Sri Lanka Institute of Information Technology

Data Warehousing and Business Intelligence

Assignment 1

2021

Name: Gavindya N.A.C

Registration No: IT20409982

Table of Contents

1. Data Set Selection 2

1.1Description 2

2. Preparation of Data Sources 3

3. Solution Architecture 4

4. Data warehouse design & development 5

5. ETL Development 5

5.1Extraction 5

5.2Transformations 8

6. ETL development – Accumulating fact tables 14

1. Data Set Selection

1.1Description

Data Set – Brooklyn Home Sales (2003-2017)

Description – This data set contains home sale prices by block in Brooklyn for the last 15 years. This data set consist of 4 tables.

Customer – Contains all the customer details

- Customer Id – Each distinct number represents one customer
- Customer name – Name of the customer who is interested in homes
- Age – Age of the customer
- Address – Permanent address of the customer

Owner – Contains all the details of the building owner

- Owner Id – Each distinct number represent one owner
- Owner name – Name of the person who owns the building
- Owner Type – Type of ownership of the building (private, public)

Building – Contains all the details of the building to be sold

- Building Id – Each distinct number represents one building
- Zip code - Five digits to identify a delivery area
- Neighborhood – Area which the building is located
- Block – Area block
- Address – Address of the building
- Tax class – Corresponding tax group which the building belongs to
- Build year – Year the building was built
- Land use – Land that is used to build the building
- Building class – Class of the building which are denoted by notations
- Building class category – Type of the building

Sales – Contains all the details of home prices

- Sales Id – Each distinct number represents one price
- Sales price – Sold price of home
- Sale date – Date which the homes were published as sold
- Land sqtft – No of square feet of the land
- Gross sqtft - No of square feet of the house

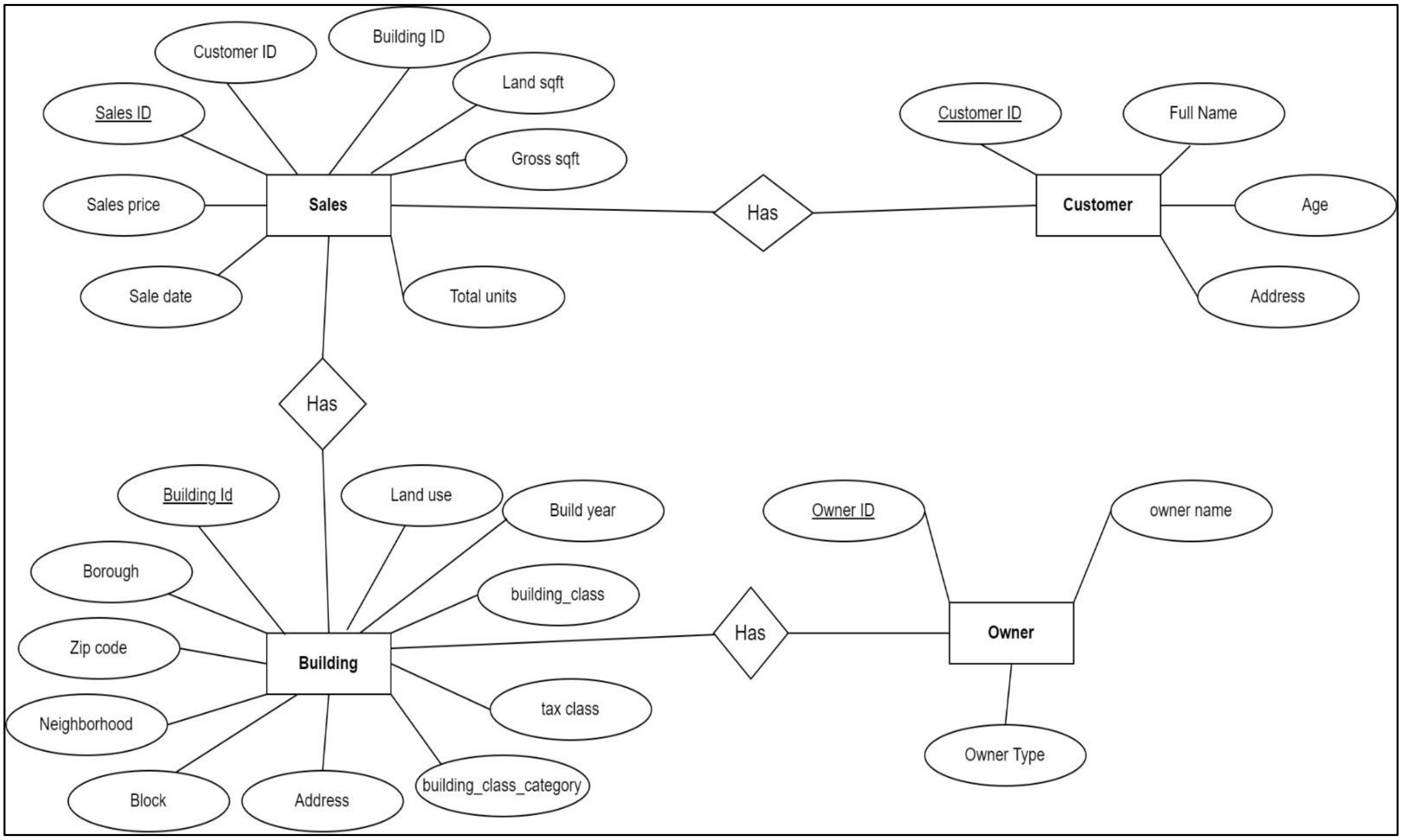


Figure 1 ER Diagram

2. Preparation of Data Sources

The initial dataset was in ‘csv’ file format, and they were separated into the following data sources, Database, Text and CSV. And they were used to create the following,

1. **Database** (.bak)

Owner table which had all the details of the building owner was converted into a database source. Owner details are OwnerID, OwnerName and OwnerType.

2. **Text** (.txt)

Customer table which has all the details of the customer who is interested in buying the building was converted into a text file

3. **CSV** (.csv)

Other two tables building and sales which had information about the building and the sales prices respectively was kept as it is.

3. Solution Architecture

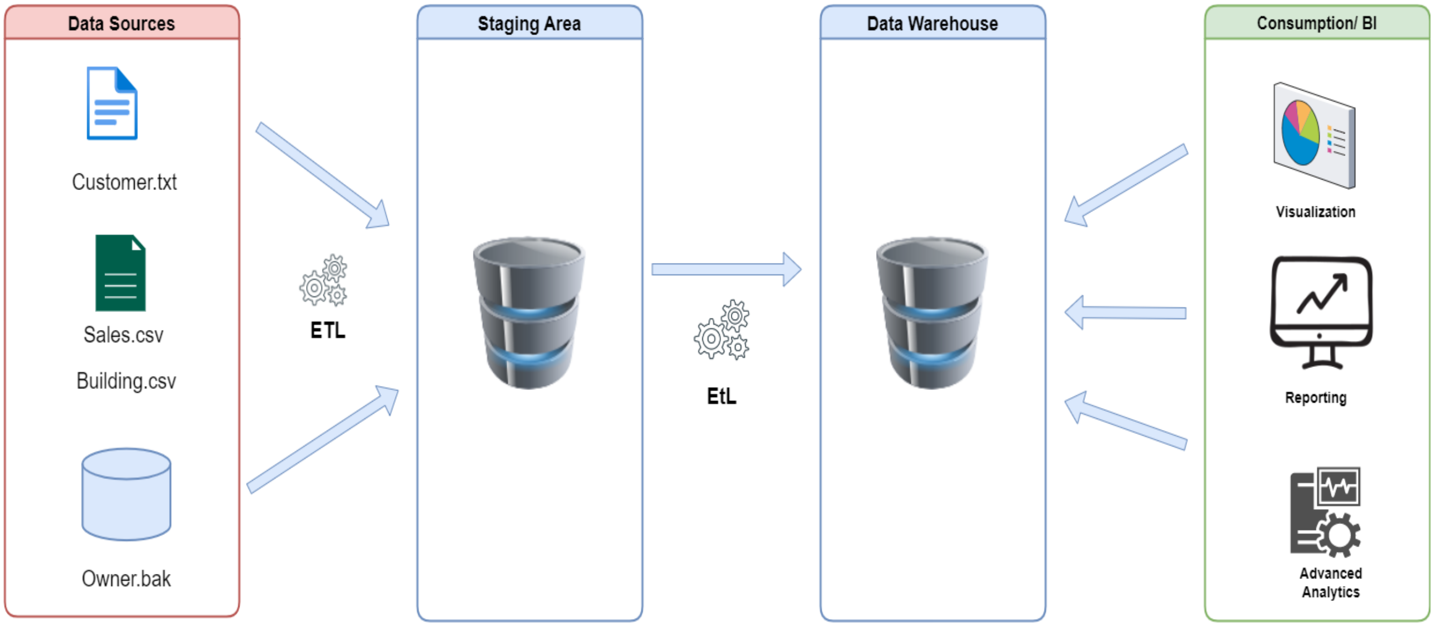


Figure 2 - 3.1 High Level Architecture

Data Source Layer
Data source also known as source System. This layer contains all the that sources we used to develop the data warehouse.
For this assignment, I got data from 3 sources (database, text file and CSV file).

Staging Layer
The Staging level consists of a database where the extracted data are stored in separate database tables.

Data Warehouse Layer
Data warehouse is one of the most important components in BI architecture. New data can be added into data warehouse regularly. But all the data stored in data warehouse are read-only. This means users are not allowed to update, over-write or delete the stored data.

Presentation Layer
This layer also known as “End User layer”. The end user layer consists of tools that display information in different formats to different users.
OLAP Cubes Screen Records/ Dashboard

4. Data warehouse design & development

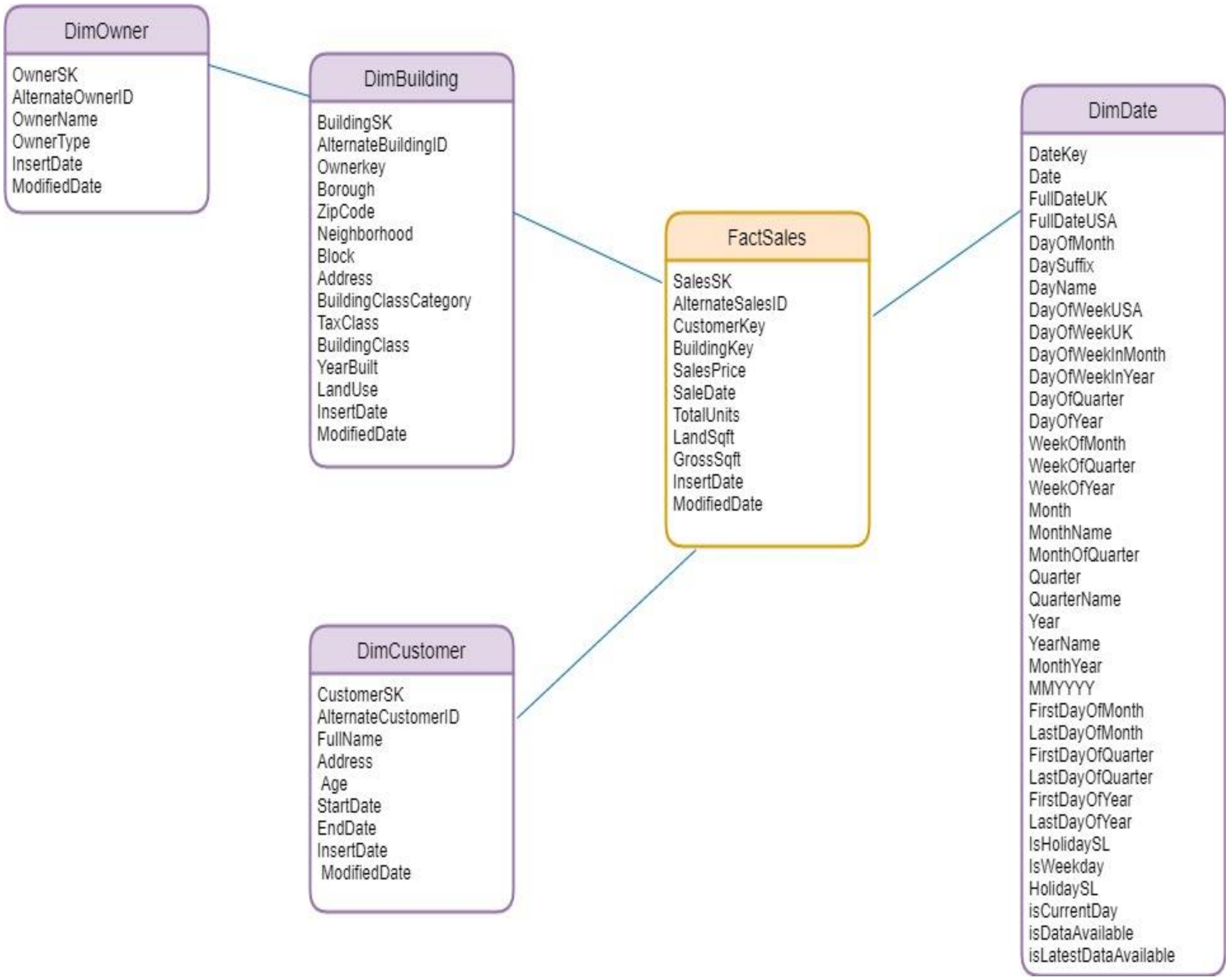


Figure 3 - 4.1 Dimensional Model

5. ETL Development

5.1Extraction

Extract data from source to staging

The first step of the SSIS ETL process is to extract data from source files. I used two types of data sources (source database, flat files)

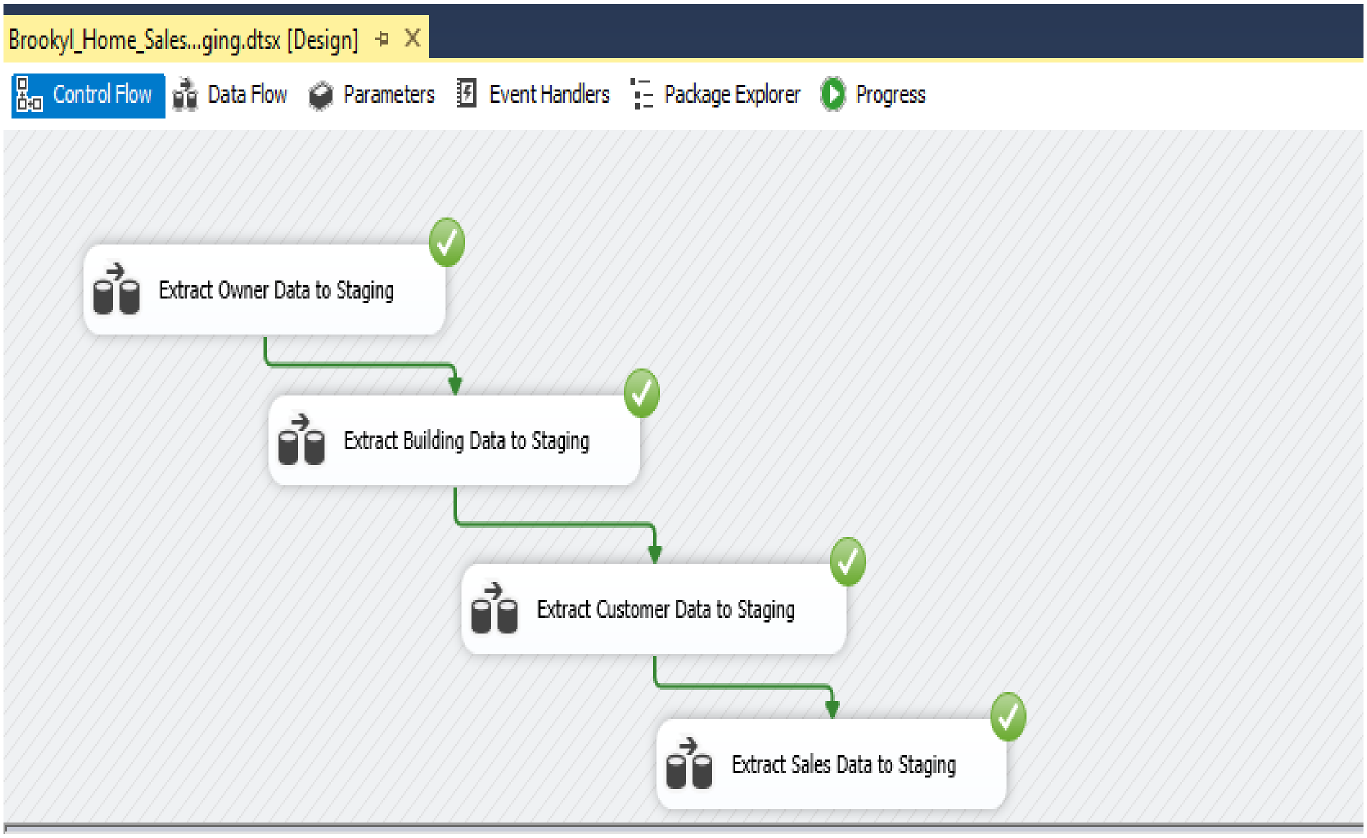

Brookyl_Home_Sales...ging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Extract Owner Data to Staging
Extract Building Data to Staging
Extract Customer Data to Staging
Extract Sales Data to Staging

Figure 4 - 5.1 Load data from Source to Staging (Control Flow)

First, I extract owner data from Brooklyn_Home_Sales_SourceDB to staging. In this process the source is a database table and destination is a staging table which is in data warehouse therefore I added “OLE DB Source” for source and “OLE DB Destination” for destination on the data flow task.

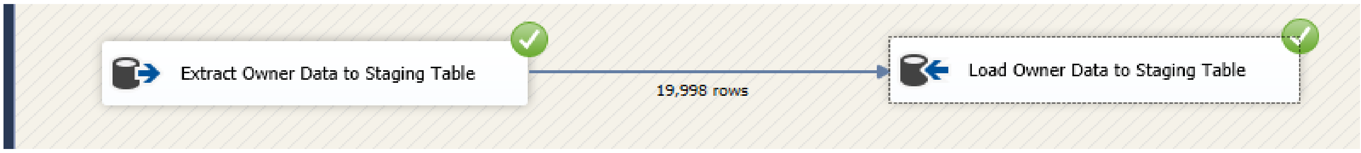

Extract Owner Data to Staging Table
19,998 rows
Load Owner Data to Staging Table

Figure 5- 5.2 Extract Owner data to Staging

My second source is a customer text file. In this case I used “Flat File Source” for source and “OLE DB Destination” for destination.

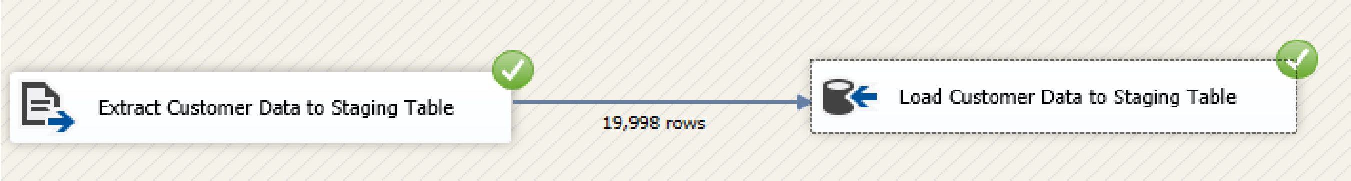

Extract Customer Data to Staging Table
19,998 rows
Load Customer Data to Staging Table

Figure 6 - 5.3 Extract Customer data to Staging

My third source is Building and Sales CSV file. In this case also I used “Flat File Source” for source and “OLE DB Destination” for destination.

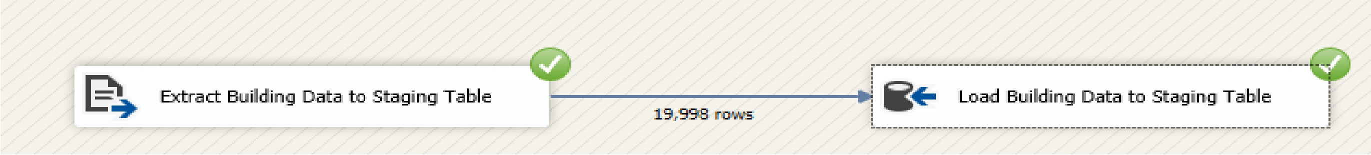

Extract Building Data to Staging Table
19,998 rows
Load Building Data to Staging Table

Figure 7 - 5.4 Extract Building data to Staging

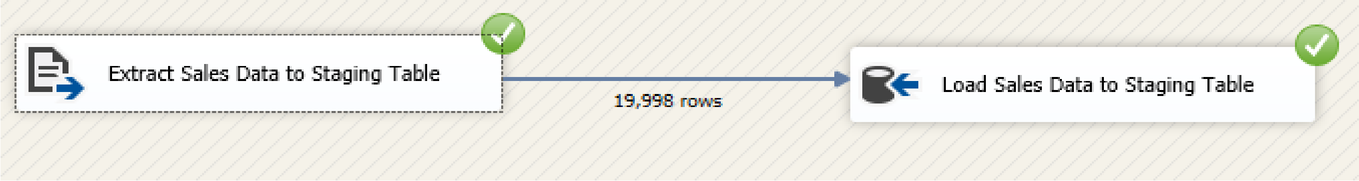

Extract Sales Data to Staging Table
19,998 rows
Load Sales Data to Staging Table

Figure 8 - 5.5 Extract Sales data to Staging

If we run the process multiple times, the staging table will be repeatedly loaded with data without truncating the data already available in the table. There for I added “Execute Sql Task” to the event handler. I repeated this process for all the tables. (Building, Sales, Customer, Owner)

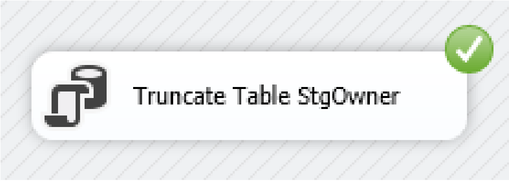


Figure 9 - 5.6 Truncate Owner Table

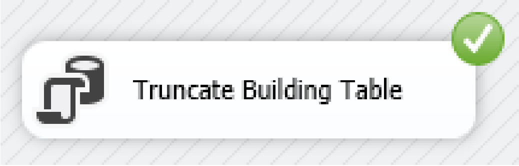


Figure 10 - 5.7 Truncate Building Table

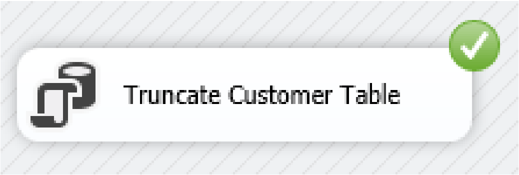


Figure 11 - 5.8 Truncate Customer Table

Figure 12 - 5.9 Truncate Sales Table

5.2Transformations

The second step of the ETL process.

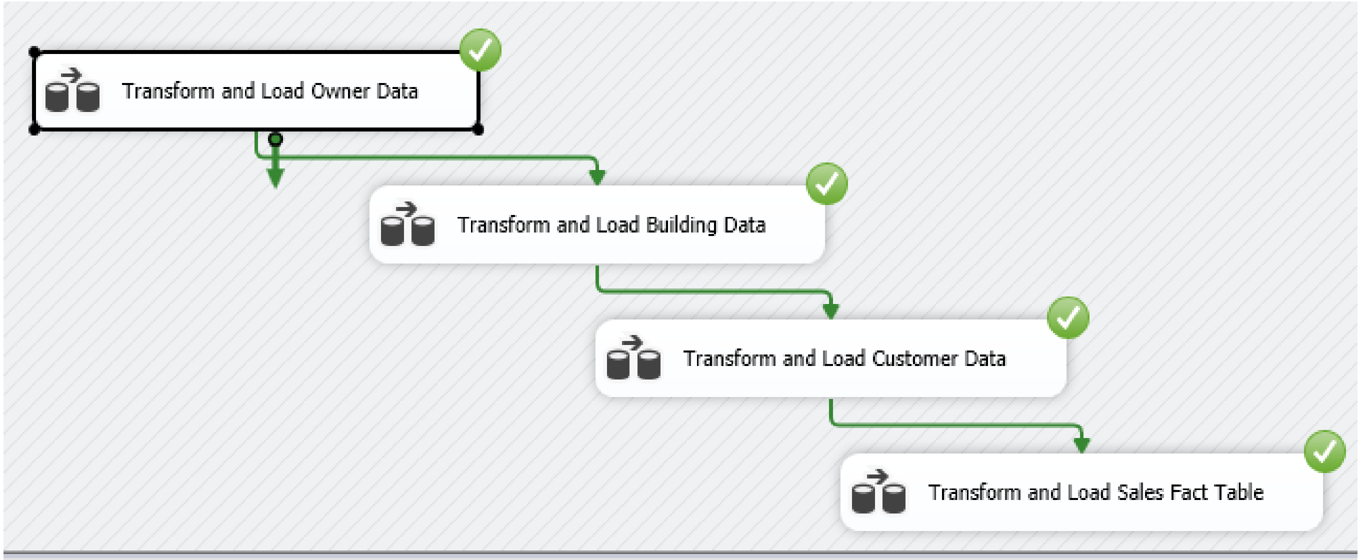

Transform and Load Owner Data
Transform and Load Building Data
Transform and Load Customer Data
Transform and Load Sales Fact Table

Figure 13 - 5.10 Transform and load data to dim tables

Transform and Load Owner Dimension

Then in the transformation process when inserting data to a hierarchy we must first load the data to the table which is in the end of the hierarchy not the table which is directly connecting to the fact table. Therefore, next I load Owner dimension.

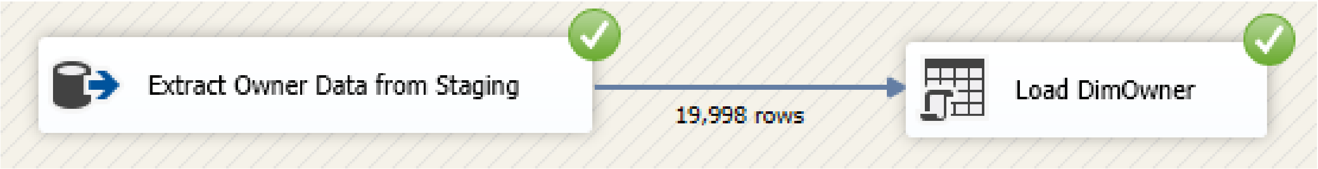
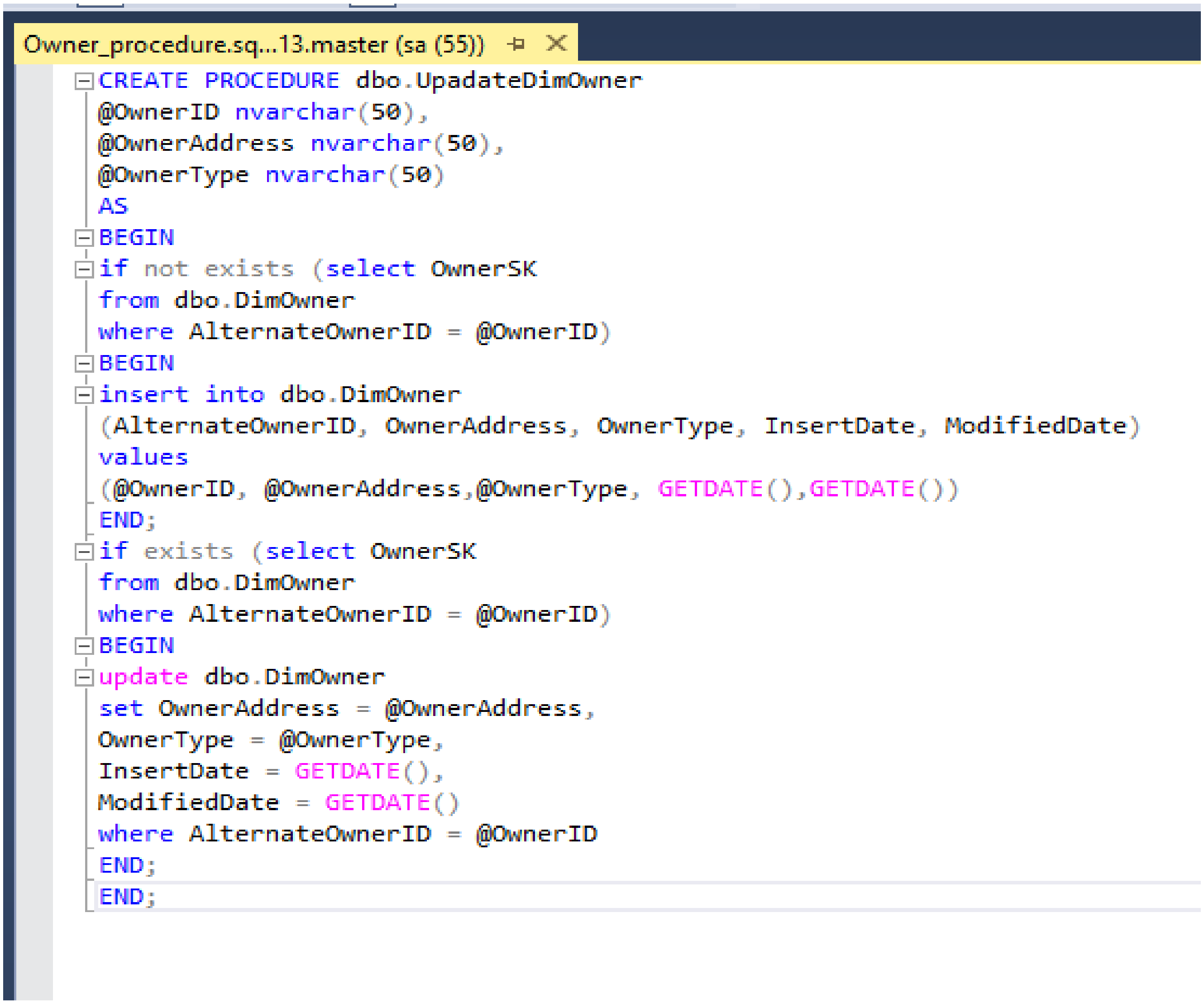


Figure 14 - 5.11 Transform and load owner dimension

I used stored procedure to check for existing data and update the table.



```
CREATE PROCEDURE dbo.UpadateDimOwner
@OwnerID nvarchar(50),
@OwnerAddress nvarchar(50),
@OwnerType nvarchar(50)
AS
BEGIN
if not exists (select OwnerSK
from dbo.DimOwner
where AlternateOwnerID = @OwnerID)
BEGIN
insert into dbo.DimOwner
(AlternateOwnerID, OwnerAddress, OwnerType, InsertDate, ModifiedDate)
values
(@OwnerID, @OwnerAddress,@OwnerType, GETDATE(),GETDATE())
END;
if exists (select OwnerSK
from dbo.DimOwner
where AlternateOwnerID = @OwnerID)
BEGIN
update dbo.DimOwner
set OwnerAddress = @OwnerAddress,
OwnerType = @OwnerType,
InsertDate = GETDATE(),
ModifiedDate = GETDATE()
where AlternateOwnerID = @OwnerID
END;
END;
```

Figure 15 - 5.12 Update procedure for dimowner

Transform and Load Building Dimension

Building dimension has owner id referred from owner dimension. Therefore, I need to get data from building staging table and owner dimension.

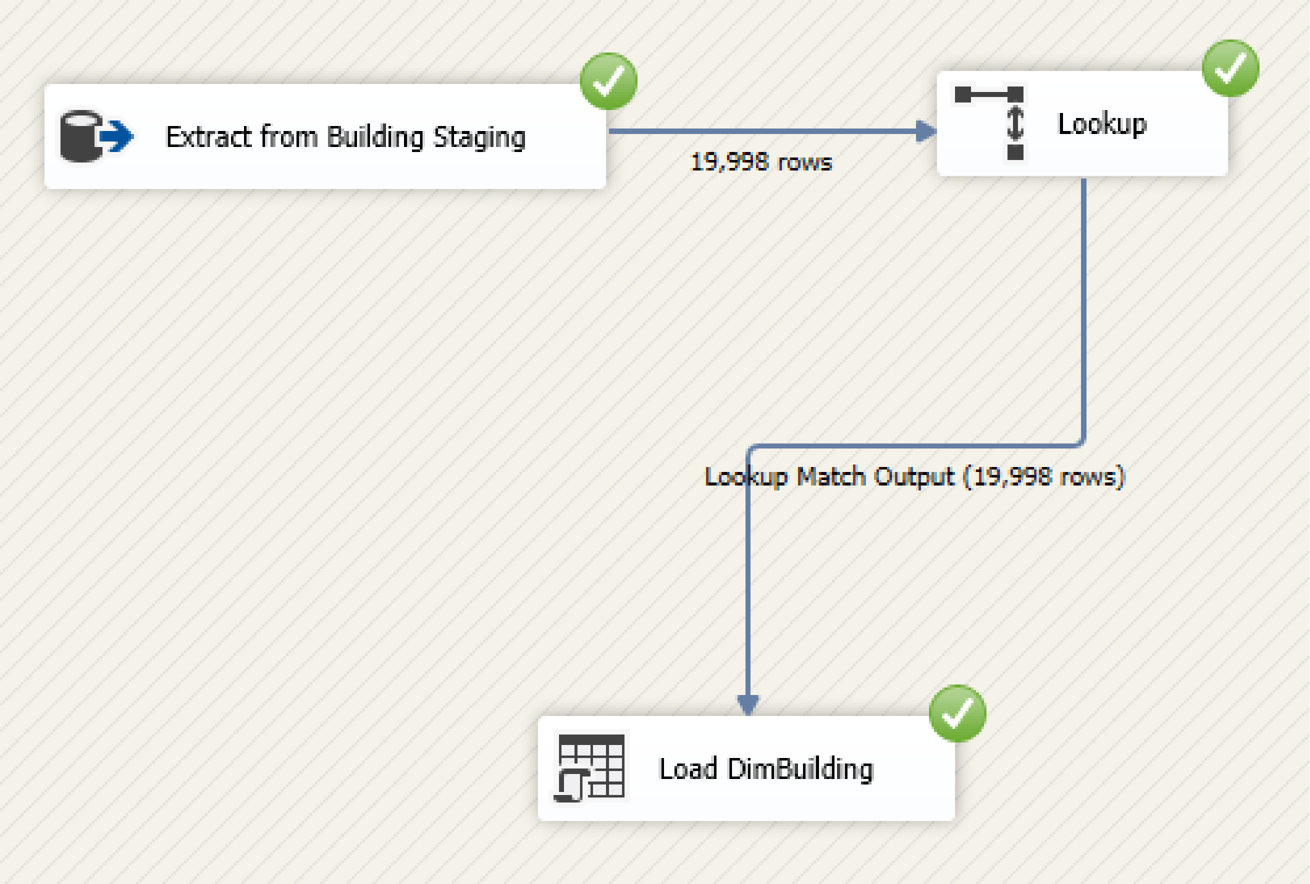
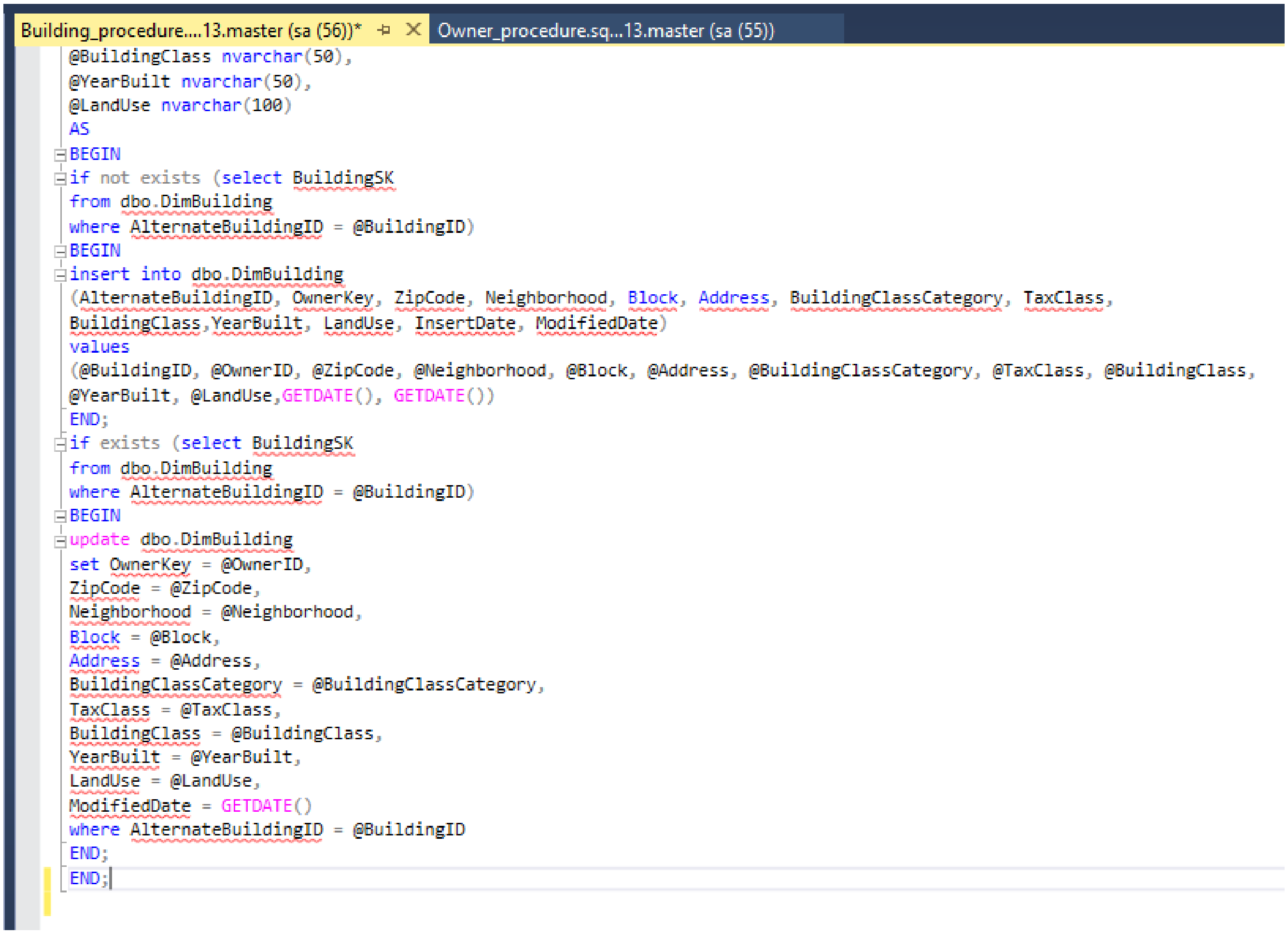

Extract from Building Staging
19,998 rows
Lookup
Lookup Match Output (19,998 rows)
Load DimBuilding

Figure 16 - 5.13 Transform and load building dimension

I used stored procedure to check for existing data and update the table


Building_procedure....13.master (sa (56))*
Owner_procedure.sq...13.master (sa (55))

```sql
@BuildingClass nvarchar(50),
@YearBuilt nvarchar(50),
@LandUse nvarchar(100)
AS
BEGIN
if not exists (select BuildingSK
from dbo.DimBuilding
where AlternateBuildingID = @BuildingID)
BEGIN
insert into dbo.DimBuilding
(AlternateBuildingID, OwnerKey, ZipCode, Neighborhood, Block, Address, BuildingClassCategory, TaxClass,
BuildingClass,YearBuilt, LandUse, InsertDate, ModifiedDate)
values
(@BuildingID, @OwnerID, @ZipCode, @Neighborhood, @Block, @Address, @BuildingClassCategory, @TaxClass, @BuildingClass,
@YearBuilt, @LandUse,GETDATE(), GETDATE())
END;
if exists (select BuildingSK
from dbo.DimBuilding
where AlternateBuildingID = @BuildingID)
BEGIN
update dbo.DimBuilding
set OwnerKey = @OwnerID,
ZipCode = @ZipCode,
Neighborhood = @Neighborhood,
Block = @Block,
Address = @Address,
BuildingClassCategory = @BuildingClassCategory,
TaxClass = @TaxClass,
BuildingClass = @BuildingClass,
YearBuilt = @YearBuilt,
LandUse = @LandUse,
ModifiedDate = GETDATE()
where AlternateBuildingID = @BuildingID
END;
END;
```

Figure 17 - 5.14 Stored procedure for building

Transform and Load Customer Dimension

Customer dimension table load data from customer staging tables. Therefore, I used OLE DB sources in the data flow. Then I used a derived column to add insert date and modified date to the Customer dimension table. Customer addresses can change time to time, and I need to keep track of their historical address. In this case I used slowly changing dimension for address.

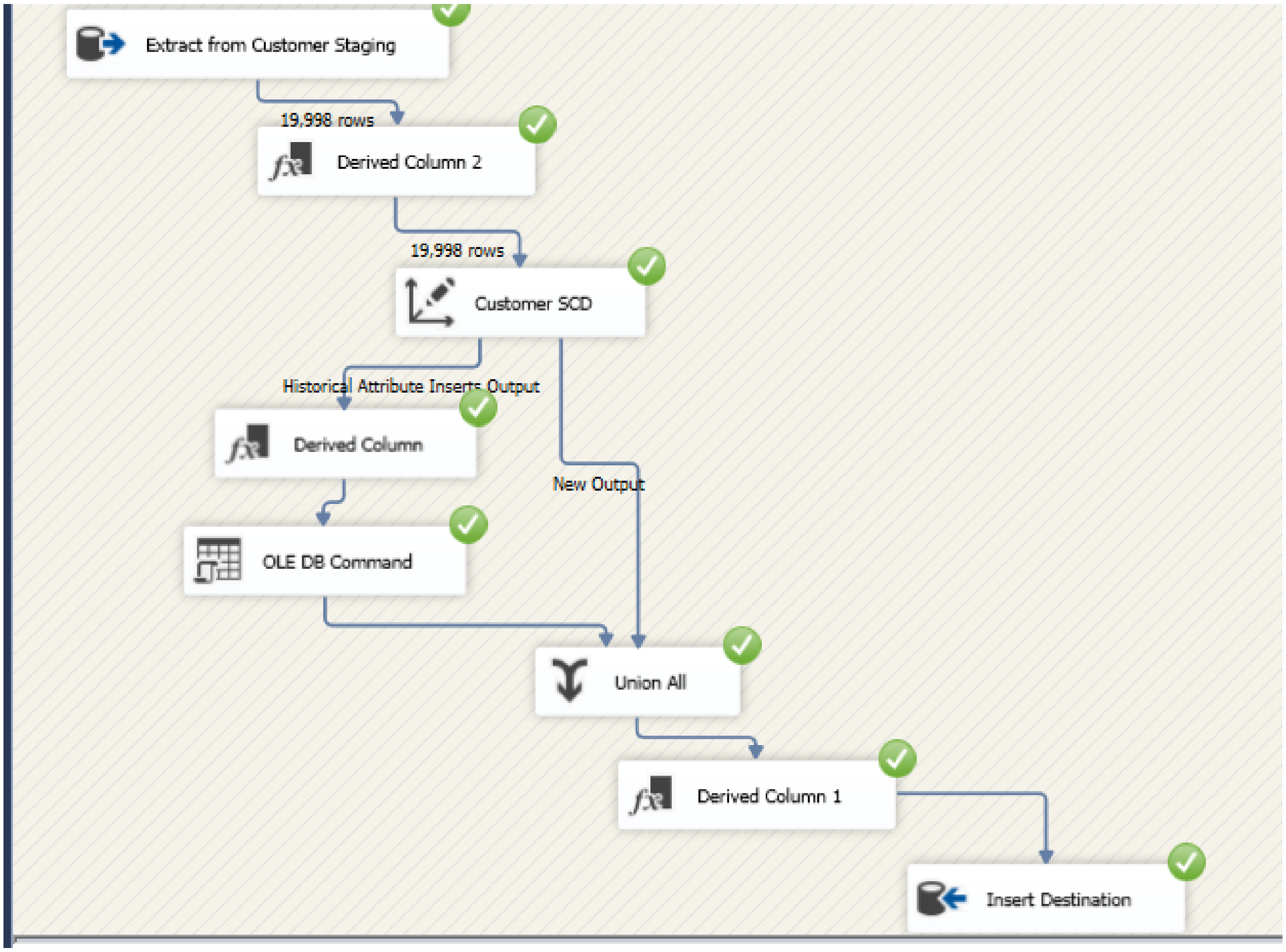


Figure 18 - 5.15 Transform and load customer dimension

Load Fact table

When loading fact tables, it is important to load the linked dimension table keys in the fact table. These keys are required to get the dimension details for the facts when generating reports based on the data.

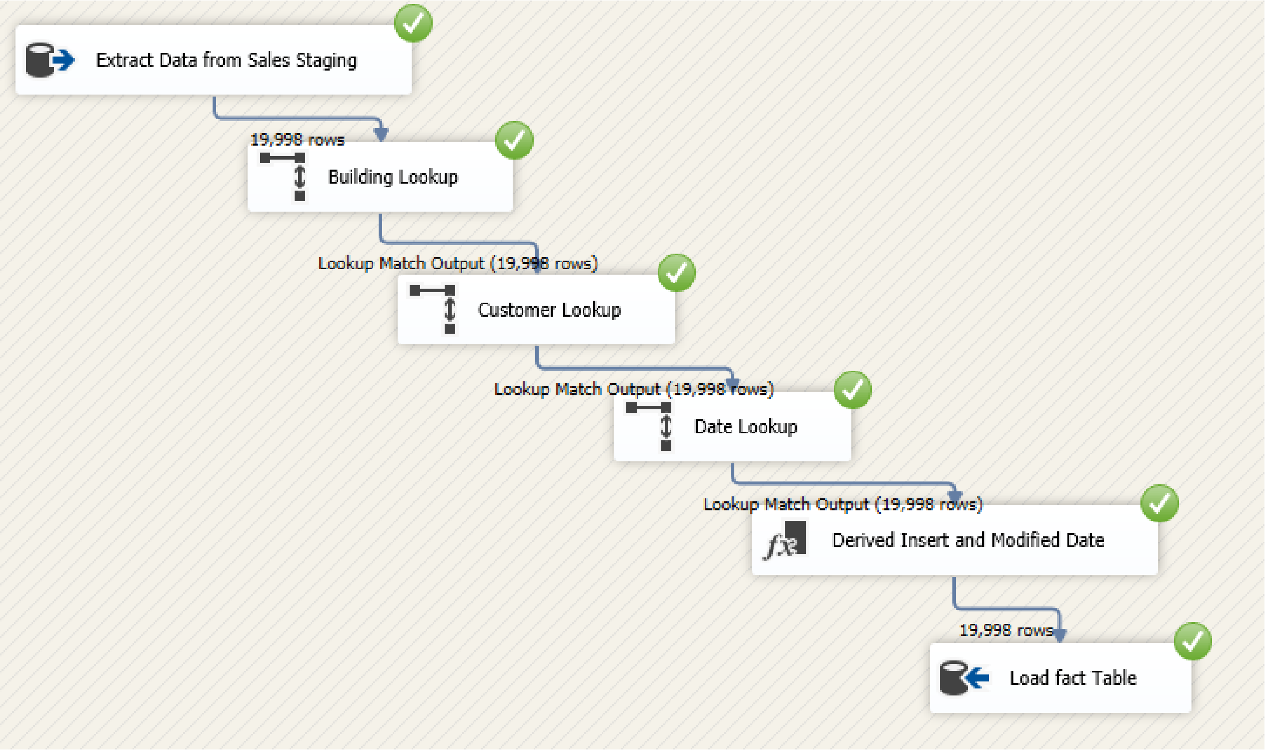
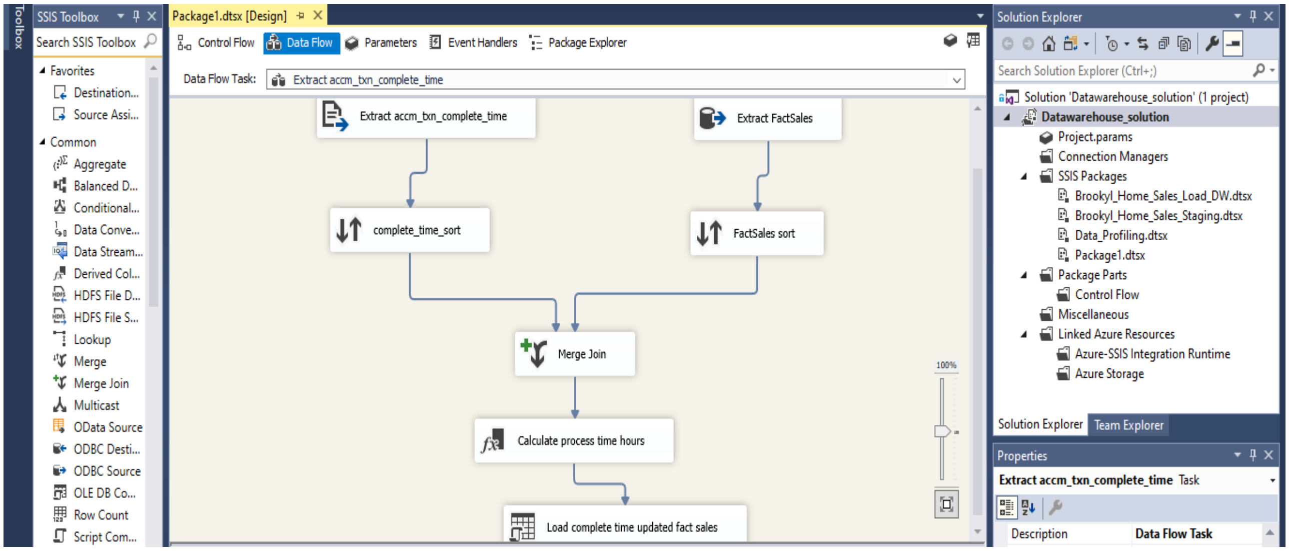

Extract Data from Sales Staging
19,998 rows
Building Lookup
Lookup Match Output (19,998 rows)
Customer Lookup
Lookup Match Output (19,998 rows)
Date Lookup
Lookup Match Output (19,998 rows)
Derived Insert and Modified Date
19,998 rows
Load fact Table

Figure 19 - 5.16 Transform and load fact table

6. ETL development – Accumulating fact tables



- First a CSV file was created with the fact table natural key and the column accm_txn_complete_time.
- Then the fact table was taken into the dataflow task by using an “OLE DB Source”.
- Then the fact table natural key column in the newly created csv file and the natural key of the fact table was sorted using Sort components.
- Then those two sources were merged using the Merge Join component.
- Then the difference between the create time column and the complete time was found by hours using DATEDIFF function and the output of it was saved to the txn_process_time_hours column.