Data Warehousing & Business Intelligent (IT)

3rd Year, 1st Semester

Assignment 1

Submitted to

Sri Lanka Institute of Information Technology

Bachelor of Science Special Honors Degree in Data Science

IT19167510 Bandara J.M.S.A

Weekday Batch

Step 1-*Description Of The Data Set*

San Francisco Building permits - kaggle.com

I selected the San Francisco Building Permits as the data set. It consists of a large CSV file with a small xlsx file. Furthermore , I have partitioned the large CSV file into small sub CSV files. The sub CSV files consists of new IDs.

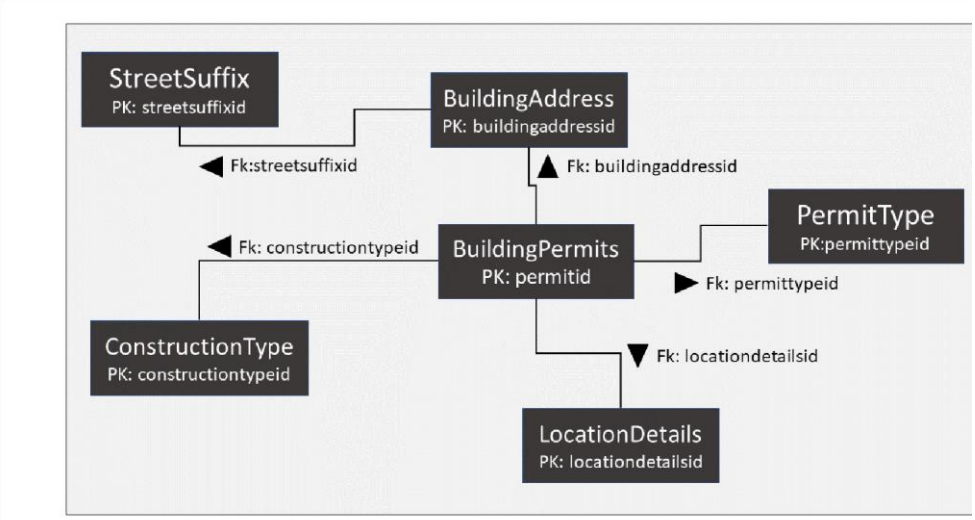
A building permit is an official approval document issued by a governmental agency that allows San Francisco's contractor to proceed with a construction or remodeling project on one's property. San Francisco has its own office related to buildings, that can do multiple functions like issuing permits, inspecting buildings to enforce safety measures, modifying rules to accommodate needs of the growing population etc.

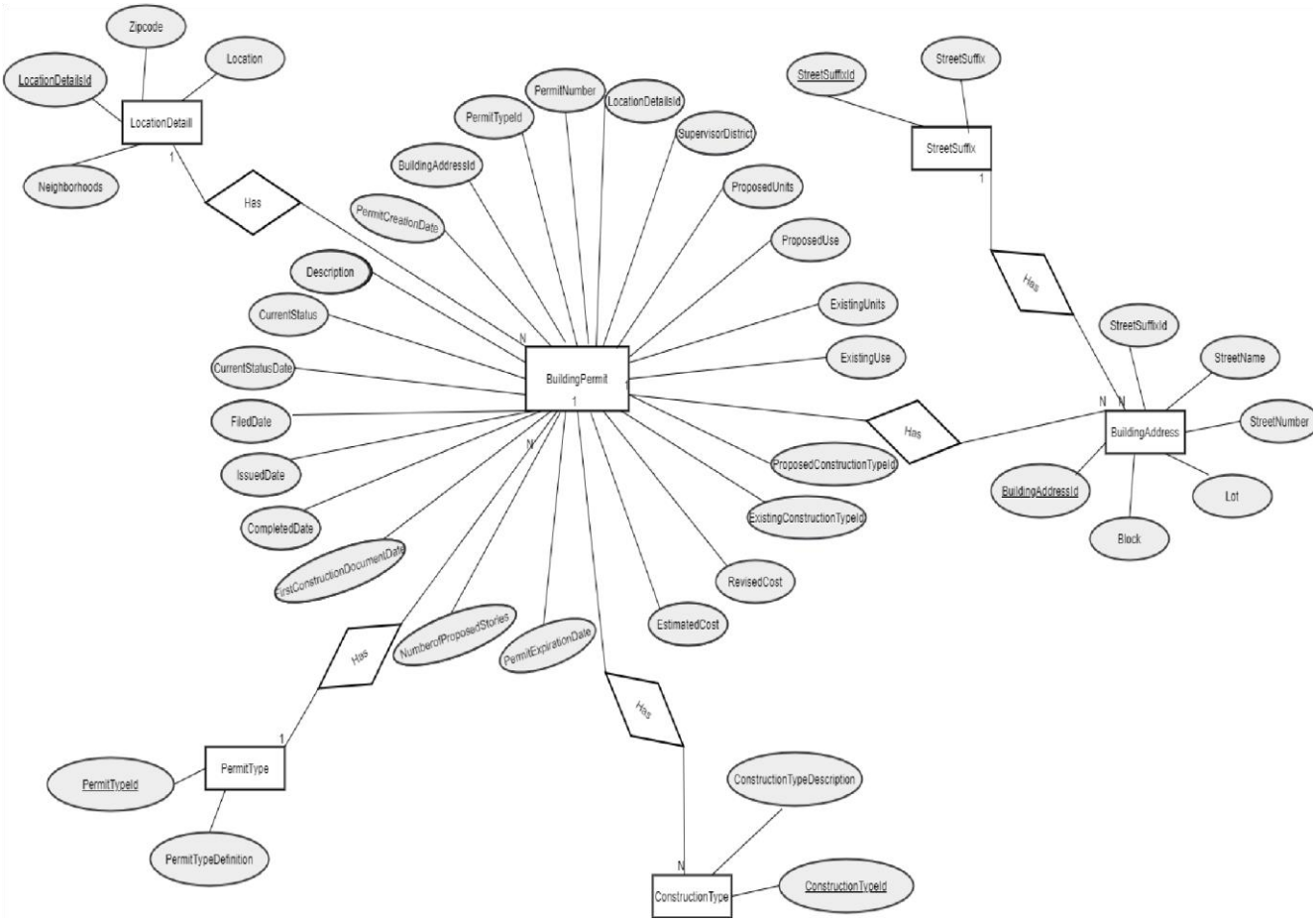
Reason for importance :- In the recent past, main discrepancy in demand and supply in real estate industry is due to delays in issuing building permits

Data set has these files named :

- Building_Permits.csv
- DataDictionaryBuildingPermit.xlsx

ER Diagram





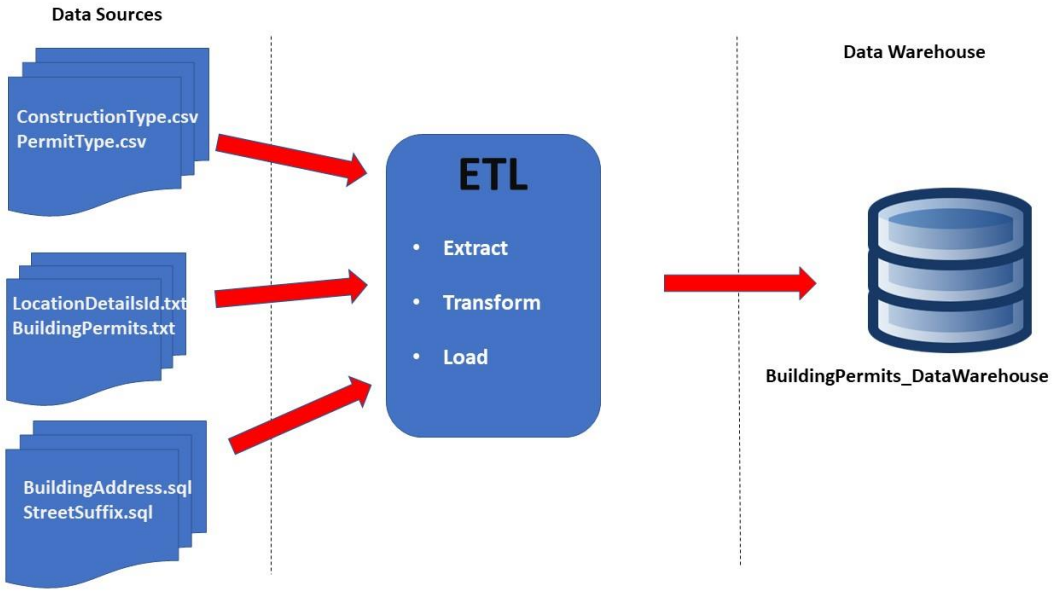
Step 2 – *Preparation of data sources*

This implement data warehouse used data set view, it is Building Permits data set. It partitioned in to separate Source files to implement data warehouse, such as *Constructiontype.csv , PermitType.csv,LocationDetailsId.txt,BuildingPermits.txt,BuildingAddress.sql and streetSuffix.sql*

Source	Source Type	Table Name	Column Name	Data Type
ConstructionType.csv	csv	ConstructionType	ConstructionTypeId	nvarchar(8)
			ConstructionTypeDescription	varchar(50)
PermitType.csv	csv	PermitType	PermitTypeId	nvarchar(8)
			PermitTypeDefinition	varchar(50)
LocationDetailsId.txt	Text File	LocationDetailsId	LocationDetailsId	nvarchar(8)
			Neighborhoods	varchar(50)
			Zipcode	numeric(18,0)
			Location	varchar(50)
BuildingPermits_Source	Microsoft SQL Database	BuildingAddress	BuildingAddressId	nvarchar(8)
			Block	nvarchar(50)
			Lot	varchar(50)
			StreetNumber	varchar(50
			StreetName	varchar(50)
			StreetSuffixId	nvarchar(8)
		StreetSuffix	StreetSuffixId	nvarchar(8)
			StreetSuffix	varchar(50)

BuildingPermits.txt	Text File	BuildingPermits	PermitNumber	varchar(50)
			PermitTypeId	int
			BuildingAddressId	int
			PermitCreationDate	varchar(50)
			Description	varchar(50)
			CurrentStatus	int
			CurrentStatusDate	int
			FiledDate	int
			IssuedDate	int
			CompletedDate	int
			FirstConstructionDocumentDa	numeric(18,0)
			NumberofExistingStories	numeric(18,0)
			NumberofProposedStories	int
			PermitExpirationDate	money
			EstimatedCost	money
			RevisedCost	int
			ExistingConstructionTypeId	int
			ProposedConstructionTypeId	varchar(50)
			ExistingUse	numeric(18,0)
			ExistingUnits	varchar(50)
			ProposedUse	numeric(18,0)
			ProposedUnits	numeric(18,0)
			SupervisorDistrict	int
			LocationDetailsId	int

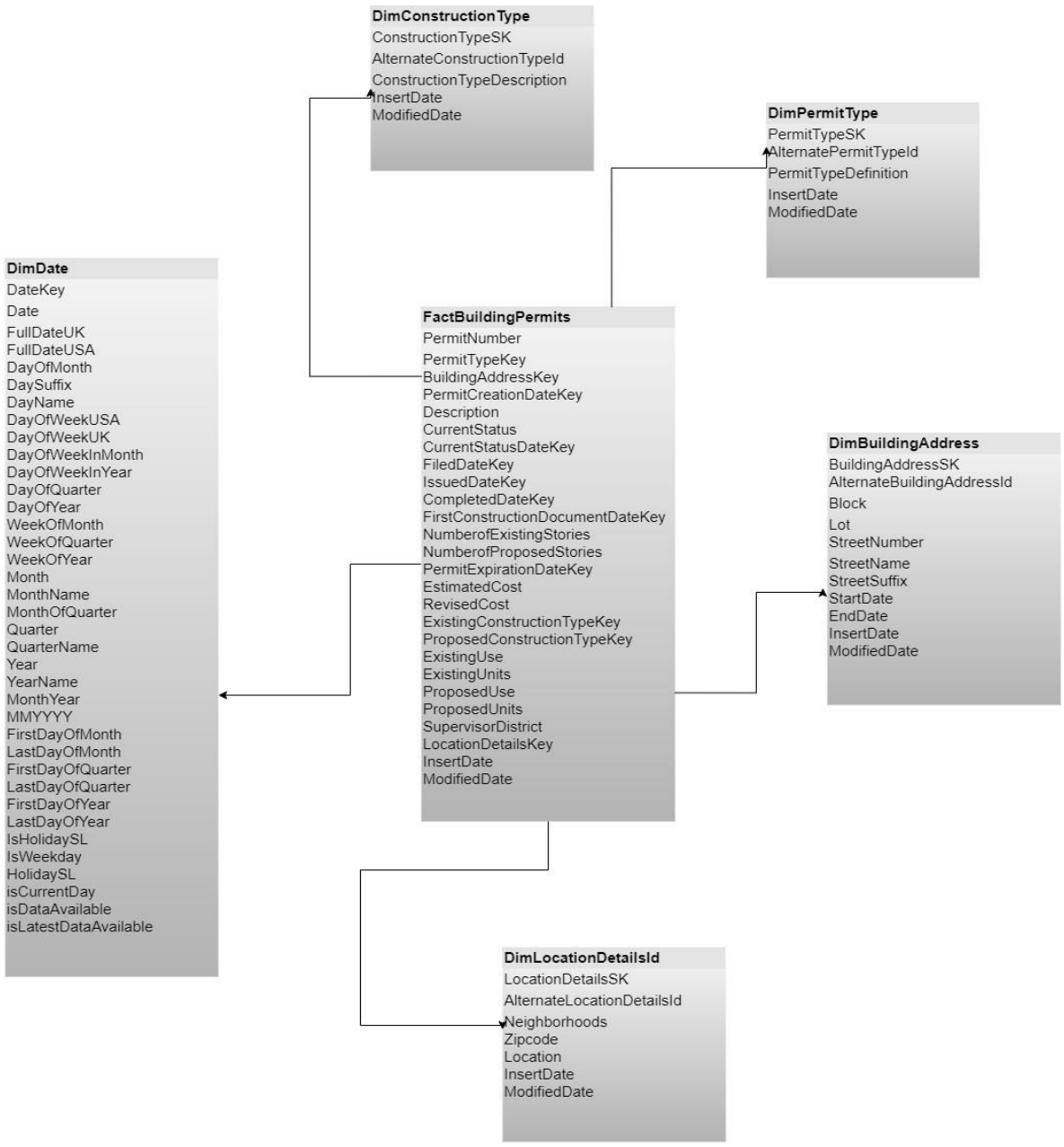
Step 3 - *Solution architecture*



The datawarehouse is the core of the BI system.We design a datawarehouse for the purpose of the analysis and reporting.

Step 4: Data warehouse design & development

This diagram show you how the fact table and dimension tables are joined in a logical manner.In this fact table,I stored the measurable values and key values.

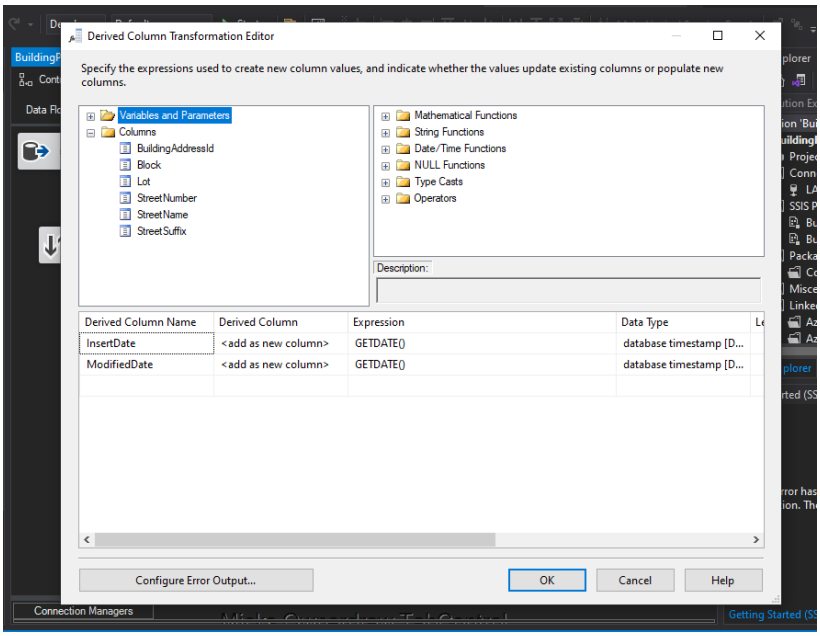


This Datawarehouse design consists of 5 different dimension table with one fact table. This is a star schema

✞ Assumption

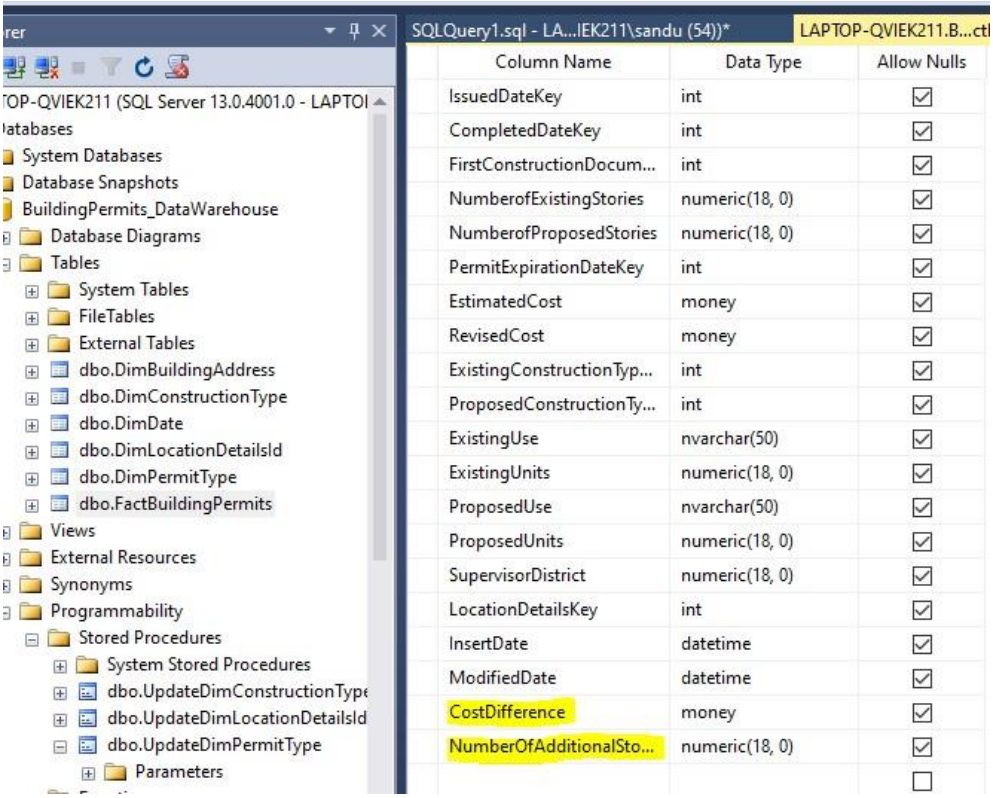
I have taken **DimBuildingAddress** as **Slowly changing dimension**,Because I assume Street name can change time to time,So we need to keep track of their historical street names. After that

I had to create two drived attributes named Insertdate and ModifiedDate when I create this slowly changing dimension.

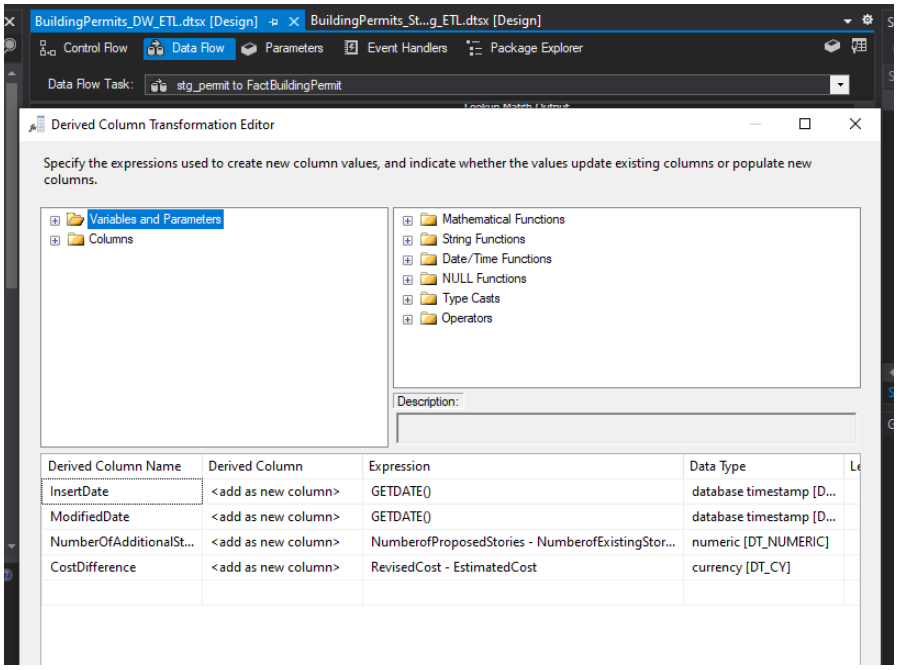


I created **NumberOfAdditionalStories** and **CostDifference** in the **factBuildingPermits** .

- **NumberOfAdditionalStories = NumberofExistingStories - NumberofProposedStories**
- **CostDifference = RevisedCost – EstimatedCost**



To create these two mesurable values firstly I created a stage called stg_permit to factBuilding Permit.Then I created its' data flow task and in a derived column component I created the formula for this values.

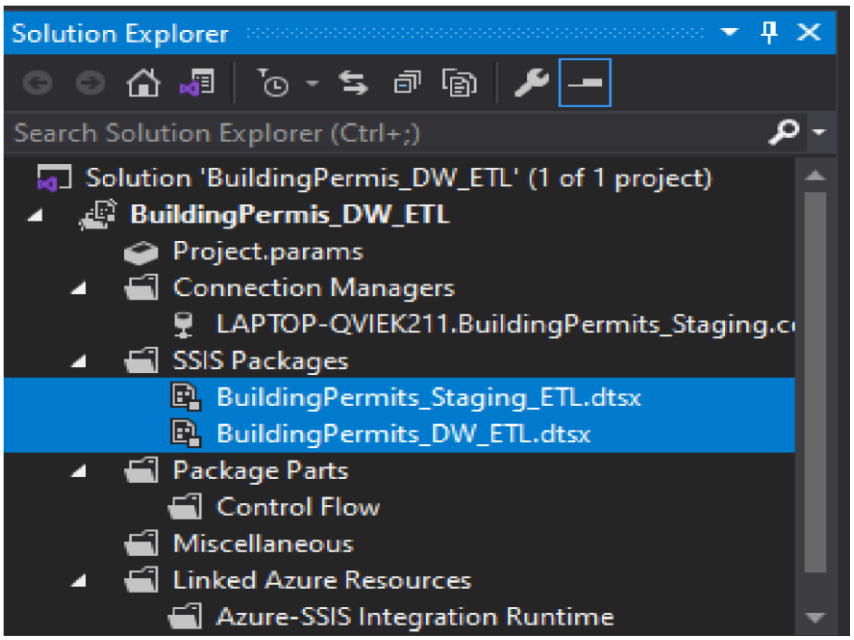


Step 5: ETL development

In the ETL process ,I created ETL process into two Packages

They are ,

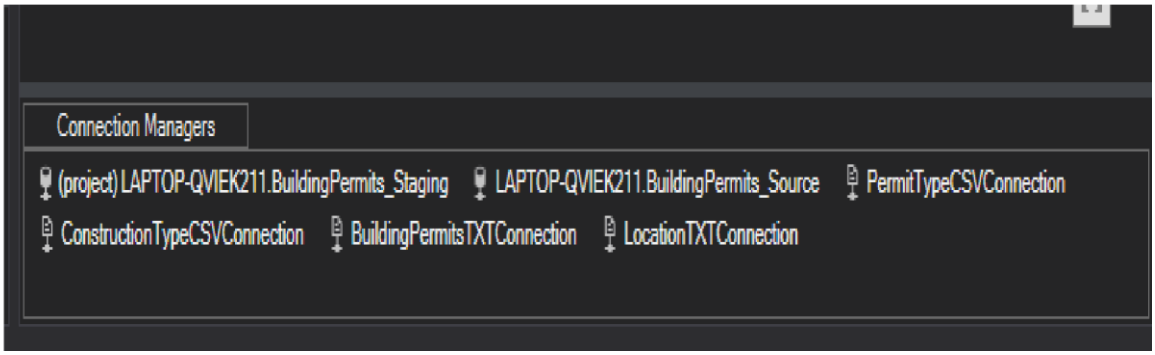
- ✦ BuildingPermits_Staging_ETL.dtsx – Source to Staging
- ✦ BuildingPermits_DW_ETL.dtsx – Staging to Datawarehouse



✞ BuildingPermits_Staging_ETL.dtsx

I created several connections in the BuildingPermits_Staging_ETL package

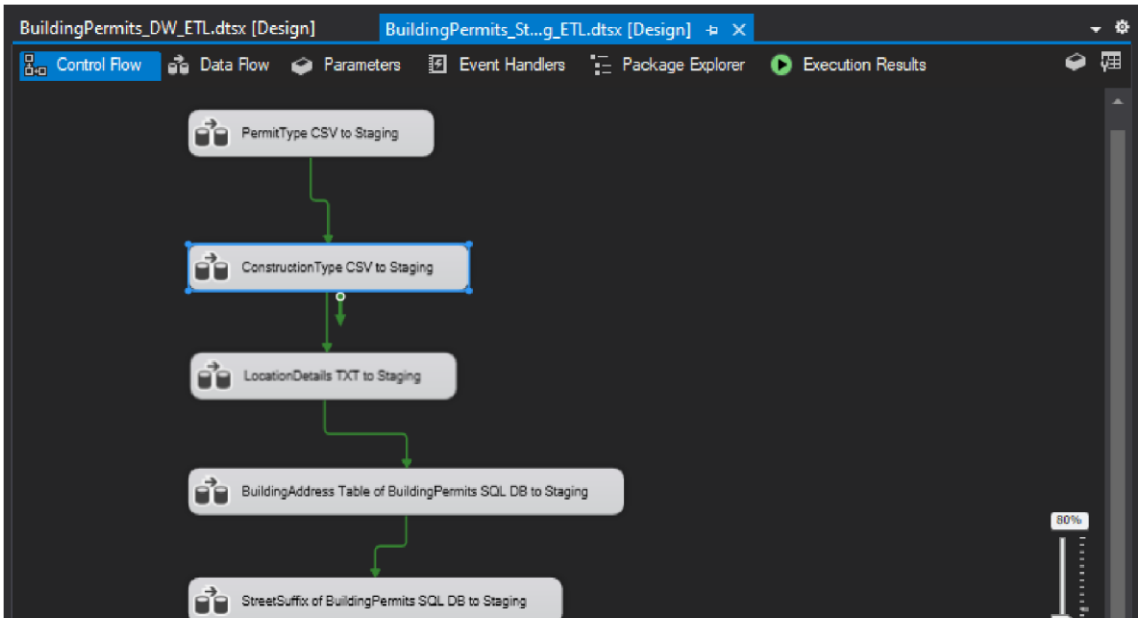
Connection	Description
BuildingPermits_Staging	Connection for OLEDB
BuildingPermits_Sources	Connection for OLEDB
PermiteTypeCSVConnection	connection for platfile
ConstructionTypeCSVConnection	connection for platfile
BuildingPermitsTXTConnection	Connection for platfile
LocationTXTConnection	Connection for platfile



First stage – PermitType CSV to Staging

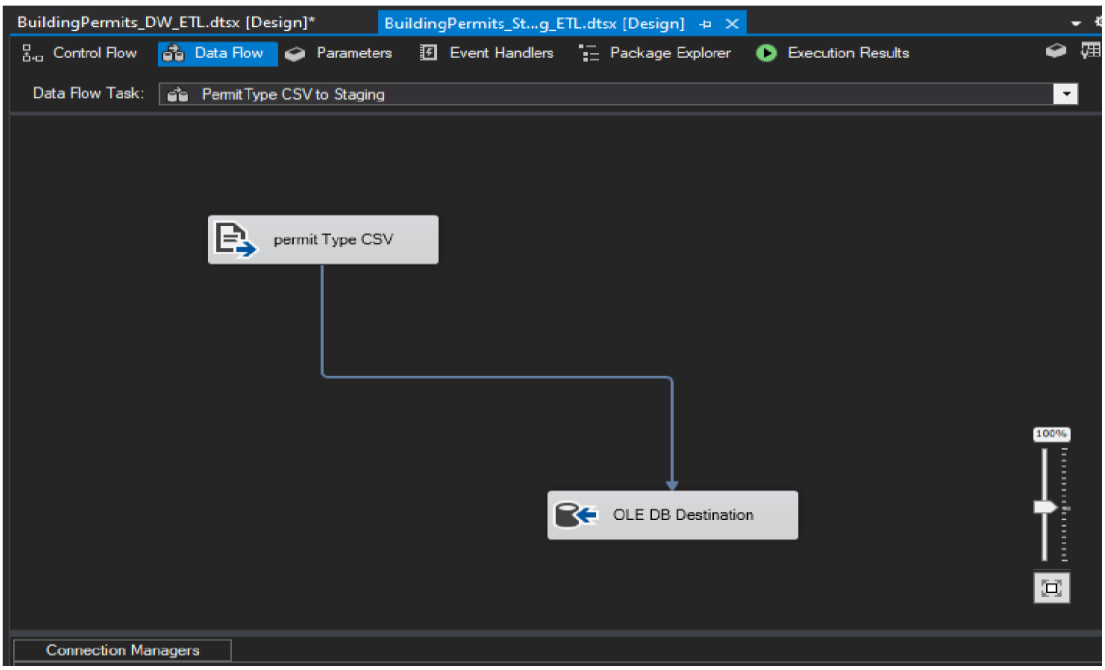
In the control Flow

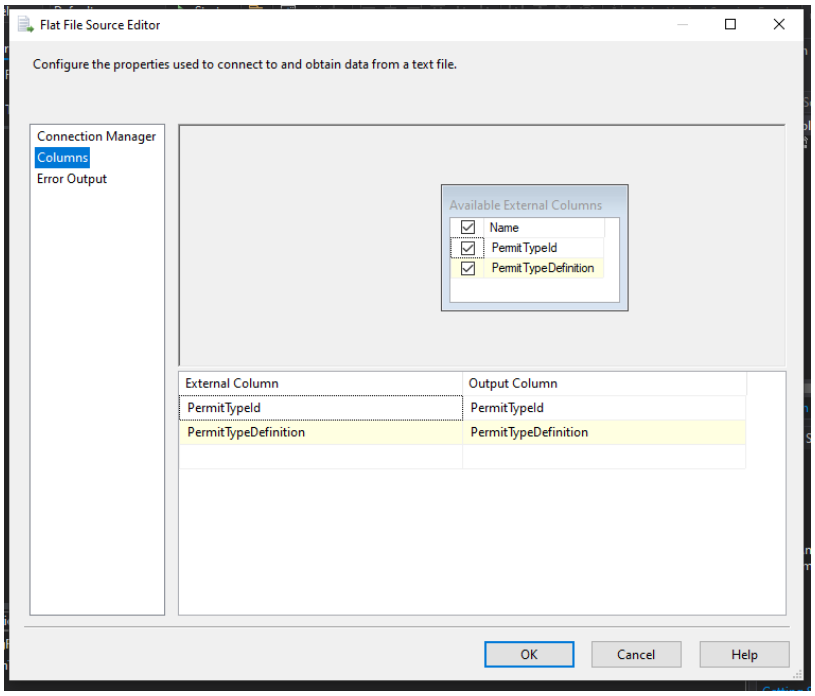
Firstly I created the control flow of the BuildingPermits_Staging_ETL.



In the Data flow

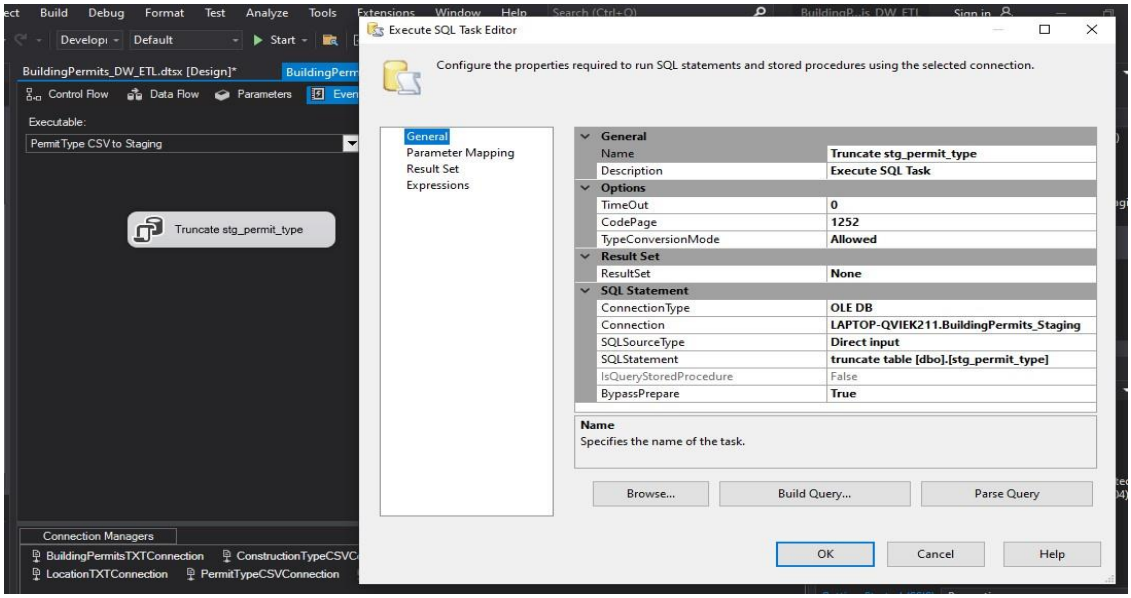
In the PermitType CSV to staging part I created its' data flow task configuring properties used to connect to and obtain data from a text file.





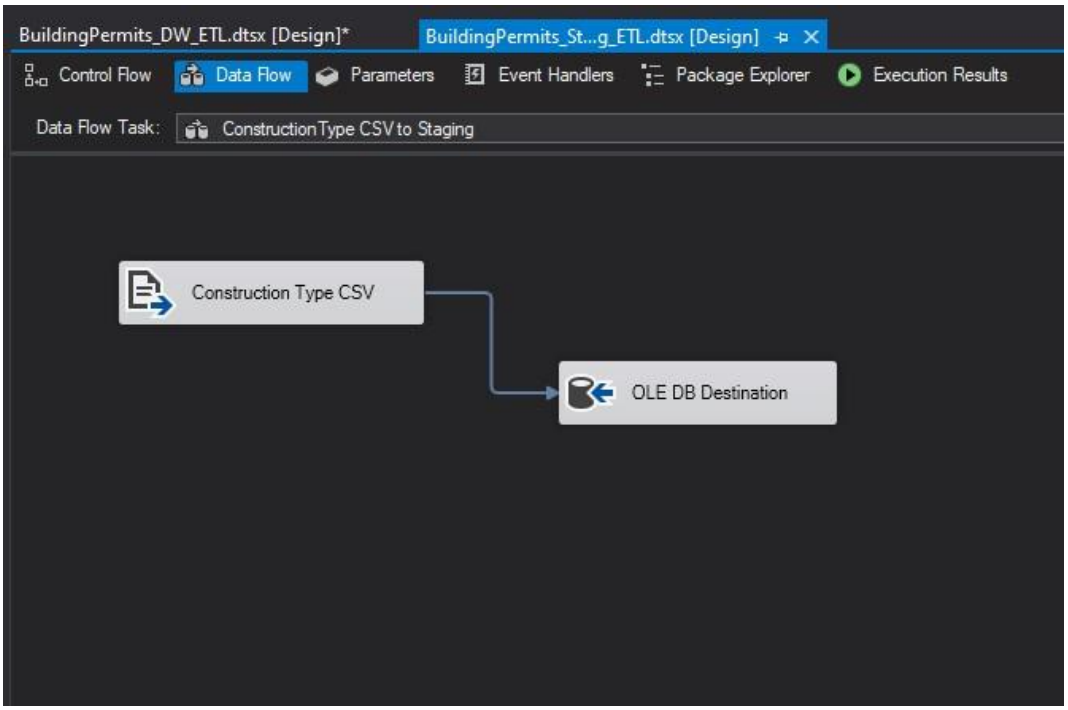
Using Event hanlder for first stage

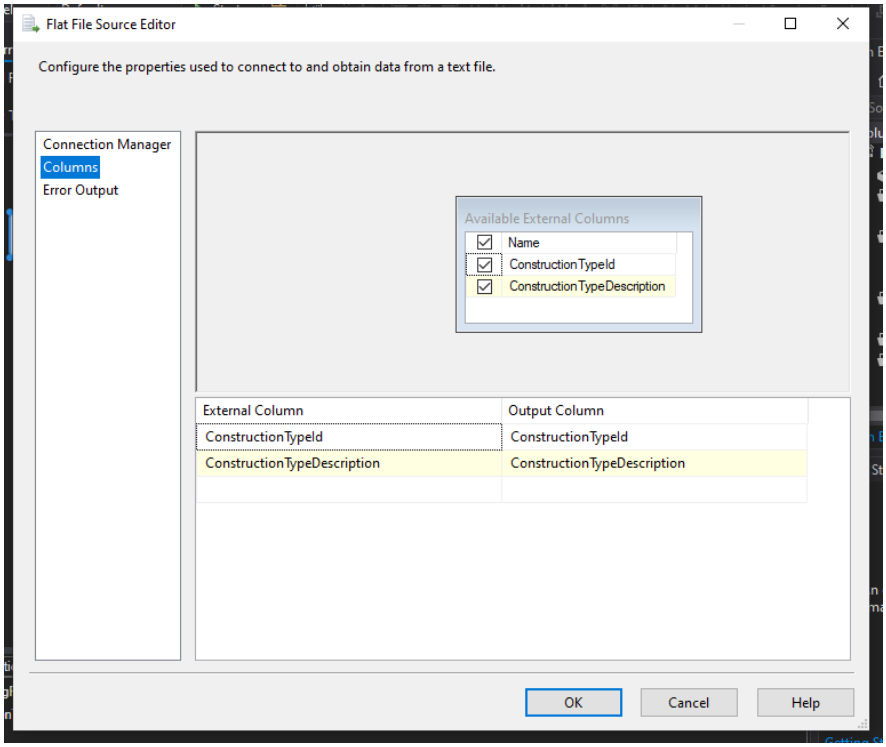
Afterthat I created an event handler For it .I truncated stg_permit_type



Second stage - ConstructionType CSV to staging

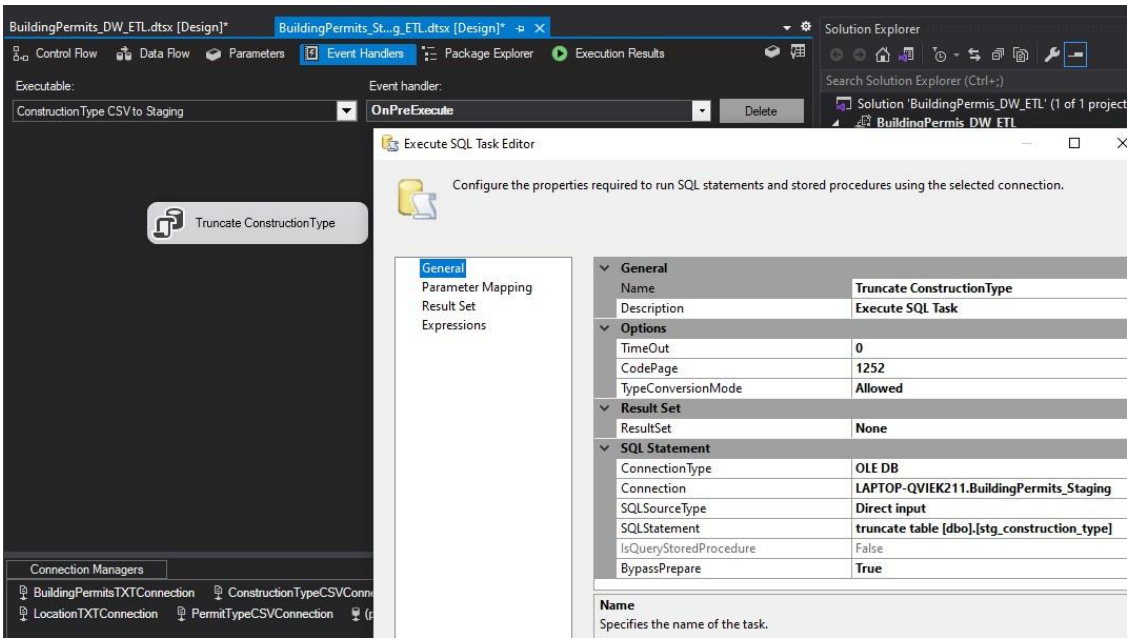
After the PermitType CSV staging I created the ConstructionType to staging, configuring properties used to connect to and obtain data from a text file.





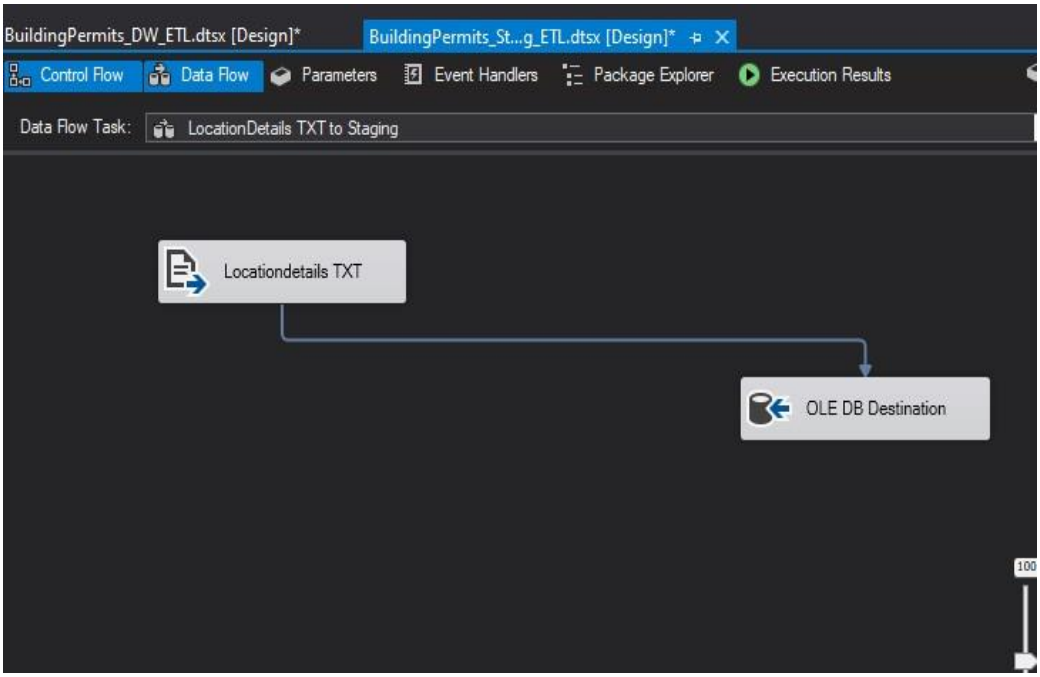
Using Event hanlder for the staging of ConstructionType

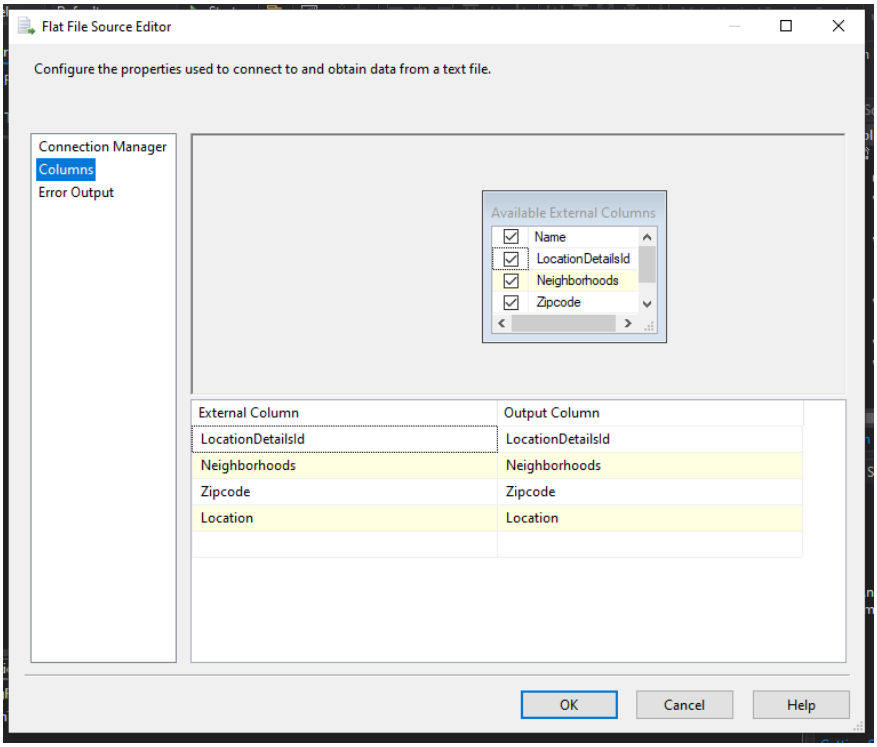
After creating Construction type staging I used an event handler For it .I truncated Construction type .



Third Stage – LocationDetails txt to staging

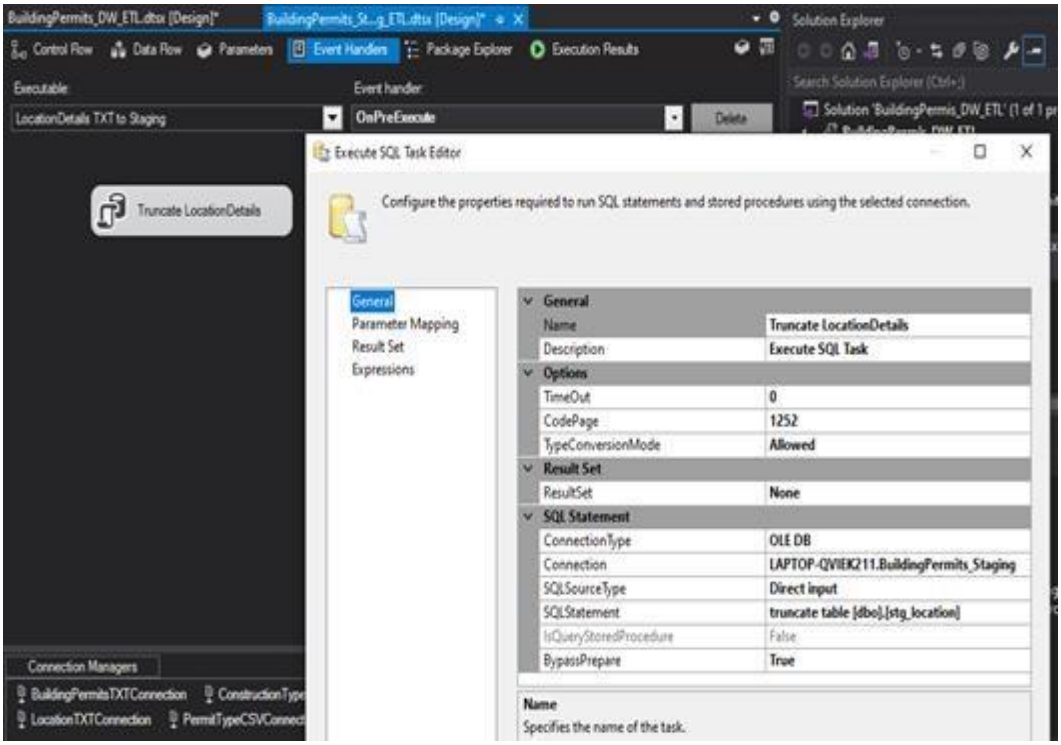
Then I created LocaionDetails txt to Staging, configuring properties used to connect to and obtain data from a text file.





Using Event hanlder for the staging of LocationDetails txt

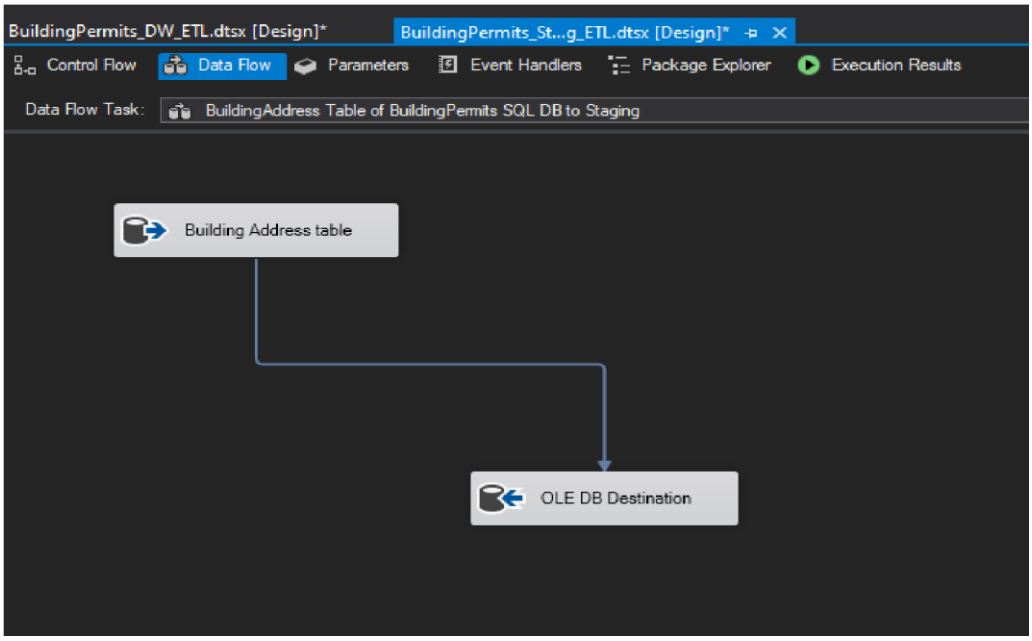
I created an event hanldler for the locationDetails staging .I truncated the location Details.

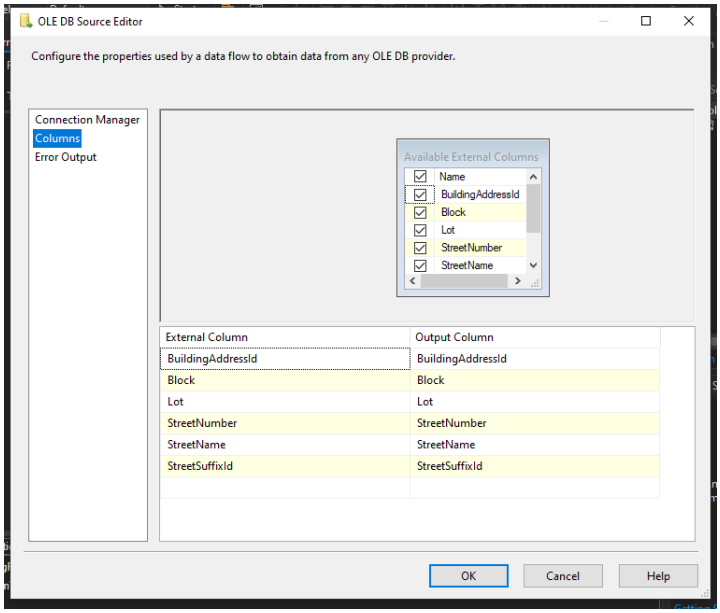


Fourth Stage - BuildingAddress Table of BuildingPermits SQL DB to Staging

In the data flow

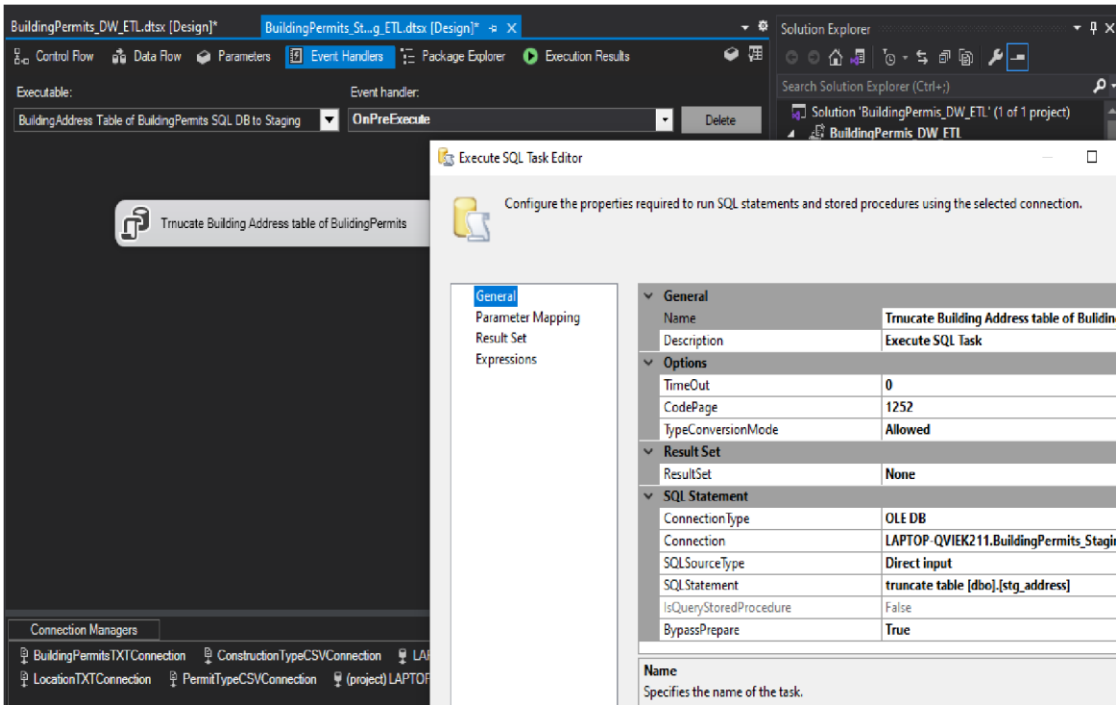
I created the data flow task of the Building address table to staging configuring properties used to connect to and obtain data from a text file.





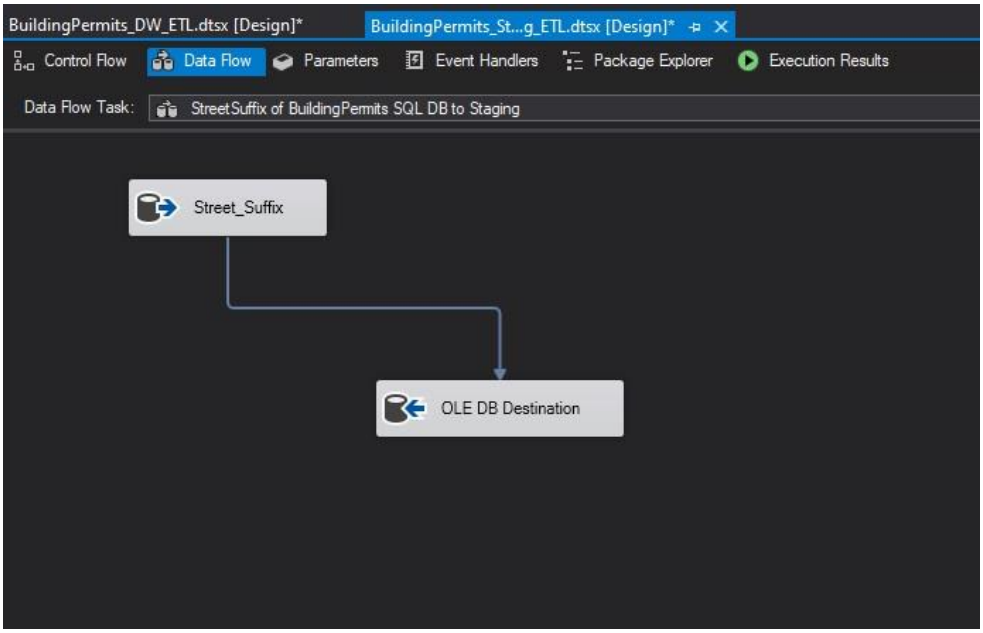
Using Event hanlder for the staging of BuildingAddress table of BuildingPermits SQL DB

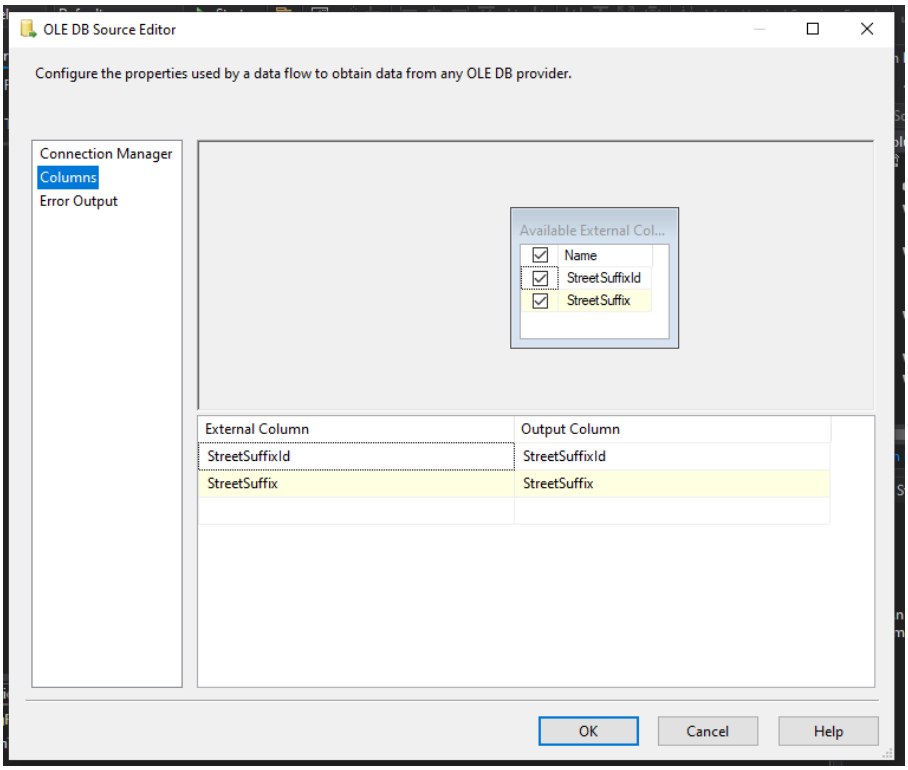
After creating of Staging of location Details ,I truncate the Building Address table of BuildingPermits SQL DB



Fifthe Stage - StreetSuffix of BuildingPermits SQL DB to Staging

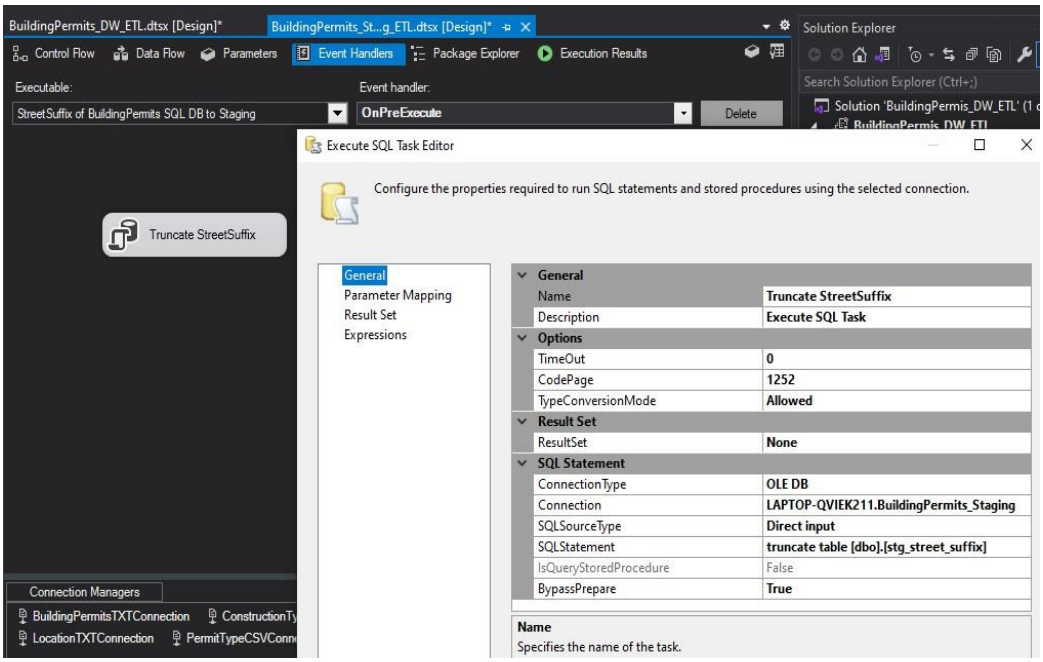
After the Staging of location Details I created streetSuffix of buldingPermits SQL DB staging.





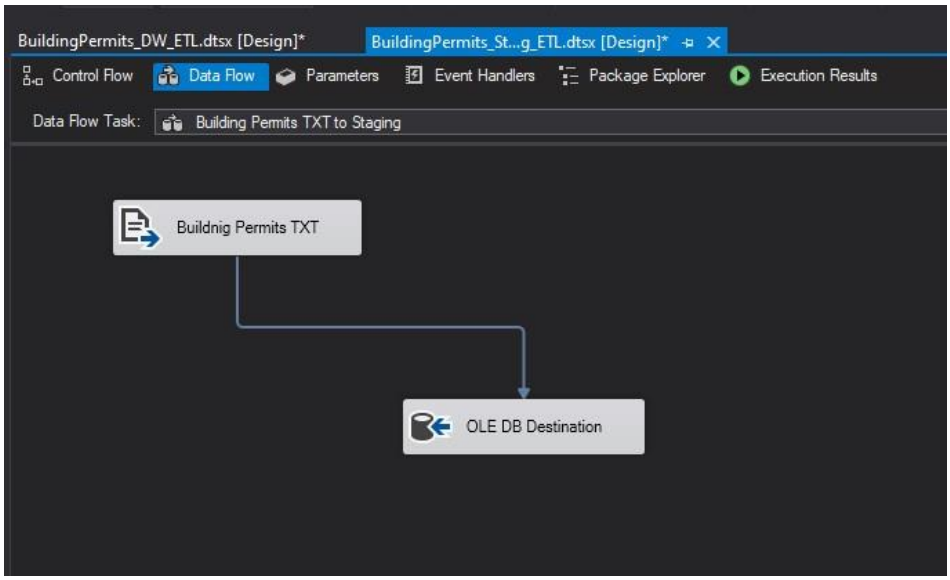
Using Event hanlder for the staging of Streetsuffix of BuildingPermits SQL DB Then

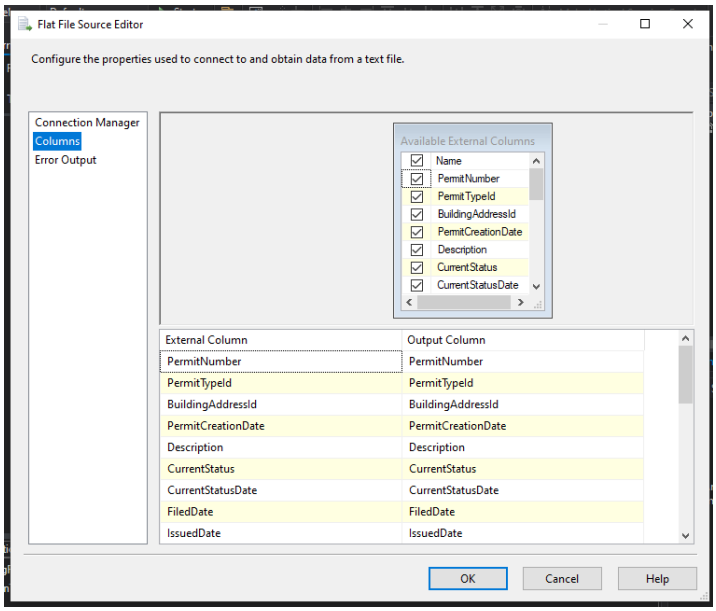
I truncated the StreetSuffix of BuildingPermits SQL DB



Sixth Stage - Building Permits TXT to Staging

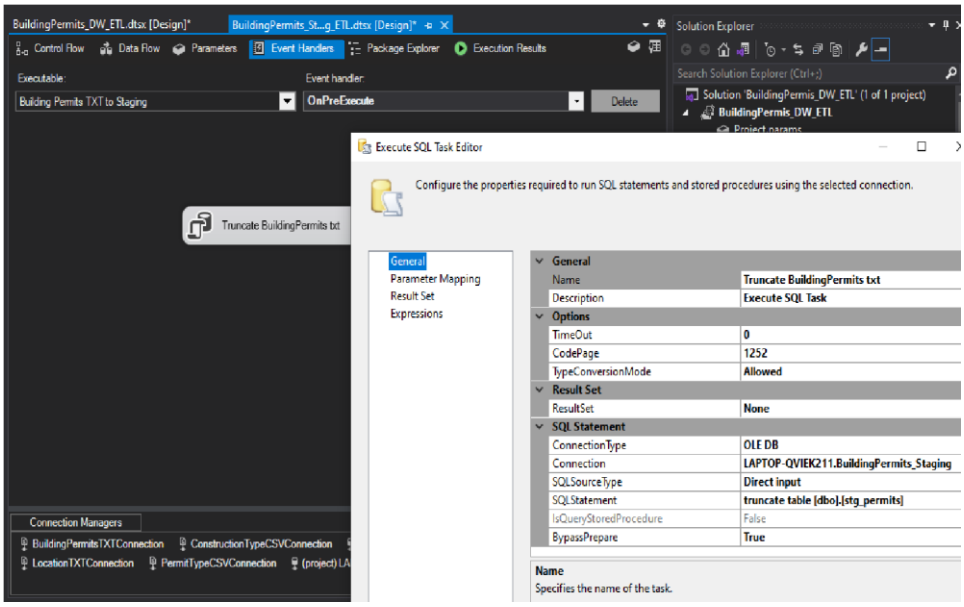
After staging of street suffix I designed the Building Permits TXT staging.





Using Event hanlder for the staging of Building Permits TXT

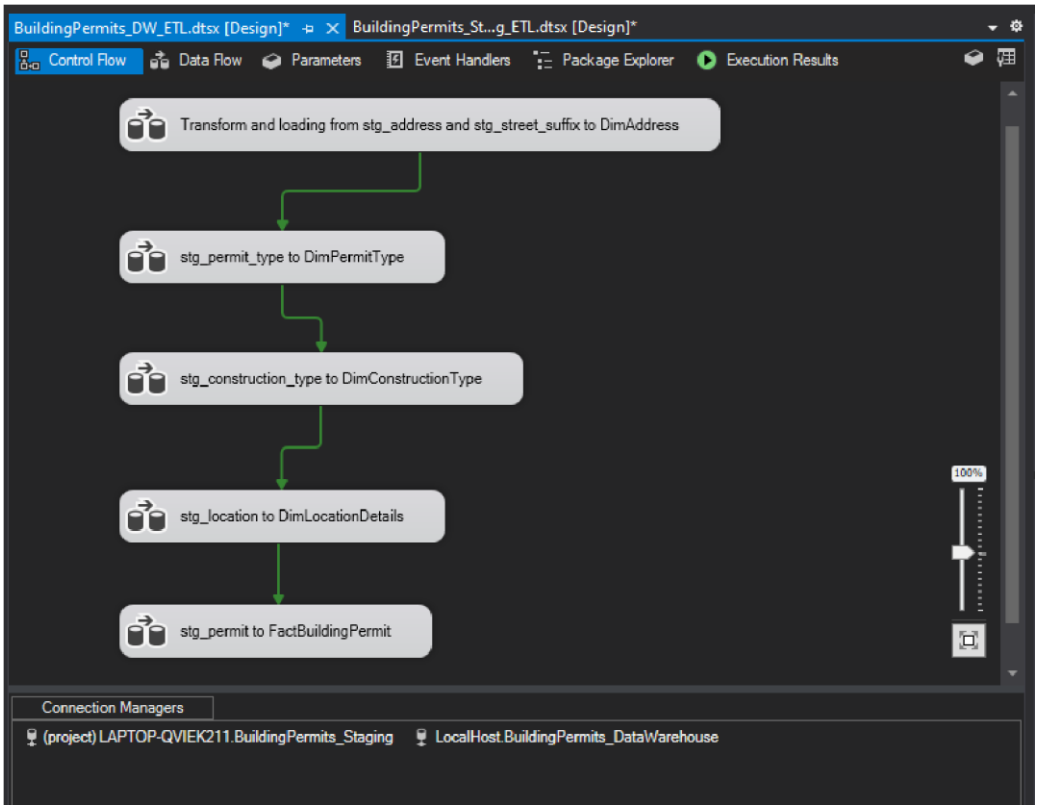
After creating the BuildingPermits staging I truncated the Building Permits



After creating staging step I create task to execute the package.

BuildingPermits_DWH_ETL.dtsx

There are two connections in the BuildingPermits_DWH_ETL package .Because I created this package for staging to warehouse.In here I designed an another controlflow for staging to datawarehouse



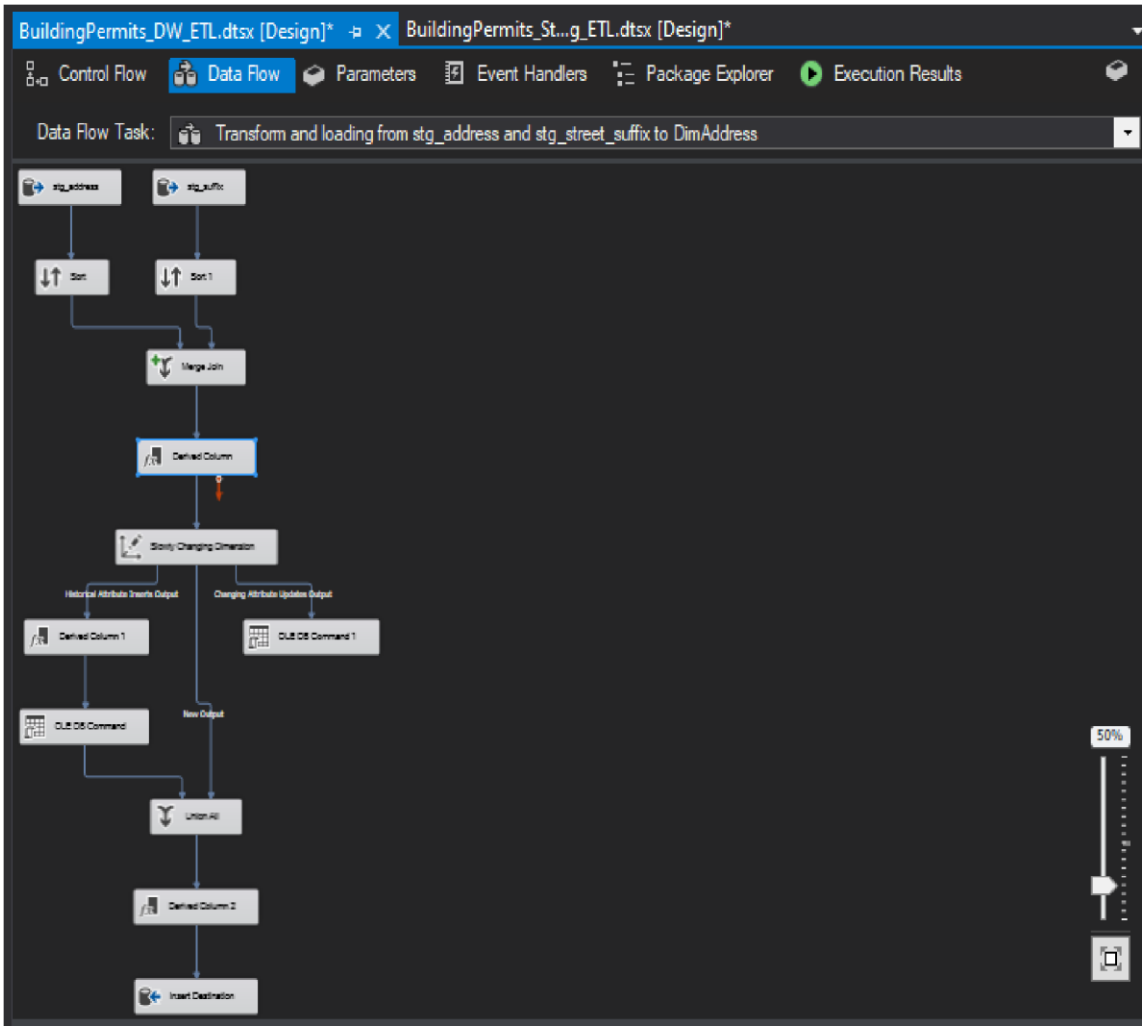
1.Transform and loading from stg_address and stg_street_suffix to DimBuildingAddress

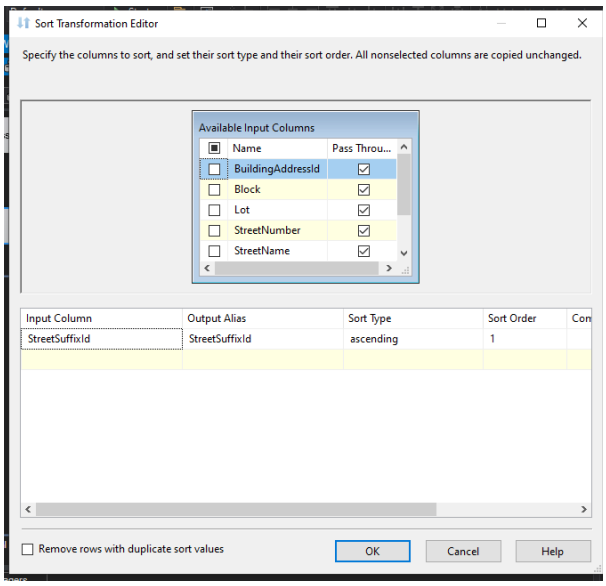
In this step I created task to transform and loading staging address and street suffix to DimBuildingAddress.

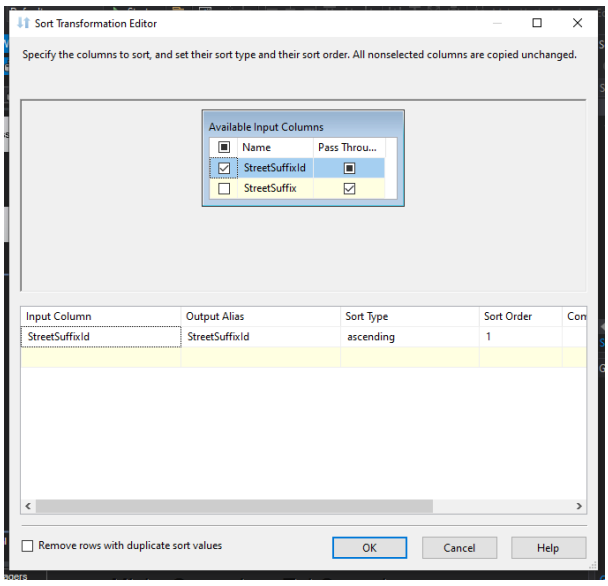
In this Transform and Loading process I have to get stg_address and stg_street_suffix tables to load the DImAddress table.So I take two **sort components** using StreetSuffixId and **merge** them(stg_address table and stg_street_suffix table) and derived columns using **derived column component.**

Then I added **slowly changing dimension component** to update the DimBuildingAddress table records.

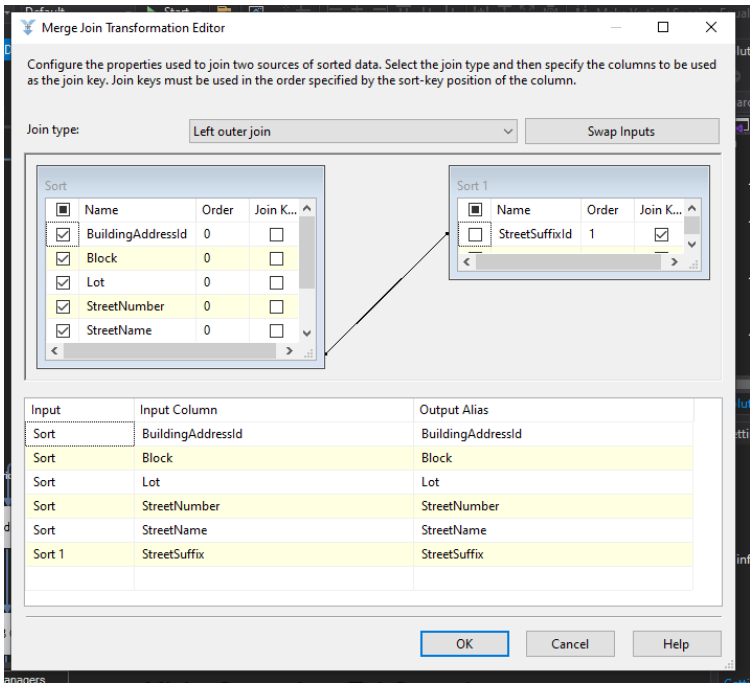
So as shown below I have transformed and loaded data


BuildingPermits_DW_ETL.dtsx [Design]*
BuildingPermits_St...g_ETL.dtsx [Design]*
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Execution Results
Data Flow Task:
Transform and loading from stg_address and stg_street_suffix to DimAddress
stg_address
stg_suffix
Sort
Sort 1
Merge Join
Derived Column
Slowly Changing Dimension
Historical Attribute Inserts Output
Changing Attribute Updates Output
Derived Column 1
OLE DB Command 1
OLE DB Command
New Output
Union All
Derived Column 2
Insert Destination
50%


Sort Transformation Editor
Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.
Available Input Columns
Name
Pass Throu...
BuildingAddressId
Block
Lot
StreetNumber
StreetName
Input Column
Output Alias
Sort Type
Sort Order
Con
StreetSuffixId
StreetSuffixId
ascending
1
Remove rows with duplicate sort values
OK
Cancel
Help

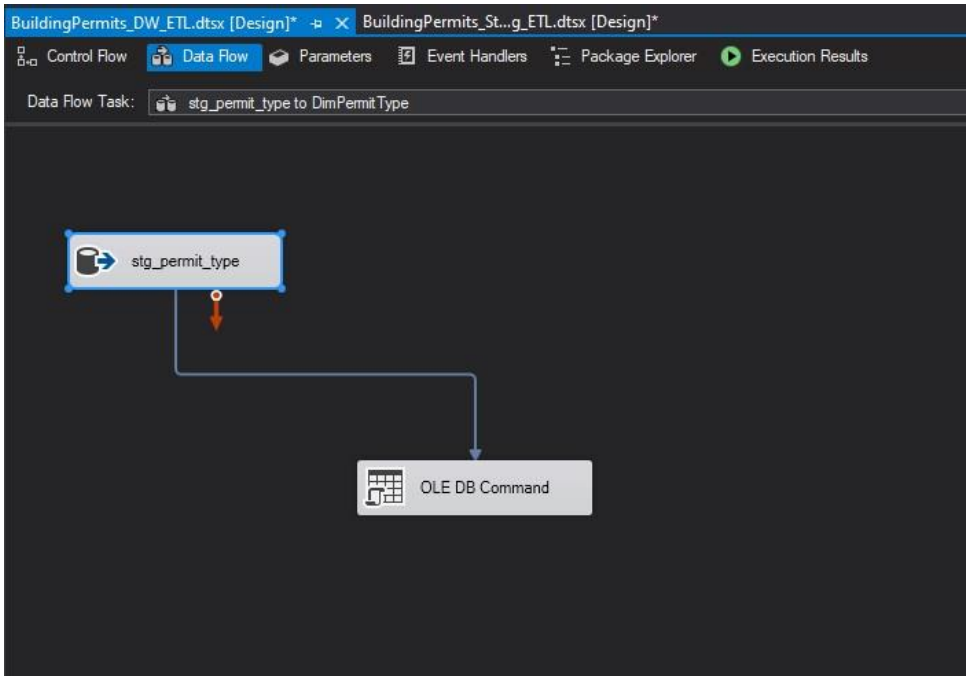

Sort Transformation Editor
Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.
Available Input Columns
Name
Pass Throu...
StreetSuffixId
StreetSuffix
Input Column
Output Alias
Sort Type
Sort Order
Con
StreetSuffixId
StreetSuffixId
ascending
1
Remove rows with duplicate sort values
OK
Cancel
Help

As shown belown I used merge join component to merge them.



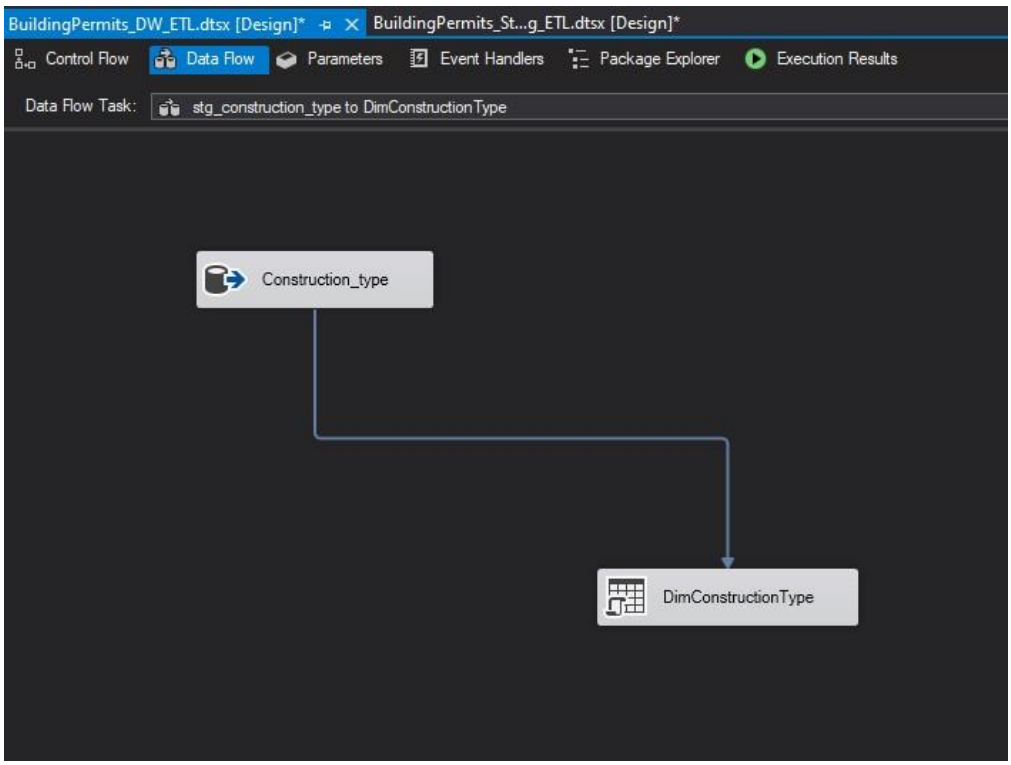
2.stg_permit_type to DimPermitType

I created the data flow task for stg_permit_type to DimPermitType



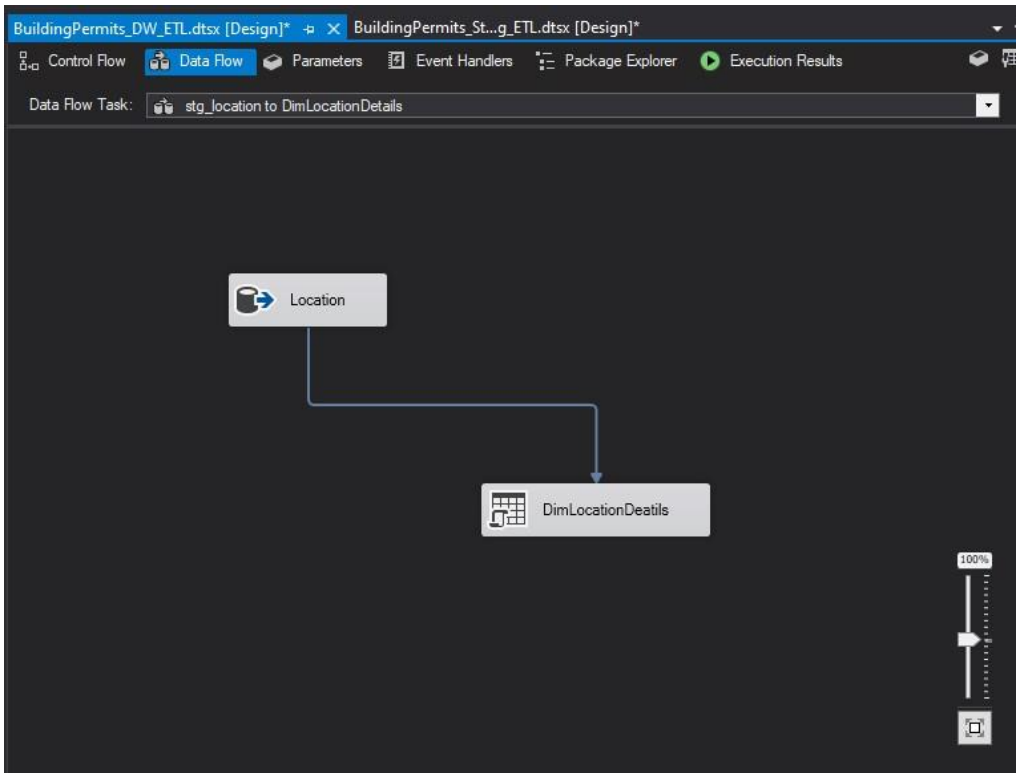
3. stg_construction_type to DimConstructionType

After creating the data flow task for stg_permit_type to DimPermitType ,I designed this task.



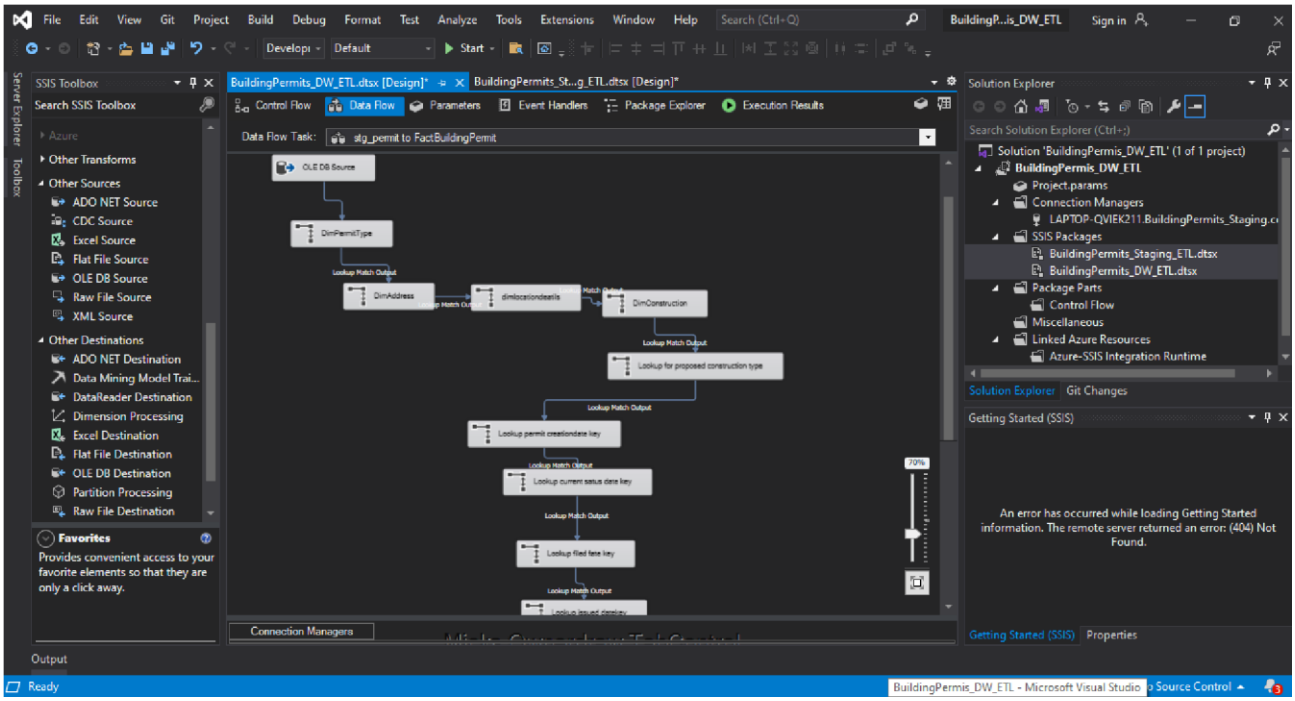
4. stg_location to DimLocationDetails

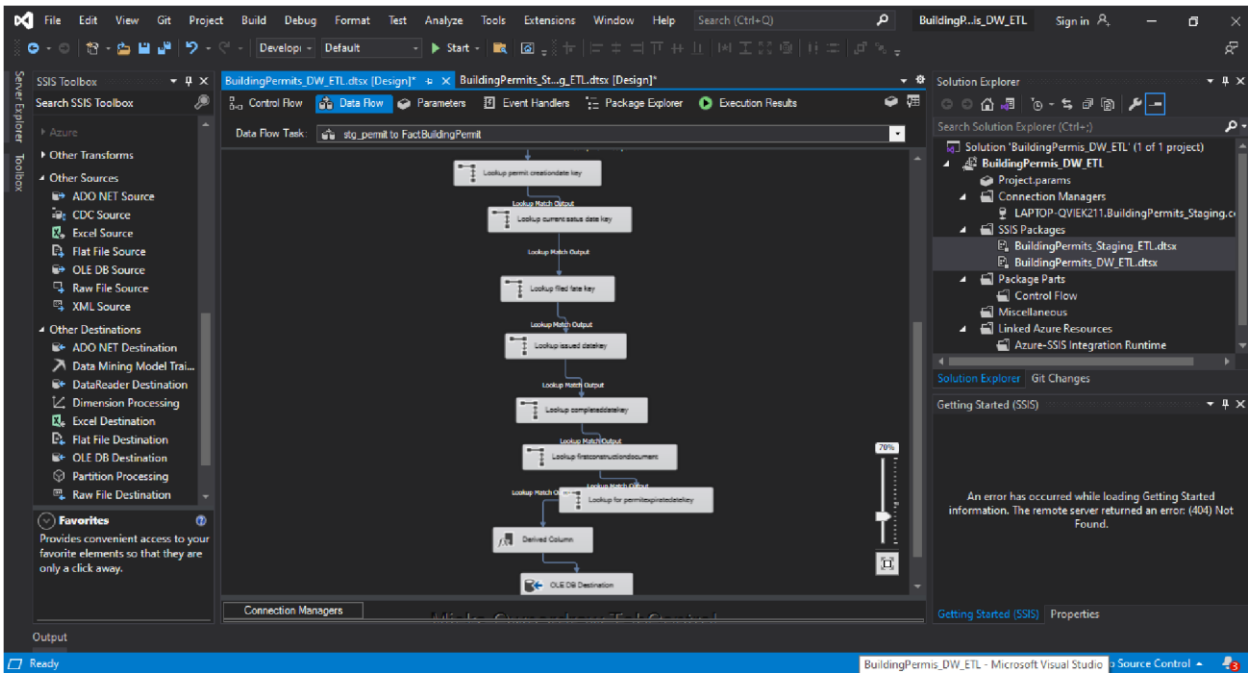
I created staging from stg_location to DimLocationDetails



5. stg_permit to FactBuildingPermit

I created staging from stg_permit to FactBuildingPermit





Stored Procedures

I have created three stored procedures in datawarehouse database as,

- UpdateDimConstructionType
- UpdateDimLocationDetailsId
- UpdateDimPermitType

UpdateDimConstructionType

```
CREATE PROCEDURE dbo.UpdateDimConstructionType

@ConstructionTypeId nvarchar(8),

@ConstructionTypeDescription nvarchar(50) AS  BEGIN

   if not exists (select ConstructionTypeSK from dbo.dimConstructionType where
AlternateConstructionTypeId = @ConstructionTypeId)      BEGIN

      insert into dbo.dimConstructionType (AlternateConstructionTypeId, ConstructionTypeDescription,
InsertDate, ModifiedDate) values (@ConstructionTypeId, @ConstructionTypeDescription, GETDATE(),
GETDATE())

   END;

   if exists (select ConstructionTypeSK from dbo.dimConstructionType where
AlternateConstructionTypeId = @ConstructionTypeId)      BEGIN

      update dbo.dimConstructionType set ConstructionTypeDescription =
@ConstructionTypeDescription, ModifiedDate = GETDATE() where AlternateConstructionTypeId =
@ConstructionTypeId

   END;

END;


EXEC dbo.UpdateDimConstructionType ?, ?
```

UpdateDimLocationDetailsId

```
CREATE PROCEDURE dbo.UpdateDimLocationDetailsId

@LocationDetailsId nvarchar(8),

@Neighborhoods nvarchar(50),

@Zipcode numeric,

@Location nvarchar(50) AS  BEGIN

   if not exists (select LocationDetailsSK from dbo.dimLocationDetailsId where
AlternateLocationDetailsId = @LocationDetailsId)      BEGIN

      insert into dbo.dimLocationDetailsId (AlternateLocationDetailsId,
Neighborhoods,Zipcode,Location , InsertDate, ModifiedDate) values
(@LocationDetailsId,@Neighborhoods,@Zipcode,@Location, GETDATE(), GETDATE())

   END;

   if exists (select LocationDetailsSK from dbo.dimLocationDetailsId where
AlternateLocationDetailsId = @LocationDetailsId)      BEGIN

      update dbo.dimLocationDetailsId set Neighborhoods =
@Neighborhoods,Zipcode=@Zipcode ,Location=@Location, ModifiedDate = GETDATE() where
```

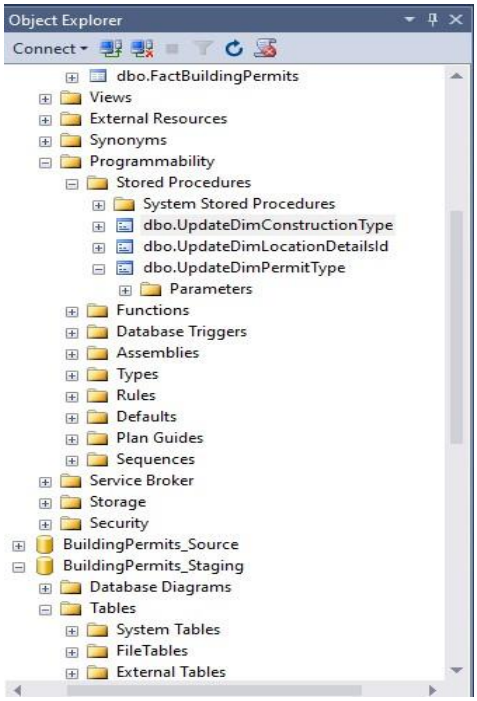
```
AlternateLocationDetailsId = @LocationDetailsId
   END;
END;
```

```
EXEC dbo.UpdateDimLocationDetailsId ?, ?, ?, ?
```

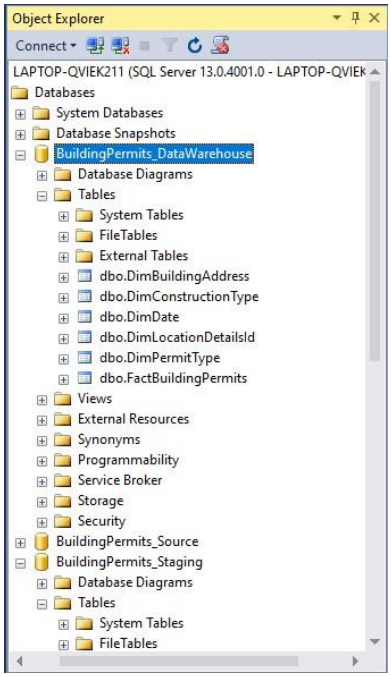
UpdateDimPermitType

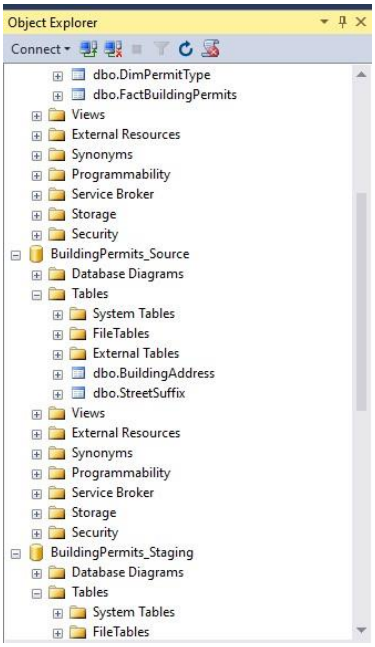
```
CREATE PROCEDURE dbo.UpdateDimPermitType
@PermitTypeId nvarchar(8),
@PermitTypeDefinition nvarchar(50) AS  BEGIN
   if not exists (select PermitTypeSK from dbo.dimPermitType where AlternatePermitTypeId = @PermitTypeId)
   BEGIN
      insert into dbo.dimPermitType (AlternatePermitTypeId, PermitTypeDefinition, InsertDate, ModifiedDate) values (@PermitTypeId, @PermitTypeDefinition, GETDATE(), GETDATE())
   END;
   if exists (select PermitTypeSK from dbo.dimPermitType where AlternatePermitTypeId = @PermitTypeId)
   BEGIN
      update dbo.dimPermitType set PermitTypeDefinition = @PermitTypeDefinition, ModifiedDate = GETDATE() where AlternatePermitTypeId = @PermitTypeId
   END;
END;
```

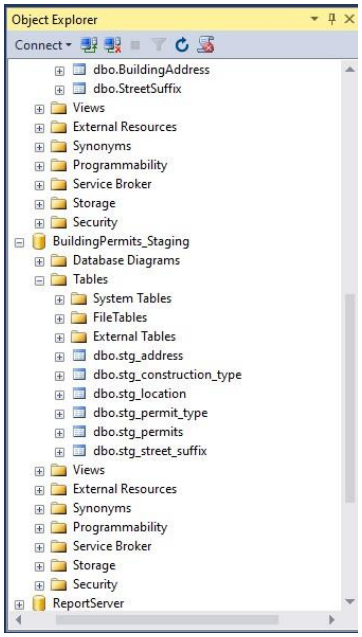
```
EXEC dbo.UpdateDimPermitType ?, ?
```



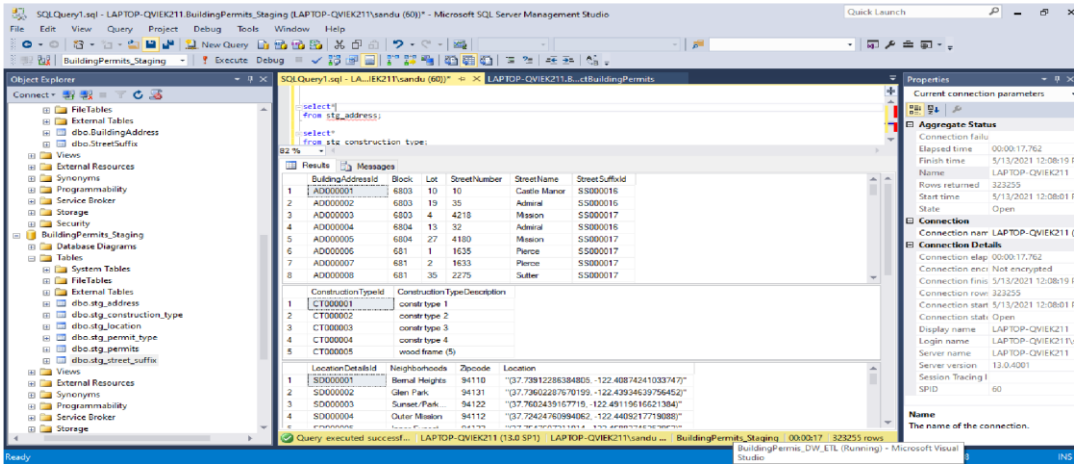
BuildingPermits_DataWarehouse BuildingPermits_Source BuildingPermits_Staging

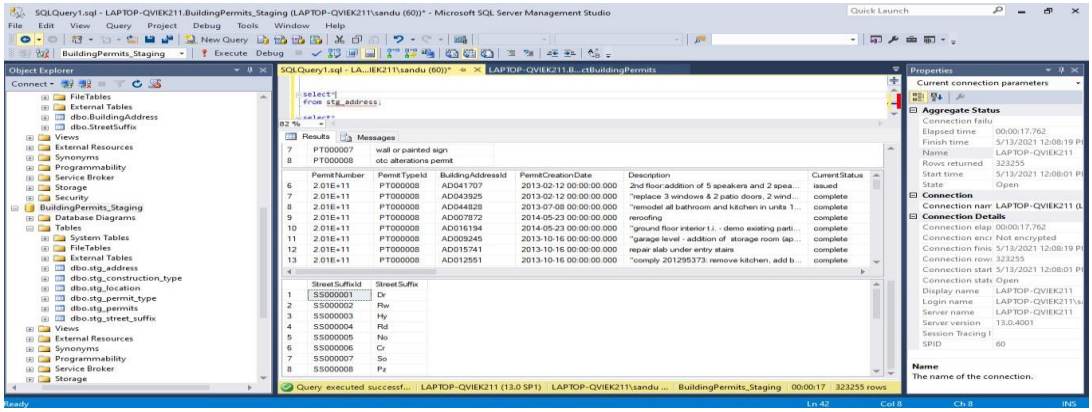


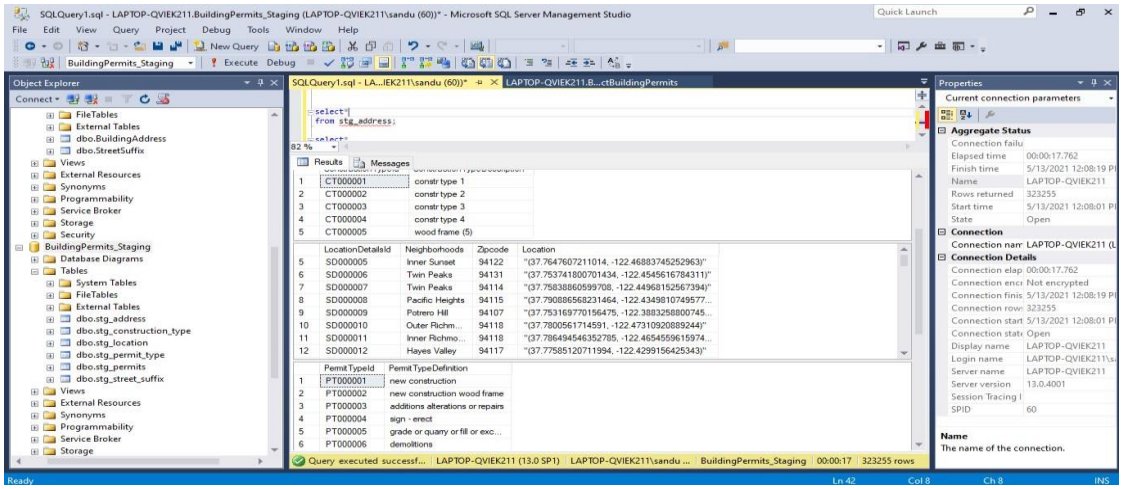




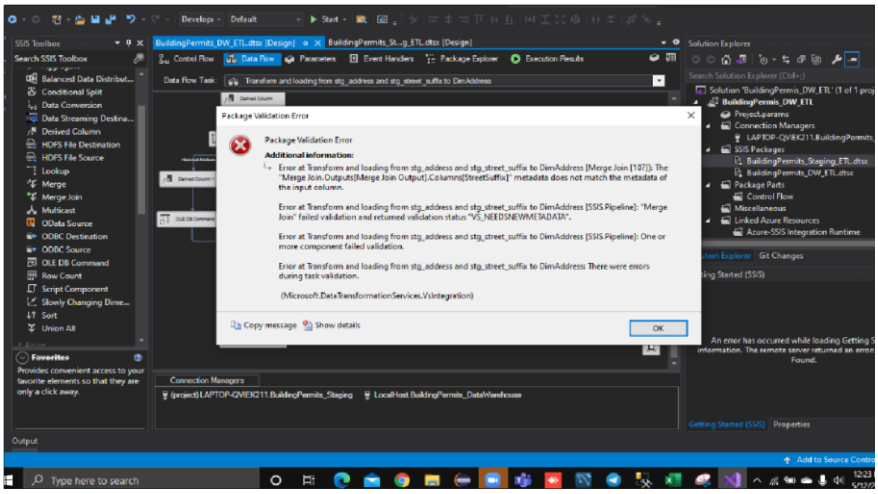
Tables of the BuildingPermits_Staging

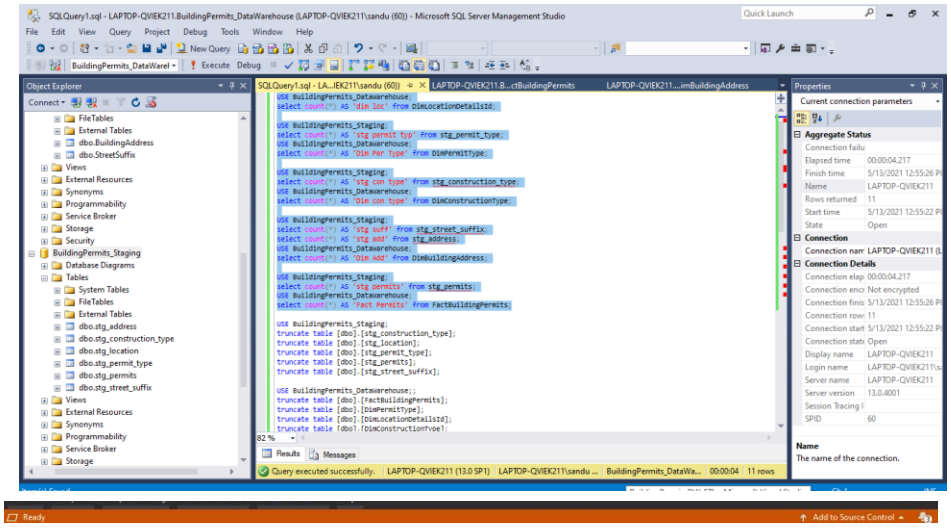


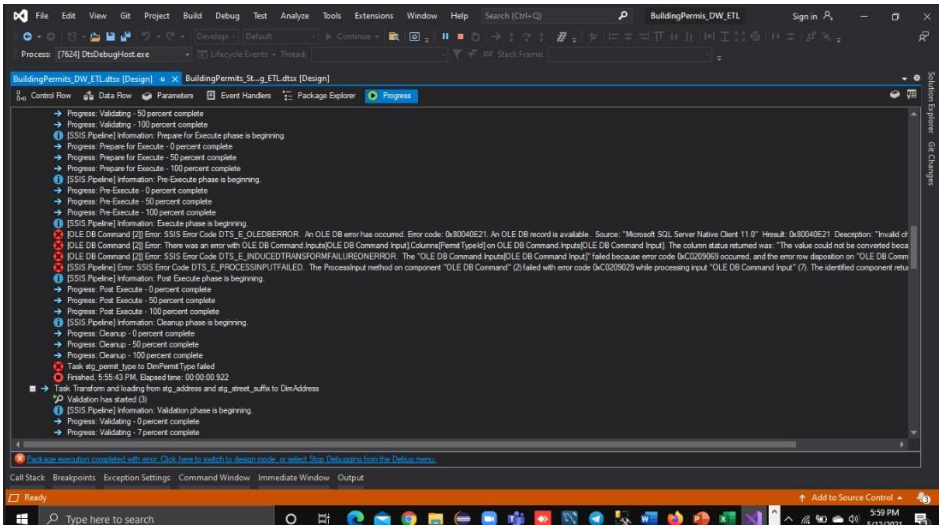




These errors occured while I was designing the datawarehouse

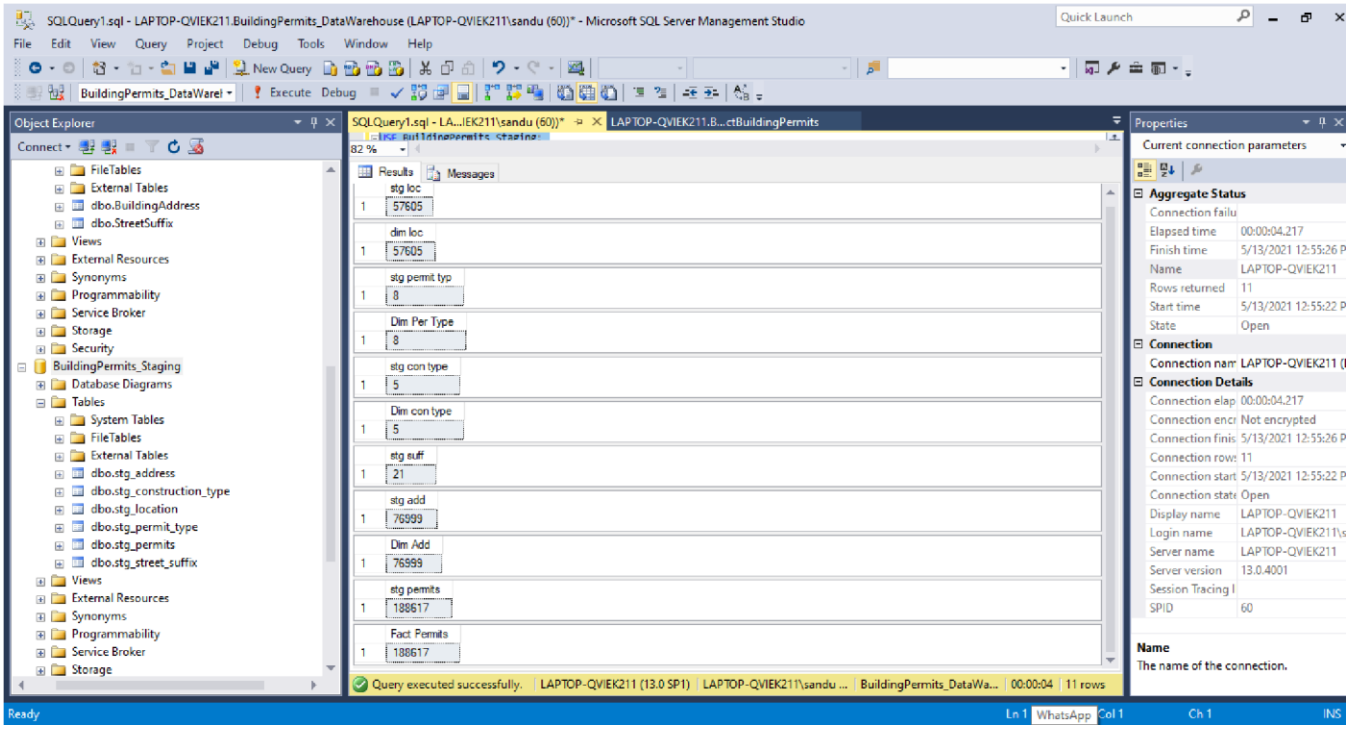




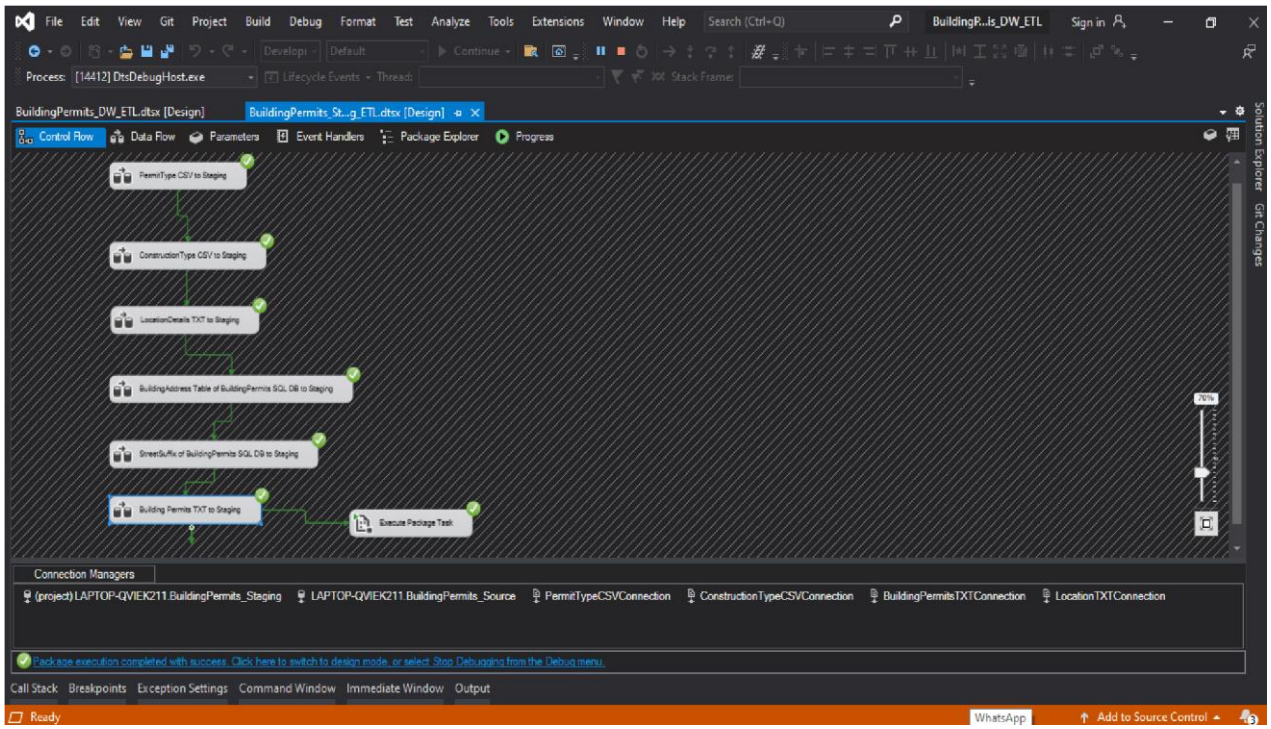


After the Excution of the full packages

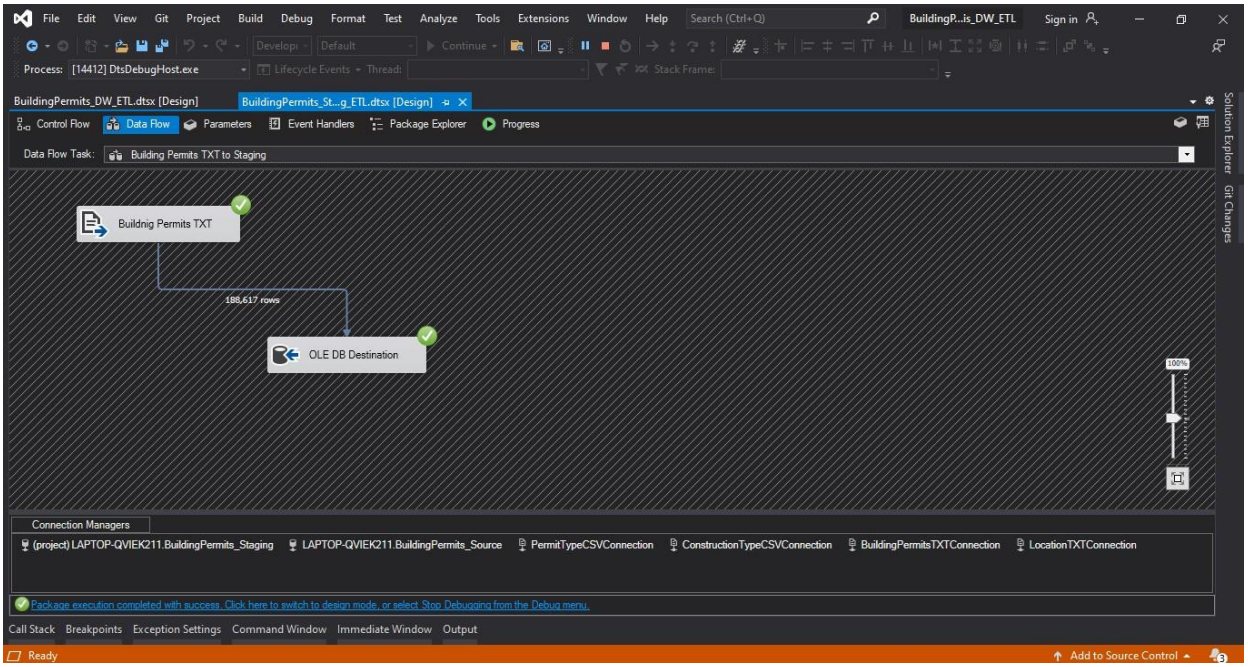
We created the datawarehouse successfully



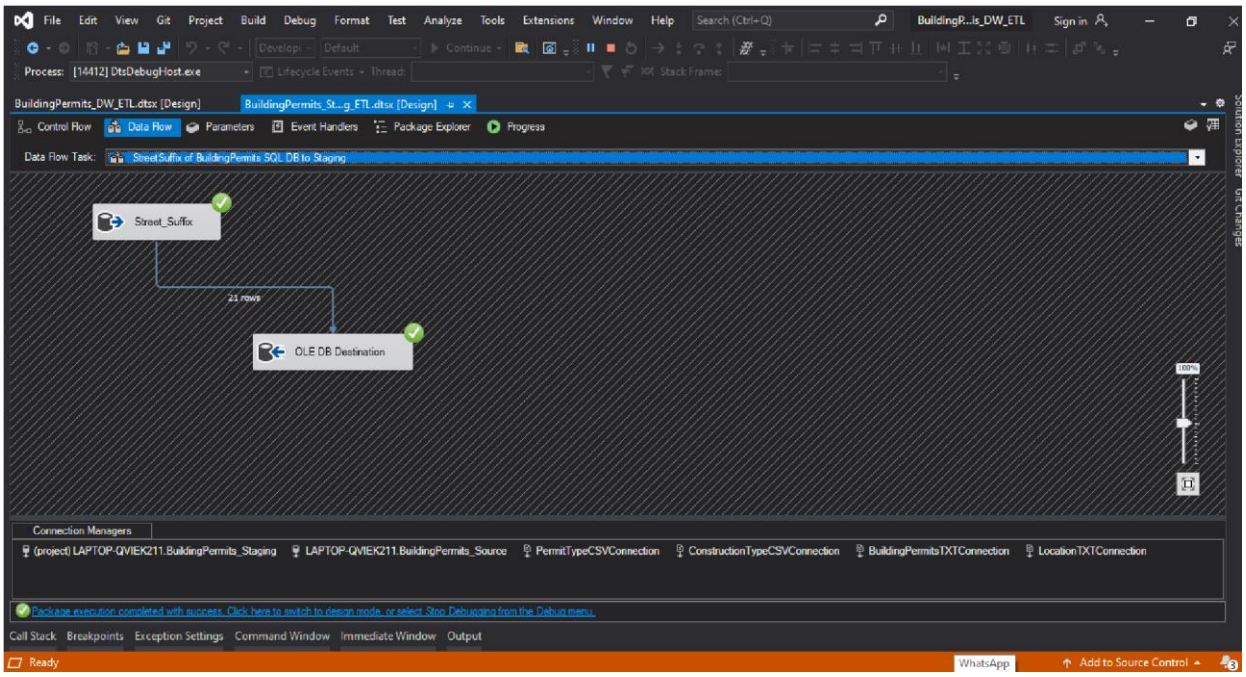
BuildingPermits_Staging_ETL.dtsx



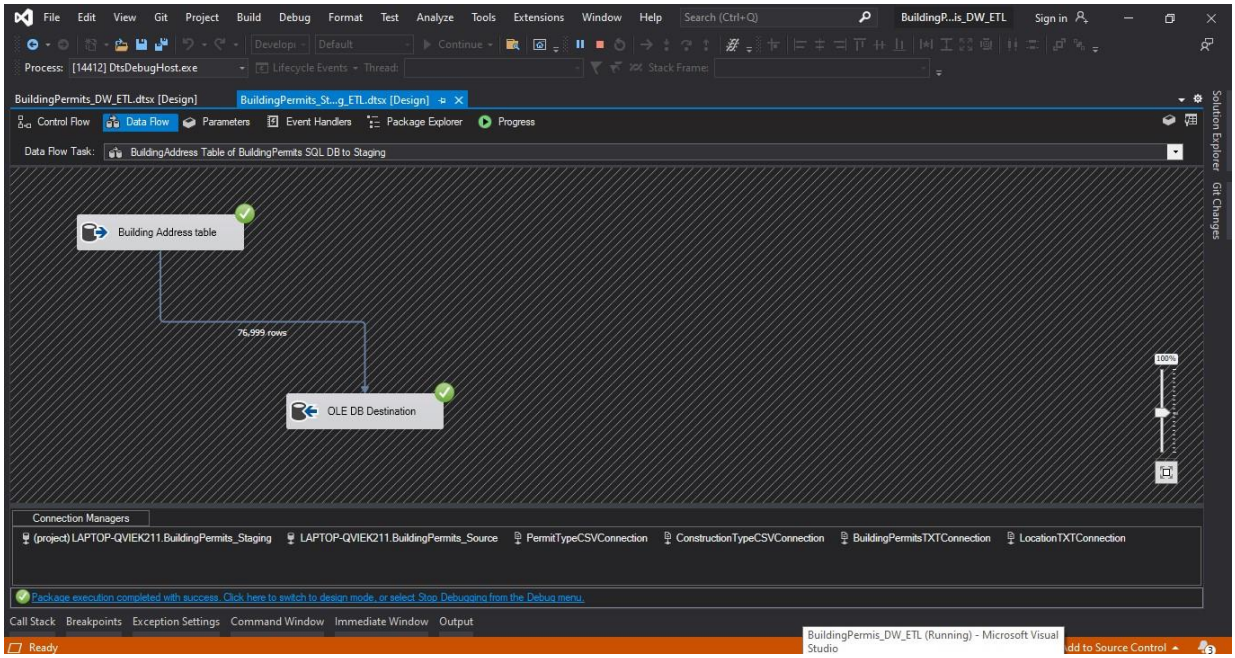
Building Permits txt to staging



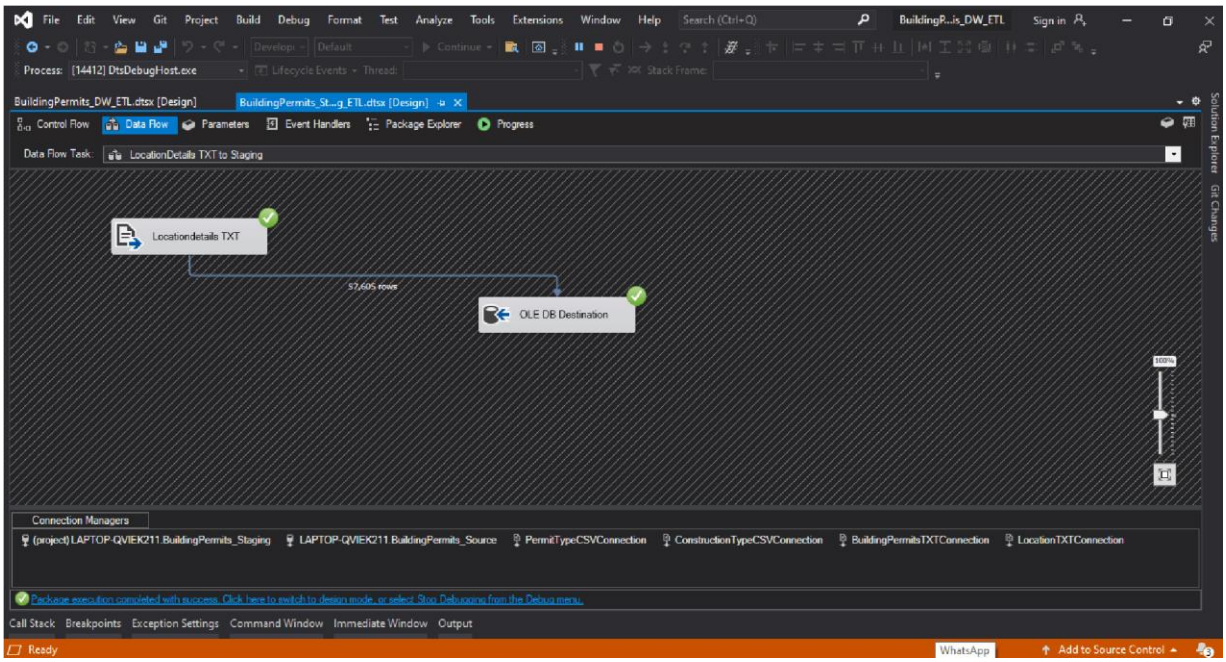
- **StreetSuffix of Building Permits SQL DB to Staging**



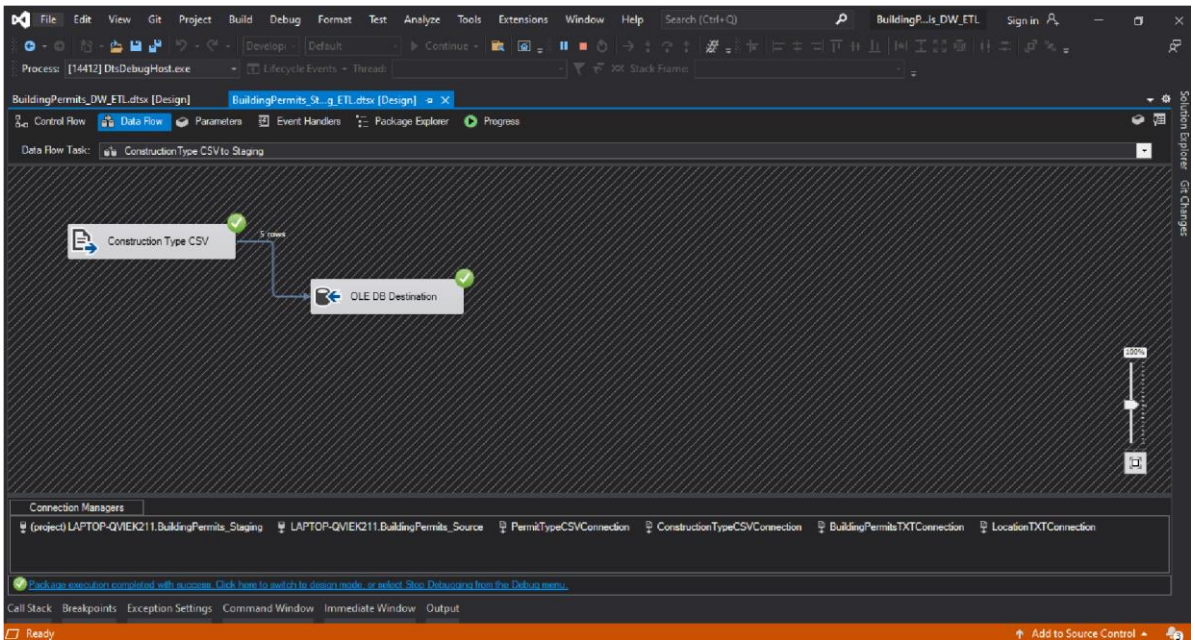
Building Address table of Building Permits SQL DB to Staging



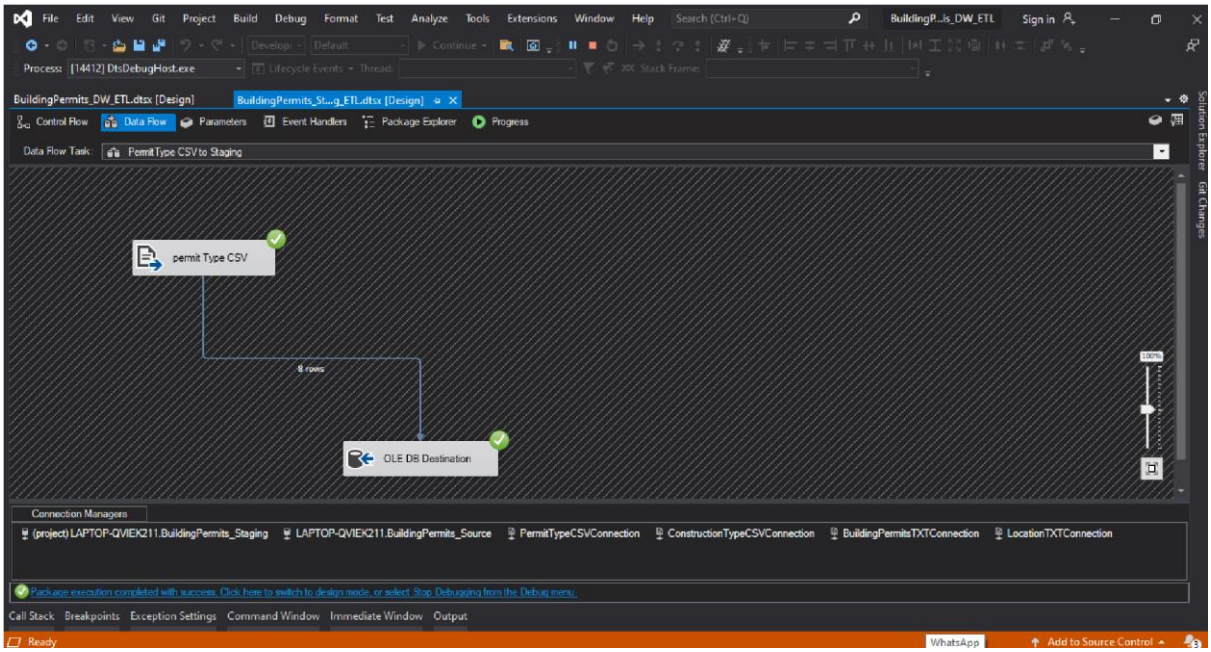
LocationFeatils txt to staging



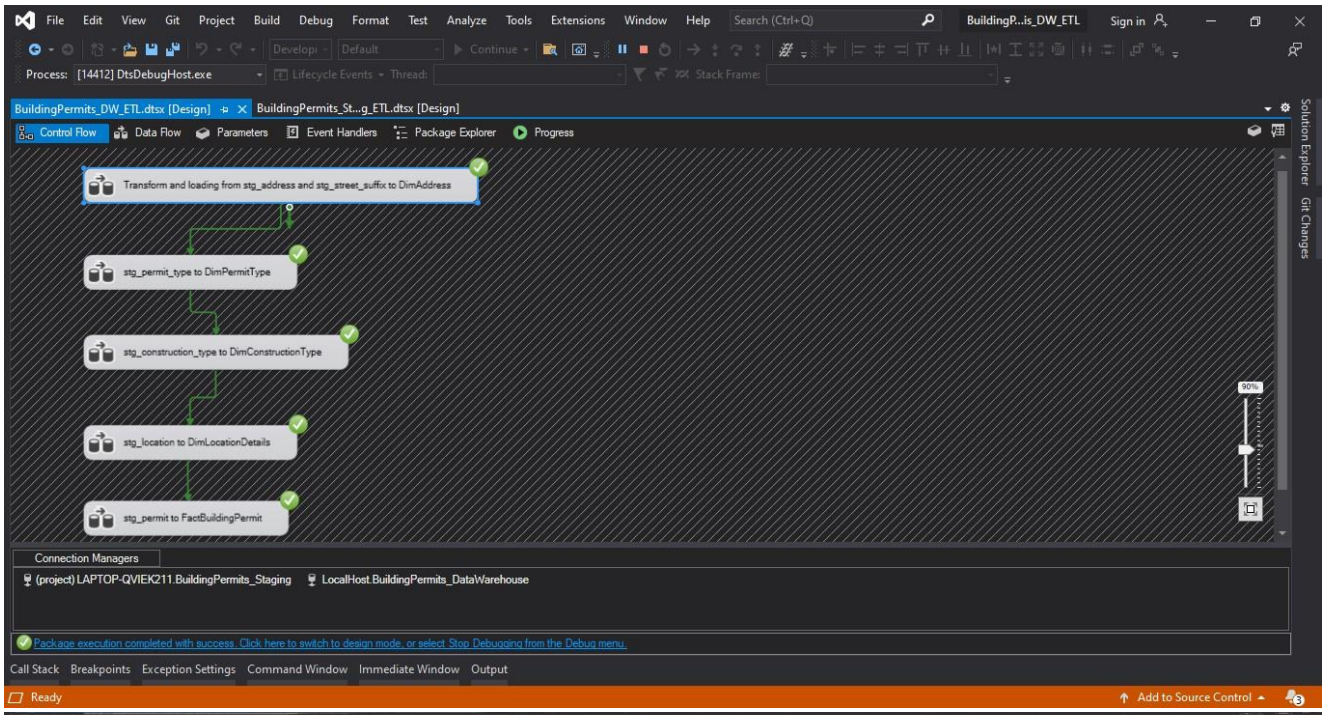
- **ConstructionType csv to staging**



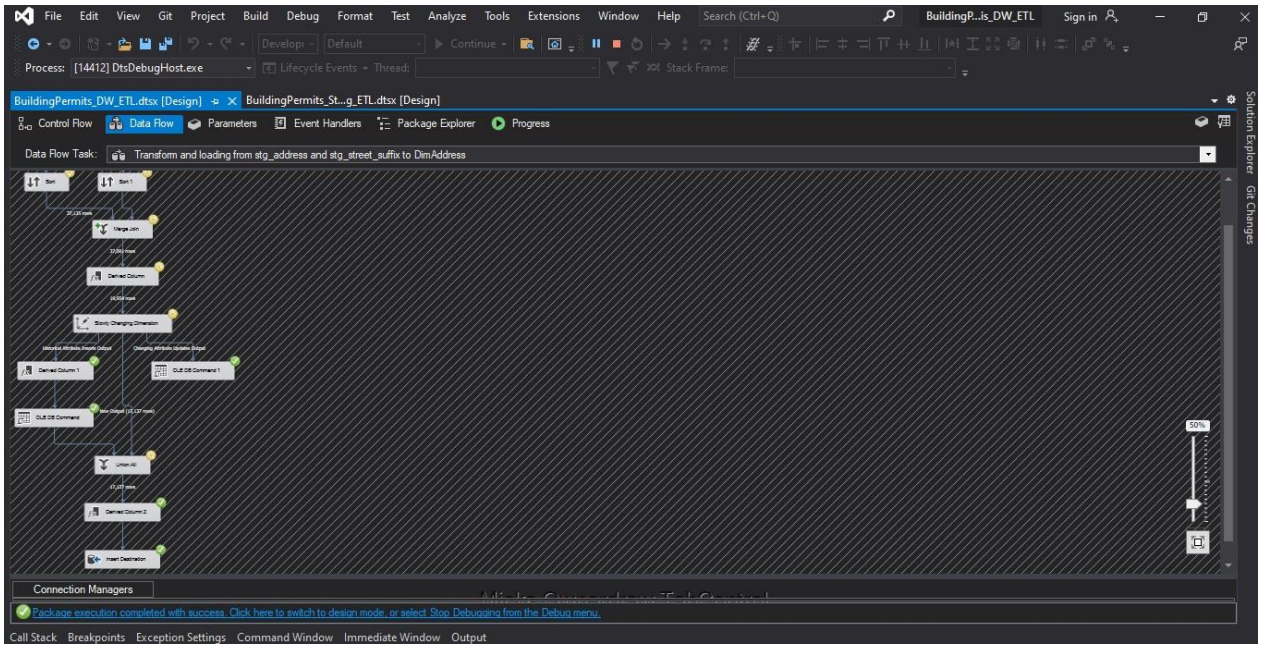
- **PermitType csv to staging**



- **BuildingPermits_DWH_ETL.dtsx**



- **Transform and loading from stg_address and stg_street_suffix to Dimaddress**



- **Stg_permit_type to DimPermit type**