CSCI 5408 ASSIGNMENT 4

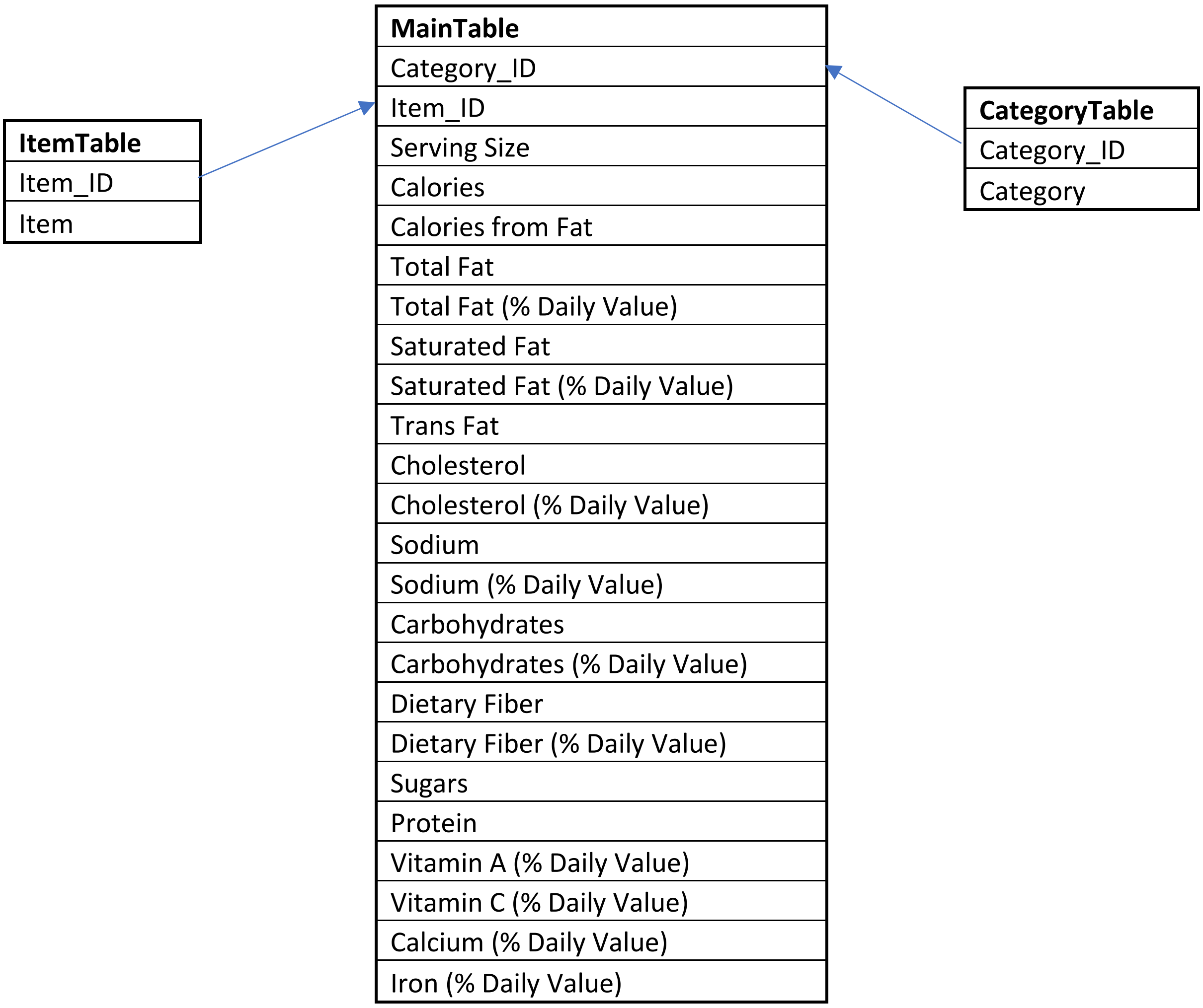
DW/OLAP & ETL PIPELINES WITH SQL SERVER 2016

1. ETL Process and Schema

The dataset chosen for this operation is "Nutrition Facts for McDonald's menu" from the weblink link https://www.kaggle.com/mcdonalds/nutrition-facts.

The dataset is Normalized into three staging tables namely MainTable, CategoryTable and ItemTable. ItemTable and CategoryTable stores Items and Categories with Item_ID and Category_ID being the Foreign Keys. The remaining columns contain the nutririon information for the particular item from the particular category.

Entity Relationship Diagram



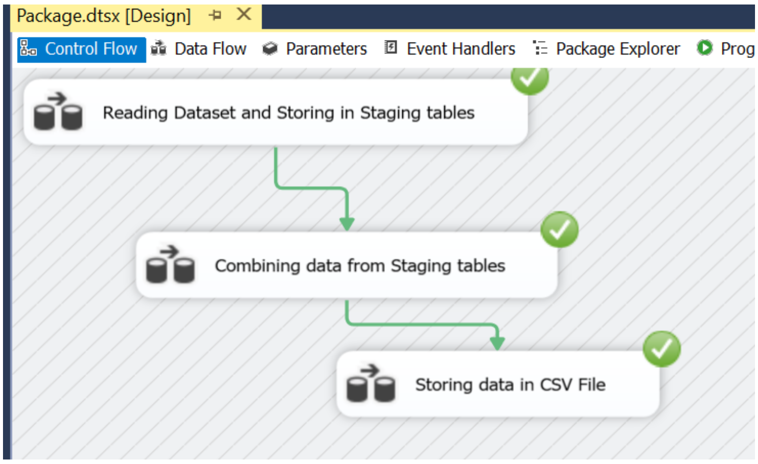
ItemTable
Item_ID
Item

MainTable
Category_ID
Item_ID
Serving Size
Calories
Calories from Fat
Total Fat
Total Fat (% Daily Value)
Saturated Fat
Saturated Fat (% Daily Value)
Trans Fat
Cholesterol
Cholesterol (% Daily Value)
Sodium
Sodium (% Daily Value)
Carbohydrates
Carbohydrates (% Daily Value)
Dietary Fiber
Dietary Fiber (% Daily Value)
Sugars
Protein
Vitamin A (% Daily Value)
Vitamin C (% Daily Value)
Calcium (% Daily Value)
Iron (% Daily Value)

CategoryTable
Category_ID
Category

2. ETL Activities

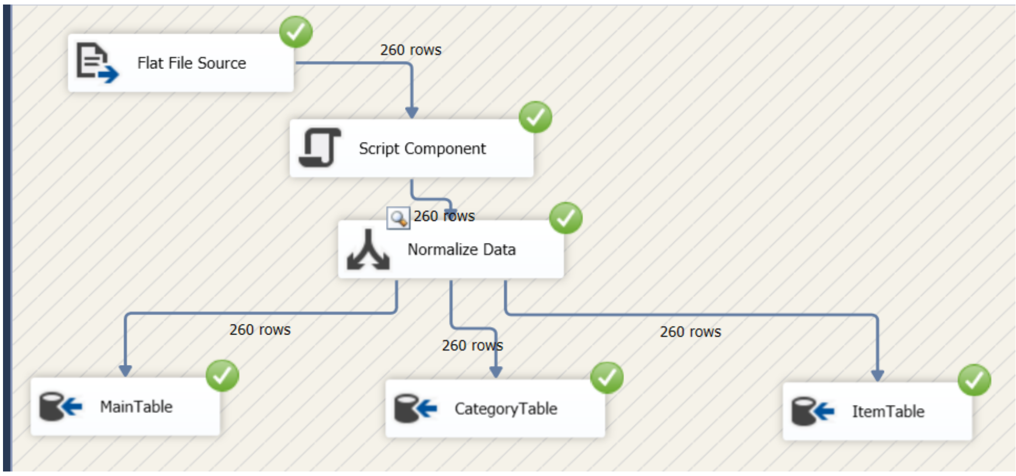
Below is the Control Flow of the ETL job. The process is divided into three stages as shown in the Control Flow.



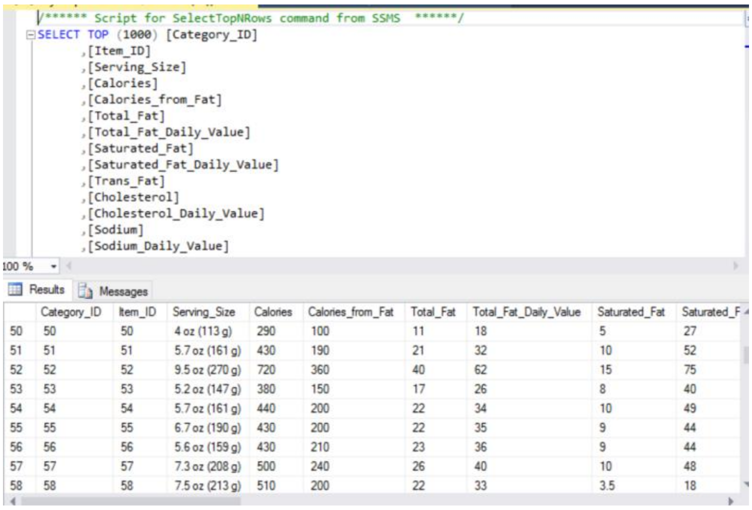
Each Data flow and its functions are as follows:

Reading the data from a dataset

- The dataset "*menu.csv*" is read using the Flat File Source.
- Script component is used to generate the Foreign Keys to normalize data.
- The Normalized data is stored into three different staging tables mentioned above.

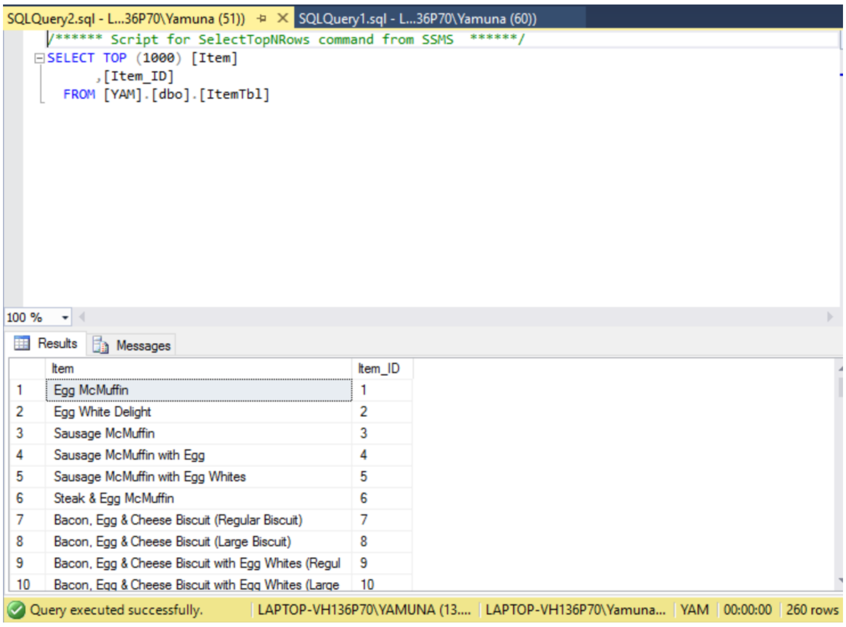


Intermediate tables



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Category_ID]
      ,[Item_ID]
      ,[Serving_Size]
      ,[Calories]
      ,[Calories_from_Fat]
      ,[Total_Fat]
      ,[Total_Fat_Daily_Value]
      ,[Saturated_Fat]
      ,[Saturated_Fat_Daily_Value]
      ,[Trans_Fat]
      ,[Cholesterol]
      ,[Cholesterol_Daily_Value]
      ,[Sodium]
      ,[Sodium_Daily_Value]
```

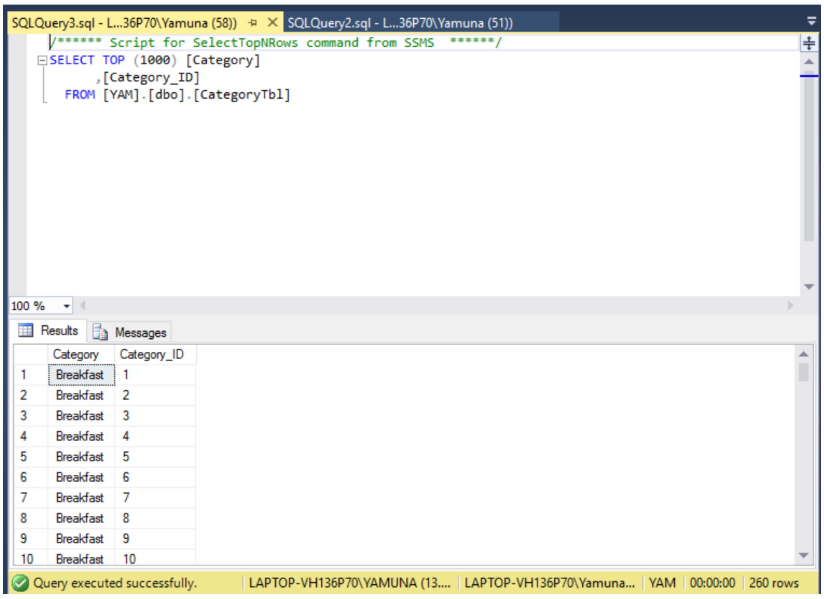
	Category_ID	Item_ID	Serving_Size	Calories	Calories_from_Fat	Total_Fat	Total_Fat_Daily_Value	Saturated_Fat	Saturated_F
50	50	50	4 oz (113 g)	290	100	11	18	5	27
51	51	51	5.7 oz (161 g)	430	190	21	32	10	52
52	52	52	9.5 oz (270 g)	720	360	40	62	15	75
53	53	53	5.2 oz (147 g)	380	150	17	26	8	40
54	54	54	5.7 oz (161 g)	440	200	22	34	10	49
55	55	55	6.7 oz (190 g)	430	200	22	35	9	44
56	56	56	5.6 oz (159 g)	430	210	23	36	9	44
57	57	57	7.3 oz (208 g)	500	240	26	40	10	48
58	58	58	7.5 oz (213 g)	510	200	22	33	3.5	18


SQLQuery2.sql - L...36P70\Yamuna (51))
SQLQuery1.sql - L...36P70\Yamuna (60))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Item]
      ,[Item_ID]
  FROM [YAM].[dbo].[ItemTbl]
```

	Item	Item_ID
1	Egg McMuffin	1
2	Egg White Delight	2
3	Sausage McMuffin	3
4	Sausage McMuffin with Egg	4
5	Sausage McMuffin with Egg Whites	5
6	Steak & Egg McMuffin	6
7	Bacon, Egg & Cheese Biscuit (Regular Biscuit)	7
8	Bacon, Egg & Cheese Biscuit (Large Biscuit)	8
9	Bacon, Egg & Cheese Biscuit with Egg Whites (Regul	9
10	Bacon, Egg & Cheese Biscuit with Egg Whites (Large	10

Query executed successfully. | LAPTOP-VH136P70\YAMUNA (13.... | LAPTOP-VH136P70\Yamuna... | YAM | 00:00:00 | 260 rows


SQLQuery3.sql - L...36P70\Yamuna (58))
SQLQuery2.sql - L...36P70\Yamuna (51))
100 %
Results
Messages

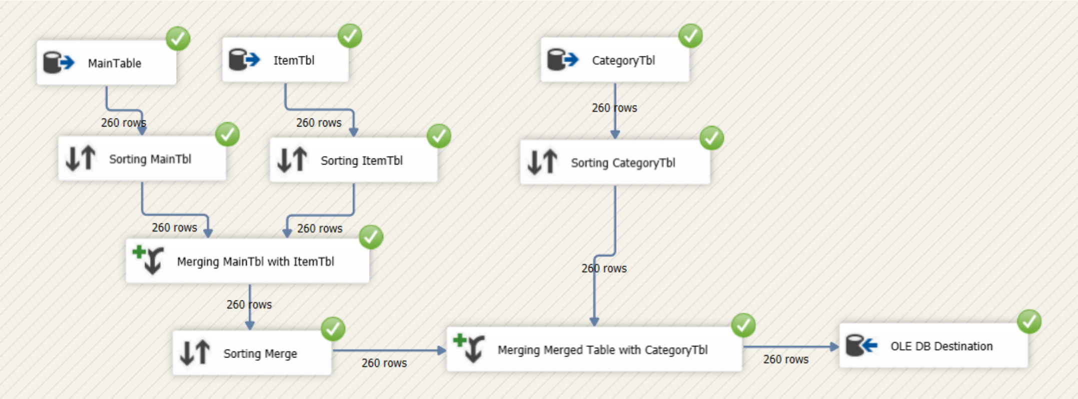
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Category]
      ,[Category_ID]
  FROM [YAM].[dbo].[CategoryTbl]
```

	Category	Category_ID
1	Breakfast	1
2	Breakfast	2
3	Breakfast	3
4	Breakfast	4
5	Breakfast	5
6	Breakfast	6
7	Breakfast	7
8	Breakfast	8
9	Breakfast	9
10	Breakfast	10

Query executed successfully. | LAPTOP-VH136P70\YAMUNA (13.... | LAPTOP-VH136P70\Yamuna... | YAM | 00:00:00 | 260 rows

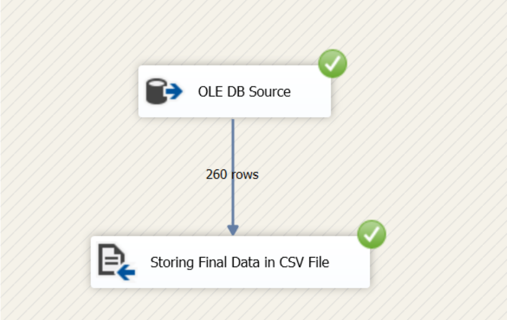
Normalising and Storing them in intermediate tables known as staging tables

- The staging tables are used as source and the data is read.
- Sort and join are performed between the tables to merge the data
- All merged data are stored in to another intermediate table FinalTable.
- The structure of this table is similar to the structure of the dataset.



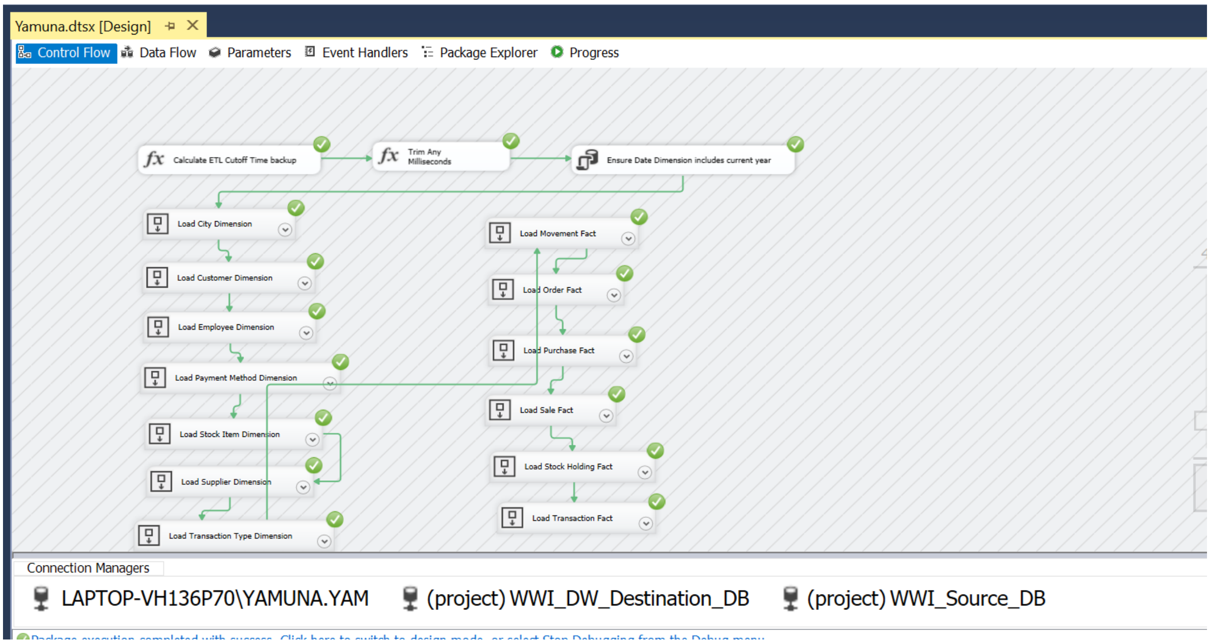
Merging the data from the staging tables and writing it back to CSV files

The FinalTable is used as the source and data from this table is written directly into the target CSV file "*output.csv*"



Note: All the files (input & output), ETL packages, SQL scripts to create intermediate tables are stored attached with this report.

3. DW design/Construction



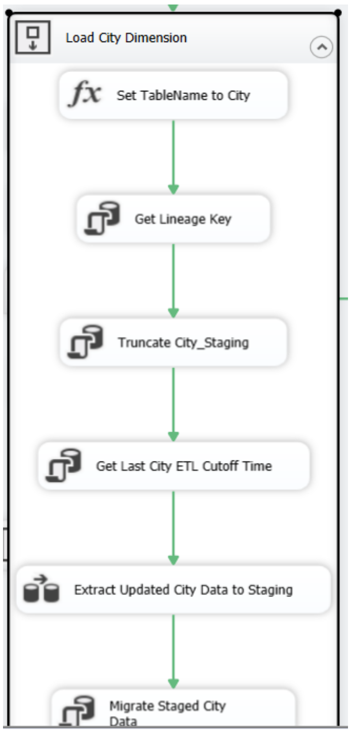
I. Sequence containers are used to process and load each Fact and Dimension tables.

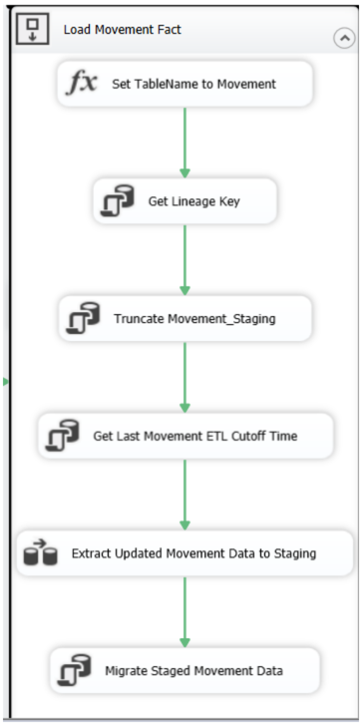
II. Inside each container, the table name to be processed is selected.

III. Existing data from the staging table is first truncated.

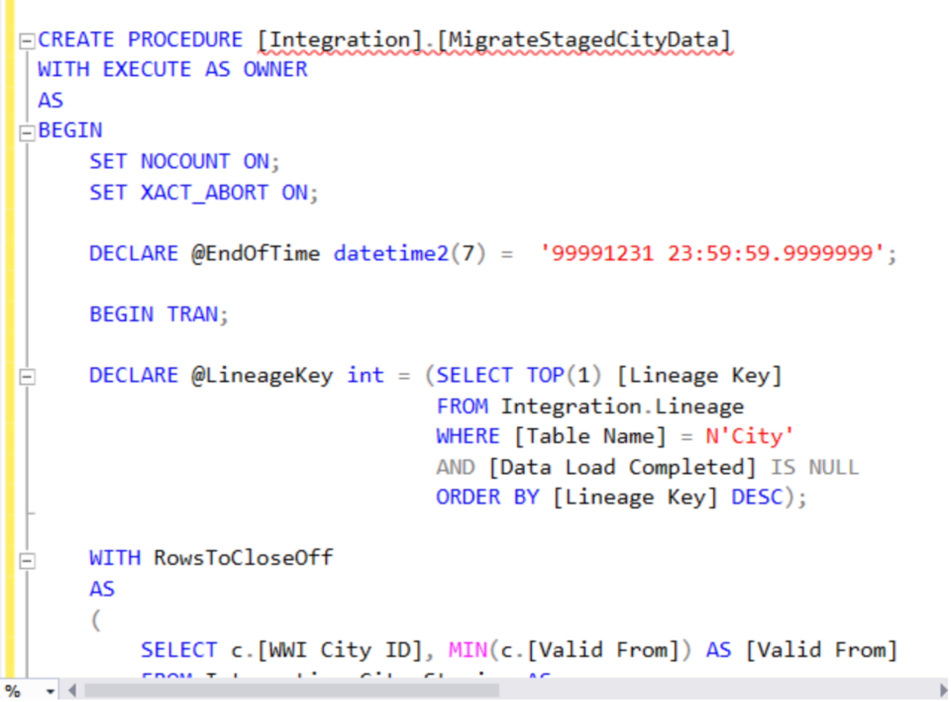
IV. The Recent updated data is moved from the Operational database to staging table.

V. Data from staging table is moved to the Dimension table by calling the corresponding stored procedures.



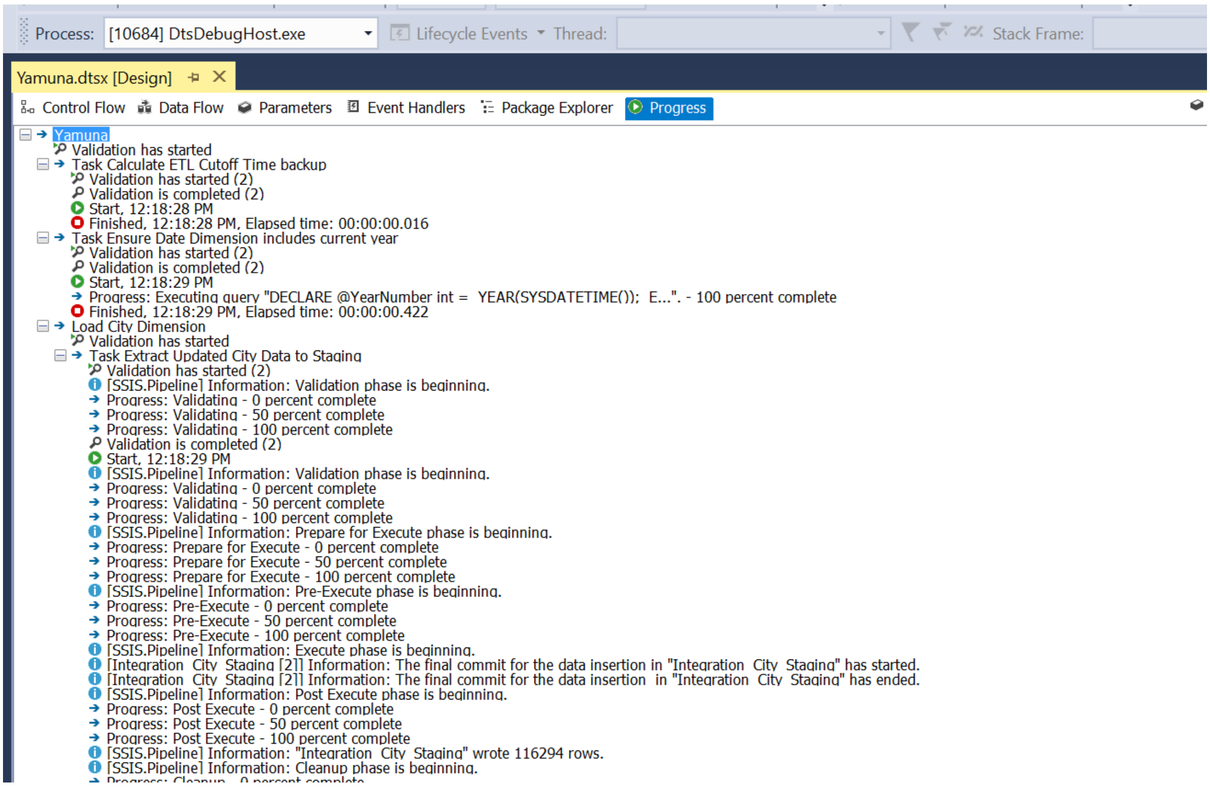

Load Movement Fact
Set TableName to Movement
Get Lineage Key
Truncate Movement_Staging
Get Last Movement ETL Cutoff Time
Extract Updated Movement Data to Staging
Migrate Staged Movement Data

Stored Procedure

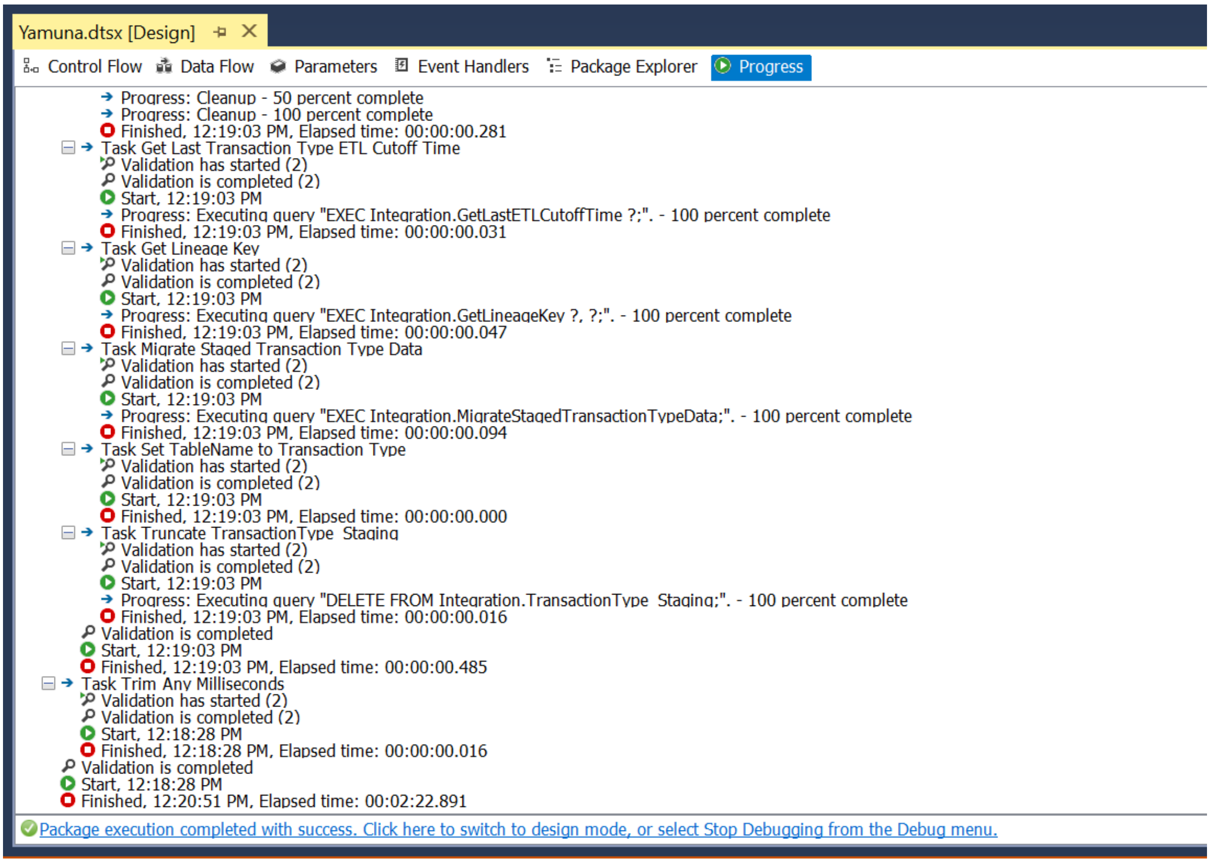


```
CREATE PROCEDURE [Integration].[MigrateStagedCityData]
WITH EXECUTE AS OWNER
AS
BEGIN
    SET NOCOUNT ON;
    SET XACT_ABORT ON;

    DECLARE @EndOfTime datetime2(7) =  '99991231 23:59:59.9999999';

    BEGIN TRAN;

    DECLARE @LineageKey int = (SELECT TOP(1) [Lineage Key]
                               FROM Integration.Lineage
                               WHERE [Table Name] = N'City'
                               AND [Data Load Completed] IS NULL
                               ORDER BY [Lineage Key] DESC);

    WITH RowsToCloseOff
    AS
    (
        SELECT c.[WWI City ID], MIN(c.[Valid From]) AS [Valid From]
```

Output screen shots creating the Data Warehouse components

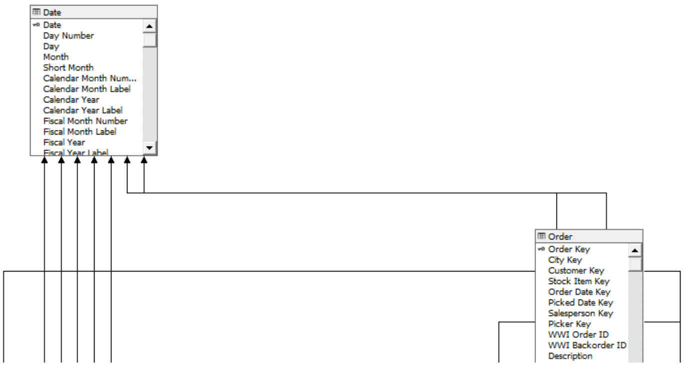


```
Yamuna
  Validation has started
  Task Calculate ETL Cutoff Time backup
    Validation has started (2)
    Validation is completed (2)
    Start, 12:18:28 PM
    Finished, 12:18:28 PM, Elapsed time: 00:00:00.016
  Task Ensure Date Dimension includes current year
    Validation has started (2)
    Validation is completed (2)
    Start, 12:18:29 PM
    Progress: Executing query "DECLARE @YearNumber int =  YEAR(SYSDATETIME());  E...". - 100 percent complete
    Finished, 12:18:29 PM, Elapsed time: 00:00:00.422
  Load City Dimension
    Validation has started
    Task Extract Updated City Data to Staging
      Validation has started (2)
      [SSIS.Pipeline] Information: Validation phase is beginning.
      Progress: Validating - 0 percent complete
      Progress: Validating - 50 percent complete
      Progress: Validating - 100 percent complete
      Validation is completed (2)
      Start, 12:18:29 PM
      [SSIS.Pipeline] Information: Validation phase is beginning.
      Progress: Validating - 0 percent complete
      Progress: Validating - 50 percent complete
      Progress: Validating - 100 percent complete
      [SSIS.Pipeline] Information: Prepare for Execute phase is beginning.
      Progress: Prepare for Execute - 0 percent complete
      Progress: Prepare for Execute - 50 percent complete
      Progress: Prepare for Execute - 100 percent complete
      [SSIS.Pipeline] Information: Pre-Execute phase is beginning.
      Progress: Pre-Execute - 0 percent complete
      Progress: Pre-Execute - 50 percent complete
      Progress: Pre-Execute - 100 percent complete
      [SSIS.Pipeline] Information: Execute phase is beginning.
      [Integration  City  Staging [2]] Information: The final commit for the data insertion in "Integration  City  Staging" has started.
      [Integration  City  Staging [2]] Information: The final commit for the data insertion  in "Integration  City  Staging" has ended.
      [SSIS.Pipeline] Information: Post Execute phase is beginning.
      Progress: Post Execute - 0 percent complete
      Progress: Post Execute - 50 percent complete
      Progress: Post Execute - 100 percent complete
      [SSIS.Pipeline] Information: "Integration  City  Staging" wrote 116294 rows.
      [SSIS.Pipeline] Information: Cleanup phase is beginning.
      Progress: Cleanup - 0 percent complete
```



```
        Progress: Cleanup - 50 percent complete
        Progress: Cleanup - 100 percent complete
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.281
      Task Get Last Transaction Type ETL Cutoff Time
        Validation has started (2)
        Validation is completed (2)
        Start, 12:19:03 PM
        Progress: Executing query "EXEC Integration.GetLastETLCutoffTime ?;". - 100 percent complete
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.031
      Task Get Lineage Key
        Validation has started (2)
        Validation is completed (2)
        Start, 12:19:03 PM
        Progress: Executing query "EXEC Integration.GetLineageKey ?, ?;". - 100 percent complete
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.047
      Task Migrate Staged Transaction Type Data
        Validation has started (2)
        Validation is completed (2)
        Start, 12:19:03 PM
        Progress: Executing query "EXEC Integration.MigrateStagedTransactionTypeData;". - 100 percent complete
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.094
      Task Set TableName to Transaction Type
        Validation has started (2)
        Validation is completed (2)
        Start, 12:19:03 PM
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.000
      Task Truncate TransactionType  Staging
        Validation has started (2)
        Validation is completed (2)
        Start, 12:19:03 PM
        Progress: Executing query "DELETE FROM Integration.TransactionType  Staging;". - 100 percent complete
        Finished, 12:19:03 PM, Elapsed time: 00:00:00.016
      Validation is completed
      Start, 12:19:03 PM
      Finished, 12:19:03 PM, Elapsed time: 00:00:00.485
    Task Trim Any Milliseconds
      Validation has started (2)
      Validation is completed (2)
      Start, 12:18:28 PM
      Finished, 12:18:28 PM, Elapsed time: 00:00:00.016
    Validation is completed
    Start, 12:18:28 PM
    Finished, 12:20:51 PM, Elapsed time: 00:02:22.891
```

Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

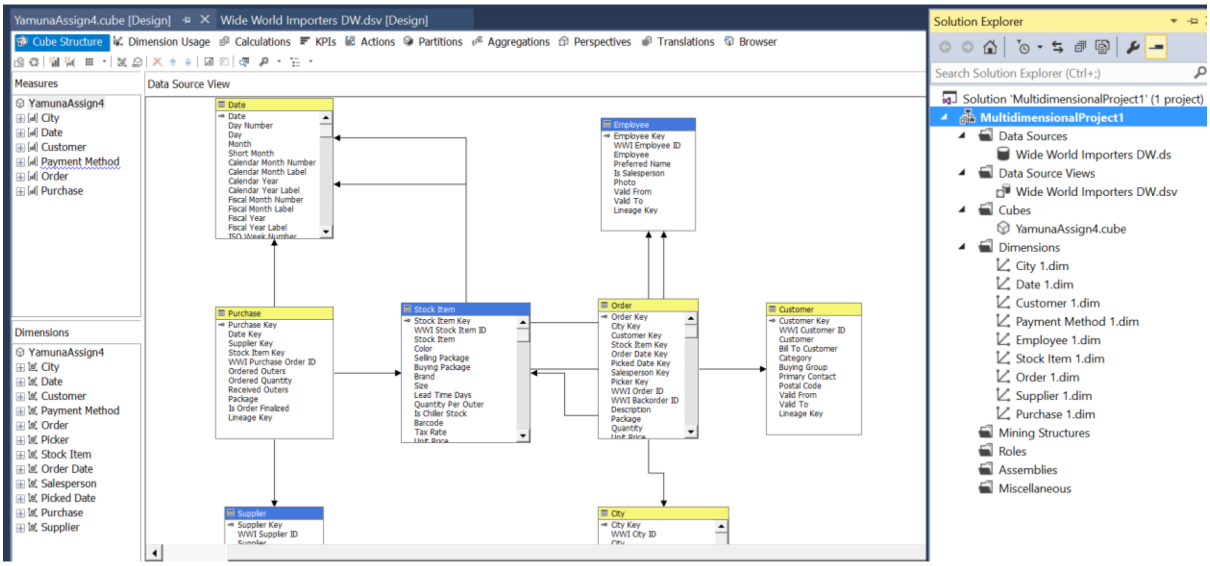


4. DW Scenario

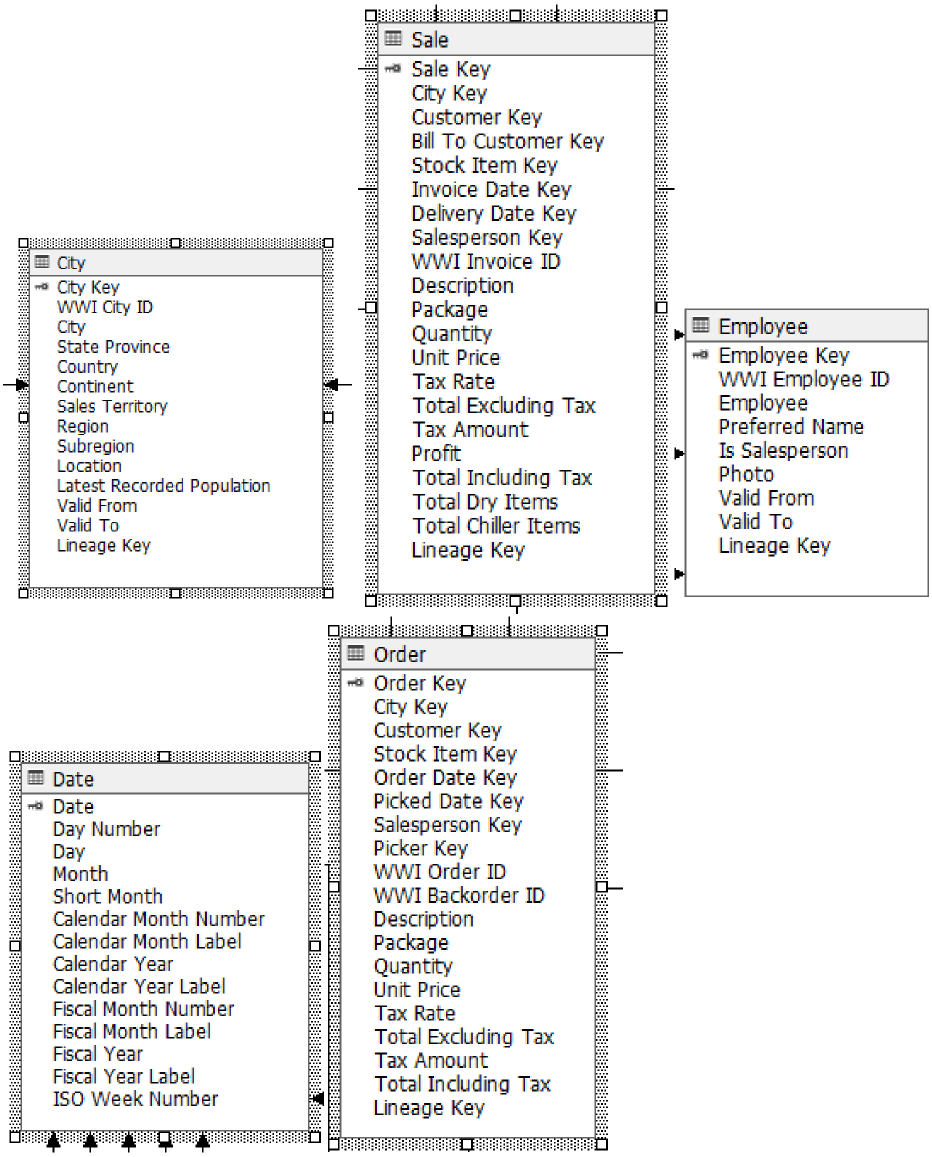
The Business Scenario for the Data Warehouse is designed to receive data on tables like Order, Customer, Date, City, Stock Item, Purchase, Supplier and Employee.

This scenario helps to analyse data on the various scenarios based on the orders placed by the customers, date, city where the order is placed and so on.

This scenario can help analyse data based on multiple sub-scenarios such as number of orders placed by customers from a particular province.



5. DW Materialization



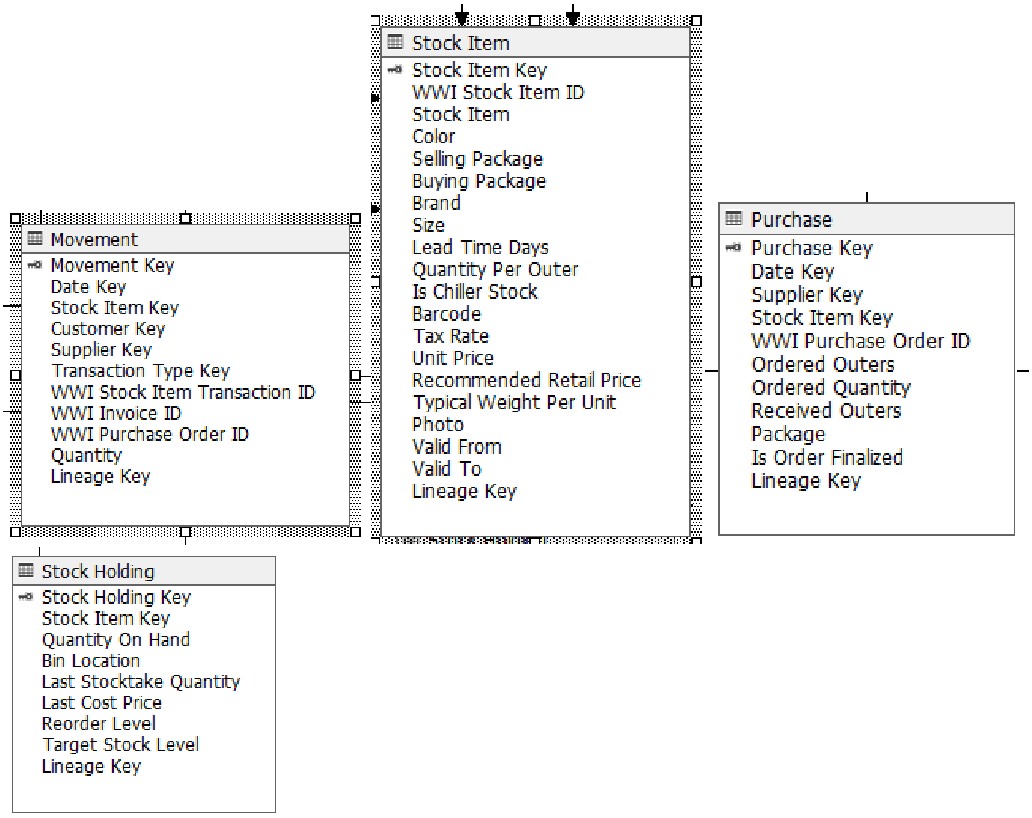
Sale
- Sale Key
- City Key
- Customer Key
- Bill To Customer Key
- Stock Item Key
- Invoice Date Key
- Delivery Date Key
- Salesperson Key
- WWI Invoice ID
- Description
- Package
- Quantity
- Unit Price
- Tax Rate
- Total Excluding Tax
- Tax Amount
- Profit
- Total Including Tax
- Total Dry Items
- Total Chiller Items
- Lineage Key

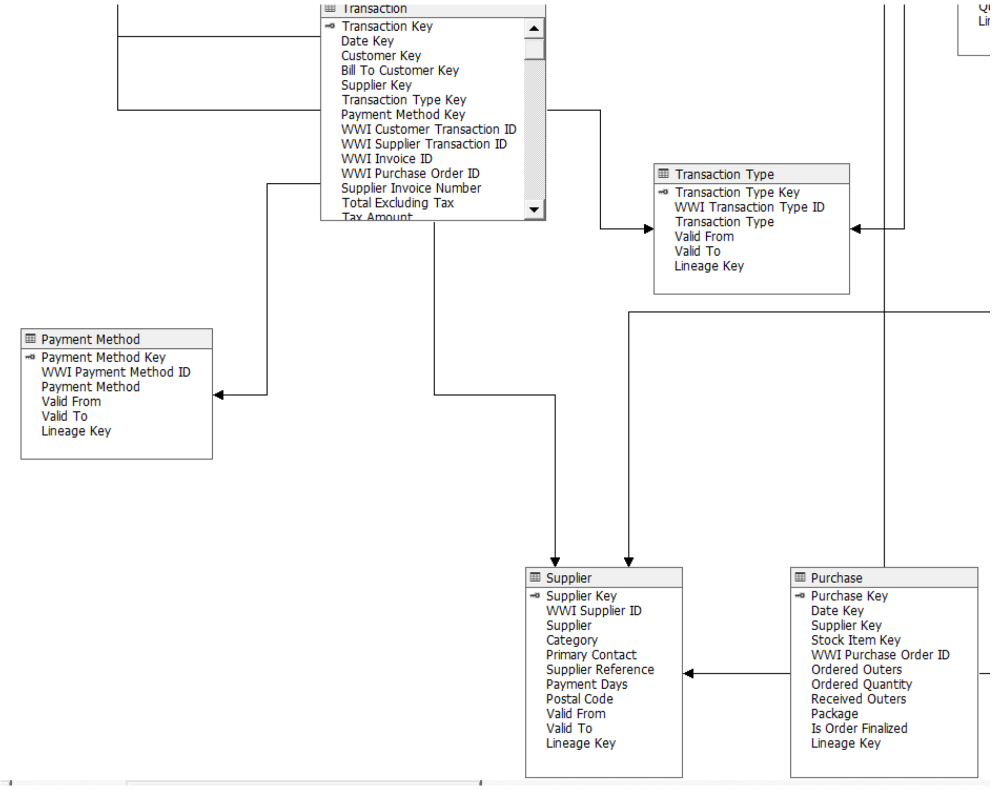
City
- City Key
- WWI City ID
- City
- State Province
- Country
- Continent
- Sales Territory
- Region
- Subregion
- Location
- Latest Recorded Population
- Valid From
- Valid To
- Lineage Key

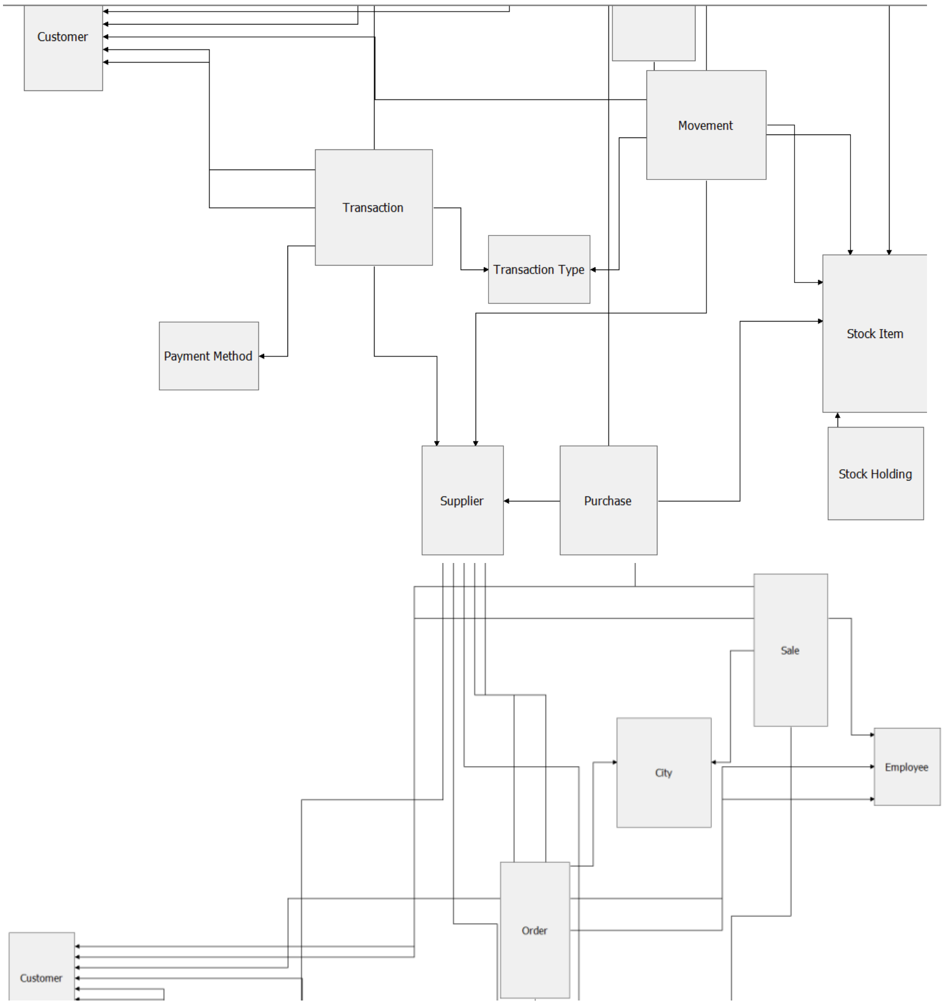
Employee
- Employee Key
- WWI Employee ID
- Employee
- Preferred Name
- Is Salesperson
- Photo
- Valid From
- Valid To
- Lineage Key

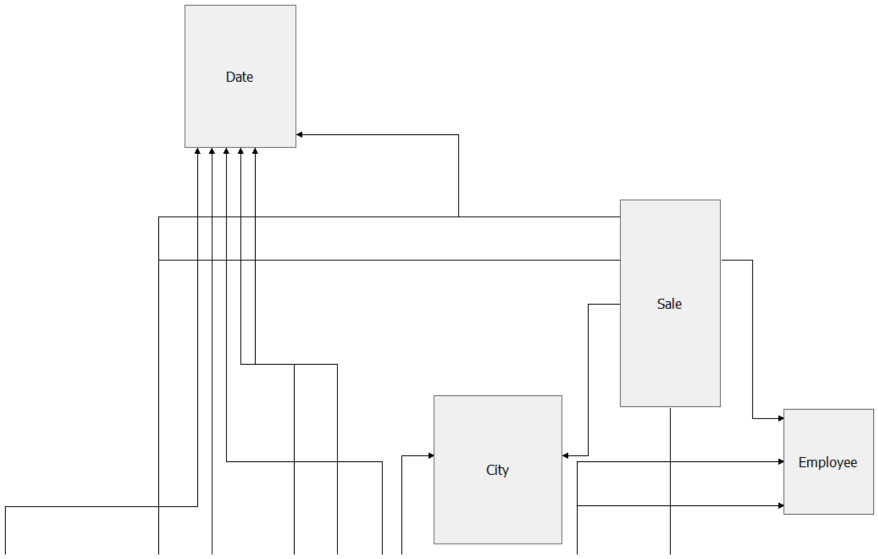
Order
- Order Key
- City Key
- Customer Key
- Stock Item Key
- Order Date Key
- Picked Date Key
- Salesperson Key
- Picker Key
- WWI Order ID
- WWI Backorder ID
- Description
- Package
- Quantity
- Unit Price
- Tax Rate
- Total Excluding Tax
- Tax Amount
- Total Including Tax
- Lineage Key

Date
- Date
- Day Number
- Day
- Month
- Short Month
- Calendar Month Number
- Calendar Month Label
- Calendar Year
- Calendar Year Label
- Fiscal Month Number
- Fiscal Month Label
- Fiscal Year
- Fiscal Year Label
- ISO Week Number

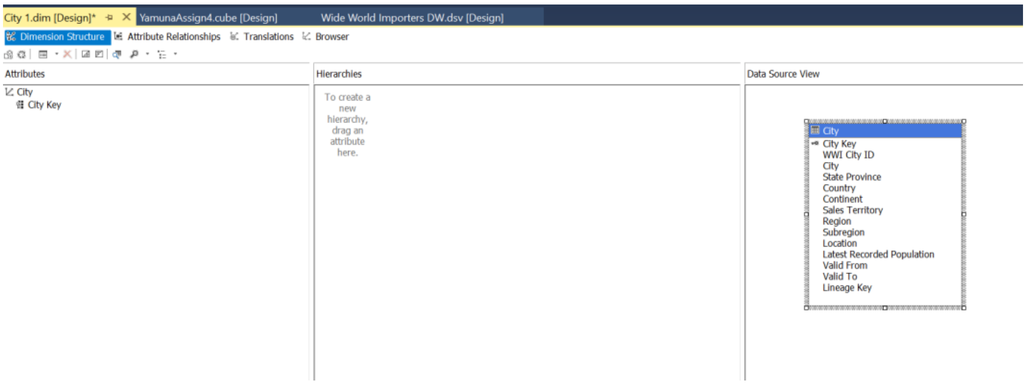

Stock Item
Stock Item Key
WWI Stock Item ID
Stock Item
Color
Selling Package
Buying Package
Brand
Size
Lead Time Days
Quantity Per Outer
Is Chiller Stock
Barcode
Tax Rate
Unit Price
Recommended Retail Price
Typical Weight Per Unit
Photo
Valid From
Valid To
Lineage Key
Movement
Movement Key
Date Key
Stock Item Key
Customer Key
Supplier Key
Transaction Type Key
WWI Stock Item Transaction ID
WWI Invoice ID
WWI Purchase Order ID
Quantity
Lineage Key
Purchase
Purchase Key
Date Key
Supplier Key
Stock Item Key
WWI Purchase Order ID
Ordered Outers
Ordered Quantity
Received Outers
Package
Is Order Finalized
Lineage Key
Stock Holding
Stock Holding Key
Stock Item Key
Quantity On Hand
Bin Location
Last Stocktake Quantity
Last Cost Price
Reorder Level
Target Stock Level
Lineage Key

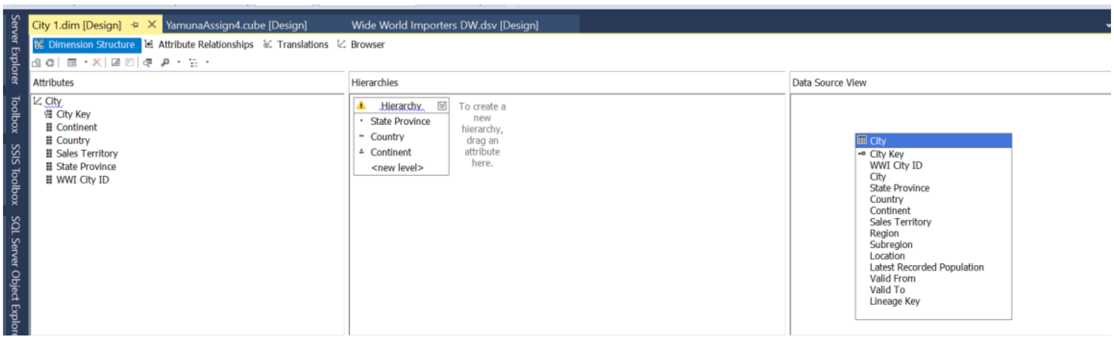

Transaction
Transaction Key
Date Key
Customer Key
Bill To Customer Key
Supplier Key
Transaction Type Key
Payment Method Key
WWI Customer Transaction ID
WWI Supplier Transaction ID
WWI Invoice ID
WWI Purchase Order ID
Supplier Invoice Number
Total Excluding Tax
Tax Amount
Transaction Type
Transaction Type Key
WWI Transaction Type ID
Transaction Type
Valid From
Valid To
Lineage Key
Payment Method
Payment Method Key
WWI Payment Method ID
Payment Method
Valid From
Valid To
Lineage Key
Supplier
Supplier Key
WWI Supplier ID
Supplier
Category
Primary Contact
Supplier Reference
Payment Days
Postal Code
Valid From
Valid To
Lineage Key
Purchase
Purchase Key
Date Key
Supplier Key
Stock Item Key
WWI Purchase Order ID
Ordered Outers
Ordered Quantity
Received Outers
Package
Is Order Finalized
Lineage Key

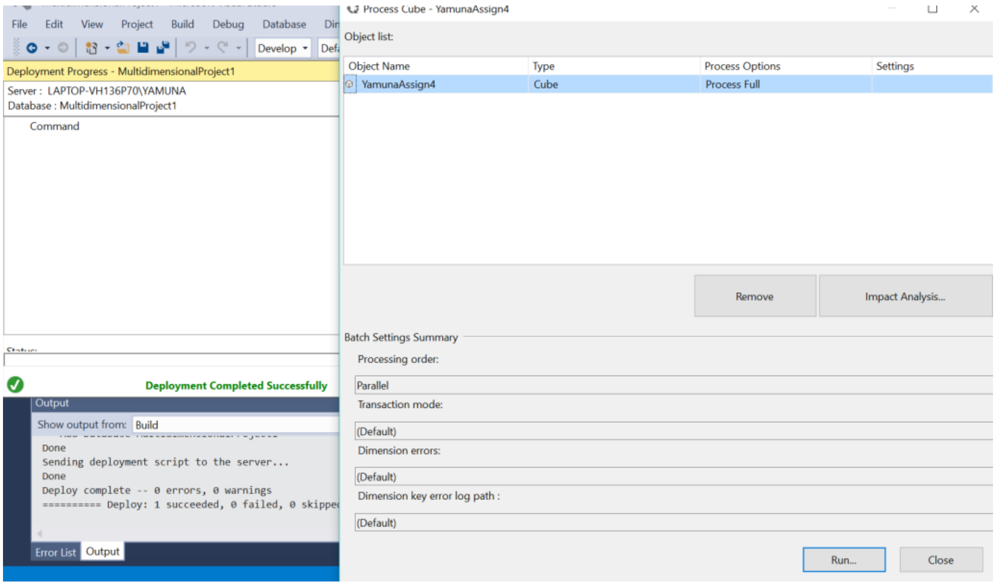

Customer
Movement
Transaction
Transaction Type
Payment Method
Stock Item
Supplier
Purchase
Stock Holding
Sale
City
Employee
Order
Customer


Date
Sale
City
Employee

Modifying Dimensions & Hierarchies







Server : LAPTOP-VH136P70\YAMUNA
Database : MultidimensionalProject1

Deployment Completed Successfully

```
Done
Sending deployment script to the server...
Done
Deploy complete -- 0 errors, 0 warnings
========= Deploy: 1 succeeded, 0 failed, 0 skippe
```

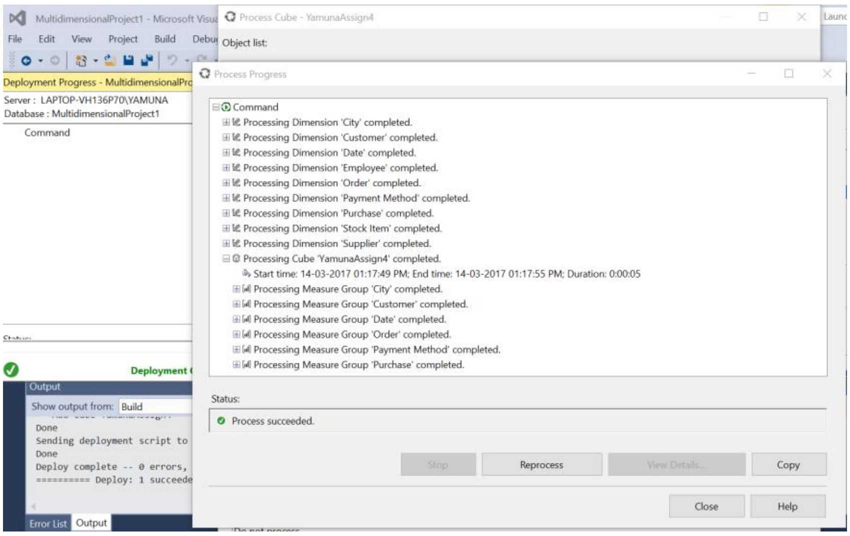
Object Name	Type	Process Options	Settings
YamunaAssign4	Cube	Process Full	

Processing order:
Parallel

Transaction mode:
(Default)

Dimension errors:
(Default)

Dimension key error log path :
(Default)


MultidimensionalProject1 - Microsoft Visua
File Edit View Project Build Debu
Process Cube - YamunaAssign4
Object list:
Process Progress
Deployment Progress - MultidimensionalPro
Stop
Reprocess
View Details...
Copy
Close
Help
Output
Show output from: Build
Error List
Output

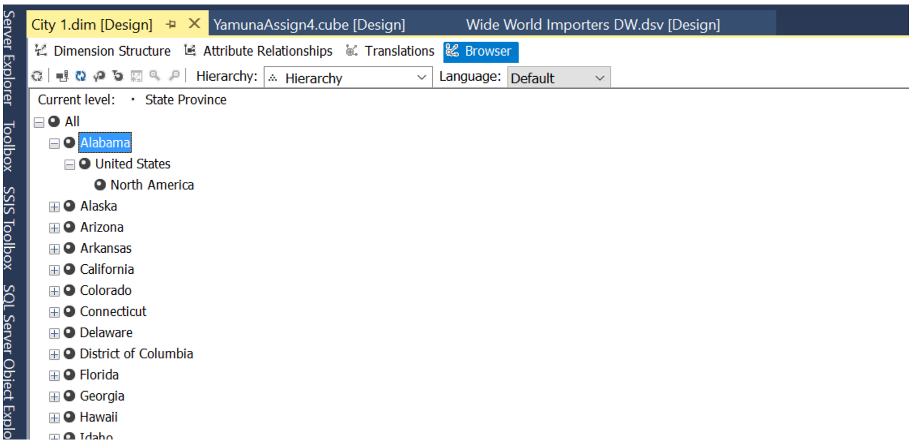
Server : LAPTOP-VH136P70\YAMUNA
Database : MultidimensionalProject1

```
Command
  Processing Dimension 'City' completed.
  Processing Dimension 'Customer' completed.
  Processing Dimension 'Date' completed.
  Processing Dimension 'Employee' completed.
  Processing Dimension 'Order' completed.
  Processing Dimension 'Payment Method' completed.
  Processing Dimension 'Purchase' completed.
  Processing Dimension 'Stock Item' completed.
  Processing Dimension 'Supplier' completed.
  Processing Cube 'YamunaAssign4' completed.
      Start time: 14-03-2017 01:17:49 PM; End time: 14-03-2017 01:17:55 PM; Duration: 0:00:05
    Processing Measure Group 'City' completed.
    Processing Measure Group 'Customer' completed.
    Processing Measure Group 'Date' completed.
    Processing Measure Group 'Order' completed.
    Processing Measure Group 'Payment Method' completed.
    Processing Measure Group 'Purchase' completed.
```

Status:

Process succeeded.

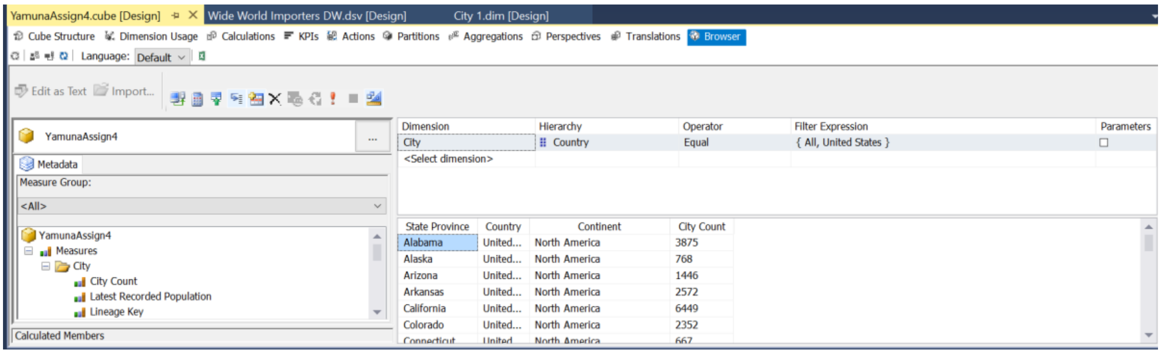
```
Done
Sending deployment script to
Done
Deploy complete -- 0 errors,
========== Deploy: 1 succeede
```



Hierarchy: Hierarchy Language: Default

Current level: • State Province

```
⊟ All
  ⊟ Alabama
    ⊟ United States
        North America
  ⊞ Alaska
  ⊞ Arizona
  ⊞ Arkansas
  ⊞ California
  ⊞ Colorado
  ⊞ Connecticut
  ⊞ Delaware
  ⊞ District of Columbia
  ⊞ Florida
  ⊞ Georgia
  ⊞ Hawaii
  ⊞ Idaho
```



Language: Default

Measure Group: <All>

```
YamunaAssign4
  ⊟ Measures
    ⊟ City
        City Count
        Latest Recorded Population
        Lineage Key
```

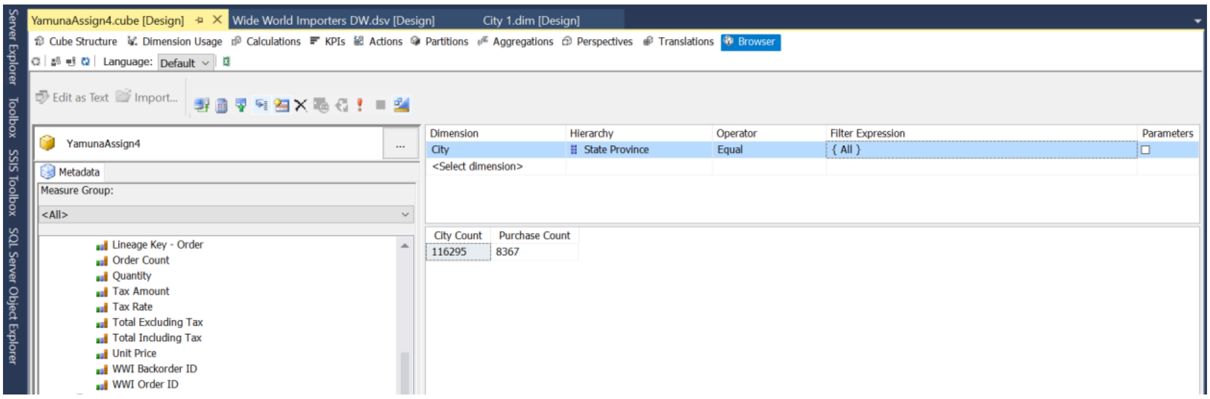
Dimension	Hierarchy	Operator	Filter Expression	Parameters
City	Country	Equal	{ All, United States }	□
<Select dimension>				

State Province	Country	Continent	City Count
Alabama	United...	North America	3875
Alaska	United...	North America	768
Arizona	United...	North America	1446
Arkansas	United...	North America	2572
California	United...	North America	6449
Colorado	United...	North America	2352
Connecticut	United...	North America	667

6. OLAP Queries

Roll-up

Find out the total number of cities and purchases in all geographic locations

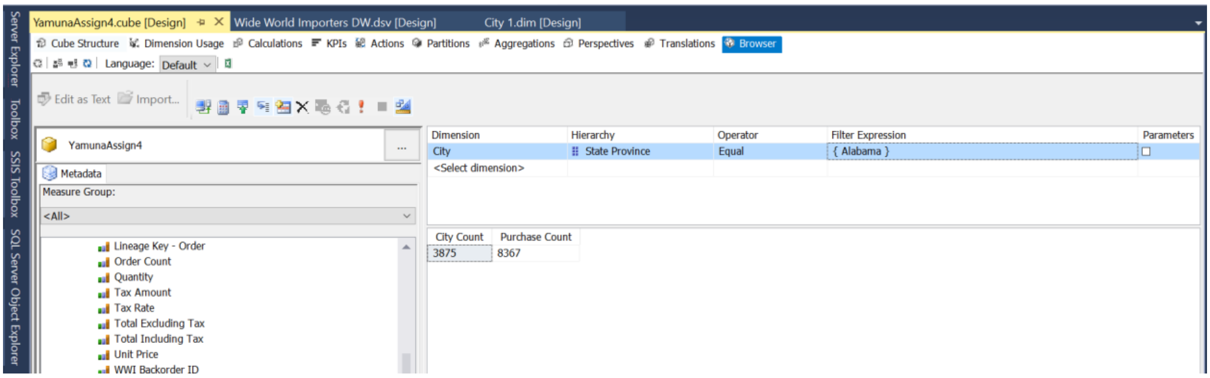


Dimension	Hierarchy	Operator	Filter Expression	Parameters
City	State Province	Equal	{ All }	□
<Select dimension>				

City Count	Purchase Count
116295	8367

Drill-down

Find the total number of cities and purchases in State Alabama

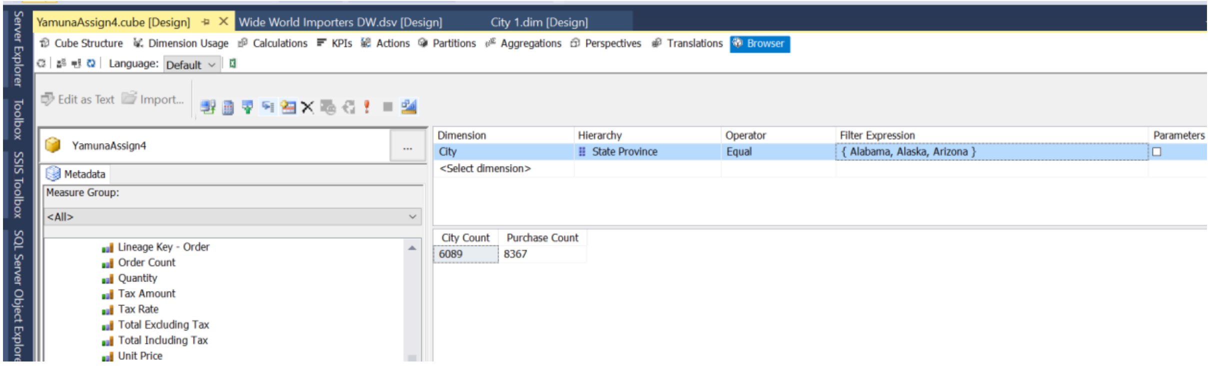


Dimension	Hierarchy	Operator	Filter Expression	Parameters
City	State Province	Equal	{ Alabama }	□
<Select dimension>				

City Count	Purchase Count
3875	8367

Slice

Find the total number of cities and purchases in States “Alabama, Alaska & Arizona”

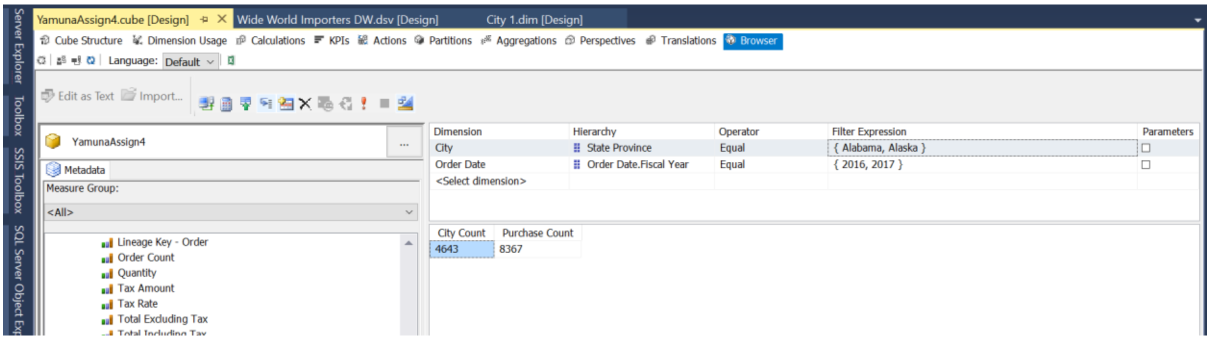


Dimension	Hierarchy	Operator	Filter Expression	Parameters
City	State Province	Equal	{ Alabama, Alaska, Arizona }	□
<Select dimension>				

City Count	Purchase Count
6089	8367

Dice

Find the total number of cities and purchases in States "Alabama & Alaska" for the year "2016 & 2017".



Dimension	Hierarchy	Operator	Filter Expression	Parameters
City	State Province	Equal	{ Alabama, Alaska }	□
Order Date	Order Date.Fiscal Year	Equal	{ 2016, 2017 }	□
<Select dimension>				

City Count	Purchase Count
4643	8367

7. Summary

SSIS is a great tool to perform ETL and Data Warehouse operations. It has a user-friendly interface and has options to perform OLAP operations as well. However, this tool is light weight and is not advanced enough like other ETL tools in the market such as Informatica & IBM InfoSphere DataStage. These tools has advanced stages and options to perform ETL tasks for large datasets.