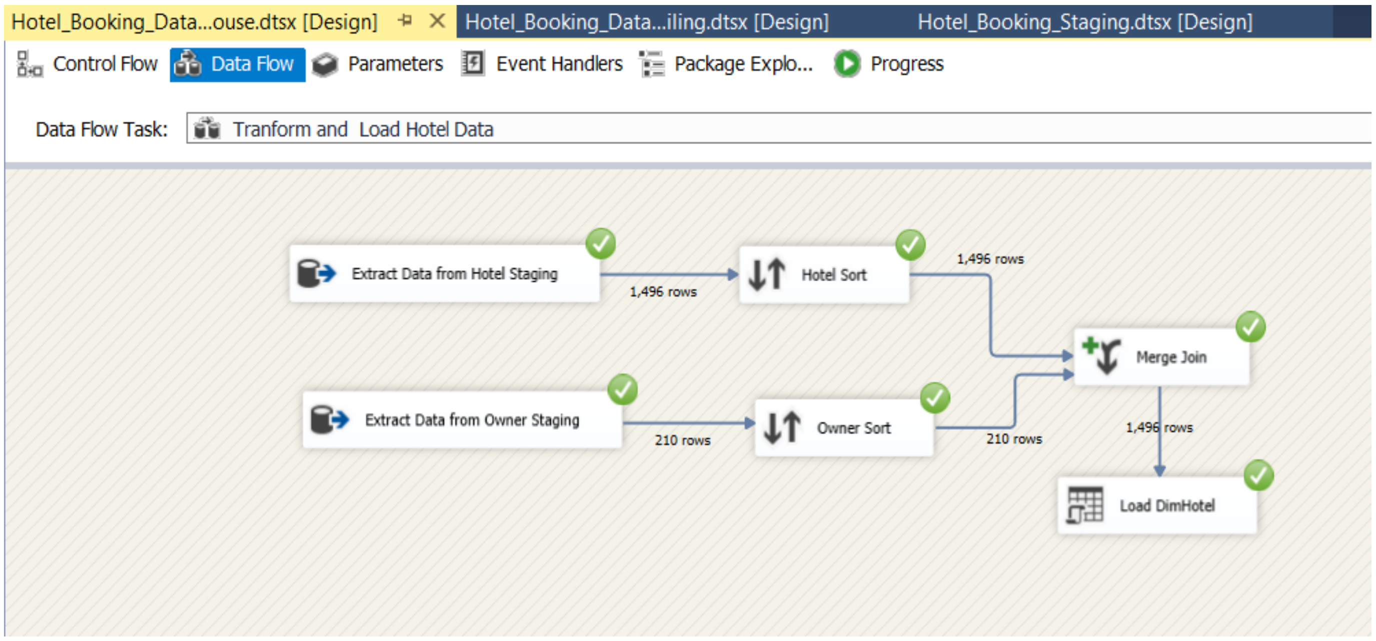
100 %

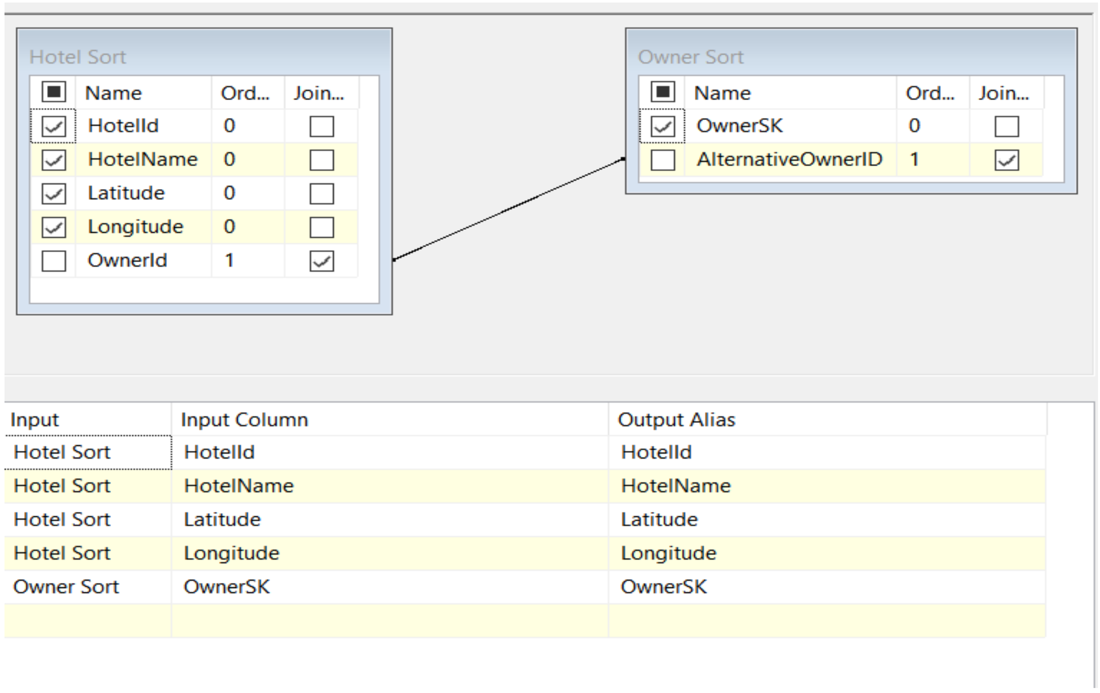
Messages

```
Commands completed successfully.
```

Hotel Data from Staging to Data Warehouse

As mentioned in above Owner is a hierarchical dimension of Hotel.To get the details of the surrogate key of the DimOwner merge has been used as below.


Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Data...iling.dtsx [Design]
Hotel_Booking_Staging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Tranform and Load Hotel Data
Extract Data from Hotel Staging
1,496 rows
Hotel Sort
1,496 rows
Merge Join
Extract Data from Owner Staging
210 rows
Owner Sort
210 rows
1,496 rows
Load DimHotel



Hotel Sort

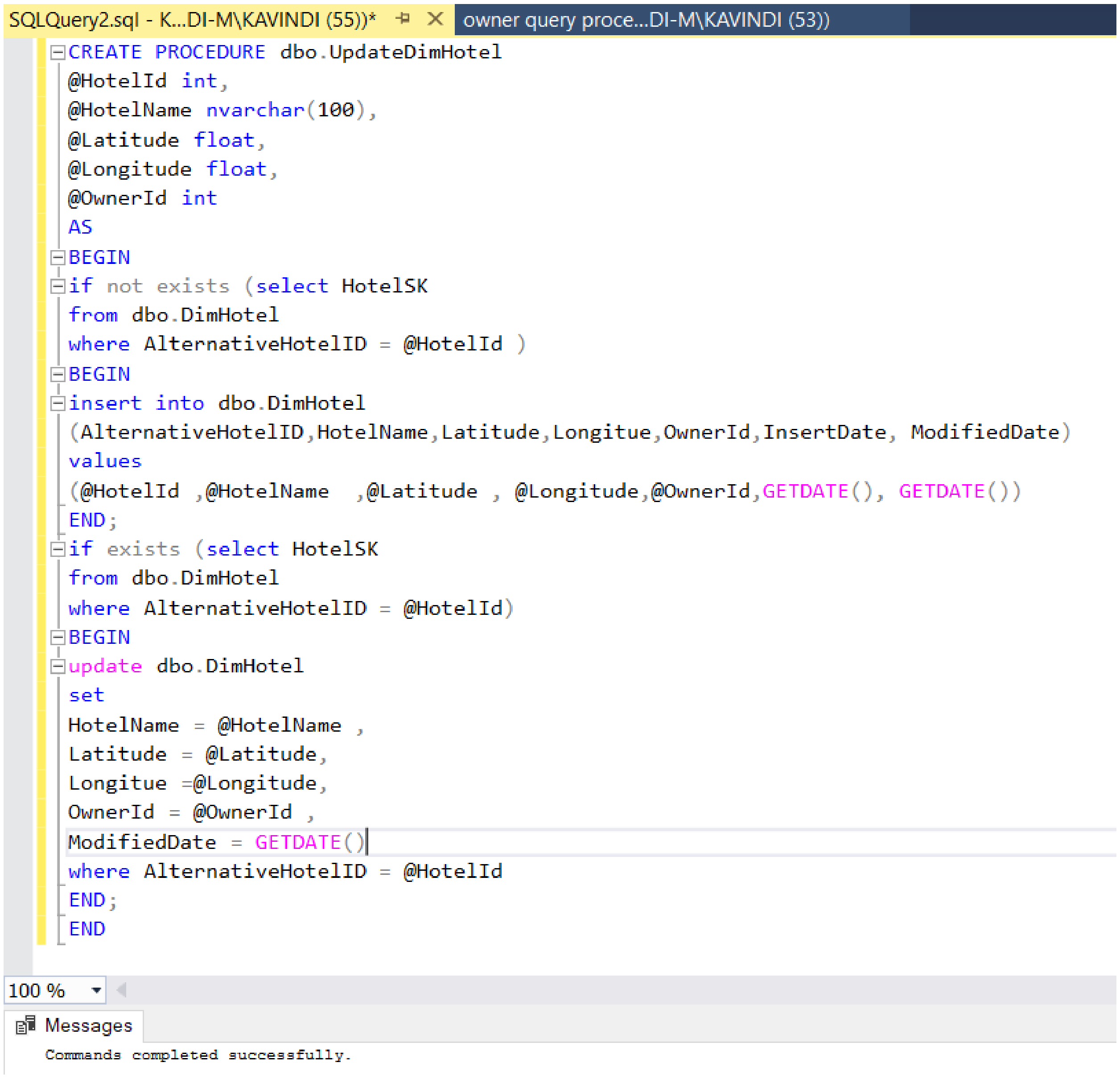
■	Name	Ord...	Join...
✓	HotelId	0	☐
✓	HotelName	0	☐
✓	Latitude	0	☐
✓	Longitude	0	☐
☐	OwnerId	1	✓

Owner Sort

■	Name	Ord...	Join...
✓	OwnerSK	0	☐
☐	AlternativeOwnerID	1	✓

Input	Input Column	Output Alias
Hotel Sort	HotelId	HotelId
Hotel Sort	HotelName	HotelName
Hotel Sort	Latitude	Latitude
Hotel Sort	Longitude	Longitude
Owner Sort	OwnerSK	OwnerSK

Hotel data has been loaded to DimHotel table in data warehouse. The following procedure is used to load data into the DimHotel table.


SQLQuery2.sql - K...DI-M\KAVINDI (55))* owner query proce...DI-M\KAVINDI (53))

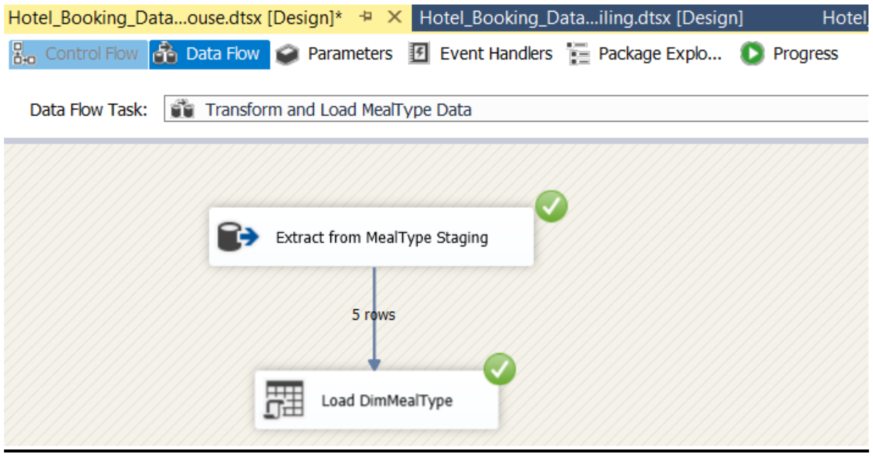
```
CREATE PROCEDURE dbo.UpdateDimHotel
@HotelId int,
@HotelName nvarchar(100),
@Latitude float,
@Longitude float,
@OwnerId int
AS
BEGIN
if not exists (select HotelSK
from dbo.DimHotel
where AlternativeHotelID = @HotelId )
BEGIN
insert into dbo.DimHotel
(AlternativeHotelID,HotelName,Latitude,Longitue,OwnerId,InsertDate, ModifiedDate)
values
(@HotelId ,@HotelName  ,@Latitude , @Longitude,@OwnerId,GETDATE(), GETDATE())
END;
if exists (select HotelSK
from dbo.DimHotel
where AlternativeHotelID = @HotelId)
BEGIN
update dbo.DimHotel
set
HotelName = @HotelName ,
Latitude = @Latitude,
Longitue =@Longitude,
OwnerId = @OwnerId ,
ModifiedDate = GETDATE()
where AlternativeHotelID = @HotelId
END;
END
```

100 %

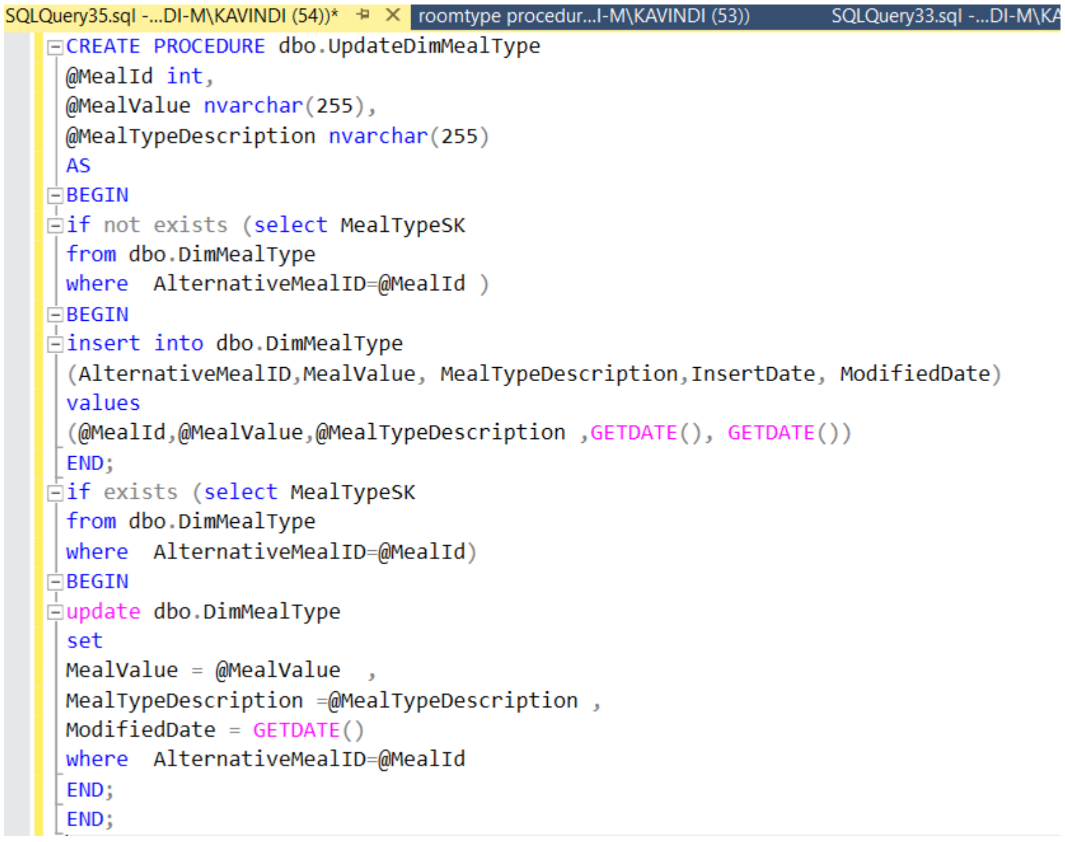
Messages

```
Commands completed successfully.
```

MealType Data from Staging to Data Warehouse

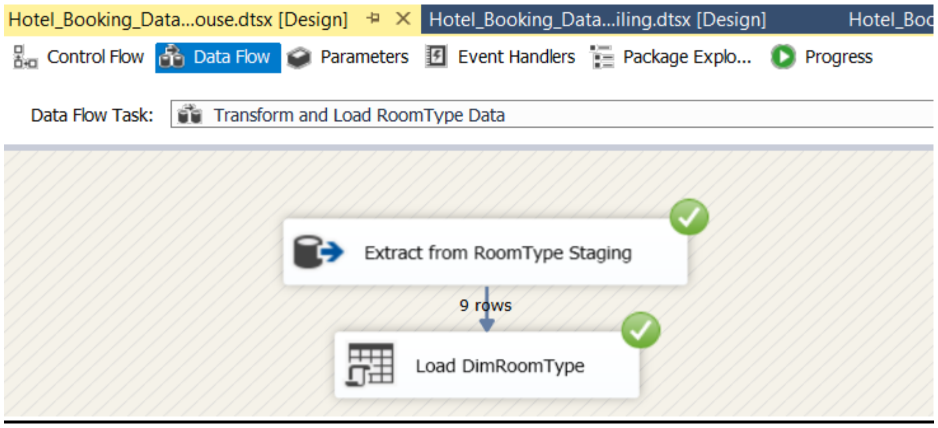

Hotel_Booking_Data...ouse.dtsx [Design]*
Hotel_Booking_Data...iling.dtsx [Design]
Hotel
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Transform and Load MealType Data
Extract from MealType Staging
5 rows
Load DimMealType

MealType data has been loaded to DimMealType table in data warehouse. The following procedure is used to load data into the DimMealType table.

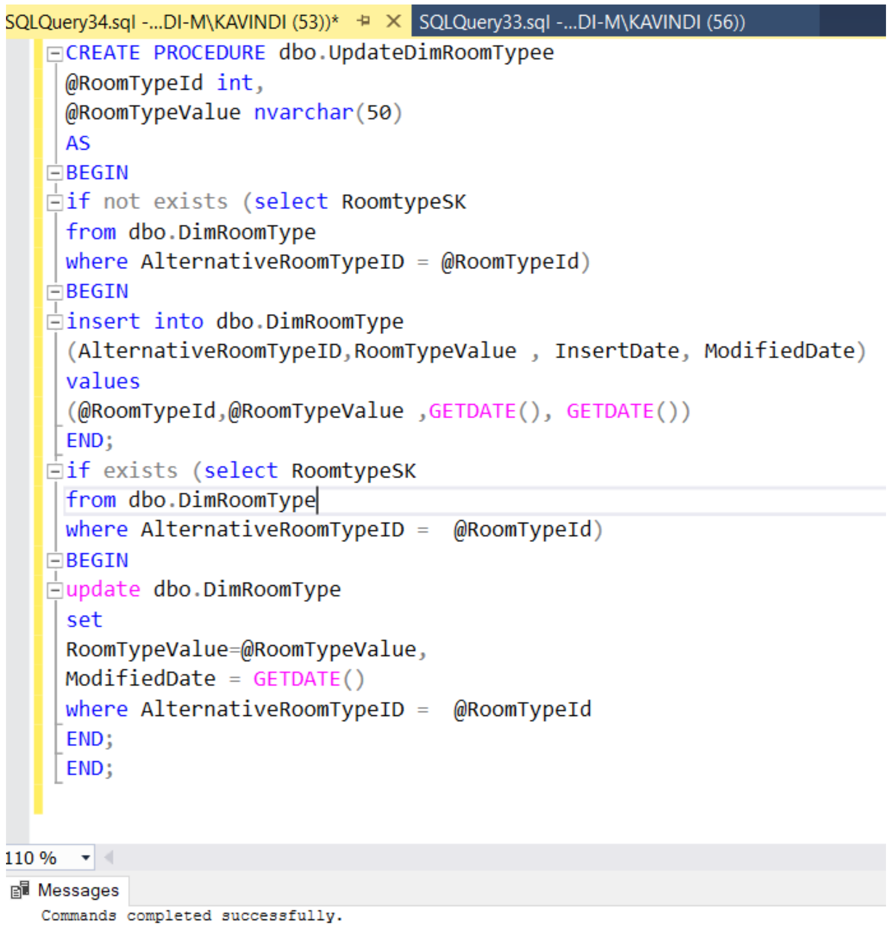

SQLQuery35.sql -...DI-M\KAVINDI (54))*
roomtype procedur...I-M\KAVINDI (53))
SQLQuery33.sql -...DI-M\KA

```sql
CREATE PROCEDURE dbo.UpdateDimMealType
@MealId int,
@MealValue nvarchar(255),
@MealTypeDescription nvarchar(255)
AS
BEGIN
if not exists (select MealTypeSK
from dbo.DimMealType
where  AlternativeMealID=@MealId )
BEGIN
insert into dbo.DimMealType
(AlternativeMealID,MealValue, MealTypeDescription,InsertDate, ModifiedDate)
values
(@MealId,@MealValue,@MealTypeDescription ,GETDATE(), GETDATE())
END;
if exists (select MealTypeSK
from dbo.DimMealType
where  AlternativeMealID=@MealId)
BEGIN
update dbo.DimMealType
set
MealValue = @MealValue  ,
MealTypeDescription =@MealTypeDescription ,
ModifiedDate = GETDATE()
where  AlternativeMealID=@MealId
END;
END;
```

RoomType Data from Staging to Data Warehouse


Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Data...iling.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Transform and Load RoomType Data
Extract from RoomType Staging
9 rows
Load DimRoomType

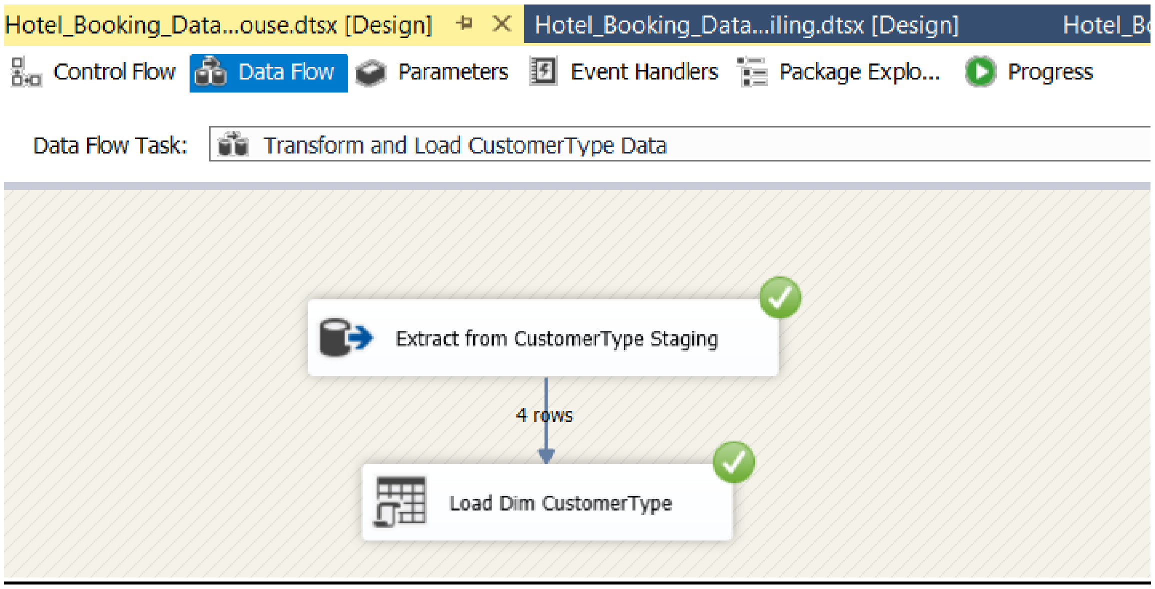
RoomType data has been loaded to DimRoomType table in data warehouse. The following procedure is used to load data into the DimRoomType table.


SQLQuery34.sql -...DI-M\KAVINDI (53))*
SQLQuery33.sql -...DI-M\KAVINDI (56))
110 %
Messages

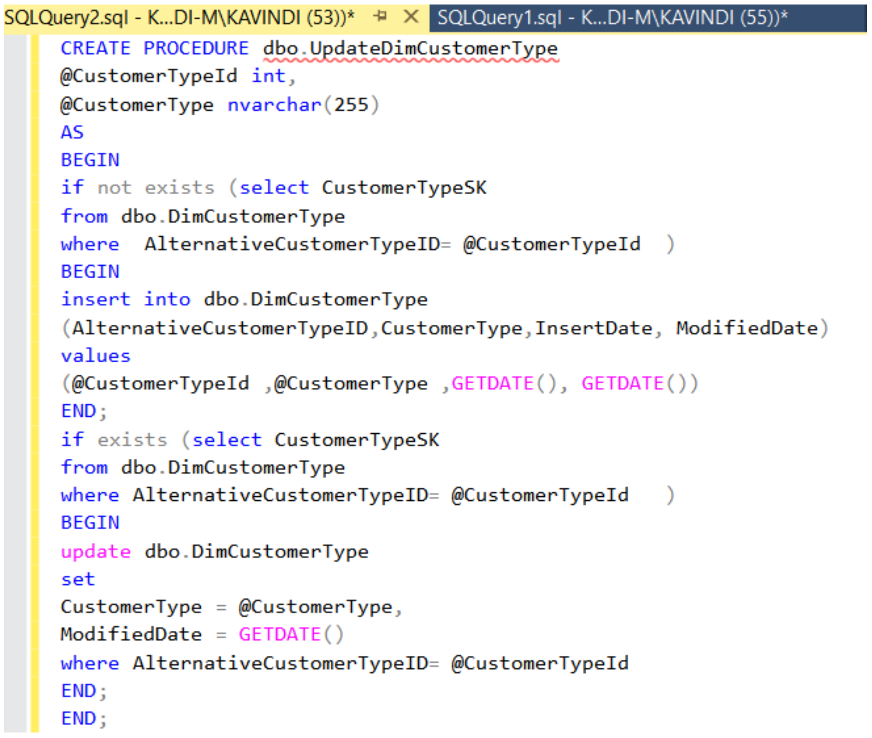
```sql
CREATE PROCEDURE dbo.UpdateDimRoomTypee
@RoomTypeId int,
@RoomTypeValue nvarchar(50)
AS
BEGIN
if not exists (select RoomtypeSK
from dbo.DimRoomType
where AlternativeRoomTypeID = @RoomTypeId)
BEGIN
insert into dbo.DimRoomType
(AlternativeRoomTypeID,RoomTypeValue , InsertDate, ModifiedDate)
values
(@RoomTypeId,@RoomTypeValue ,GETDATE(), GETDATE())
END;
if exists (select RoomtypeSK
from dbo.DimRoomType
where AlternativeRoomTypeID =  @RoomTypeId)
BEGIN
update dbo.DimRoomType
set
RoomTypeValue=@RoomTypeValue,
ModifiedDate = GETDATE()
where AlternativeRoomTypeID =  @RoomTypeId
END;
END;
```

```
Commands completed successfully.
```

CustomerType Data from Staging to Data Warehouse


Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Data...iling.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Data Flow Task:
Transform and Load CustomerType Data
Extract from CustomerType Staging
4 rows
Load Dim CustomerType

CustomerType data has been loaded to DimCustomerType table in data warehouse. The following procedure is used to load data into the DimCustomerType table.


SQLQuery2.sql - K...DI-M\KAVINDI (53))*
SQLQuery1.sql - K...DI-M\KAVINDI (55))*

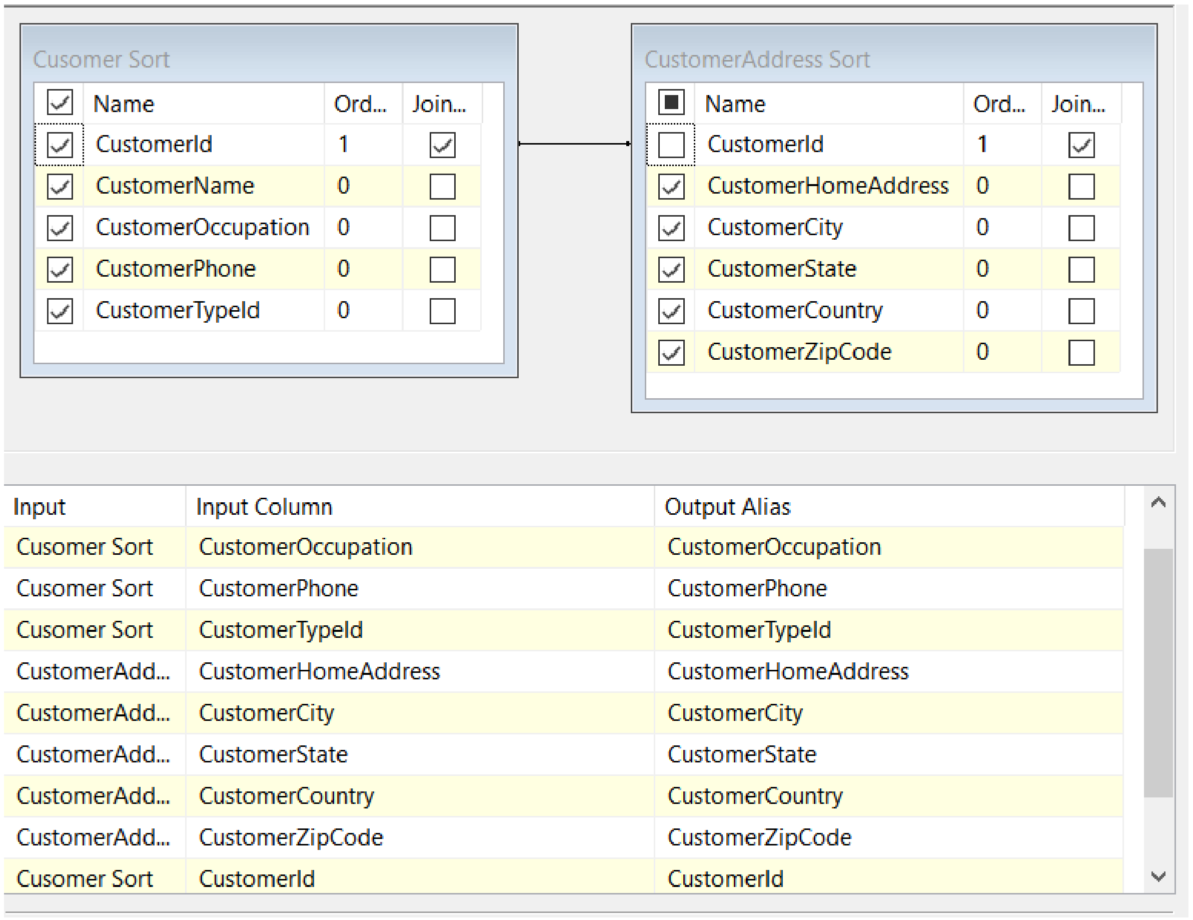
```sql
CREATE PROCEDURE dbo.UpdateDimCustomerType
@CustomerTypeId int,
@CustomerType nvarchar(255)
AS
BEGIN
if not exists (select CustomerTypeSK
from dbo.DimCustomerType
where  AlternativeCustomerTypeID= @CustomerTypeId   )
BEGIN
insert into dbo.DimCustomerType
(AlternativeCustomerTypeID,CustomerType,InsertDate, ModifiedDate)
values
(@CustomerTypeId ,@CustomerType ,GETDATE(), GETDATE())
END;
if exists (select CustomerTypeSK
from dbo.DimCustomerType
where AlternativeCustomerTypeID= @CustomerTypeId    )
BEGIN
update dbo.DimCustomerType
set
CustomerType = @CustomerType,
ModifiedDate = GETDATE()
where AlternativeCustomerTypeID= @CustomerTypeId
END;
END;
```

(b)Load Slowly changing dimensions

Customer Data from Staging to Data Warehouse

CustomerAddress staging table has been merged into Customer table. Merge has been done by sorting the common attribute of both the tables which is CustomerId.Surrogate key of the hierarchical dimension of Customer, which is DimCustomerType has been obtained through a lookup.

Merge Join



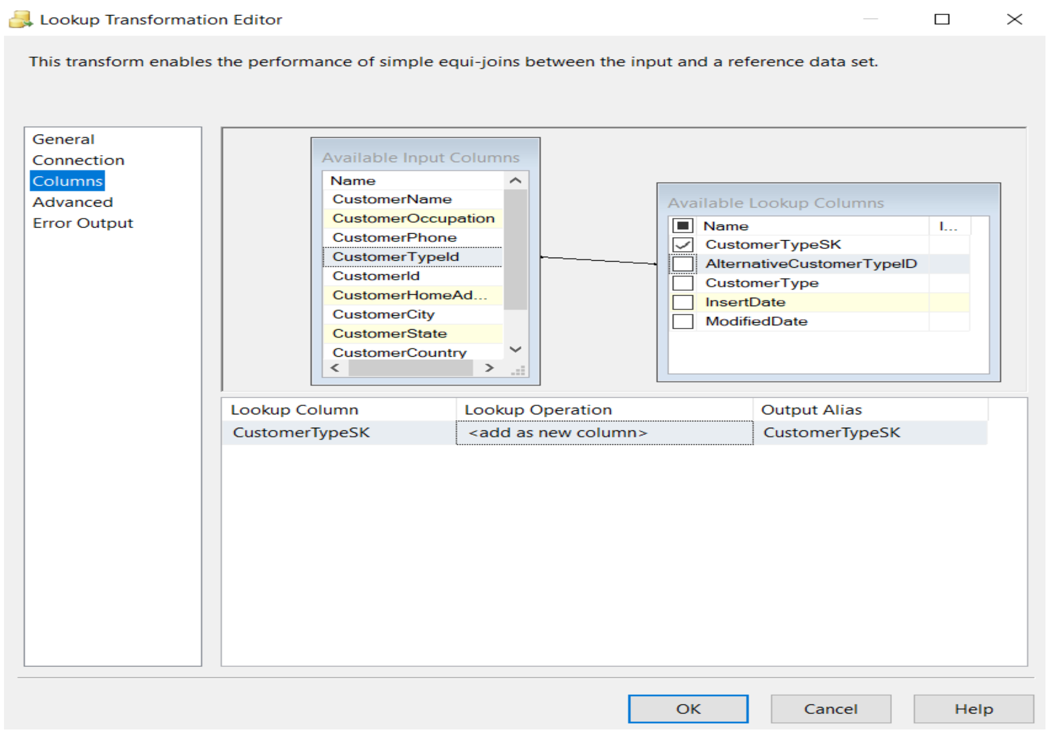
Cusomer Sort

✓	Name	Ord...	Join...
✓	CustomerId	1	✓
✓	CustomerName	0	☐
✓	CustomerOccupation	0	☐
✓	CustomerPhone	0	☐
✓	CustomerTypeId	0	☐

CustomerAddress Sort

■	Name	Ord...	Join...
☐	CustomerId	1	✓
✓	CustomerHomeAddress	0	☐
✓	CustomerCity	0	☐
✓	CustomerState	0	☐
✓	CustomerCountry	0	☐
✓	CustomerZipCode	0	☐

Input	Input Column	Output Alias
Cusomer Sort	CustomerOccupation	CustomerOccupation
Cusomer Sort	CustomerPhone	CustomerPhone
Cusomer Sort	CustomerTypeId	CustomerTypeId
CustomerAdd...	CustomerHomeAddress	CustomerHomeAddress
CustomerAdd...	CustomerCity	CustomerCity
CustomerAdd...	CustomerState	CustomerState
CustomerAdd...	CustomerCountry	CustomerCountry
CustomerAdd...	CustomerZipCode	CustomerZipCode
Cusomer Sort	CustomerId	CustomerId


Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Available Input Columns

Name
CustomerName
CustomerOccupation
CustomerPhone
CustomerTypeId
CustomerId
CustomerHomeAd...
CustomerCity
CustomerState
CustomerCountry

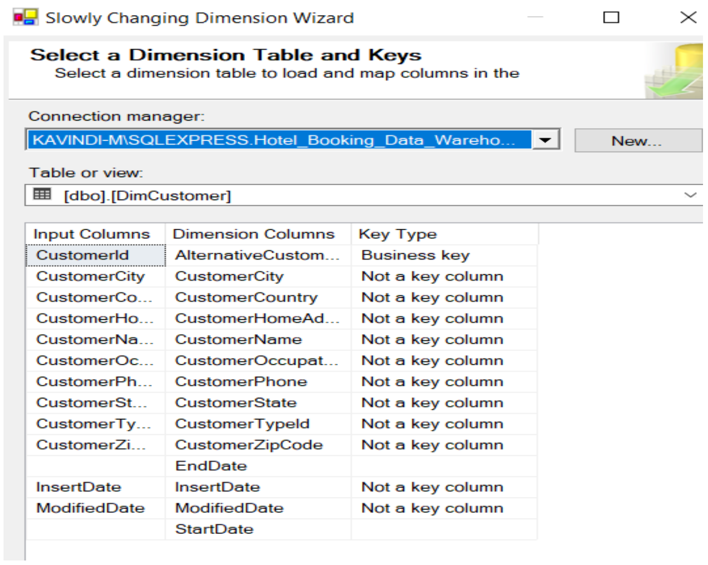
Available Lookup Columns

	Name	I...
✓	CustomerTypeSK	
	AlternativeCustomerTypeID	
	CustomerType	
	InsertDate	
	ModifiedDate	

Lookup Column	Lookup Operation	Output Alias
CustomerTypeSK	<add as new column>	CustomerTypeSK

OK Cancel Help

- As explained above, DimCustomer dimension has been identified as a slowly changing dimension. Hence necessary steps has been followed to make DimCustomer a slowly changing dimension

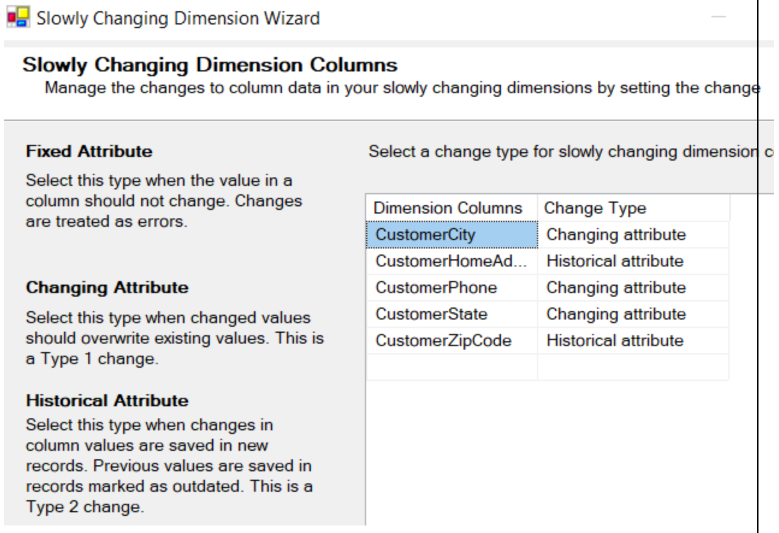

Slowly Changing Dimension Wizard

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

Connection manager:
KAVINDI-M\SQLEXPRESS.Hotel_Booking_Data_Wareho...
New...

Table or view:
[dbo].[DimCustomer]

Input Columns	Dimension Columns	Key Type
CustomerId	AlternativeCustom...	Business key
CustomerCity	CustomerCity	Not a key column
CustomerCo...	CustomerCountry	Not a key column
CustomerHo...	CustomerHomeAd...	Not a key column
CustomerNa...	CustomerName	Not a key column
CustomerOc...	CustomerOccupat...	Not a key column
CustomerPh...	CustomerPhone	Not a key column
CustomerSt...	CustomerState	Not a key column
CustomerTy...	CustomerTypeId	Not a key column
CustomerZi...	CustomerZipCode	Not a key column
	EndDate	
InsertDate	InsertDate	Not a key column
ModifiedDate	ModifiedDate	Not a key column
	StartDate	


Slowly Changing Dimension Wizard

Slowly Changing Dimension Columns
Manage the changes to column data in your slowly changing dimensions by setting the change

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

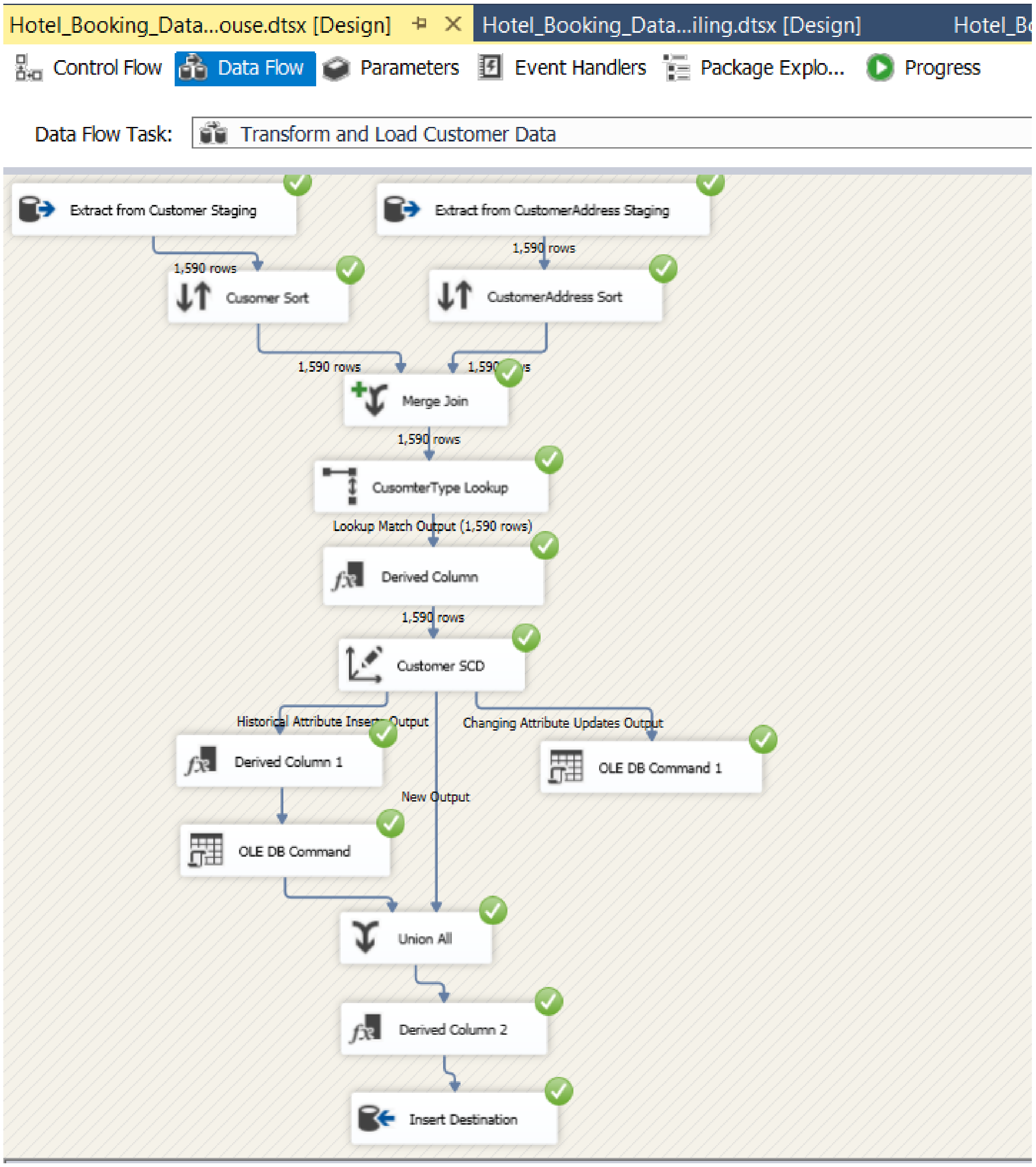
Changing Attribute
Select this type when changed values should overwrite existing values. This is a Type 1 change.

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

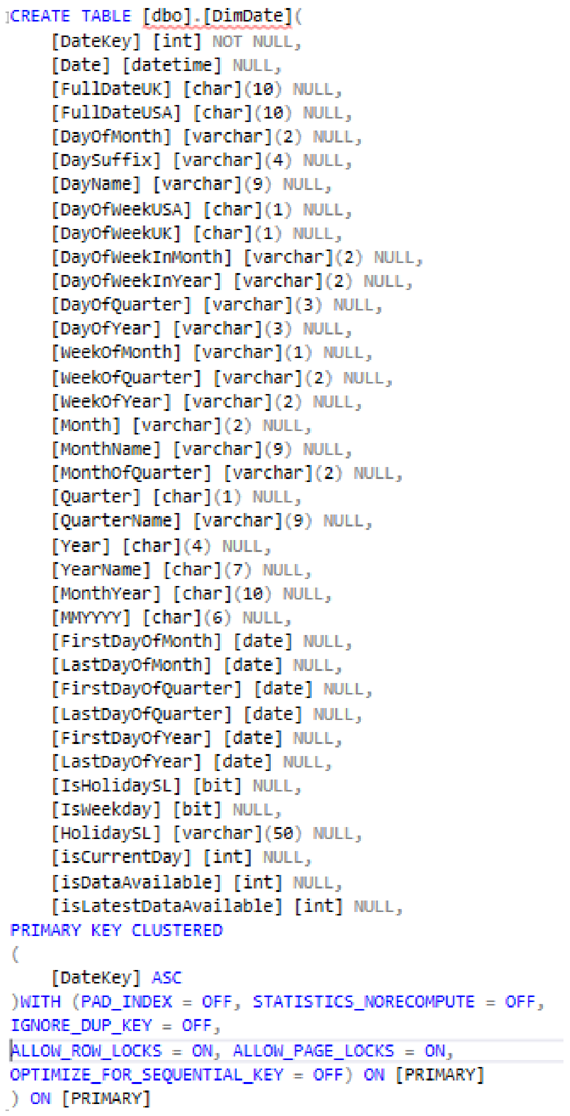
Select a change type for slowly changing dimension c

Dimension Columns	Change Type
CustomerCity	Changing attribute
CustomerHomeAd...	Historical attribute
CustomerPhone	Changing attribute
CustomerState	Changing attribute
CustomerZipCode	Historical attribute

Following all the steps ,finally Customer data has been Loaded to DimCustomer table in data warehouse.

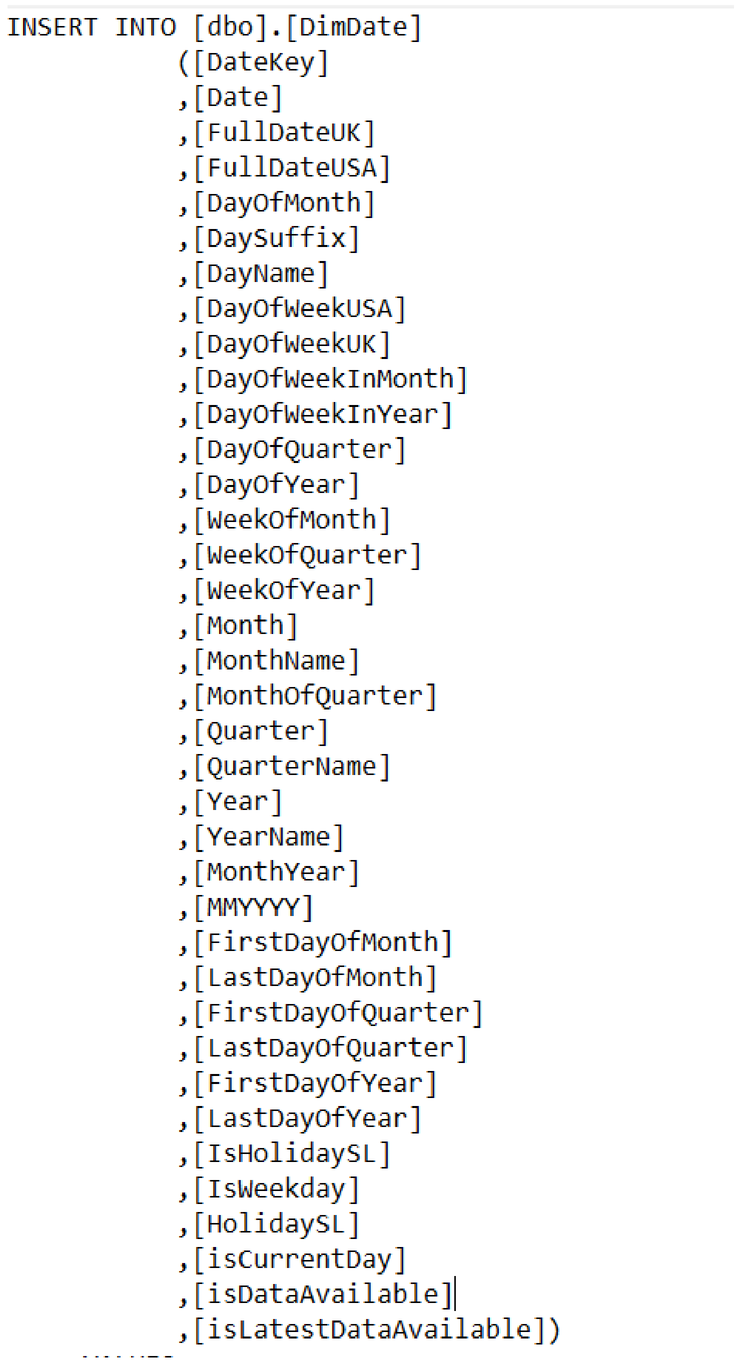


Dim Date Creation

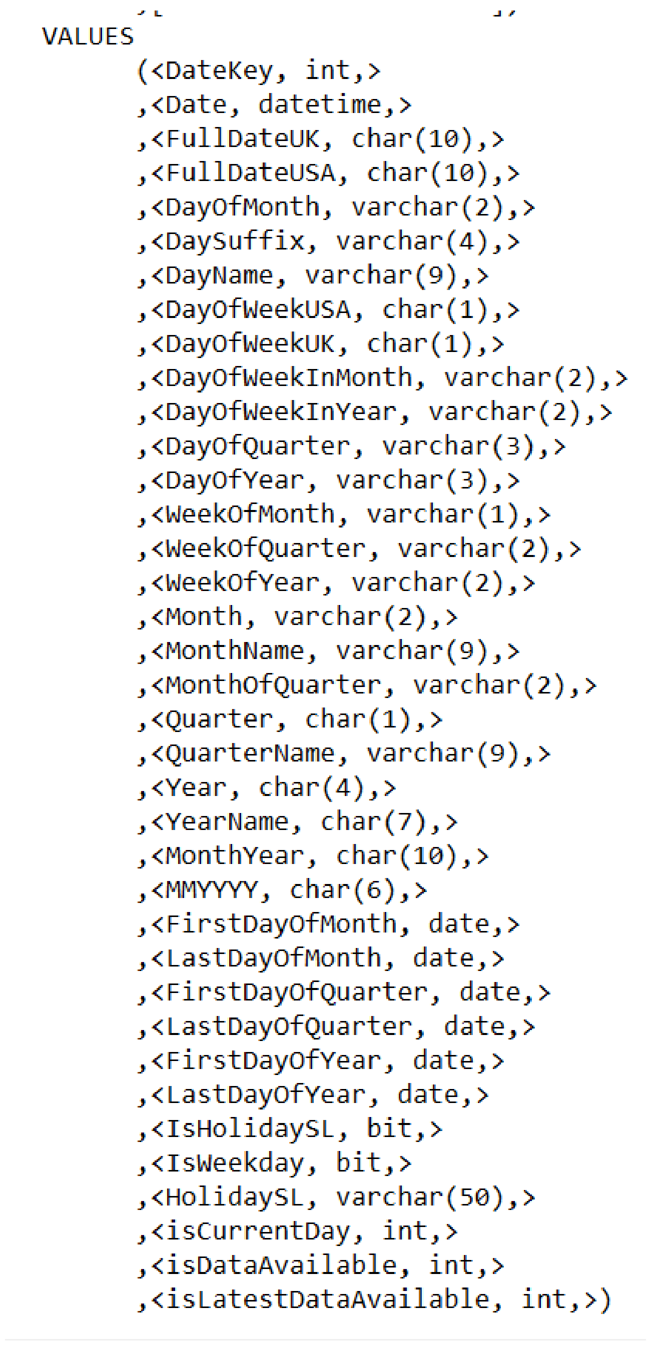


```
CREATE TABLE [dbo].[DimDate](
    [DateKey] [int] NOT NULL,
    [Date] [datetime] NULL,
    [FullDateUK] [char](10) NULL,
    [FullDateUSA] [char](10) NULL,
    [DayOfMonth] [varchar](2) NULL,
    [DaySuffix] [varchar](4) NULL,
    [DayName] [varchar](9) NULL,
    [DayOfWeekUSA] [char](1) NULL,
    [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,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

DimDate Data inserting

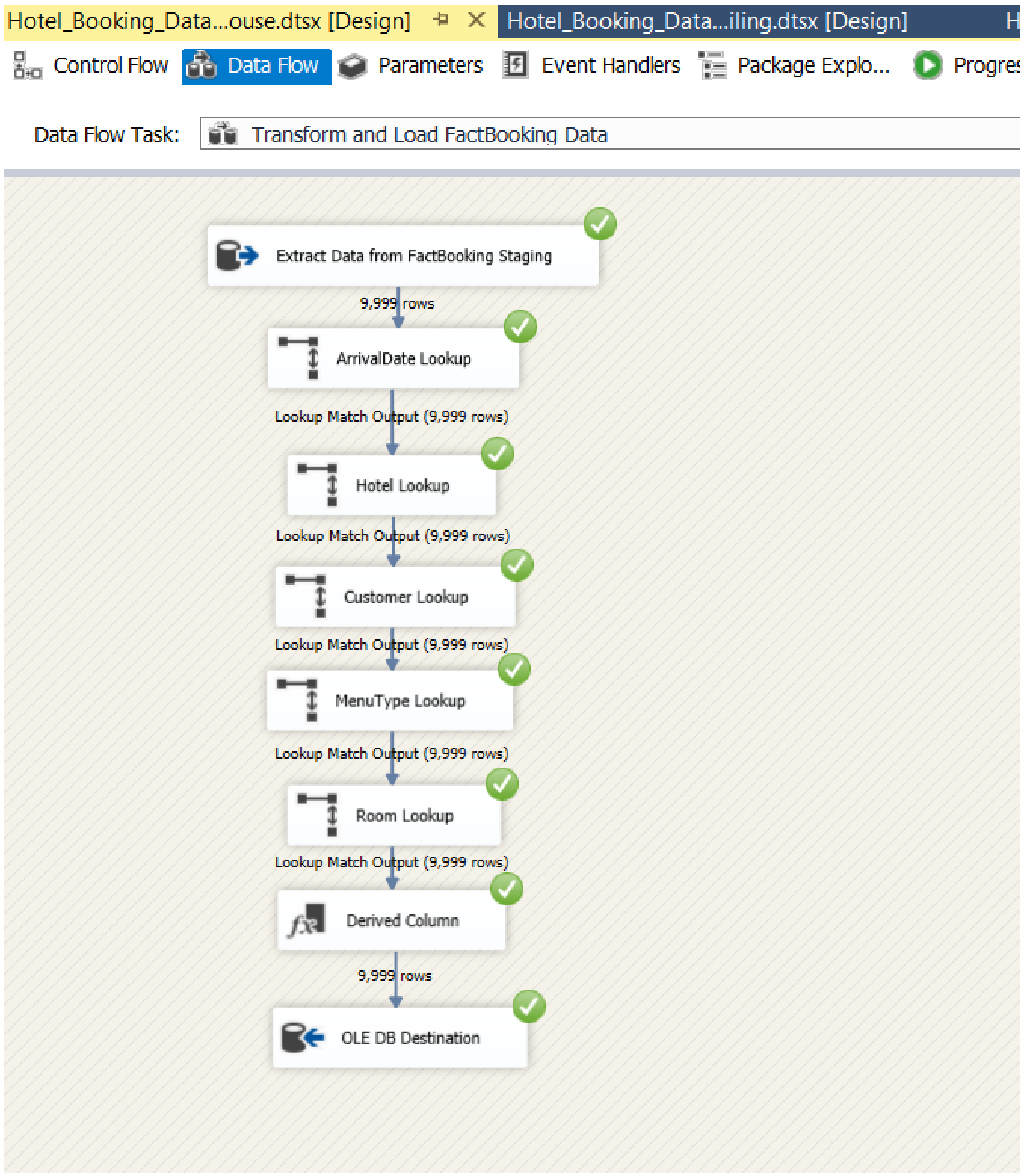


```
INSERT INTO [dbo].[DimDate]
           ([DateKey]
           ,[Date]
           ,[FullDateUK]
           ,[FullDateUSA]
           ,[DayOfMonth]
           ,[DaySuffix]
           ,[DayName]
           ,[DayOfWeekUSA]
           ,[DayOfWeekUK]
           ,[DayOfWeekInMonth]
           ,[DayOfWeekInYear]
           ,[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])
```

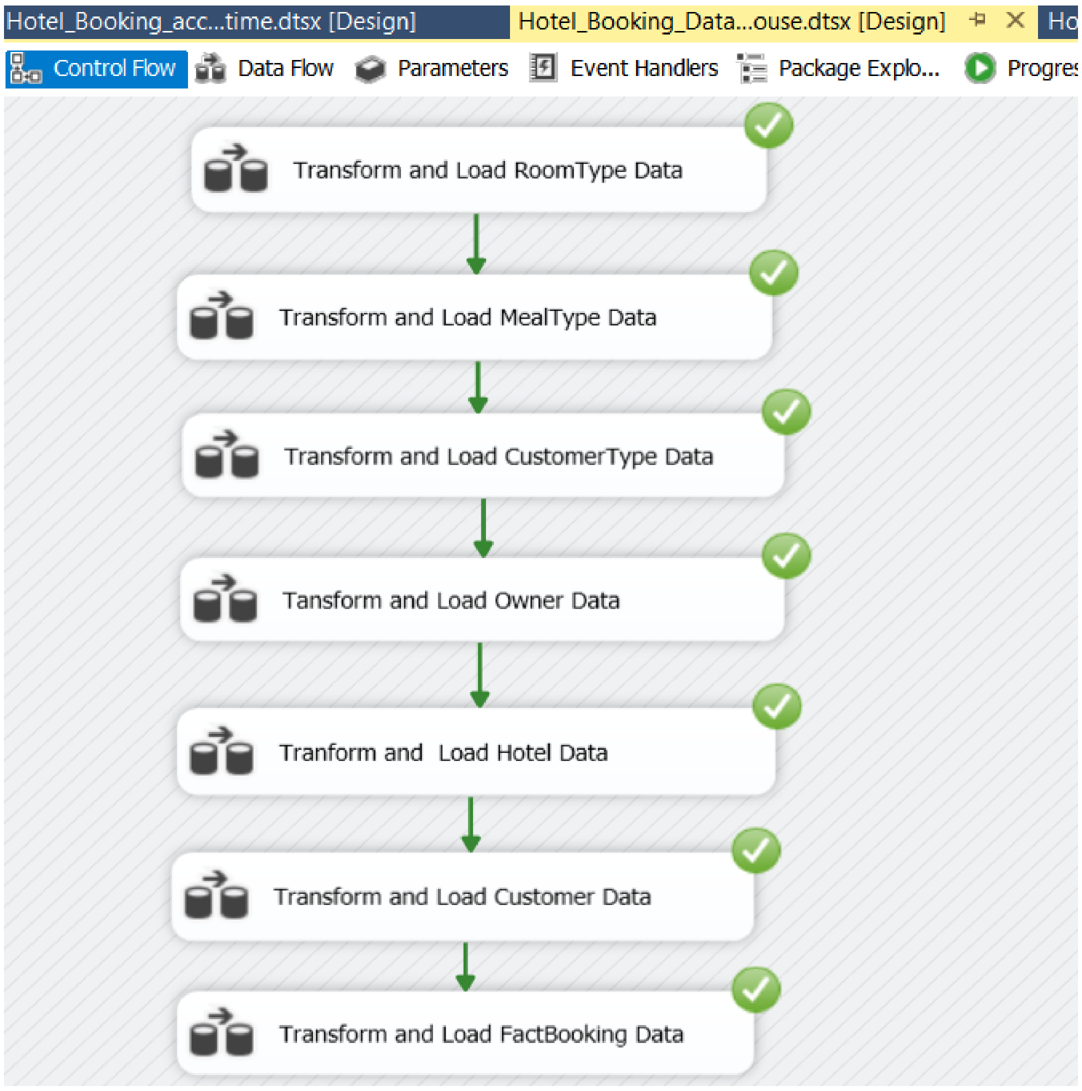


```
     VALUES
           (<DateKey, int,>
           ,<Date, datetime,>
           ,<FullDateUK, char(10),>
           ,<FullDateUSA, char(10),>
           ,<DayOfMonth, varchar(2),>
           ,<DaySuffix, varchar(4),>
           ,<DayName, varchar(9),>
           ,<DayOfWeekUSA, char(1),>
           ,<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,>)
```

(C)Load Fact Table

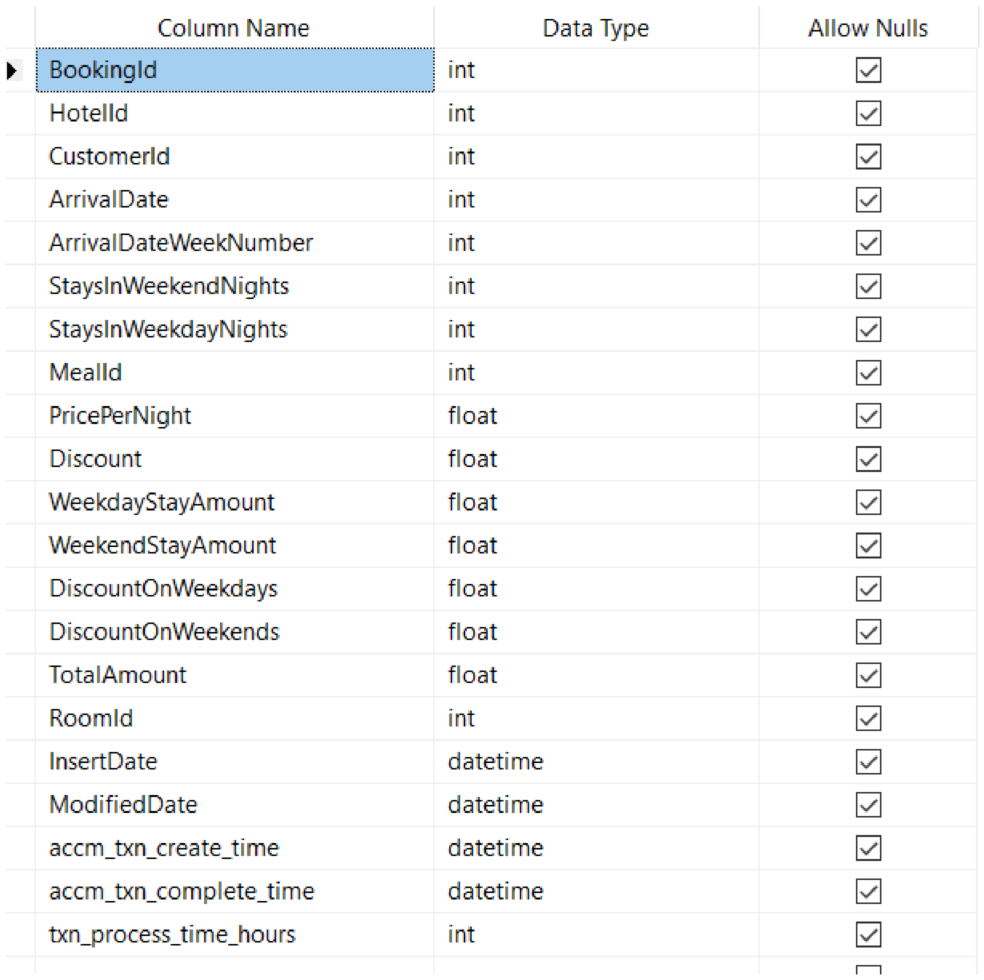

Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Booking_Data...iling.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progres
Data Flow Task:
Transform and Load FactBooking Data
Extract Data from FactBooking Staging
9,999 rows
ArrivalDate Lookup
Lookup Match Output (9,999 rows)
Hotel Lookup
Lookup Match Output (9,999 rows)
Customer Lookup
Lookup Match Output (9,999 rows)
MenuType Lookup
Lookup Match Output (9,999 rows)
Room Lookup
Lookup Match Output (9,999 rows)
Derived Column
9,999 rows
OLE DB Destination

Overall ETL to data warehouse


Hotel_Booking_acc...time.dtsx [Design]
Hotel_Booking_Data...ouse.dtsx [Design]
Ho
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progres
Transform and Load RoomType Data
Transform and Load MealType Data
Transform and Load CustomerType Data
Transform and Load Owner Data
Tranform and Load Hotel Data
Transform and Load Customer Data
Transform and Load FactBooking Data

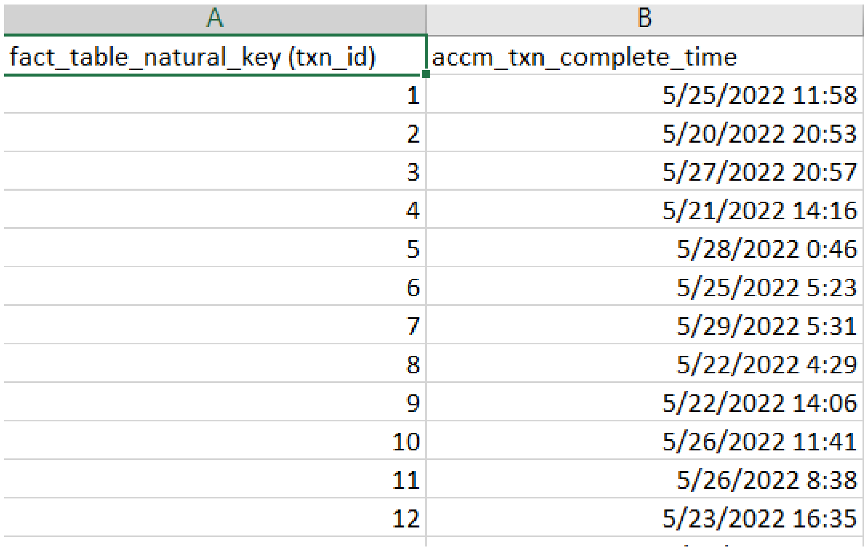
(6)ETL development – Accumilating fact tables

The fact table was extended by adding the last 3 rows as shown below.



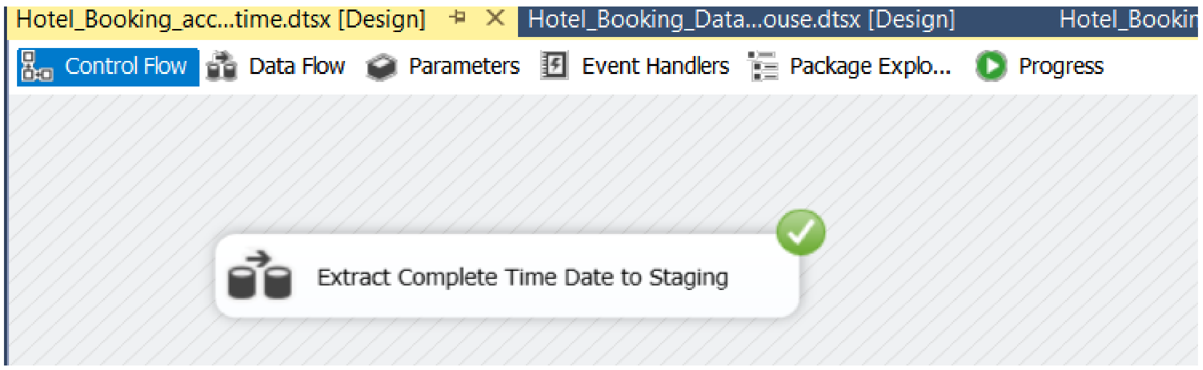
Column Name	Data Type	Allow Nulls
BookingId	int	✓
HotelId	int	✓
CustomerId	int	✓
ArrivalDate	int	✓
ArrivalDateWeekNumber	int	✓
StaysInWeekendNights	int	✓
StaysInWeekdayNights	int	✓
MealId	int	✓
PricePerNight	float	✓
Discount	float	✓
WeekdayStayAmount	float	✓
WeekendStayAmount	float	✓
DiscountOnWeekdays	float	✓
DiscountOnWeekends	float	✓
TotalAmount	float	✓
RoomId	int	✓
InsertDate	datetime	✓
ModifiedDate	datetime	✓
accm_txn_create_time	datetime	✓
accm_txn_complete_time	datetime	✓
txn_process_time_hours	int	✓

Then a separate data source (csv file) named CompletionTimw.csv was created and the structure of this file is shown below

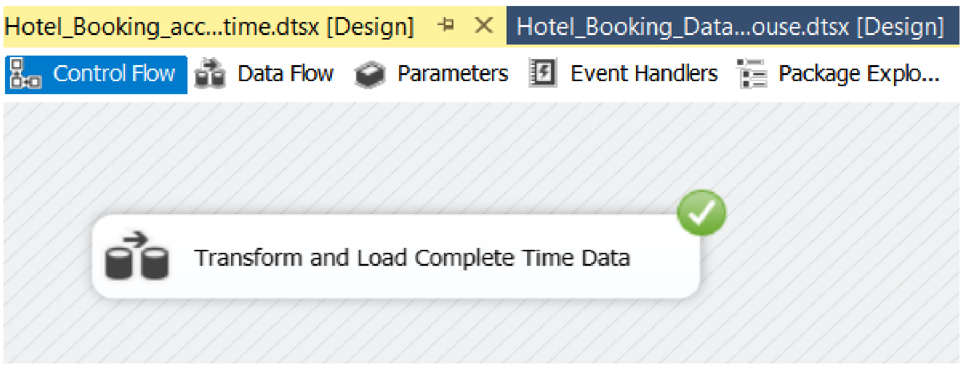


A	B
fact_table_natural_key (txn_id)	accm_txn_complete_time
1	5/25/2022 11:58
2	5/20/2022 20:53
3	5/27/2022 20:57
4	5/21/2022 14:16
5	5/28/2022 0:46
6	5/25/2022 5:23
7	5/29/2022 5:31
8	5/22/2022 4:29
9	5/22/2022 14:06
10	5/26/2022 11:41
11	5/26/2022 8:38
12	5/23/2022 16:35

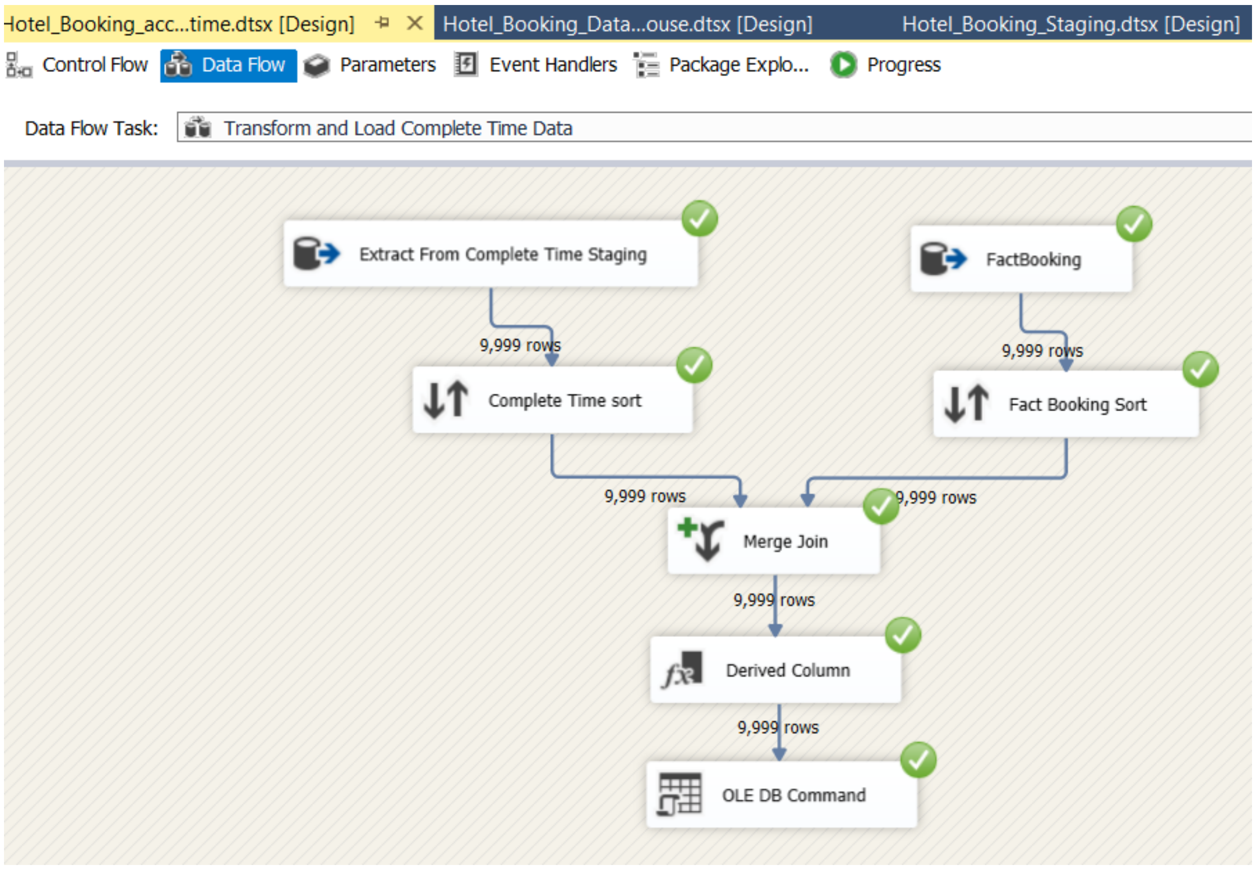
Extract CompleteTime Data to Staging area


Hotel_Booking_acc...time.dtsx [Design]
Hotel_Booking_Data...ouse.dtsx [Design]
Hotel_Bookin
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Progress
Extract Complete Time Date to Staging

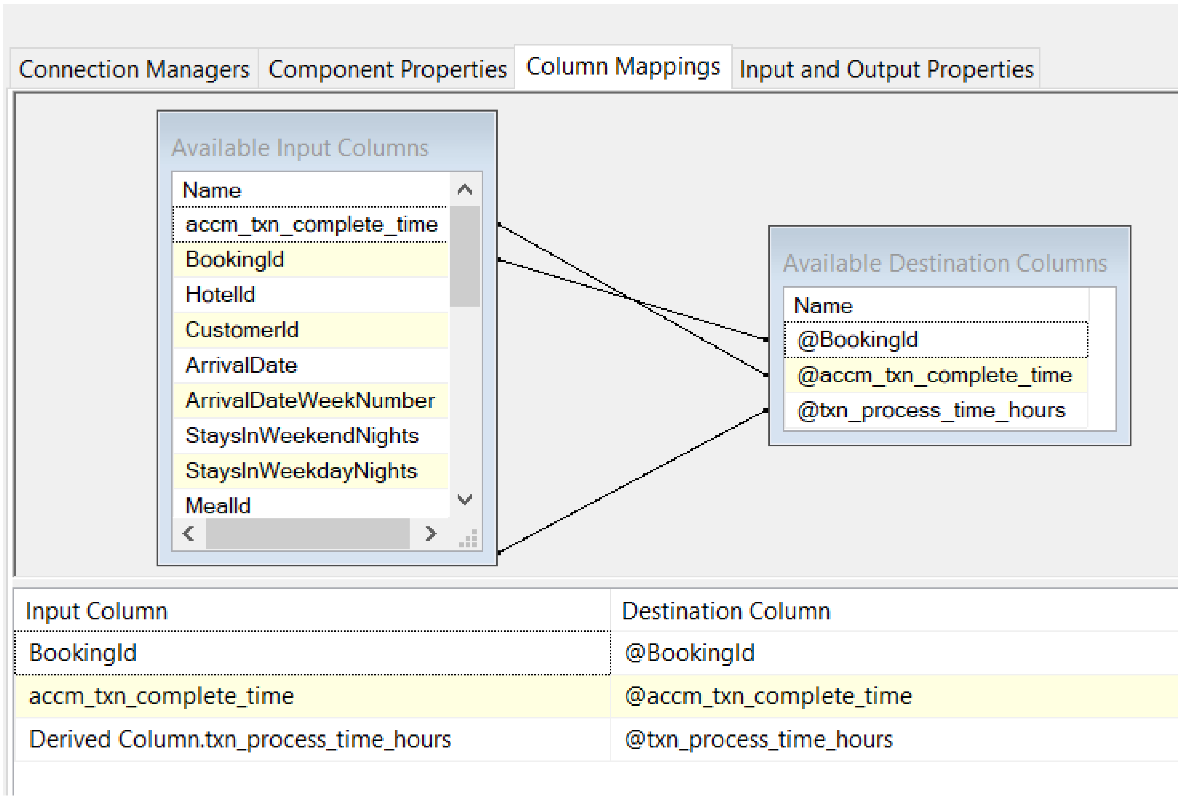
Hotel Data from Staging to Data Warehouse


Hotel_Booking_acc...time.dtsx [Design]
Hotel_Booking_Data...ouse.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Transform and Load Complete Time Data

Data from the Complete Time staging table and FactBooking fact table were extracted and merged. Merge has been performed by sorting both the tables using the fields fact_table_natural_key (txn_id) & BookingId respectively. Finally, the merged data was loaded to the FactBooking fact table in the data warehouse.

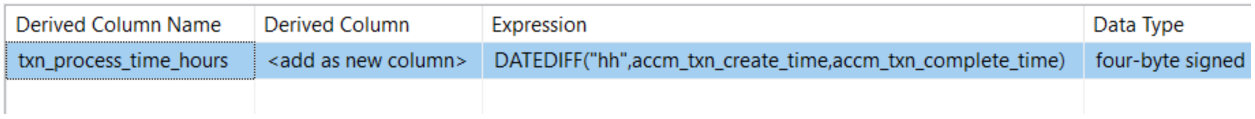


Relevant column mappings is shown below



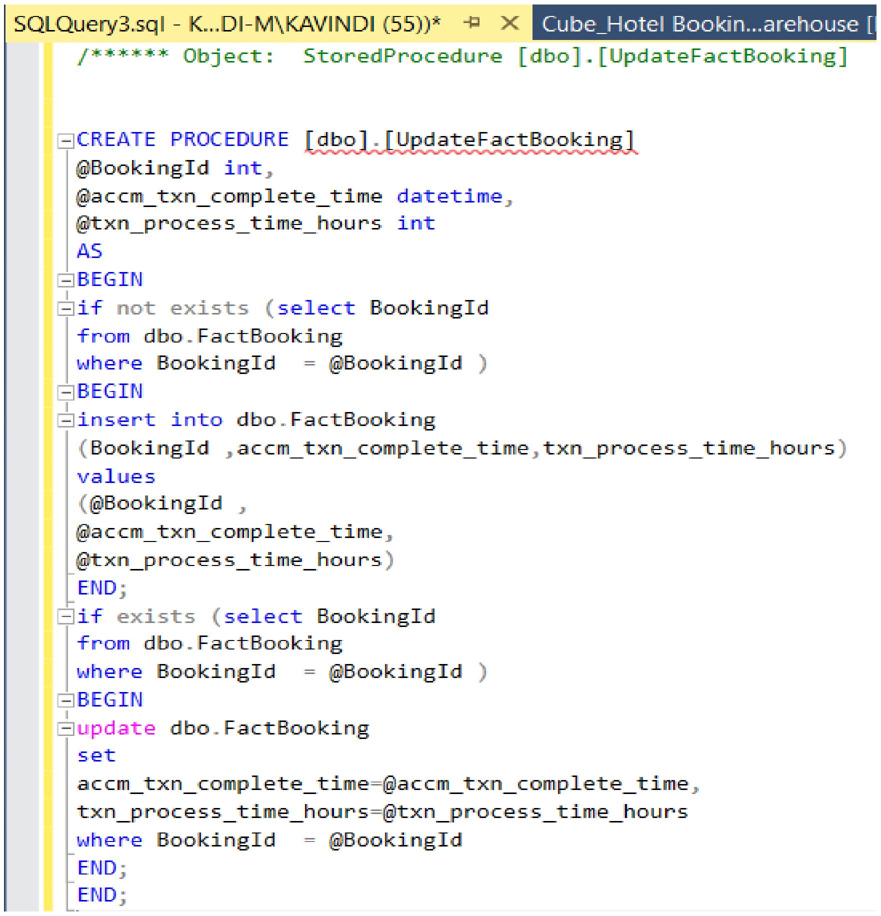
Input Column	Destination Column
BookingId	@BookingId
accm_txn_complete_time	@accm_txn_complete_time
Derived Column.txn_process_time_hours	@txn_process_time_hours

A Derived Columns task has been used to derive the values for txn_process_time_hours column by getting the date difference of accn_txn_complete_time & accn_txn_create_time



Derived Column Name	Derived Column	Expression	Data Type
txn_process_time_hours	<add as new column>	DATEDIFF("hh",accm_txn_create_time,accm_txn_complete_time)	four-byte signed

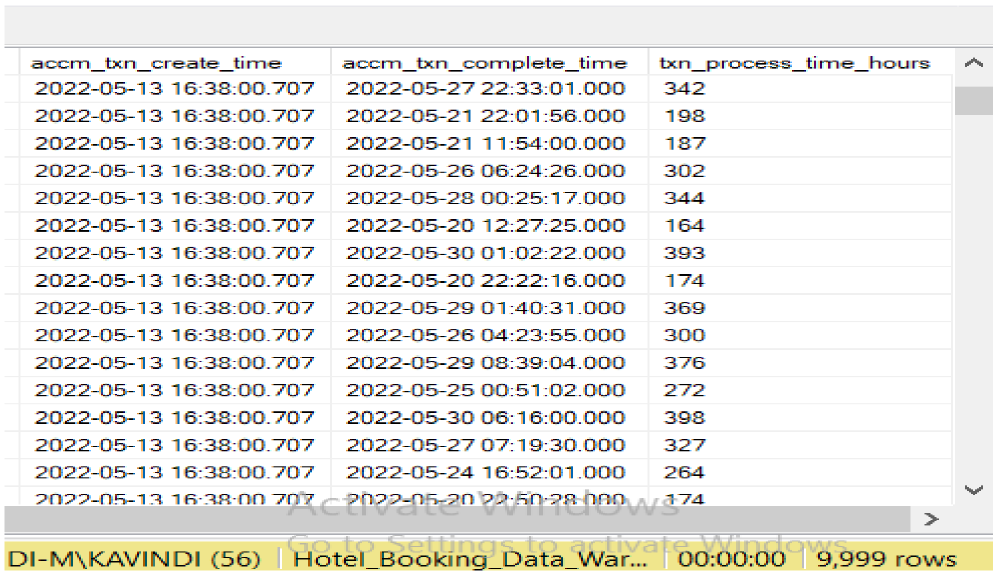
The following procedure is used in order to load the data


SQLQuery3.sql - K...DI-M\KAVINDI (55))* Cube_Hotel Bookin...arehouse [

```
/****** Object:  StoredProcedure [dbo].[UpdateFactBooking]


CREATE PROCEDURE [dbo].[UpdateFactBooking]
@BookingId int,
@accm_txn_complete_time datetime,
@txn_process_time_hours int
AS
BEGIN
if not exists (select BookingId
from dbo.FactBooking
where BookingId  = @BookingId )
BEGIN
insert into dbo.FactBooking
(BookingId ,accm_txn_complete_time,txn_process_time_hours)
values
(@BookingId ,
@accm_txn_complete_time,
@txn_process_time_hours)
END;
if exists (select BookingId
from dbo.FactBooking
where BookingId  = @BookingId )
BEGIN
update dbo.FactBooking
set
accm_txn_complete_time=@accm_txn_complete_time,
txn_process_time_hours=@txn_process_time_hours
where BookingId  = @BookingId
END;
END;
```

A screenshot of the FactBookings fact table of the data warehouse after accumulating (Completing Step 6) is shown below.



accm_txn_create_time	accm_txn_complete_time	txn_process_time_hours
2022-05-13 16:38:00.707	2022-05-27 22:33:01.000	342
2022-05-13 16:38:00.707	2022-05-21 22:01:56.000	198
2022-05-13 16:38:00.707	2022-05-21 11:54:00.000	187
2022-05-13 16:38:00.707	2022-05-26 06:24:26.000	302
2022-05-13 16:38:00.707	2022-05-28 00:25:17.000	344
2022-05-13 16:38:00.707	2022-05-20 12:27:25.000	164
2022-05-13 16:38:00.707	2022-05-30 01:02:22.000	393
2022-05-13 16:38:00.707	2022-05-20 22:22:16.000	174
2022-05-13 16:38:00.707	2022-05-29 01:40:31.000	369
2022-05-13 16:38:00.707	2022-05-26 04:23:55.000	300
2022-05-13 16:38:00.707	2022-05-29 08:39:04.000	376
2022-05-13 16:38:00.707	2022-05-25 00:51:02.000	272
2022-05-13 16:38:00.707	2022-05-30 06:16:00.000	398
2022-05-13 16:38:00.707	2022-05-27 07:19:30.000	327
2022-05-13 16:38:00.707	2022-05-24 16:52:01.000	264
2022-05-13 16:38:00.707	2022-05-20 22:50:28.000	174

DI-M\KAVINDI (56) | Hotel_Booking_Data_War... | 00:00:00 | 9,999 rows

Thank You!