

**Department of Management Science and Technology  
Athens University of Economics and Business  
MSc Business Analytics  
Data Warehousing, Visualization (Assignment#2)**



## **New York City Bike Share**

**Maria Zafeiropoulou (AM: P2822113)  
&  
Marianna Konstantopoulou (AM: P2822122)**

**Athens, Greece  
12/12/2021**

## Contents

<b>Data Warehousing, Visualization (Assignment#2)</b>	1
1.    Business goals and description of the dataset	3
2.    Data cleaning & Relational Schema	4
3.    Importing data and building the database	7
4.    Deploying the SSIS Package	16
5.    Defining the Data Cube	18
6.    Exporting Cube Data to Excel	23
7.    Reporting in SSRS	24
8.    Power Bi	27
9.    Power Point Presentation	30

## 1. Business goals and description of the dataset

Bicycle sharing is a service that allows users to rent a bike for a short-term use. These systems operate by allowing users to return the bike at any of the station's locations. Since these systems are very popular in various cities around the world, they have been referred to as the "next generation of transportation". Over 100 cities in the world have already deployed or are planning to implement bike sharing programs. These programs provide a variety of transportation options and are beneficial for the environment. In this report we will particularly study New York's City Bike Share system.

The City Bike Company based in New York approached us and asked the following questions: "Which times are the busiest for the system? What's the most common ride duration? Which stations are the most popular? What is the age and gender of the riders?". Their goal is to find answers to these questions through our analysis in order to decide whether they can add more bikes in certain bike stations or at particular time periods. They are also interested in the age and gender of their consumers so they can choose the appropriate marketing plan for their system or even try to attract a different age group with a new approach.

The New York City Bike Share makes regular open data releases and we were able to find this data at Kaggle (<https://www.kaggle.com/akkithetechie/new-york-city-bike-share-dataset>). The dataset contains 735.502 anonymised trips information made from January 2015 to June 2017 and it contains the following information for each ride:

- Trip Duration (seconds)
- Start Date
- Start Time
- Stop Date
- Stop Time
- Start Station Name
- End Station Name
- Station ID
- Station Latitude
- Station Longitude
- Bike ID
- User Type (Customer = 24-hour pass or single ride user; Subscriber = Annual Member)
- Gender (0=unknown; 1=male; 2=female)
- Year of Birth

The raw dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Trip Duration	Start Time	Stop Time	Start Station ID	Start Station Name	Start Station Latitude	Start Station Longitude	End Station ID	End Station Name	Station Latitude	Station Longitude	Bike ID	User Type	Birth Year	Gender	Trip Duration_in_min	
2	0	376	01/10/2015 00:16	01/10/2015 00:22	3212	Christ Hospital	40.73478582	-74.05044364	3207	Oakland Ave	40.7376	-74.0525	24470	Subscriber	1960	1	6
3	1	739	01/10/2015 00:27	01/10/2015 00:39	3207	Oakland Ave	40.7376037	-74.0524783	3212	Christ Hospital	40.73479	-74.0504	24481	Subscriber	1960	1	12
4	2	2714	01/10/2015 00:32	01/10/2015 01:18	3193	Lincoln Park	40.7246051	-74.07840595	3193	Lincoln Park	40.72461	-74.0784	24628	Subscriber	1983	1	45
5	3	275	01/10/2015 00:34	01/10/2015 00:39	3199	Newport Pkwy	40.7287448	-74.0321082	3187	Warren St	40.72112	-74.0381	24613	Subscriber	1975	1	5
6	4	561	01/10/2015 00:40	01/10/2015 00:49	3183	Exchange Place	40.7162469	-74.0334588	3192	Liberty Light Rail	40.71124	-74.0557	24668	Customer	1984	0	9
7	5	365	01/10/2015 00:41	01/10/2015 00:47	3198	Heights Elevator	40.74671595	-74.0404433	3215	Central Ave	40.74673	-74.0493	24644	Customer	1984	0	6
8	6	139	01/10/2015 00:43	01/10/2015 00:46	3206	Hilltop	40.7311689	-74.0575736	3195	Sip Ave	40.73074	-74.0638	24482	Subscriber	1988	1	2
9	7	1299	01/10/2015 00:51	01/10/2015 01:31	3197	North St	40.752559	-74.044725	3215	Central Ave	40.74673	-74.0493	24550	Customer	1984	0	22
10	8	647	01/10/2015 02:01	01/10/2015 02:12	3213	Van Vorst Park	40.71848892	-74.04772663	3190	Garfield Ave Statio	40.71047	-74.07	24650	Subscriber	1988	1	11
11	9	233	01/10/2015 04:43	01/10/2015 04:47	3194	McGinley Square	40.72533993	-74.06762213	3195	Sip Ave	40.73074	-74.0638	24584	Subscriber	1978	2	4
12	10	352	01/10/2015 04:43	01/10/2015 04:49	3215	Central Ave	40.7467299	-74.049250	3197	North St	40.75256	-74.0447	24550	Customer	1984	0	6
13	11	1013	01/10/2015 05:31	01/10/2015 05:48	3193	Lincoln Park	40.7246051	-74.07840595	3193	Lincoln Park	40.72461	-74.0784	24471	Subscriber	1980	1	17
14	12	592	01/10/2015 05:41	01/10/2015 05:51	3196	Riverview Park	40.7443187	-74.043990	3202	Newport PATH	40.72722	-74.0338	24406	Subscriber	1983	2	10
15	13	308	01/10/2015 05:44	01/10/2015 05:49	3209	Brwick St	40.7241765	-74.0506564	3186	Grove St PATH	40.71959	-74.0431	24723	Subscriber	1980	1	5
16	14	667	01/10/2015 05:52	01/10/2015 06:03	3207	Oakland Ave	40.7376037	-74.0524783	3185	City Hall	40.71773	-74.0438	24516	Subscriber	1984	2	11
17	15	312	01/10/2015 06:25	01/10/2015 06:31	3214	Essex Light Rail	40.7127742	-74.0364857	3186	Grove St PATH	40.71959	-74.0431	19086	Subscriber	1984	1	5
18	16	588	01/10/2015 06:26	01/10/2015 06:36	3210	Pershing Field	40.74267714	-74.05178863	3195	Sip Ave	40.73074	-74.0638	24561	Subscriber	1972	1	10
19	17	615	01/10/2015 06:26	01/10/2015 06:37	3209	Brunswick St	40.7741765	-74.0506564	3183	Exchange Place	40.71625	-74.0335	24515	Subscriber	1970	1	10
20	18	1009	01/10/2015 06:28	01/10/2015 06:44	3196	Riverview Park	40.7443187	-74.0439909	3183	Exchange Place	40.71625	-74.0335	24639	Subscriber	1976	1	17
21	19	1113	01/10/2015 06:29	01/10/2015 06:48	3217	Bayside Park	40.69865054	-74.08207968	3183	Exchange Place	40.71625	-74.0335	24718	Subscriber	1977	1	19
22	20	492	01/10/2015 06:33	01/10/2015 06:41	3207	Oakland Ave	40.7376037	-74.0524783	3195	Sip Ave	40.73074	-74.0638	24422	Subscriber	1983	2	8
23	21	331	01/10/2015 06:49	01/10/2015 06:54	3207	Oakland Ave	40.7376037	-74.0524783	3195	Sip Ave	40.73074	-74.0638	24452	Subscriber	1977	1	6
24	22	18715	01/10/2015 06:58	01/10/2015 12:10	3185	City Hall	40.7177325	-74.0438485	3184	Paulus Hook	40.71415	-74.0338	24516	Subscriber	1980	1	312
25	23	213	01/10/2015 06:59	01/10/2015 07:03	3186	Grove St PATH	40.71958612	-74.04311746	3213	Van Vorst Park	40.71849	-74.0477	19086	Subscriber	1970	2	4
26	24	775	01/10/2015 07:00	01/10/2015 07:13	3210	Pershing Field	40.74267714	-74.05178863	3203	Hamilton Park	40.7276	-74.0442	24401	Subscriber	1964	2	13
27	25	306	01/10/2015 07:00	01/10/2015 07:05	3209	Brunswick St	40.7241765	-74.0506564	3186	Grove St PATH	40.71959	-74.0431	24496	Subscriber	1989	1	5
28	26	182	01/10/2015 07:03	01/10/2015 07:06	3214	Essex Light Rail	40.7127742	-74.0364857	3183	Exchange Place	40.71625	-74.0335	24391	Subscriber	1984	2	3

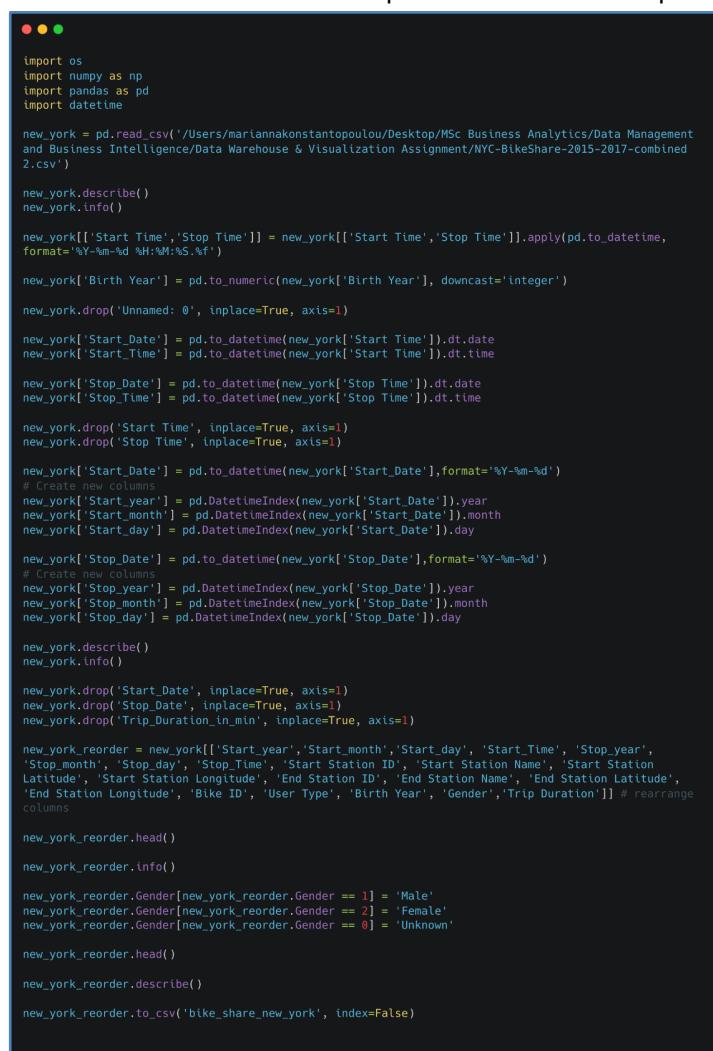
## 2. Data cleaning & Relational Schema

The first problem that we had to manage was to process the CSV file and manipulate the data in order to make it organised and easier to read. For the data cleaning and data manipulation part we used the programming language Python. These are the steps we followed for this process:

- We imported libraries such as os, numpy, pandas and datetime that would be useful for the next steps
- Read the CSV file using the panda library and we used the commands describe() and info() to gain more insight about the names and the type of the columns

After figuring out the way we want this dataset to look like in order to build the star schema later on we performed the following actions:

- We changed the Start Time and Stop Time type to datetime format and then we split date (year, month, day) and time to different columns
- The Birth Year column was converted to be numeric and we added subsequently the gender names (unknown, male and female) in the Gender column. Additionally, we separated columns (using Excel this time) for Hours, Minutes and Seconds we extracted from Start\_Time and Stop\_Time columns
- The first column with no name was dropped and we also dropped the column Trip Duration in minutes as it portrayed the same information with Trip Duration in seconds column
- Last step was to reorder the columns in the preferred order and export the results to CSV



```
import os
import numpy as np
import pandas as pd
import datetime

new_york = pd.read_csv('/Users/mariannakonstantopoulou/Desktop/MSc Business Analytics/Data Management and Business Intelligence/Data Warehouse & Visualization Assignment/NYC-BikeShare-2015-2017-combined2.csv')

new_york.describe()
new_york.info()

new_york[['Start Time','Stop Time']] = new_york[['Start Time','Stop Time']].apply(pd.to_datetime,
format='%Y-%m-%d %H:%M:%S.%f')

new_york['Birth Year'] = pd.to_numeric(new_york['Birth Year'], downcast='integer')

new_york.drop('Unnamed: 0', inplace=True, axis=1)

new_york['Start Date'] = pd.to_datetime(new_york['Start Time']).dt.date
new_york['Start Time'] = pd.to_datetime(new_york['Start Time']).dt.time

new_york['Stop Date'] = pd.to_datetime(new_york['Stop Time']).dt.date
new_york['Stop Time'] = pd.to_datetime(new_york['Stop Time']).dt.time

new_york.drop('Start Time', inplace=True, axis=1)
new_york.drop('Stop Time', inplace=True, axis=1)

new_york['Start Date'] = pd.to_datetime(new_york['Start Date'],format='%Y-%m-%d')
# Create new columns
new_york['Start_year'] = pd.DatetimeIndex(new_york['Start Date']).year
new_york['Start_month'] = pd.DatetimeIndex(new_york['Start Date']).month
new_york['Start_day'] = pd.DatetimeIndex(new_york['Start Date']).day

new_york['Stop Date'] = pd.to_datetime(new_york['Stop Date'],format='%Y-%m-%d')
# Create new columns
new_york['Stop_year'] = pd.DatetimeIndex(new_york['Stop Date']).year
new_york['Stop_month'] = pd.DatetimeIndex(new_york['Stop Date']).month
new_york['Stop_day'] = pd.DatetimeIndex(new_york['Stop Date']).day

new_york.describe()
new_york.info()

new_york.drop('Start Date', inplace=True, axis=1)
new_york.drop('Stop Date', inplace=True, axis=1)
new_york.drop('Trip Duration_in_min', inplace=True, axis=1)

new_york_reorder = new_york[['Start_year','Start_month','Start_day','Start Time','Stop year',
'Stop month','Stop day','Stop Time','Start Station ID','Start Station Name','Start Station
Latitude','Start Station Longitude','End Station ID','End Station Name','End Station Latitude',
'End Station Longitude','Bike ID','User Type','Birth Year','Gender','Trip Duration']] # rearrange
columns

new_york_reorder.head()
new_york_reorder.info()

new_york_reorder.Gender[new_york_reorder.Gender == 1] = 'Male'
new_york_reorder.Gender[new_york_reorder.Gender == 2] = 'Female'
new_york_reorder.Gender[new_york_reorder.Gender == 0] = 'Unknown'

new_york_reorder.head()
new_york_reorder.describe()

new_york_reorder.to_csv('bike_share_new_york', index=False)
```

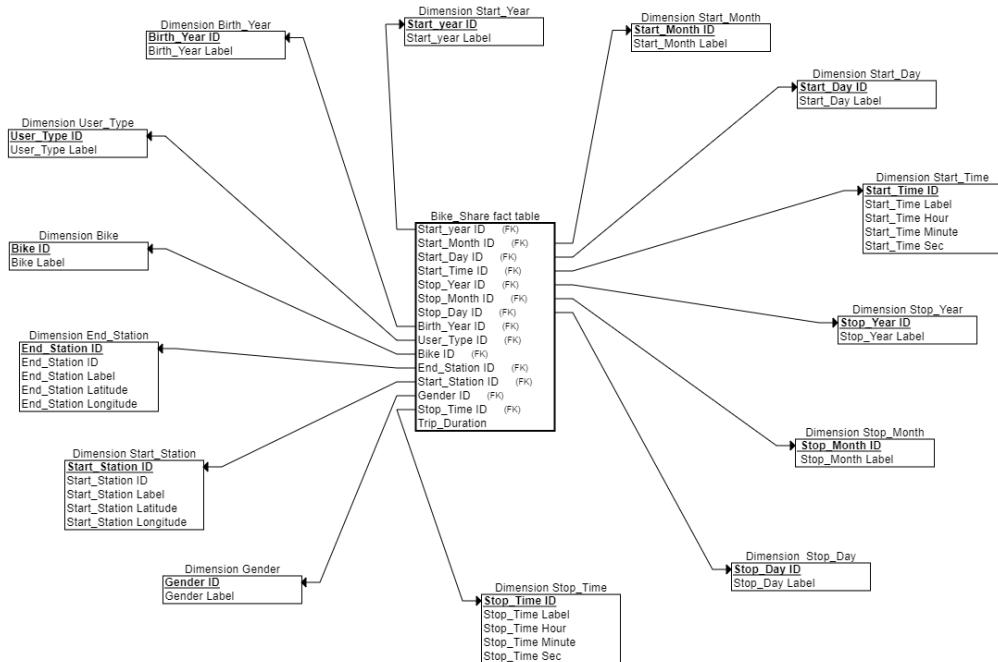
The final structure of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 735502 entries, 0 to 735501
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Start_year      735502 non-null   int64  
 1   Start_month     735502 non-null   int64  
 2   Start_day       735502 non-null   int64  
 3   Start_Time      735502 non-null   object  
 4   Stop_year       735502 non-null   int64  
 5   Stop_month      735502 non-null   int64  
 6   Stop_day        735502 non-null   int64  
 7   Stop_Time       735502 non-null   object  
 8   Start Station ID 735502 non-null   int64  
 9   Start Station Name 735502 non-null   object  
 10  Start Station Latitude 735502 non-null   float64 
 11  Start Station Longitude 735502 non-null   float64 
 12  End Station ID   735502 non-null   int64  
 13  End Station Name 735502 non-null   object  
 14  End Station Latitude 735502 non-null   float64 
 15  End Station Longitude 735502 non-null   float64 
 16  Bike ID          735502 non-null   int64  
 17  User Type        735502 non-null   object  
 18  Birth Year       735502 non-null   int16  
 19  Gender           735502 non-null   object  
 20  Trip Duration    735502 non-null   int64  
dtypes: float64(4), int16(1), int64(10), object(6)
memory usage: 113.6+ MB
```

(The Hour, Minutes and Second separate columns were added later in Excel, using the hour(), minute(), second() functions)

Start_year	Start_month	Start_day	Start_Time	Stop_year	Stop_month	Stop_day	Stop_Time	ID	Start Station Name	Start Station Latitude	Start Station Longitude	End Station ID	End Station Name
2015	10	1	00:16:26	2015	10	1	00:22:42	3212	Christ Hospital	40.73478582	-74.0504436	3207	Oakland Ave
2015	10	1	00:27:12	2015	10	1	00:39:32	3207	Oakland Ave	40.7376037	-74.0524783	3212	Christ Hospital
2015	10	1	00:32:46	2015	10	1	01:18:01	3193	Lincoln Park	40.7246051	-74.078406	3193	Lincoln Park
2015	10	1	00:34:31	2015	10	1	00:39:06	3199	Newport Pkwy	40.7287448	-74.0321082	3187	Warren St
2015	10	1	00:40:12	2015	10	1	00:49:33	3183	Exchange Place	40.7162469	-74.0334588	3192	Liberty Light Rail
End Station Latitude	End Station Longitude	Bike ID	User Type	Birth Year	Gender	Trip Duration	Start_Hour	Start_Minute	Start_Sec	Stop_Hour	Stop_Minute	Stop_Sec	
40.7376037	-74.0524783	24470	Subscriber	1960	Male	376	0	16	26	0	22	42	
40.73478582	-74.0504436	24481	Subscriber	1960	Male	739	0	27	12	0	39	32	
40.7246051	-74.078406	24628	Subscriber	1983	Male	2714	0	32	46	1	18	1	
40.7211236	-74.038051	24613	Subscriber	1975	Male	275	0	34	31	0	39	6	
40.7112423	-74.0557013	24668	Customer	1984	Unknown	561	0	40	12	0	49	33	

For the relational design we are following the star schema structure:

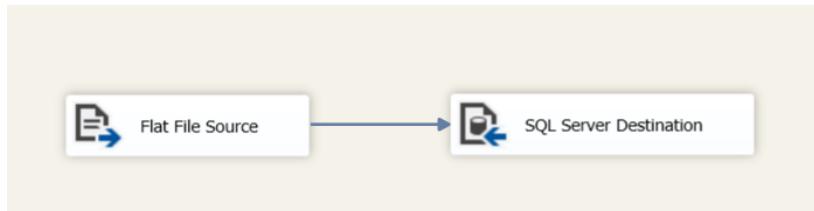


In the above demonstration, **Bike\_Share** is a fact table having a measure attribute **Trip\_Duration** and attributes i.e. (Start\_Year ID, Start\_Month ID, Start\_Day ID, Start\_Time ID, Stop\_Year ID, Stop\_Month ID, Stop\_Day ID, Start\_Station ID, End\_Station ID, Birth\_Year ID, Gender ID, Bike ID, User\_Type ID) which references to the dimension tables.

1. Start\_Year dimension table contains the attributes: Start\_Year ID, Start\_Year Label
2. Start\_Month dimension table contains the attributes: Start\_Month ID, Start\_Month Label
3. Start\_Day dimension table contains the attributes: Start\_Day ID, Start\_Day Label
4. Start\_Time dimension table contains the attributes: Start\_Time ID, Start\_Time Label, Start\_Time Hour, Start\_Time Minute, Start\_Time Sec
5. Stop\_Year dimension table contains the attributes: Stop\_Year ID, Stop\_Year Label
6. Stop\_Month dimension table contains the attributes: Stop\_Month ID, Stop\_Month Label
7. Stop\_Day dimension table contains the attributes: Stop\_Day ID, Stop\_Day Label
8. Stop\_Time dimension table contains the attributes: Stop\_Time ID, Stop\_Time Label, Stop\_Time Hour, Stop\_Time Minute, Stop\_Time Sec
9. Start\_Station dimension table contains the attributes: Start\_Station ID, Start\_Station ID Label, Start\_Station Name, Start\_Station\_Latitude, Start\_Station Longitude
10. End\_Station dimension table contains the attributes: End\_Station ID, End\_Station ID Label, End\_Station Name, End\_Station\_Latitude, End\_Station Longitude
11. Birth\_Year dimension table contains the attributes: Birth\_Year ID, Birth\_Year Label
12. User\_Type dimension table contains the attributes: User\_Type ID, User\_Type Label
13. Gender dimension table contains the attributes: Gender ID, Gender Label
14. Bike dimension table contains the attributes: Bike ID, Bike Label

### 3. Importing data and building the database

After creating the database “Bike\_Share” in the SQL server we made a Data Flow in Visual Studio so that we could import the CSV in the SQL Server.



Since the CSV was connected to the server we created the staging table that contained all the columns from the CSV and we made sure that the type of each column was the correct one.

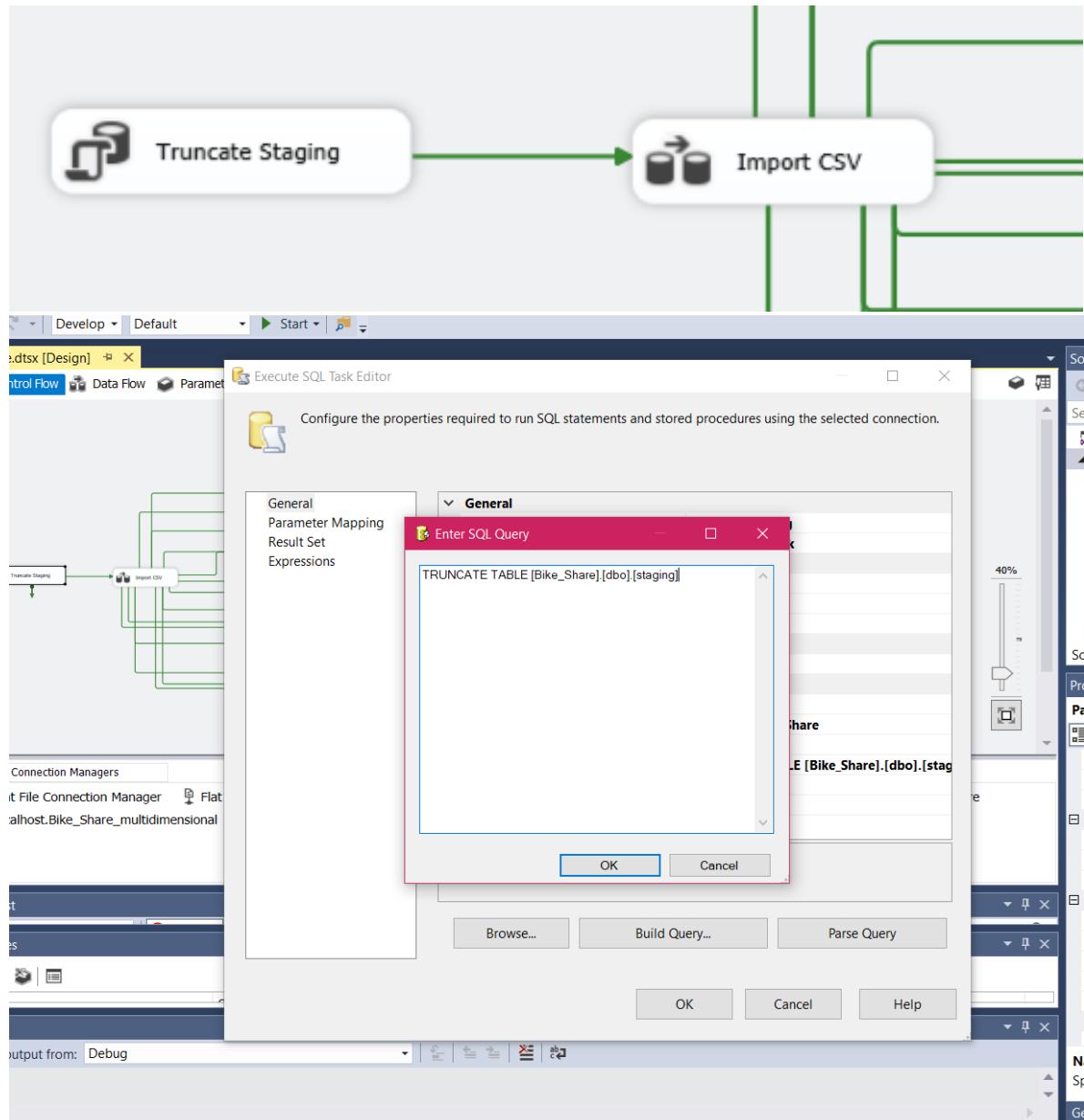
The screenshot shows the 'SQL Destination Editor' window within the Visual Studio Data Flow Task designer. The main area displays the 'Create Table' script for the 'SQL Server Destination':

```
CREATE TABLE [SQL Server Destination]
(
    [Start_year] bigint,
    [Start_month] bigint,
    [Start_day] bigint,
    [Start_Time] time(0),
    [Stop_year] bigint,
    [Stop_month] bigint,
    [Stop_day] bigint,
    [Stop_Time] time(0),
    [Start Station ID] bigint,
    [Start Station Name] varchar(200),
    [Start Station Latitude] float,
    [Start Station Longitude] float,
    [End Station ID] bigint,
    [End Station Name] varchar(200),
    [End Station Latitude] float,
    [End Station Longitude] float,
    [Bike ID] bigint,
    [User Type] varchar(200),
    [Birth Year] bigint,
    [Gender] varchar(200),
    [Trip Duration] bigint,
    [Start_Hour] bigint,
    [Start_Minute] bigint,
    [Start_Sec] bigint,
    [Stop_Hour] bigint,
    [Stop_Minute] bigint,
    [Stop_Sec] bigint
)
```

The 'Control Flow' tab is selected in the top navigation bar. On the left, the 'Connection Managers' section shows a 'Flat File Connection Manager' named 'localhost.Bike\_Share\_multidimensional'. The 'Mappings' and 'Advanced' tabs are also visible. The bottom pane shows the 'Results' and 'Messages' tabs, with the 'Results' tab displaying a preview of the imported data:

	Start_year	Start_month	Start_day	Start_Time	Stop_year	Stop_month	Stop_day	Stop_Time	Start Station ID	Start Station Name	Start Station Latitude	Start Station Longitude	End Station ID	End Station Name	End Station Latitude	End Station Longitude
1	2015	10	1	00:16:26	2015	10	1	00:22:42	3212	Christ Hospital	40.73478582	-74.05044364	3207	Oakland Ave	40.7376037	-74.052471
2	2015	10	1	00:27:12	2015	10	1	00:39:32	3207	Oakland Ave	40.7376037	-74.0524783	3212	Christ Hospital	40.73478582	-74.05044
3	2015	10	1	00:32:46	2015	10	1	01:18:01	3193	Lincoln Park	40.7246051	-74.07840595	3193	Lincoln Park	40.7246051	-74.078401
4	2015	10	1	00:34:31	2015	10	1	00:39:06	3199	Newport Pkwy	40.7287448	-74.0321082	3187	Warren St	40.7211236	-74.038051
5	2015	10	1	00:40:12	2015	10	1	00:49:33	3183	Exchange Place	40.7162469	-74.034588	3192	Liberty Light Rail	40.7112423	-74.05570
6	2015	10	1	00:41:46	2015	10	1	00:47:51	3198	Heights Elevator	40.7487195	-74.0404433	3215	Central Ave	40.7467299	-74.04925
7	2015	10	1	00:43:44	2015	10	1	00:46:03	3206	Hilltop	40.731168	-74.0575736	3195	Sip Ave	40.73074263	-74.06370
8	2015	10	1	01:10:10	2015	10	1	01:31:50	3197	North St	40.752559	-74.044725	3215	Central Ave	40.7467299	-74.04925
9	2015	10	1	02:01:36	2015	10	1	02:12:24	3213	Van Vorst Park	40.71848892	-74.04772663	3190	Garfield Ave Station	40.71046702	-74.070031
10	2015	10	1	04:43:33	2015	10	1	04:47:27	3194	McGinley Square	40.72533993	-74.06762213	3195	Sip Ave	40.73074263	-74.06378
11	2015	10	1	04:43:42	2015	10	1	04:49:34	3215	Central Ave	40.7467299	-74.0492509	3197	North St	40.752559	-74.04472
12	2015	10	1	05:31:13	2015	10	1	05:48:06	3193	Lincoln Park	40.7246051	-74.07840595	3193	Lincoln Park	40.7246051	-74.078401
13	2015	10	1	05:41:17	2015	10	1	05:51:10	3196	Riverview Park	40.7441317	-74.0439909	3202	Newport PATH	40.7272235	-74.03375
14	2015	10	1	05:44:29	2015	10	1	05:49:38	3209	Brunswick St	40.7241765	-74.0506564	3186	Grove St PATH	40.71958612	-74.04311
15	2015	10	1	05:52:21	2015	10	1	06:03:29	3207	Oakland Ave	40.7376037	-74.0524783	3185	City Hall	40.7177325	-74.04384
16	2015	10	1	06:25:54	2015	10	1	06:31:07	3214	Esex Light Rail	40.7127742	-74.0364857	3186	Grove St PATH	40.71958612	-74.04311
17	2015	10	1	06:26:37	2015	10	1	06:36:25	3210	Pershing Field	40.74267714	-74.05178863	3195	Sip Ave	40.73074263	-74.06378
18	2015	10	1	06:26:48	2015	10	1	06:37:04	3209	Brunswick St	40.7241765	-74.0506564	3183	Exchange Place	40.7162469	-74.03345
19	2015	10	1	06:28:03	2015	10	1	06:44:53	3196	Riverview Park	40.7441317	-74.0439909	3183	Exchange Place	40.7162469	-74.03345
20	2015	10	1	06:29:54	2015	10	1	06:48:28	3217	Bayside Park	40.69865054	-74.08207968	3183	Exchange Place	40.7162469	-74.03345
21	2015	10	1	06:33:19	2015	10	1	06:41:32	3207	Oakland Ave	40.7376037	-74.0524783	3195	Sip Ave	40.73074263	-74.06378

The next step was to create a new SQL Execute Task in Control Flow that will run before the Import CSV stage called “Truncate Staging”. This task not only truncates the staging table every time the Control Flow starts but it also resets any unique ids that we might have.



Following that we started creating the dimensions. This is the procedure we followed in order to create the dimensions (we will provide an example and continue with the output for the rest of our dimensions)

### **Creation of Dimension Start\_Year**

**Step 1.** Design the dimension by adding the id and label (their data type as well) and choosing to not allow nulls

**Step 2.** Make the id auto-increment and also making it the primary key

Column Name	Data Type	Allow Nulls
id_Start_Year	int	<input type="checkbox"/>
label_Start_year	int	<input type="checkbox"/> <input type="checkbox"/>

Column Properties

(General) (Name) id\_Start\_Year  
Allow Nulls No  
Data Type int  
Default Value or Binding  
Table Designer Collation <database default>  
Computed Column Specification  
Condensed Data Type int  
Description Deterministic Yes  
DTS-published No  
Full-text Specification No  
Has Non-SQL Server Subscriber No  
Identity Specification Is Identity Yes  
Identity Increment 1  
Identity Seed 1  
Indexable Yes  
Is Columnset No  
Is Sparse No  
Deterministic

Step 3. For the label we add a new index/key and choose to make it unique and ignore duplicates

Column Name	Data Type	Allow Nulls
id_Start_Year	int	<input type="checkbox"/>
label_Start_year	int	<input type="checkbox"/>

Indexes/Keys

Selected Primary/Unique Key or Index:  
IX\_Dim\_Start\_Year  
PK\_Dim\_Start\_Year

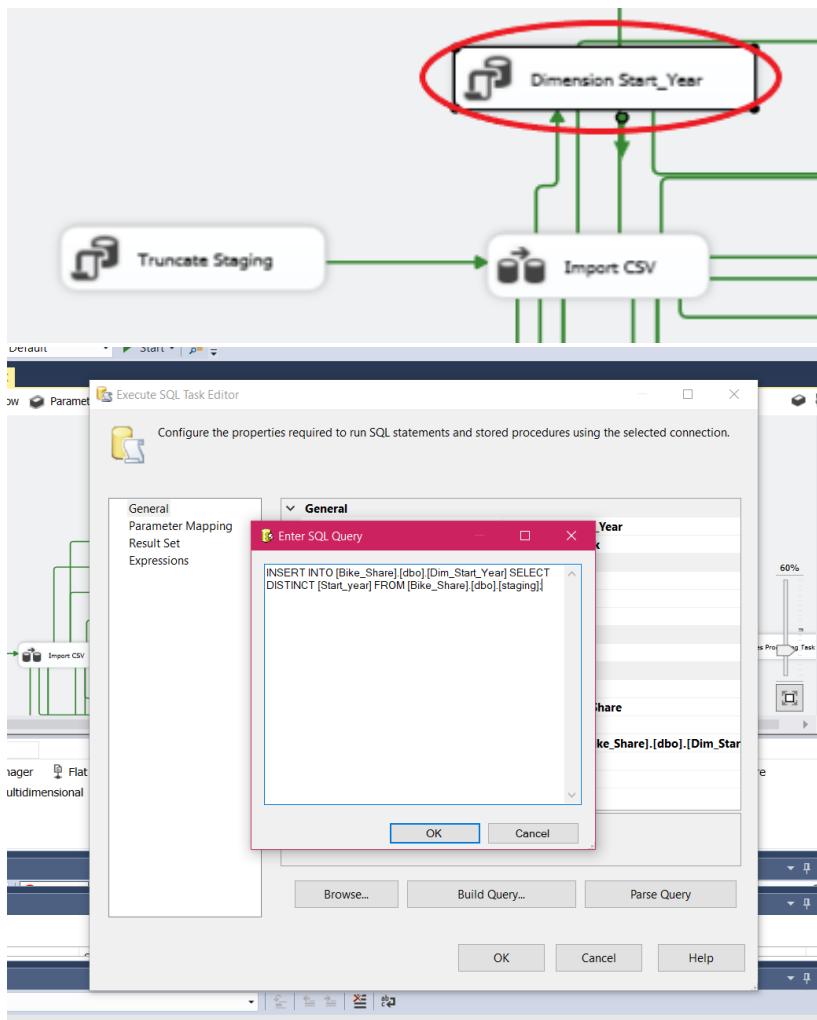
Editing properties for existing primary/unique key or index.

(General) Columns label\_Start\_year (ASC)  
Is Unique Yes  
Type Index  
Identity (Name) IX\_Dim\_Start\_Year  
Description  
Table Designer Create As Clustered No  
Data Space Specification PRIMARY  
Fill Specification  
Ignore Duplicate Keys Yes

Column Properties Add Delete Close

(General) (Name) id\_Start\_Year  
Allow Nulls No  
Data Type int

Step 4. The Dimension needs to be created in Visual Studio as well, after the Import CSV we add an “Execute SQL Task” with the name of the Dimension and add an SQL query to insert the values from staging



Step 5. We check the output to make sure everything is running smoothly

Results		Messages	
	id_Start_Year	label_Start_Year	
1	3	2015	
2	1	2016	
3	2	2017	

We followed the above steps for all of our dimensions and these are the output we received:

Results		Messages	
	id_Gender	label_Gender	
1	3	Female	
2	1	Male	
3	2	Unknown	

Results		Messages	
	id_Stop_Year	label_Stop_Year	
1	3	2015	
2	1	2016	
3	2	2017	

Results		Messages	
	id_User_Type	label_User_Type	
1	1	Customer	
2	2	Subscriber	

- Dimension Bike\_ID, Birth\_Year, Stop\_Month, Start\_Day, Stop\_Day, Start\_Month

	Results		Messages	
<b>Bike_ID</b>	<b>id_Bike_ID</b>	<b>label_Bike_ID</b>	<b>id_Birth_Year</b>	<b>label_Birth_Year</b>
1	162	14552	1	61
2	545	14632	2	49
3	163	14705	3	46
4	81	14717	4	25
5	242	14786	5	4
6	630	14872	6	33
7	544	14970	7	47
8	323	15084	8	34
9	36	15166	9	24
10	251	16021	10	5
11	369	16644	11	53
12	223	16754	12	51
13	126	17101	13	48
14	412	17103	14	42
15	496	17156	15	59
16	286	17653	16	10
17	353	17904	17	41
18	322	17936	18	15
19	628	17941	19	40
20	497	18062	20	3
21	673	18245	21	37
22	527	18268	22	54
23	438	18384	23	12
24	582	18402	24	58
25	674	18470	25	56
26	281	18605	26	60
27	287	18648	27	22
28	668	18772	28	30
29	333	18777	29	18
30	677	18812	30	14
31	539	19086	31	29
32	121	19276	32	45

	Results		Messages	
<b>Start_Day</b>	<b>id_Start_Day</b>	<b>label_Start_Day</b>	<b>id_Stop_Day</b>	<b>label_Stop_Day</b>
1	24	1	1	7
2	3	2	2	21
3	26	3	3	10
4	27	4	4	11
5	29	5	5	14
6	6	6	6	28
7	8	7	7	31
8	14	8	8	23
9	30	9	9	15
10	9	10	10	29
11	21	11	11	6
12	10	12	12	20
13	19	13	13	3
14	16	14	14	17
15	1	15	15	4
16	31	16	16	16
17	13	17	17	24
18	17	18	18	1
19	18	19	19	2
20	7	20	20	30
21	4	21	21	26
22	11	22	22	19
23	28	23	23	13
24	20	24	24	5
25	15	25	25	25
26	22	26	26	9
27	2	27	27	22
28	5	28	28	27
29	12	29	29	18
30	25	30	30	12
31	23	31	31	8

	Results		Messages	
<b>Stop_Month</b>	<b>id_Stop_Month</b>	<b>label_Stop_Month</b>	<b>id_Start_Month</b>	<b>label_Start_Month</b>
1	2	1	1	7
2	8	2	2	2
3	3	3	3	4
4	4	4	4	5
5	5	5	5	6
6	10	6	6	10
7	12	7	7	12
8	9	8	8	9
9	6	9	9	6
10	11	10	10	11
11	1	11	11	1
12	7	12	12	7

	Results		Messages	
<b>Start_Month</b>	<b>id_Start_Month</b>	<b>label_Start_Month</b>	<b>id_Birth_Year</b>	<b>label_Birth_Year</b>
1	7	1	1	61
2	2	2	2	49
3	4	3	3	46
4	5	4	4	25
5	8	5	5	4
6	3	6	6	33
7	12	7	7	47
8	10	8	8	34
9	1	9	9	24
10	11	10	10	59
11	6	11	11	10
12	9	12	12	45

- Dimension Start\_Station, End\_Station

	id_Start_Station	label_Start_Station_ID	label_Start_Station_Name	label_Start_Station_Latitude	label_Start_Station_Longitude
1	1	3207	Oakland Ave	40.7376037	-74.0524783
2	2	3272	Jersey and 3rd	40.72333159	-74.04595256
3	3	3195	Sip Ave	40.73074263	-74.06378388
4	4	3189	West Side Light Rail	40.714402	-74.0887723
5	5	3269	Brunswick and 6th	40.72601173	-74.05038893
6	6	3267	Morris Canal	40.71241882	-74.03852552
7	7	3273	Manila and 1st	40.72165072	-74.04288411
8	8	3210	Pershing Field	40.74267714	-74.05178863
9	9	3276	Marin Light Rail	40.71458404	-74.04281706
10	10	3187	Warren St	40.7211236	-74.03805095
11	11	3202	Newport PATH	40.7272235	-74.0337589
12	12	3184	Paulus Hook	40.7141454	-74.0335519
13	13	3200	MLK Light Rail	40.7111305	-74.0788855
14	14	3216	Columbia Park	40.6970299	-74.0969366
15	15	3217	Bayside Park	40.69865054	-74.08207968
16	16	3191	Union St	40.7182113	-74.0836394
17	17	3426	JCBS Depot	40.70965083	-74.06860113
18	18	3214	Essex Light Rail	40.7127742	-74.0364857
19	19	3196	Riverview Park	40.7443187	-74.0439909
20	20	3192	Liberty Light Rail	40.7112423	-74.0557013
21	21	3193	Lincoln Park	40.7246051	-74.07840595
22	22	3188	NJCU	40.7101087	-74.0858489
23	23	3211	Newark Ave	40.72152515	-74.04630454
24	24	3277	Communipaw and Berry ...	40.71435837	-74.06661093
25	25	3268	Lafayette Park	40.71346383	-74.06285852
26	26	3194	McGinley Square	40.72533993	-74.06762213
27	27	3215	Central Ave	40.7467299	-74.0492509
28	28	3280	Astor Place	40.7192822	-74.07126188
29	29	3281	Leonard Gordon Park	40.74590997	-74.05727148

	id_End_Station	label_End_Station_ID	label_End_Station_Name	label_End_Station_Latitude	label_End_Station_Longitude
1	1	247	Perry St and Bleecker St	40.73535398	-74.00483091
2	2	297	E 15 St and 3 Ave	40.734232	-73.986923
3	3	319	Fulton St and Broadway	40.711066	-74.009447
4	4	380	W 4 St and 7 Ave S	40.73401143	-74.00293877
5	5	448	W 37 St and 10 Ave	40.75660359	-73.9979009
6	6	501	FDR Drive and E 35 St	40.744219	-73.97121214
7	7	3173	Riverside Blvd and W 67 St	40.77750703	-73.98888588
8	8	3183	Exchange Place	40.7162469	-74.0334588
9	9	3186	Grove St PATH	40.71958612	-74.04311746
10	10	3197	North St	40.752559	-74.044725
11	11	3198	Heights Elevator	40.74871595	-74.0404433
12	12	3199	Newport Pkwy	40.7287448	-74.0321082
13	13	3203	Hamilton Park	40.72759597	-74.04424731
14	14	3215	Central Ave	40.7467299	-74.0492509
15	15	3220	5 Corners Library	40.73496102	-74.05950308
16	16	3265	E 2 St and 2 Ave E	40.72456343	-73.98944378
17	17	3270	Jersey and 6th St	40.72528911	-74.04557168
18	18	3274	Bethune Center	40.70495752	-74.08593088
19	19	3278	Monmouth and 6th	40.72568548	-74.04879034
20	20	3279	Dixon Mills	40.72163014	-74.04996783
21	21	3280	Astor Place	40.7192822	-74.07126188
22	22	3281	Leonard Gordon Park	40.74590997	-74.05727148
23	23	3442	Indiana	0	0
24	24	152	Warren St and Church St	40.71473993	-74.00910627
25	25	173	Broadway and W 49 St	40.76068327	-73.98452729
26	26	225	W 14 St and The High Line	40.74195138	-74.00803013
27	27	267	Broadway and LN 26 St	40.75007711	-73.00765429

- Dimension Start\_Time & Stop\_Time

	id_Start_Time	label_Start_Time	label_Start_Hour	label_Start_Minute	label_Start_Sec
1	1	00:00:00	0	0	0
2	2	00:00:01	0	0	1
3	3	00:00:04	0	0	4
4	4	00:00:05	0	0	5
5	5	00:00:06	0	0	6
6	6	00:00:07	0	0	7
7	7	00:00:08	0	0	8
8	8	00:00:10	0	0	10
9	9	00:00:11	0	0	11
10	10	00:00:12	0	0	12
11	11	00:00:13	0	0	13
12	12	00:00:14	0	0	14
13	13	00:00:18	0	0	18
14	14	00:00:19	0	0	19
15	15	00:00:22	0	0	22
16	16	00:00:26	0	0	26
17	17	00:00:27	0	0	27
18	18	00:00:29	0	0	29
19	19	00:00:32	0	0	32
20	20	00:00:34	0	0	34
21	21	00:00:35	0	0	35
22	22	00:00:36	0	0	36
23	23	00:00:37	0	0	37
24	24	00:00:38	0	0	38
25	25	00:00:39	0	0	39
26	26	00:00:40	0	0	40
27	27	00:00:41	0	0	41
28	28	00:00:43	0	0	43
29	29	nn:nn:44	n	n	44

	id_Stop_Time	label_Stop_Time	label_Stop_Hour	label_Stop_Minute	label_Stop_Sec
1	1	00:00:00	0	0	0
2	2	00:00:02	0	0	2
3	3	00:00:03	0	0	3
4	4	00:00:05	0	0	5
5	5	00:00:06	0	0	6
6	6	00:00:07	0	0	7
7	7	00:00:12	0	0	12
8	8	00:00:13	0	0	13
9	9	00:00:14	0	0	14
10	10	00:00:15	0	0	15
11	11	00:00:16	0	0	16
12	12	00:00:17	0	0	17
13	13	00:00:18	0	0	18
14	14	00:00:19	0	0	19
15	15	00:00:20	0	0	20
16	16	00:00:22	0	0	22
17	17	00:00:23	0	0	23
18	18	00:00:24	0	0	24
19	19	00:00:25	0	0	25
20	20	00:00:27	0	0	27
21	21	00:00:29	0	0	29
22	22	00:00:30	0	0	30
23	23	00:00:32	0	0	32
24	24	00:00:33	0	0	33
25	25	00:00:34	0	0	34
26	26	00:00:36	0	0	36
27	27	00:00:38	0	0	38

In addition, we designed the fact table “Bike\_Share” :

MARIANNA.Bike_Share.o.Bike_Share_fact		SQLQuery4.sql - M...Marianna Kon (62)	SQLQuery3.s
Column Name	Data Type	Allow Nulls	
Start_Year	int	<input type="checkbox"/>	
Start_Month	int	<input type="checkbox"/>	
Start_Day	int	<input type="checkbox"/>	
Start_Time	int	<input type="checkbox"/>	
Stop_Year	int	<input type="checkbox"/>	
Stop_Month	int	<input type="checkbox"/>	
Stop_Day	int	<input type="checkbox"/>	
Stop_Time	int	<input type="checkbox"/>	
Start_Station	int	<input type="checkbox"/>	
End_Station	int	<input type="checkbox"/>	
Bike_ID	int	<input type="checkbox"/>	
User_Type	int	<input type="checkbox"/>	
Birth_Year	int	<input type="checkbox"/>	
Gender	int	<input type="checkbox"/>	
Trip_Duration	int	<input checked="" type="checkbox"/>	

It contains the ids from the Dimensions we created as foreign keys and the measure Trip\_Duration (we had to add the foreign key relationships when creating the table in MS SQL)

Additionally, we added the “Update Fact Table” stage at the Control Flow where we had to make the appropriate joins in order to insert the values.

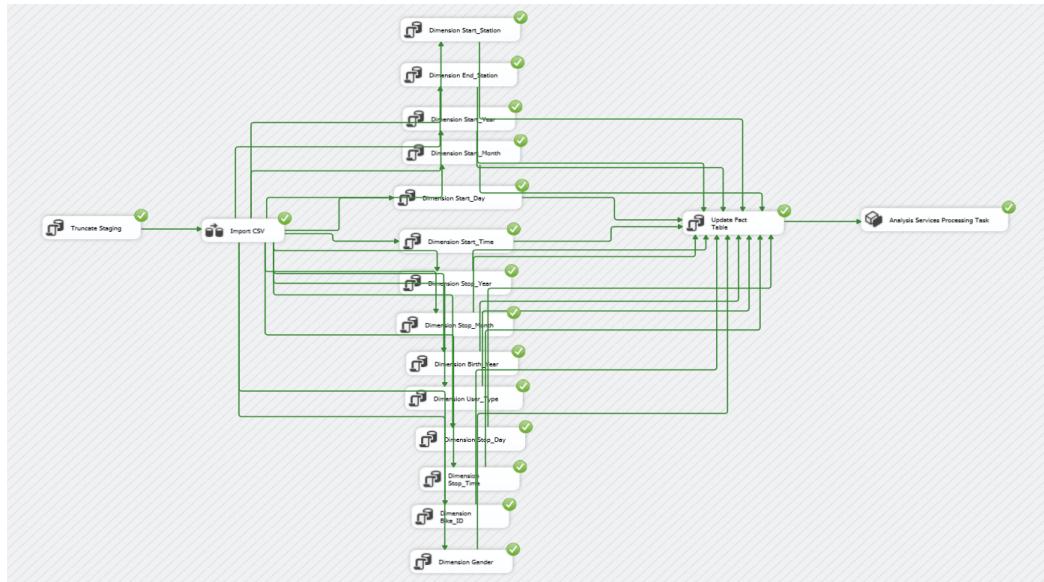
```
Enter SQL Query
INSERT INTO [Bike_Share].[dbo].[Bike_Share_fact]
SELECT [Dim_Start_Month].[Start_Year] AS [Start_Year], [Bike_Share].[dbo].[Dim_Start_Month].[id_Start_Month] AS [Start_Month], [Bike_Share].[dbo].[Dim_Start_Day].[id_Start_Day] AS [Start_Day],
[Bike_Share].[dbo].[Dim_Start_Time].[id_Start_Time] AS [Start_Time], [Bike_Share].[dbo].[Dim_Stop_Year] AS [Stop_Year], [Bike_Share].[dbo].[Dim_Stop_Month].[id_Stop_Month] AS [Stop_Month], [Bike_Share].[dbo].[Dim_Stop_Day] AS [Stop_Day],
[Bike_Share].[dbo].[Dim_Stop_Time].[id_Stop_Time] AS [Stop_Time], [Bike_Share].[dbo].[Dim_Start_Station].[id_Start_Station] AS [Start_Station], [Bike_Share].[dbo].[Dim_End_Station].[id_End_Station] AS [End_Station],
[Bike_Share].[dbo].[Dim_Bike_ID].[id_Bike_ID] AS [Bike_ID], [Bike_Share].[dbo].[Dim_User_Type].[id_User_Type] AS [User_Type], [Bike_Share].[dbo].[Dim_Birth_Year].[id_Birth_Year] AS [Birth_Year],
[Bike_Share].[dbo].[Dim_Gender].[id_Gender] AS [Gender], [Bike_Share].[dbo].[staging].[Trip Duration] AS [Trip_Duration]
FROM [Bike_Share].[dbo].[staging]
```

INNER JOIN [Bike\_Share].[dbo].[Dim\_Bike\_ID] ON [Bike\_Share].[dbo].[staging].[Bike ID] = [Bike\_Share].[dbo].[Dim\_Bike\_ID].[label\_Bike\_ID]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Birth\_Year] ON [Bike\_Share].[dbo].[staging].[Birth Year] = [Bike\_Share].[dbo].[Dim\_Birth\_Year].[label\_Birth\_Year]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Gender] ON [Bike\_Share].[dbo].[staging].[Gender] = [Bike\_Share].[dbo].[Dim\_Gender].[label\_Gender]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Start\_Day] ON [Bike\_Share].[dbo].[staging].[Start\_Day] = [Bike\_Share].[dbo].[Dim\_Start\_Day].[label\_Start\_Day]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Start\_Time] ON [Bike\_Share].[dbo].[staging].[Start\_Time] = [Bike\_Share].[dbo].[Dim\_Start\_Time].[label\_Start\_Time]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Start\_Month] ON [Bike\_Share].[dbo].[staging].[Start\_month] = [Bike\_Share].[dbo].[Dim\_Start\_Month].[label\_Start\_Month]
INNER JOIN [Bike\_Share].[dbo].[Dim\_End\_Station] ON [Bike\_Share].[dbo].[staging].[End\_Station] = [Bike\_Share].[dbo].[Dim\_End\_Station].[label\_End\_Station]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Start\_Station] ON [Bike\_Share].[dbo].[staging].[Start\_Station] = [Bike\_Share].[dbo].[Dim\_Start\_Station].[label\_Start\_Station\_ID]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Stop\_Day] ON [Bike\_Share].[dbo].[staging].[Stop\_Day] = [Bike\_Share].[dbo].[Dim\_Stop\_Day].[label\_Stop\_Day]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Stop\_Time] ON [Bike\_Share].[dbo].[staging].[Stop\_Time] = [Bike\_Share].[dbo].[Dim\_Stop\_Time].[label\_Stop\_Time]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Stop\_Month] ON [Bike\_Share].[dbo].[staging].[Stop\_month] = [Bike\_Share].[dbo].[Dim\_Stop\_Month].[label\_Stop\_Month]
INNER JOIN [Bike\_Share].[dbo].[Dim\_Stop\_Year] ON [Bike\_Share].[dbo].[staging].[Stop\_Year] = [Bike\_Share].[dbo].[Dim\_Stop\_Year].[label\_Stop\_Year]
INNER JOIN [Bike\_Share].[dbo].[Dim\_End\_Station] ON [Bike\_Share].[dbo].[staging].[End\_Station] = [Bike\_Share].[dbo].[Dim\_End\_Station].[label\_End\_Station\_ID]
INNER JOIN [Bike\_Share].[dbo].[Dim\_User\_Type] ON [Bike\_Share].[dbo].[staging].[User Type] = [Bike\_Share].[dbo].[Dim\_User\_Type].[label\_User\_Type]

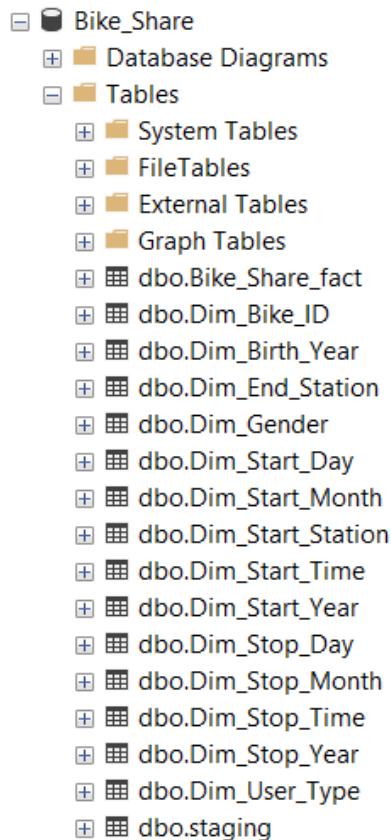
While running the Control Flow the message of correct number of rows affected appeared and this was the output of the table:

Results		Messages														
Start_Year	Start_Month	Start_Day	Start_Time	Stop_Year	Stop_Month	Stop_Day	Stop_Time	Start_Station	End_Station	Bike_ID	User_Type	Birth_Year	Gender	Trip_Duration		
1	1	8	14	56775	1	5	23	58123	40	12	1	2	26	1	1290	
2	1	1	28	16300	1	6	13	16861	20	9	1	2	35	3	475	
3	1	10	8	32941	1	9	31	33379	23	75	1	2	16	1	370	
4	1	1	29	27527	1	6	28	14800	45	106	1	1	20	2	73383	
5	1	1	5	48677	1	6	27	49145	32	8	1	2	55	1	409	
6	3	9	8	16172	3	7	31	17059	19	9	1	2	8	1	800	
7	1	12	5	48833	1	12	27	49106	11	12	1	2	32	1	213	
8	1	8	14	63068	1	5	23	65217	30	12	1	2	63	1	2173	
9	1	8	18	14819	1	5	2	15332	20	9	1	2	39	1	426	
10	3	11	20	55698	3	11	5	55957	51	46	1	2	27	1	200	
11	1	5	7	42138	1	4	30	42994	31	47	1	2	48	1	793	
12	1	12	12	69454	1	12	18	69871	11	66	1	2	36	1	293	
13	1	2	14	49909	1	8	23	50206	3	43	1	2	31	1	237	
14	2	2	18	27200	2	8	2	27493	23	13	1	2	35	3	222	
15	3	6	29	14631	3	1	14	14996	51	9	1	2	39	1	278	
16	1	3	15	19873	1	10	25	20153	51	46	1	2	27	1	194	
17	1	3	15	20481	1	10	25	20847	23	108	1	2	20	3	279	
18	1	10	18	57992	1	9	2	58520	11	47	1	2	50	1	467	
19	1	3	1	18275	1	10	4	19077	3	103	1	2	20	1	717	

This is the successfully run Visual Studio Control Flow, with all dimensions and the fact table:



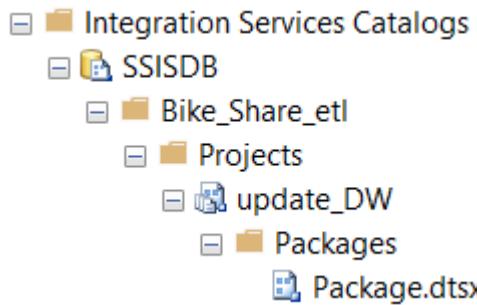
Furthermore, here is the completed database:



#### 4. Deploying the SSIS Package

The last step before creating the Cube is to deploy the SSIS Package for our Control Flow to be executed everyday at 2a.m.

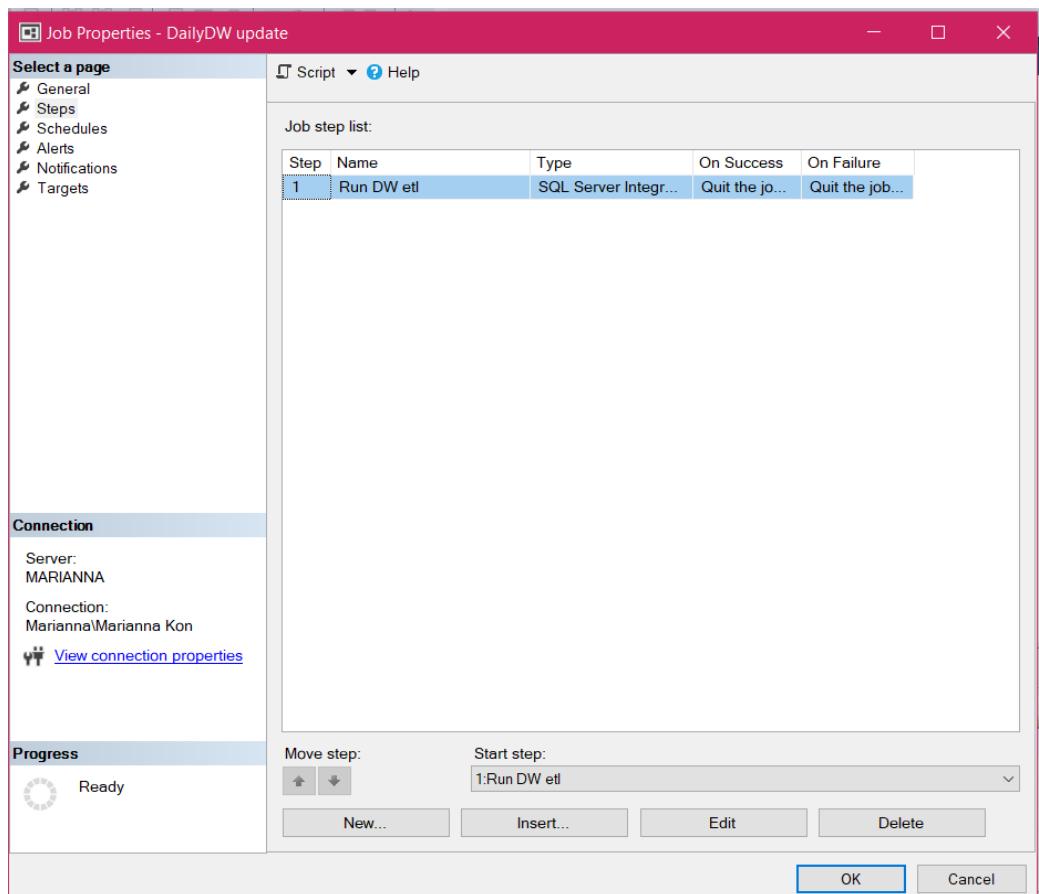
We added the Bike\_Share ETL to the Integration Services Catalogs where we created a new project called “update\_DW” which helps execute our package without Visual Studio.



In order to do daily updates we need to set a schedule we will need to use the SQL Server Agent. We named this job “DailyDW Update”.



We set the Steps:



And then we set the schedule for every night at 2 am.

Job Properties - DailyDW update

Select a page: General, Steps, Schedules, Alerts, Notifications, Targets

Script ▾ Help

Schedule list:

ID	Name	Enabled	Description	J...
11	Daily Run	Yes	Occurs every day at 2:00:00 AM. Schedule will b...	V...

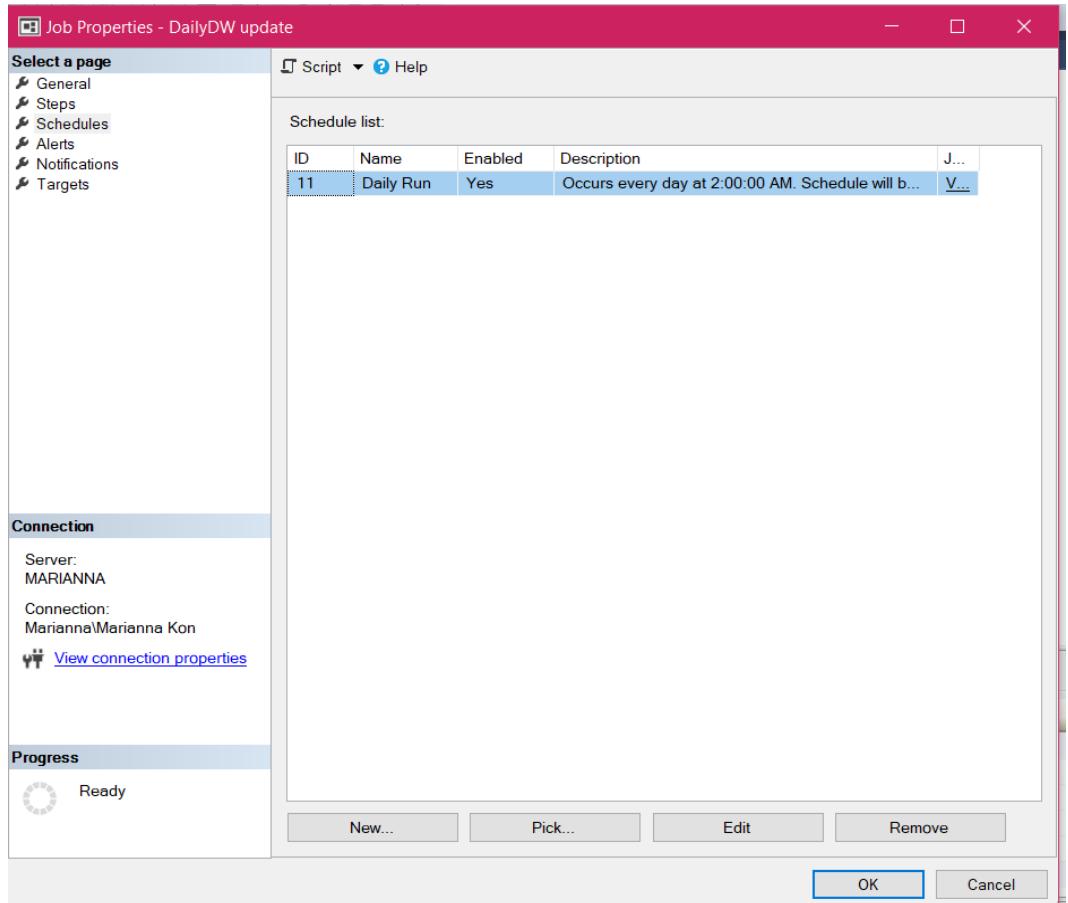
Connection:

Server: MARIANNA  
Connection: MariannaMarianna Kon  
[View connection properties](#)

Progress:

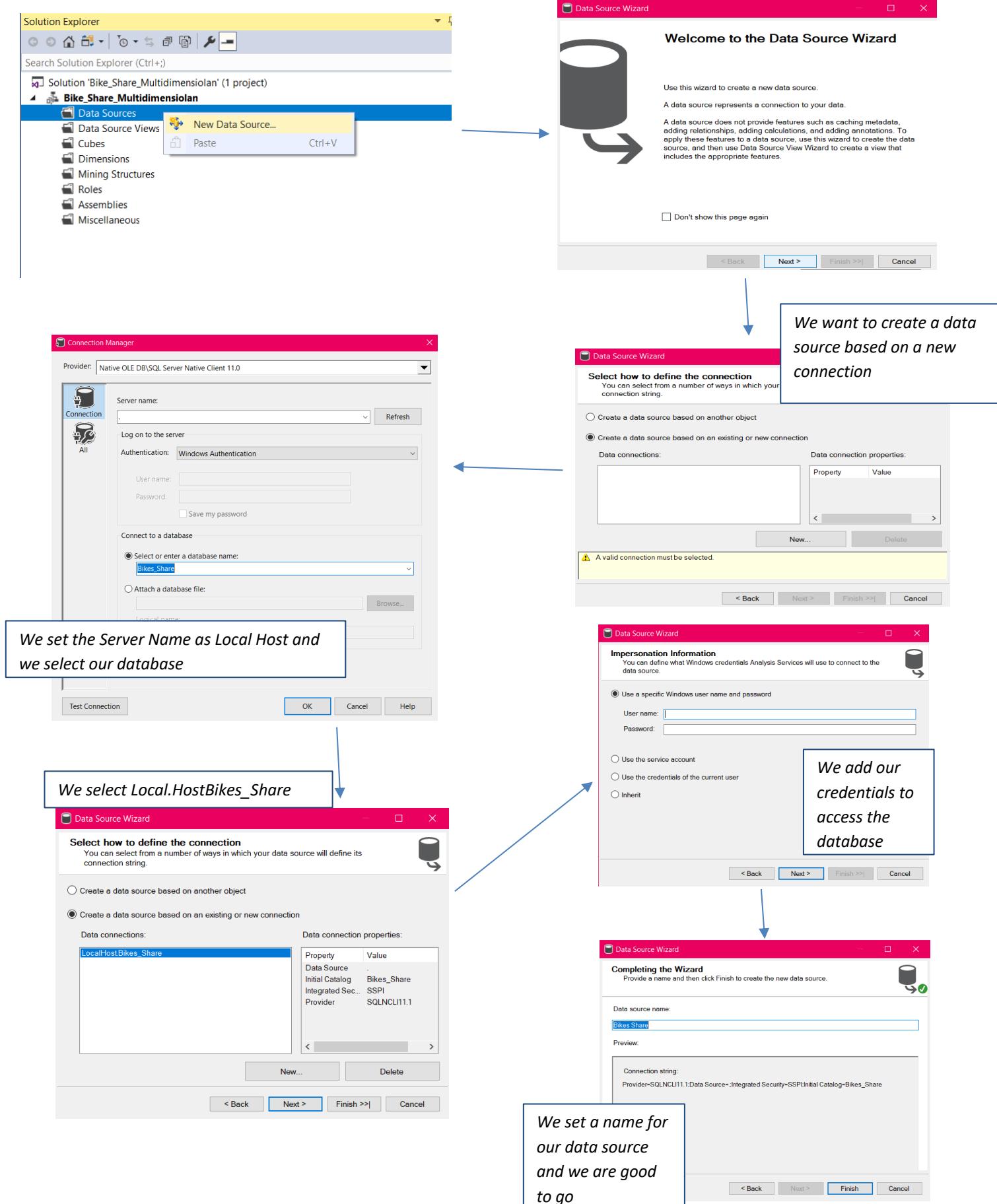
Ready

New... Pick... Edit Remove OK Cancel



## 5. Defining the Data Cube

- In order to Create the cube, the first step is to add the Data Source



- Next, we want to create a new **Data Source View**.

**Solution Explorer**

**Data Source View Wizard**

Welcome to the Data Source View Wizard

Use this wizard to create a new data source view. You create a data source view from tables and views in a relational database. A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.

Don't show this page again

< Back Next > Finish >> Cancel

**We select the fact table and then "Add Related Tables"**

**Data Source View Wizard**

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type
Dim_Start_Station (dbo)	Table
Dim_Start_Time (dbo)	Table
Dim_Start_Year (dbo)	Table
Dim_Stop_Day (dbo)	Table
Dim_Stop_Month (dbo)	Table
Dim_Stop_Year (dbo)	Table
Dim_User_Type (dbo)	Table
Staging (dbo)	Table

Included objects:

Name	Type
Fact_Bikes (dbo)	Table

Add Related Tables

Show system objects

< Back Next > Finish >> Cancel

**We select our data source**

**Data Source View Wizard**

Select a Data Source

Select an existing relational data source or create a new one.

Relational data sources:

Bikes Share
-------------

Data source properties:

Property	Value
Data Source	Bikes_Share
Initial Catalog	SPSI
Integrated Security	SSPI
Provider	SQLNCLI11.1

New Data Source... Advanced...

< Back Next > Finish >> Cancel

**Finally, we provide a name and we create the data source**

**Data Source View Wizard**

Completing the Wizard

Provide a name, and then click Finish to create the new data source view.

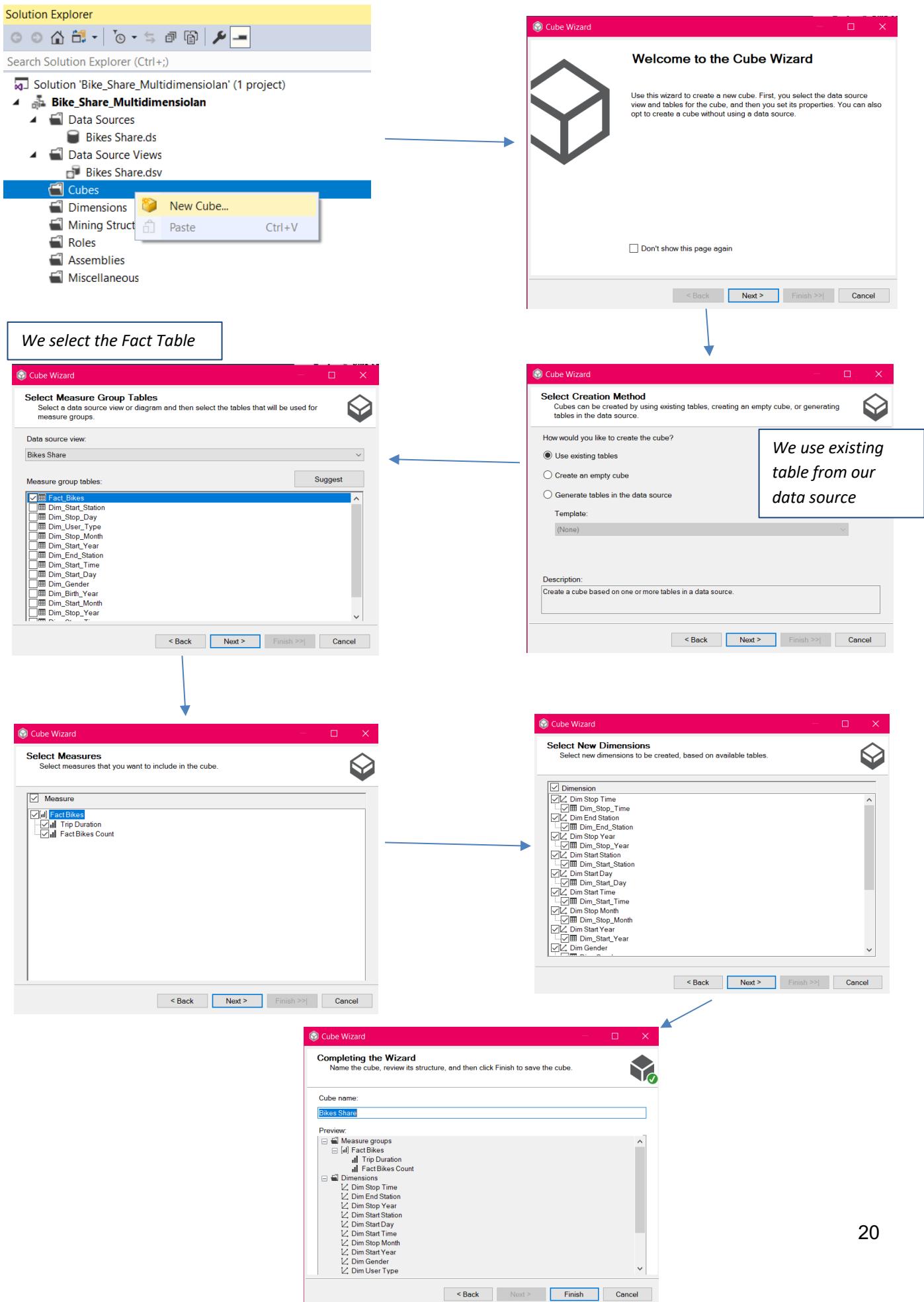
Name:

Preview:

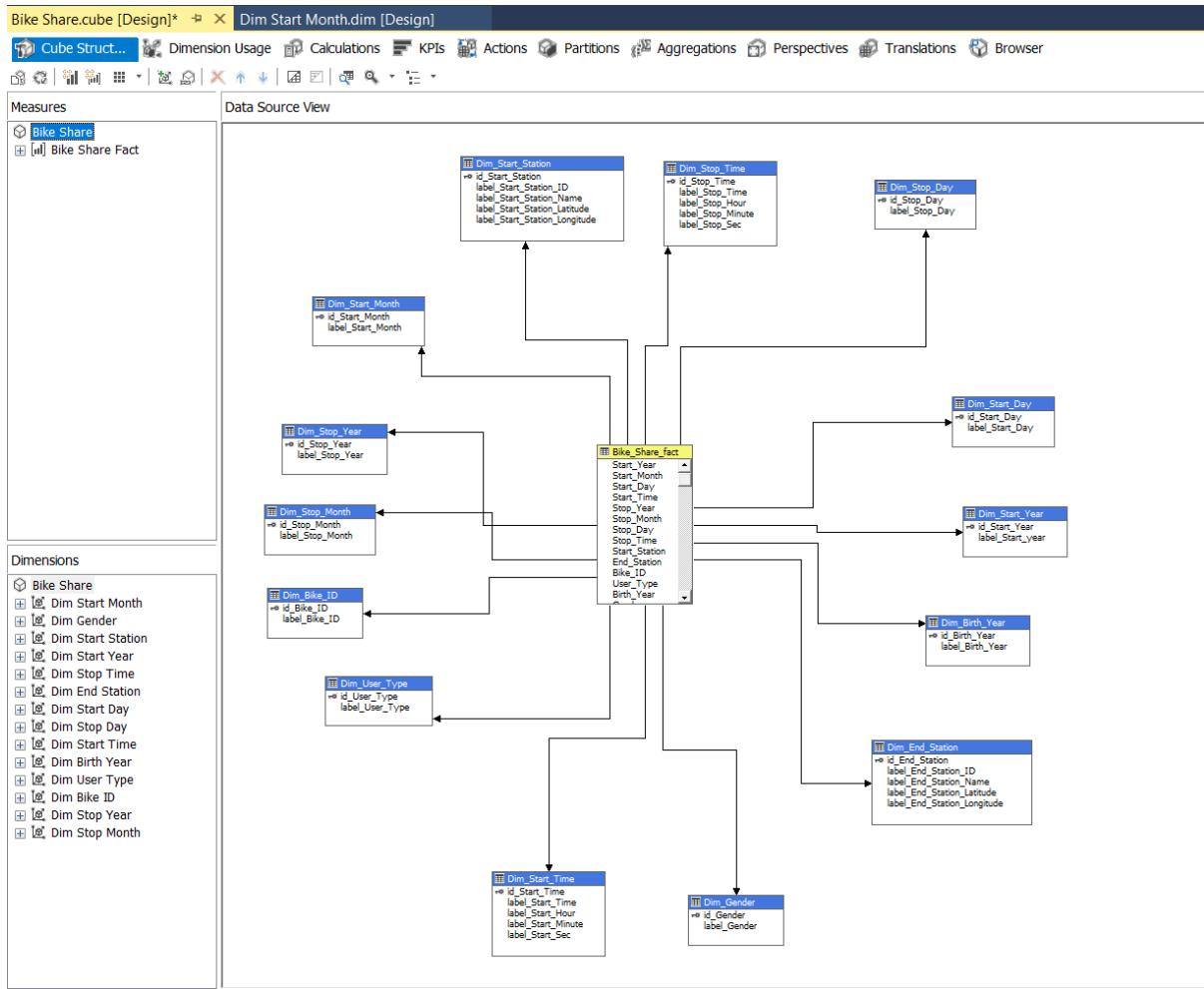
Bikes Share	
Fact_Bikes (dbo)	
Dim_Start_Station (dbo)	
Dim_Stop_Day (dbo)	
Dim_Stop_Month (dbo)	
Dim_Start_Year (dbo)	
Dim_End_Station (dbo)	
Dim_Start_Time (dbo)	
Dim_Start_Day (dbo)	
Dim_Gender (dbo)	
Dim_Birth_Year (dbo)	
Dim_Start_Month (dbo)	
Dim_Stop_Year (dbo)	
Dim_Stop_Time (dbo)	
Dim_Bike_ID (dbo)	

< Back Next > Finish Cancel

- Now, we are ready to create the **Cube**.



- Below we can see our Cube Structure-Schema.



- And our Dimension Tables

**Dimensions**

- ↳ Bike Share
- ↳ Dim Start Month
  - ↳ Edit Dim Start Month
  - ↳ Attributes
    - ↳ Id Start Month
    - ↳ Label Start Month
- ↳ Dim Gender
  - ↳ Edit Dim Gender
  - ↳ Attributes
    - ↳ Id Gender
    - ↳ Label Gender
- ↳ Dim Start Station
  - ↳ Edit Dim Start Station
  - ↳ Attributes
    - ↳ Id Start Station
    - ↳ Label Start Station ID
    - ↳ Label Start Station Latitude
    - ↳ Label Start Station Longitude
    - ↳ Label Start Station Name
- ↳ Dim Start Year
  - ↳ Edit Dim Start Year
  - ↳ Attributes
    - ↳ Id Start Year
    - ↳ Label Start Year
- ↳ Dim Stop Time
  - ↳ Edit Dim Stop Time
  - ↳ Attributes
    - ↳ Id Stop Time
    - ↳ Label Stop Hour
    - ↳ Label Stop Minute
    - ↳ Label Stop Sec
    - ↳ Label Stop Time

**Dimensions**

- ↳ Dim End Station
  - ↳ Edit Dim End Station
  - ↳ Attributes
    - ↳ Id End Station
    - ↳ Label End Station ID
    - ↳ Label End Station Latitude
    - ↳ Label End Station Longitude
    - ↳ Label End Station Name
- ↳ Dim Start Day
  - ↳ Edit Dim Start Day
  - ↳ Attributes
    - ↳ Id Start Day
    - ↳ Label Start Day
- ↳ Dim Stop Day
  - ↳ Edit Dim Stop Day
  - ↳ Attributes
    - ↳ Id Stop Day
    - ↳ Label Stop Day
- ↳ Dim Start Time
  - ↳ Edit Dim Start Time
  - ↳ Attributes
    - ↳ Id Start Time
    - ↳ Label Start Hour
    - ↳ Label Start Minute
    - ↳ Label Start Sec
    - ↳ Label Start Time
- ↳ Dim Birth Year
  - ↳ Edit Dim Birth Year
  - ↳ Attributes
    - ↳ Id Birth Year
    - ↳ Label Birth Year
- ↳ Dim User Type
  - ↳ Edit Dim User Type
  - ↳ Attributes
    - ↳ Id User Type
    - ↳ Label User Type

↳ Dim Bike ID

- ↳ Edit Dim Bike ID
- ↳ Attributes
  - ↳ Id Bike ID
  - ↳ Label Bike ID

↳ Dim Stop Year

- ↳ Edit Dim Stop Year
- ↳ Attributes
  - ↳ Id Stop Year
  - ↳ Label Stop Year

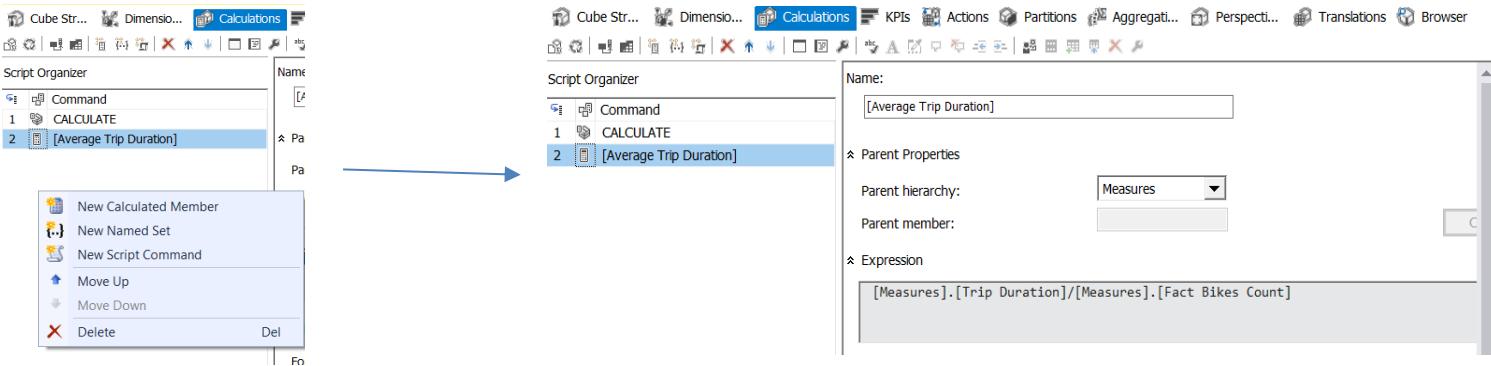
↳ Dim Stop Month

- ↳ Edit Dim Stop Month
- ↳ Attributes
  - ↳ Id Stop Month
  - ↳ Label Stop Month

We want to make same farther calculations so we create,

- Calculated Members:

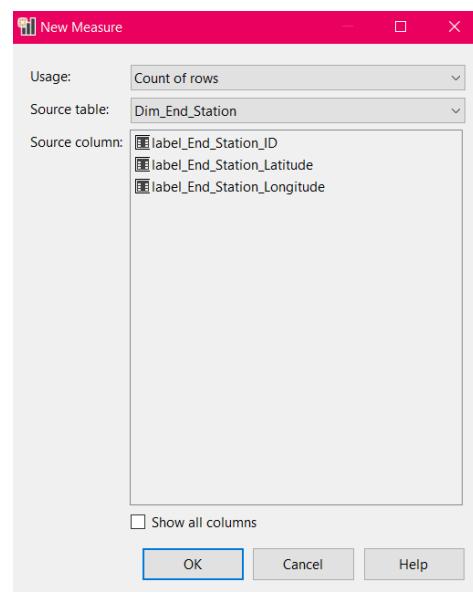
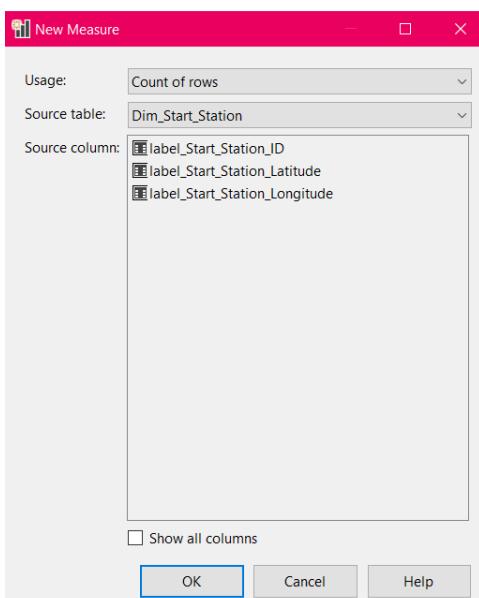
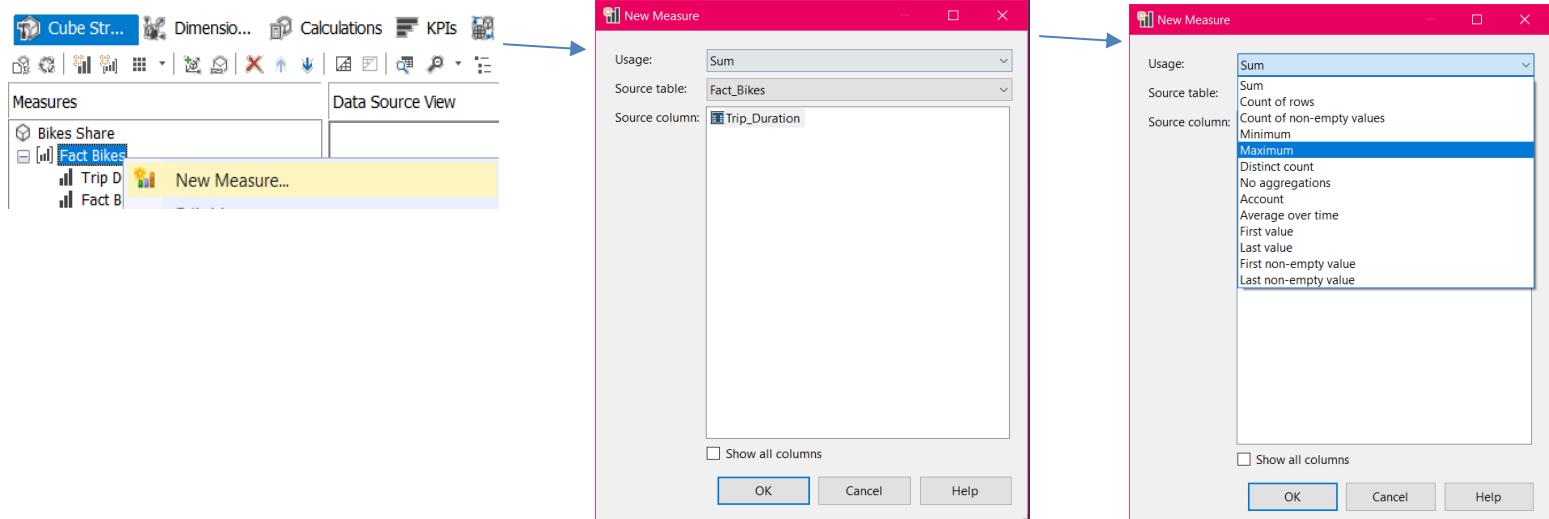
We have created a new calculated member to estimate the [Average Trip Duration](#)



And we add,

- New Measures:

We have Add new Measures to calculate the Maximum-Minimum Trip Duration and the number of Start and End Stations



## 6. Exporting Cube Data to Excel

The screenshot shows the Analysis Services Management Studio interface. At the top, there's a ribbon bar with various tabs like 'Cube S...', 'Dimensions...', 'Calculated...', 'KPIs', 'Actions', 'Partitions', 'Aggregates', 'Perspectives', 'Translations', and 'Browser'. The 'Browser' tab is selected. Below the ribbon, there's a toolbar with icons for 'Edit as Text', 'Import...', 'MDX', and 'Analyze in Excel'. The 'Analyze in Excel' icon is circled in blue. A large downward arrow points from this icon to a screenshot of an Excel spreadsheet.

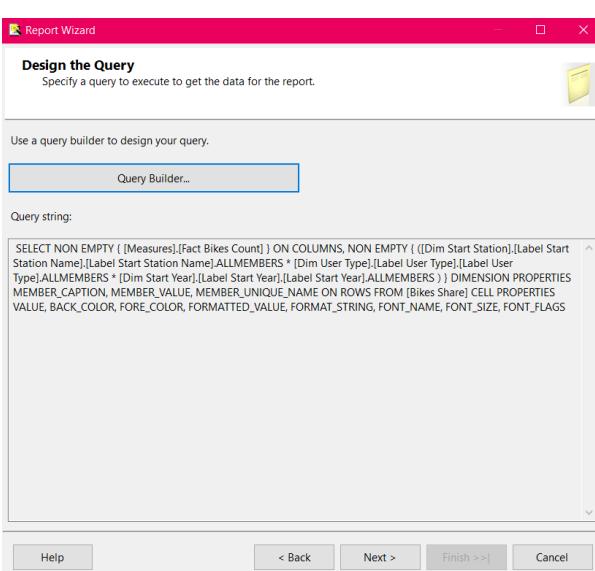
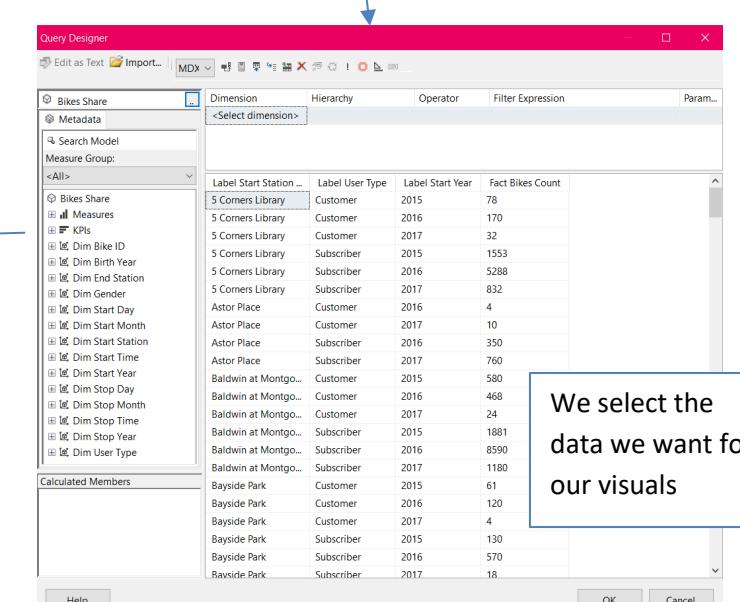
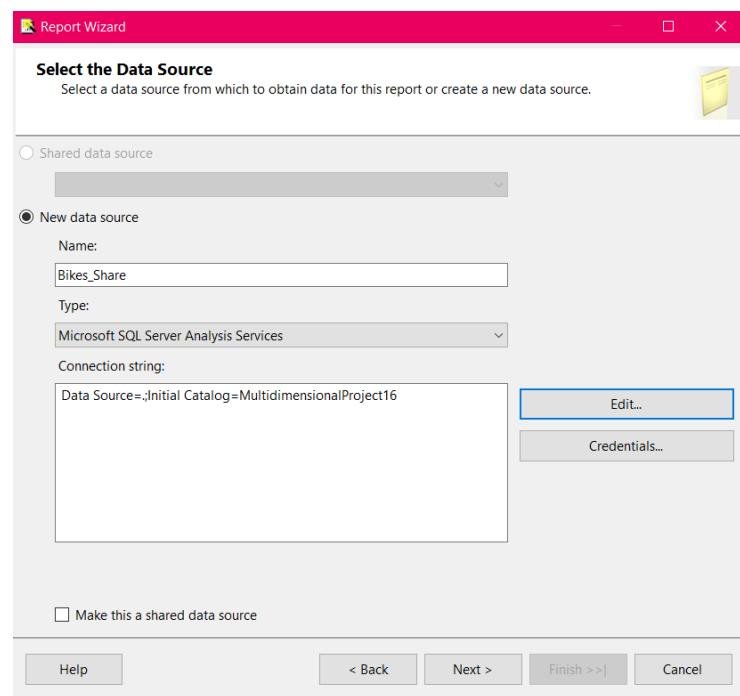
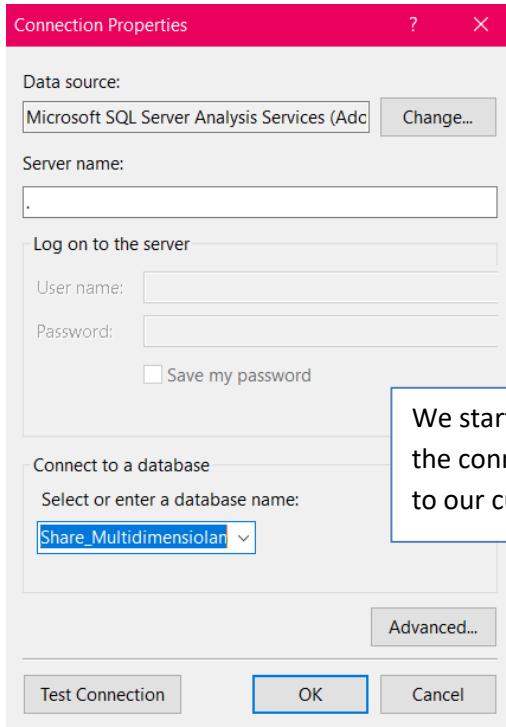
**Excel Spreadsheet Content:**

A	B	C	D	E	F	G	H	I	J
1 Label Start Year	2016								
2									
3 Επικέντες γραμμής	Fact Bikes Count								
4 Customer	31042								
5 Subscriber	464126								
6 Γενικό Αθροισμα	495168								
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									

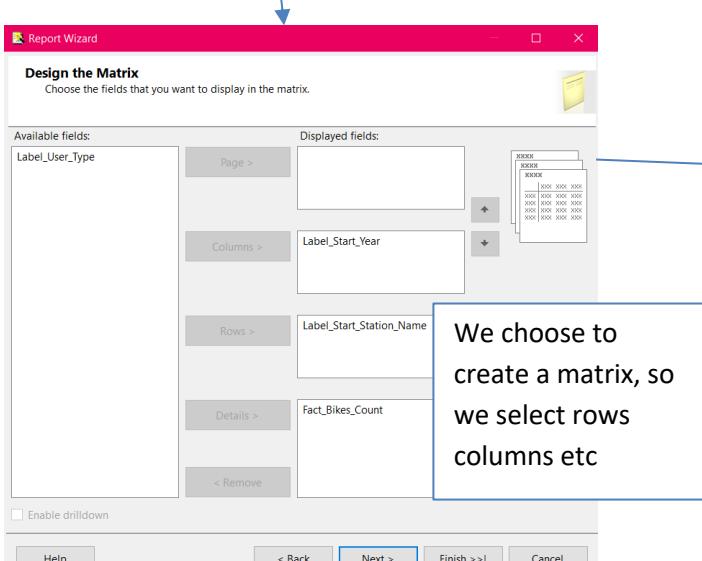
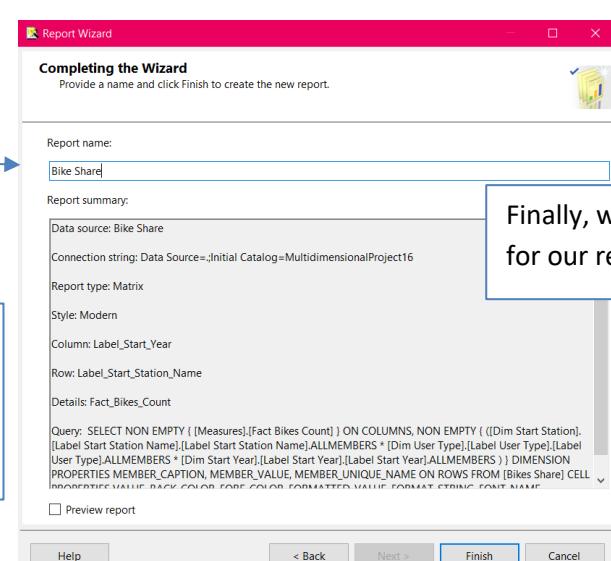
**Analysis Services Filter Panel:**

- Πεδία Συγκεντρωτικού Πίνακα
- Επιλογή πεδίων για προσθήκη στην αναφορά:
- Αναζήτηση
- Σ Σ Fact Bikes
  - Fact Bikes Count
  - Trip Duration
- Dim Bike
  - Id Bike
  - Label Bike ID
- Dim Birth Year
  - Id Birth Year
- Σύρετε τα πεδία μεταξύ των παρακάτω περιοχών:
- Φίλτρα
- Στήλες
- Σειρές
- Τιμές

## 7. Reporting in SSRS



We choose to create a matrix, so we select rows columns etc



Below we can see the final result

Bike Share.rdl [Design] X

Design Preview

1 of 1 | Find | Next

**Bike Share**

User Type	2015	2016	2017	Rank	Total Count
5 Corners Library	1631	5458	864	1	7953
Astor Place		354	770	2	1124
Baldwin at Montgomery	2461	9058	1204	3	12723
Bayside Park	191	690	22	4	903
Bethune Center		58	16	5	74
Brunswick and 6th		1718	2412	6	4130
Brunswick St	7365	19146	2826	7	29337
Central Ave	1306	4468	746	8	6520
Christ Hospital	2231	4732	732	9	7695
City Hall	4986	15582	3408	10	23976
Columbia Park	169	384	156	11	709
Columbus Drive		5540	1990	12	7530
Communipaw and Berry Lane		68	222	13	290
Danforth Light Rail		120	90	14	210
Dey St	1352	4382	456	15	6190
Dixon Mills		5476	2708	16	8184
Essex Light Rail	7131	18924	2404	17	28459
Exchange Place	10776	38028	7478	18	56282
Garfield Ave Station	503	1848	402	19	2753
Grove St PATH	15727	57472	12956	20	86155
Hamilton Park	10556	30600	6454	21	47610
Heights Elevator	979	3586	542	22	5107
Hilltop	2120	5186	772	23	8078
JC Medical Center	3276	9374	1480	24	14130
JCBS Depot		2		25	2

Bike Share.rdl [Design] X

Design Preview

1 of 1 | Find | Next

**Bike Share**

User Type	2015	2016	2017	Rank	Total Count
5 Corners Library	Customer	78	170	32	1 280
Astor Place	Subscriber	1553	5288	832	2 7673
Baldwin at Montgomery	Customer			4	10 14
Bayside Park	Subscriber		350	760	4 1110
Bethune Center	Customer	580	468	24	5 1072
Brunswick and 6th	Subscriber	1881	8590	1180	6 11651
Brunswick St	Customer	61	120	4	7 185
Central Ave	Subscriber	130	570	18	8 718
Christ Hospital	Customer		38	8	9 46
City Hall	Subscriber		20	8	10 28
Communipaw	Customer		26	24	11 50
Danforth Light Rail	Subscriber		1692	2388	12 4080
Dey St	Customer	403	506	24	13 933
Dixon Mills	Subscriber	6962	18640	2802	14 28404
Essex Light Rail	Customer	105	106	36	15 247
Exchange Place	Subscriber	1201	4362	710	16 6273
Garfield Ave Station	Customer	173	218		17 391
Grove St PATH	Subscriber	2058	4514	732	18 7304
Hamilton Park	Customer	521	1002	72	19 1595
Heights Elevator	Subscriber	4465	14580	3336	20 22381
Hilltop	Customer	13	70		21 83
JC Medical Center	Subscriber	156	314	156	22 626
JCBS Depot	Customer		288	36	23 324
	Subscriber		5252	1954	24 7206
	Customer			2	25 2

## Drill down/roll up creation:

1st step : Add Child Group

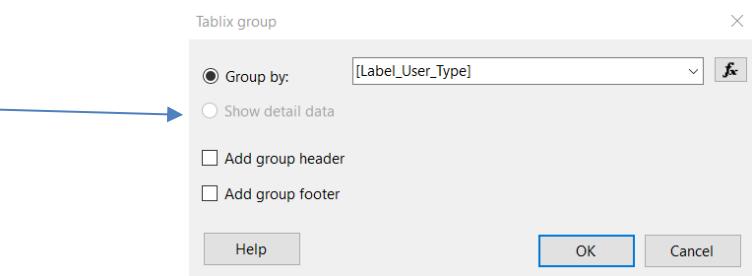
Row Groups

Column Groups

matrix1\_Label\_Start\_Station\_Name

matrix1\_Label\_Start\_Year

- Add Group
- Add Total
- Delete Group
- Group Properties...



2nd step : Change Group Properties

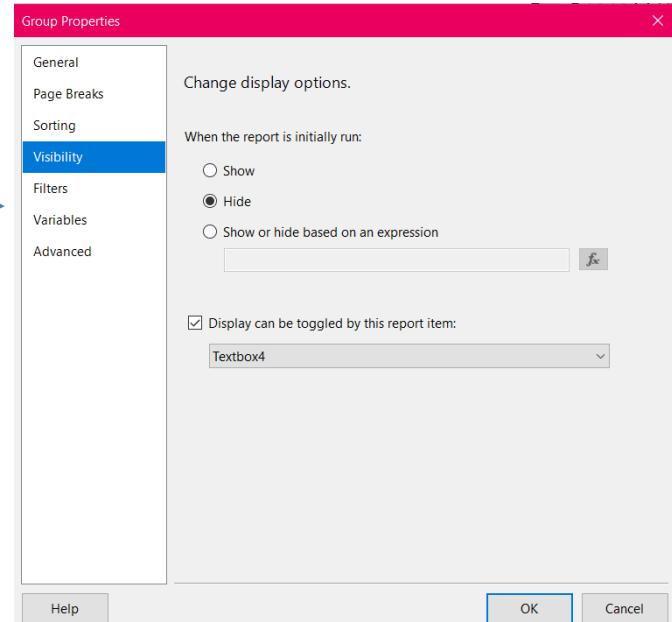
Row Groups

matrix1\_Label\_Start\_Station\_Name

matrix

Label\_User\_Type

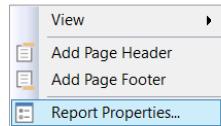
- Add Group
- Add Total
- Delete Group
- Group Properties...



- Rank Creation

\*The Rank refers to the Start Station alphabetical order

Bike Share			
Station Name	User Type	Label Start Year	Rank
bel User Type		[Sum(Fact_Bike)]	

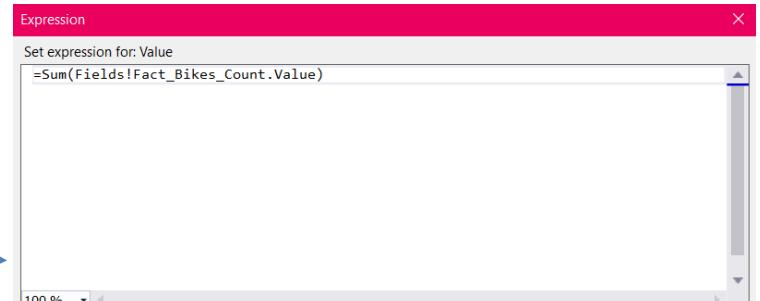
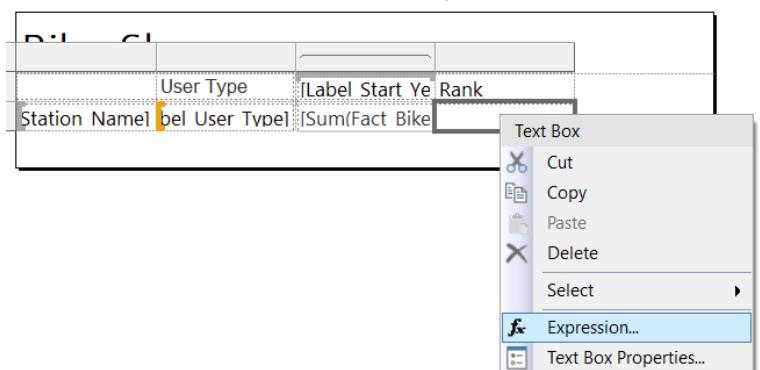
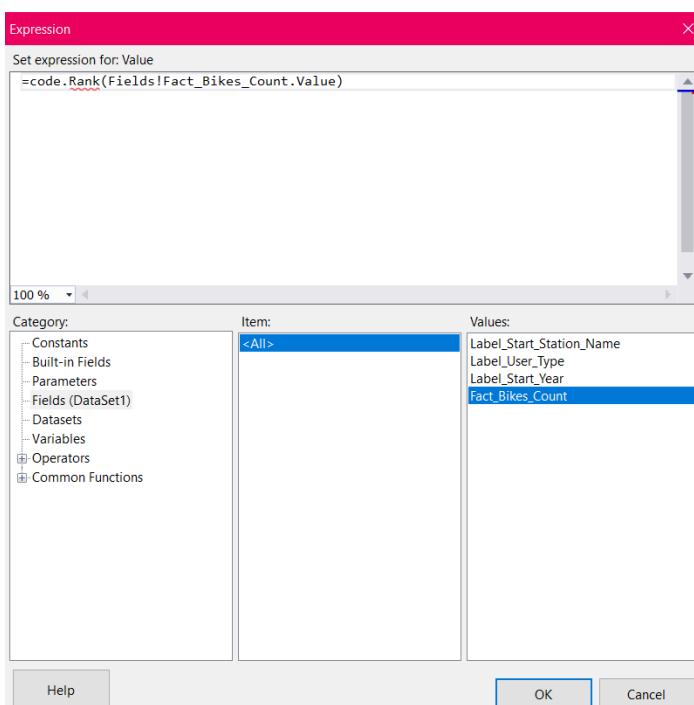


```

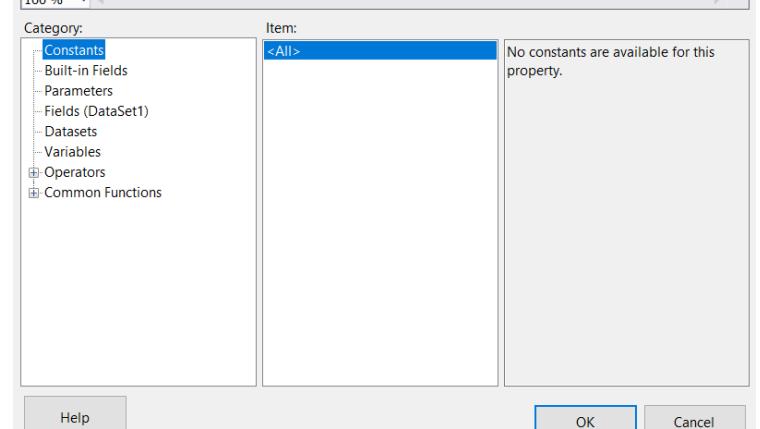
Report Properties
  Page Setup
  Code
  References
  Variables

Custom code:
Public dim myvalue as Integer = 0
Public Function Rank( sourcevalue as Integer ) As Integer
myvalue = myvalue + 1
return myvalue
End Function

```

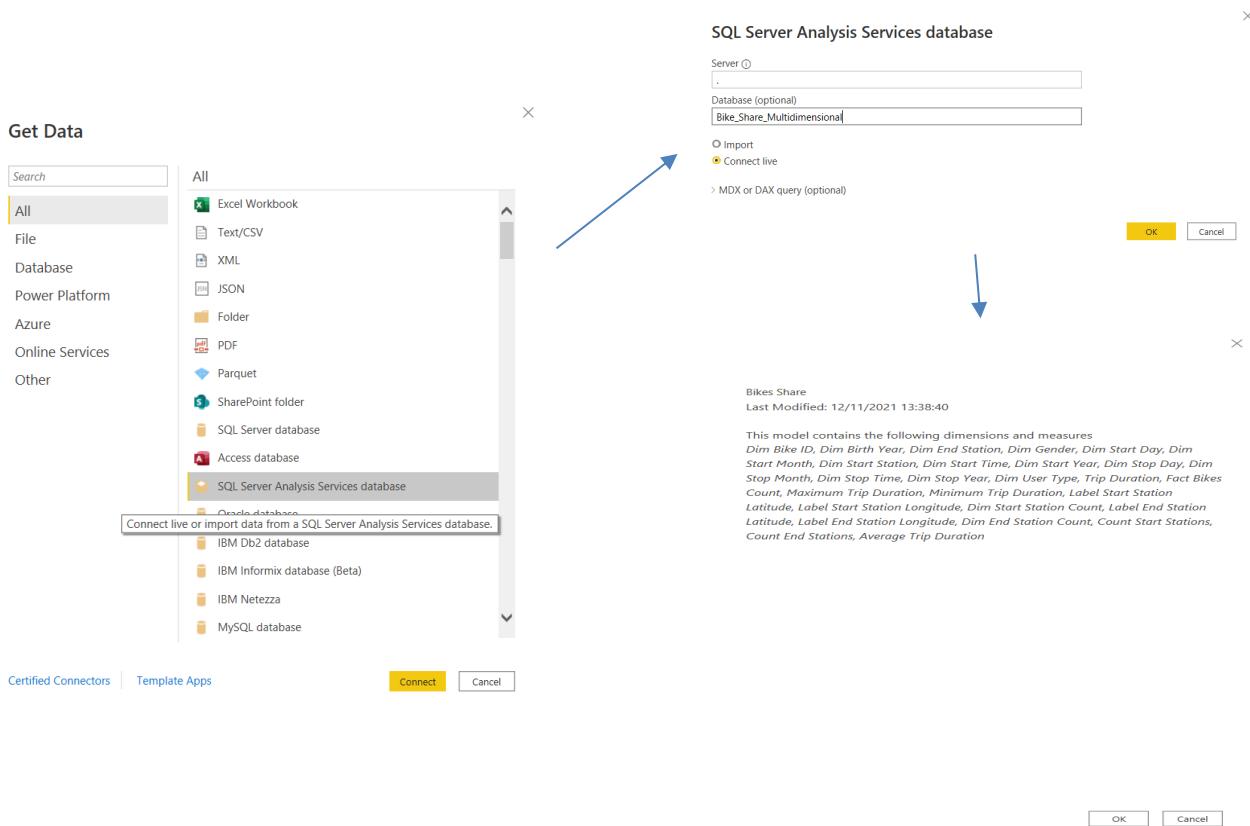


Finally, for the 'Total Count' we have used the expression:



## 8. Power Bi

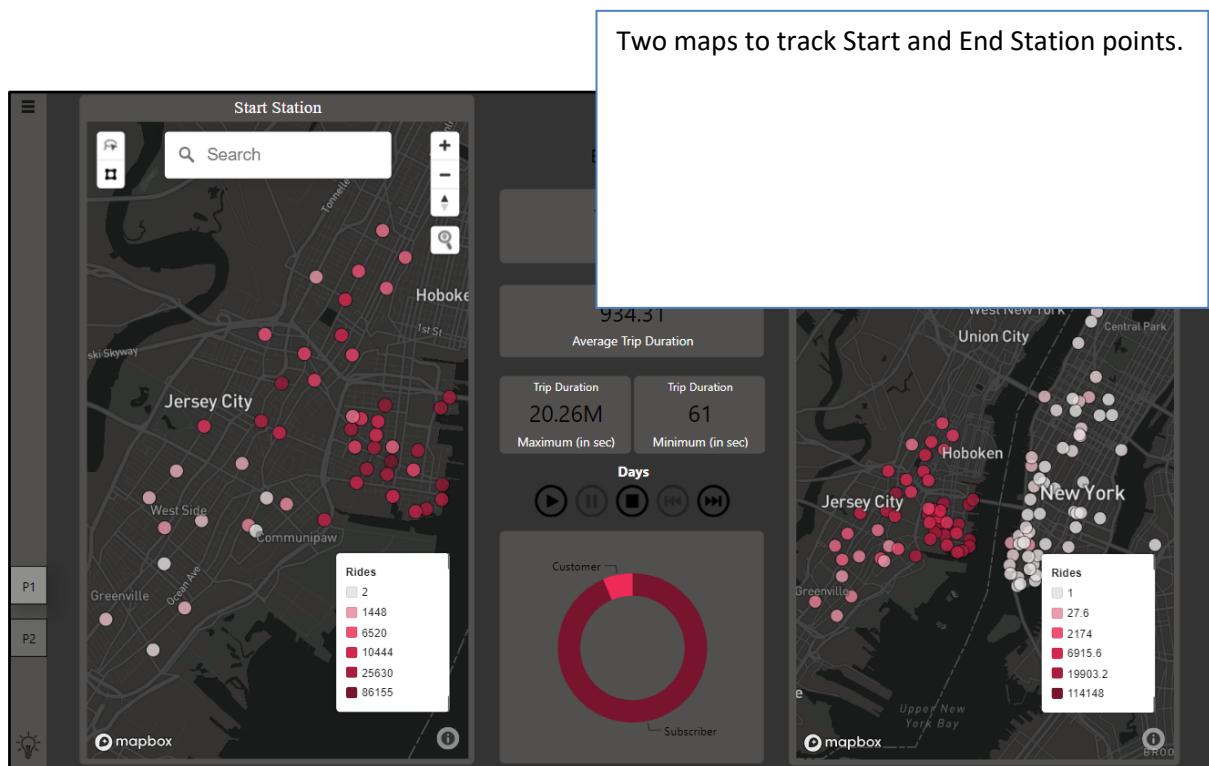
The first step is to get data by connecting our cube to Power bi.



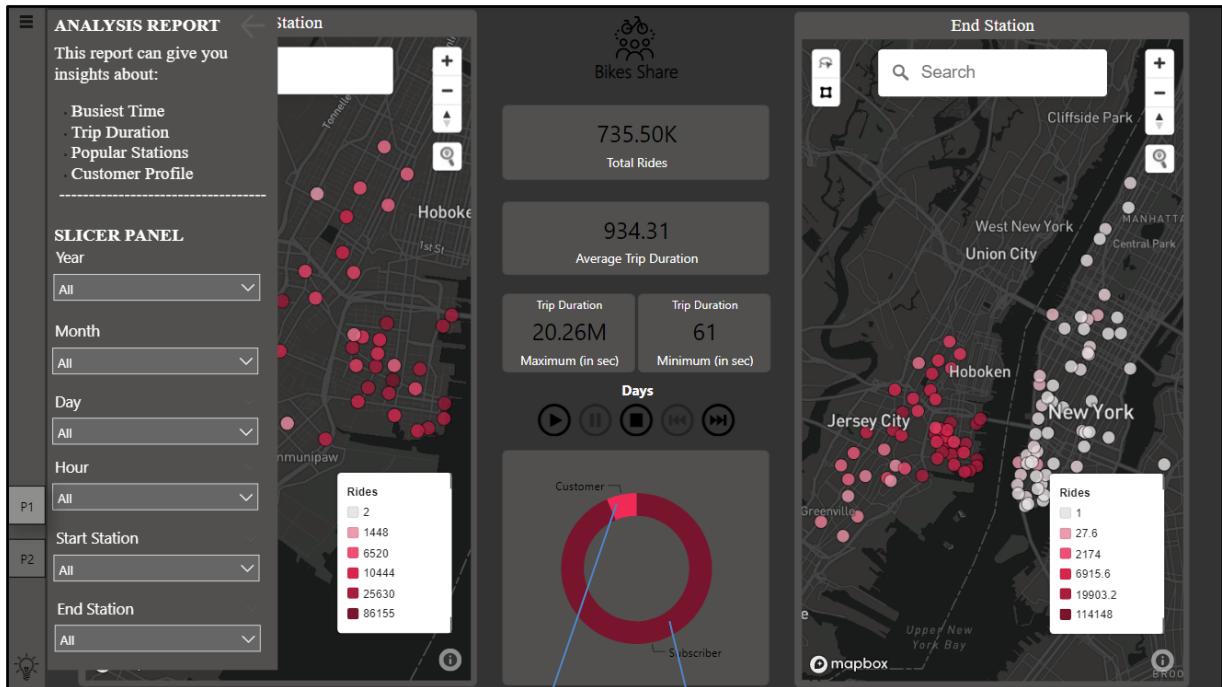
Finally, we are ready to create our report, visualize our data and answer the business questions

We have created a two-page report.

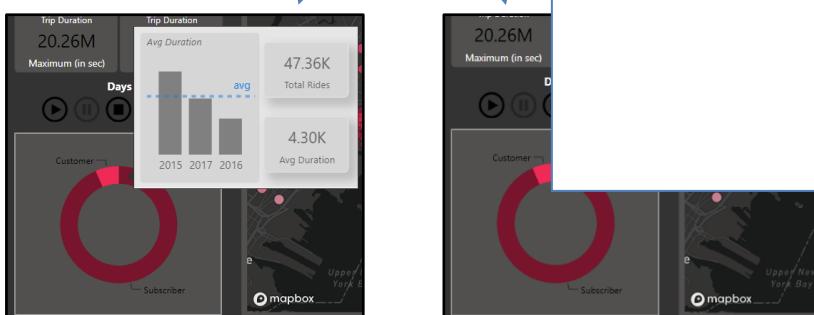
- First page:



Slicer Panel, contains the filters and a sort report description.

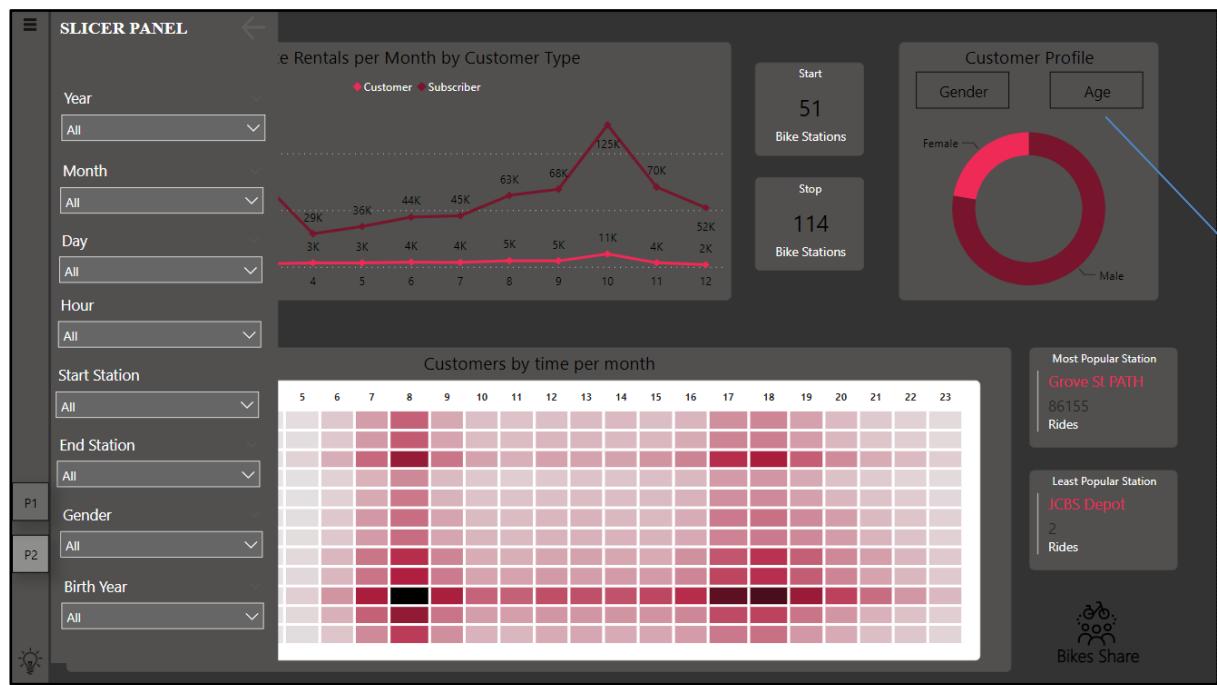


Custom Tooltip to get more insights about Customers-Subscribers

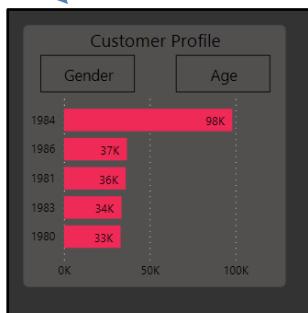


## Second Page:

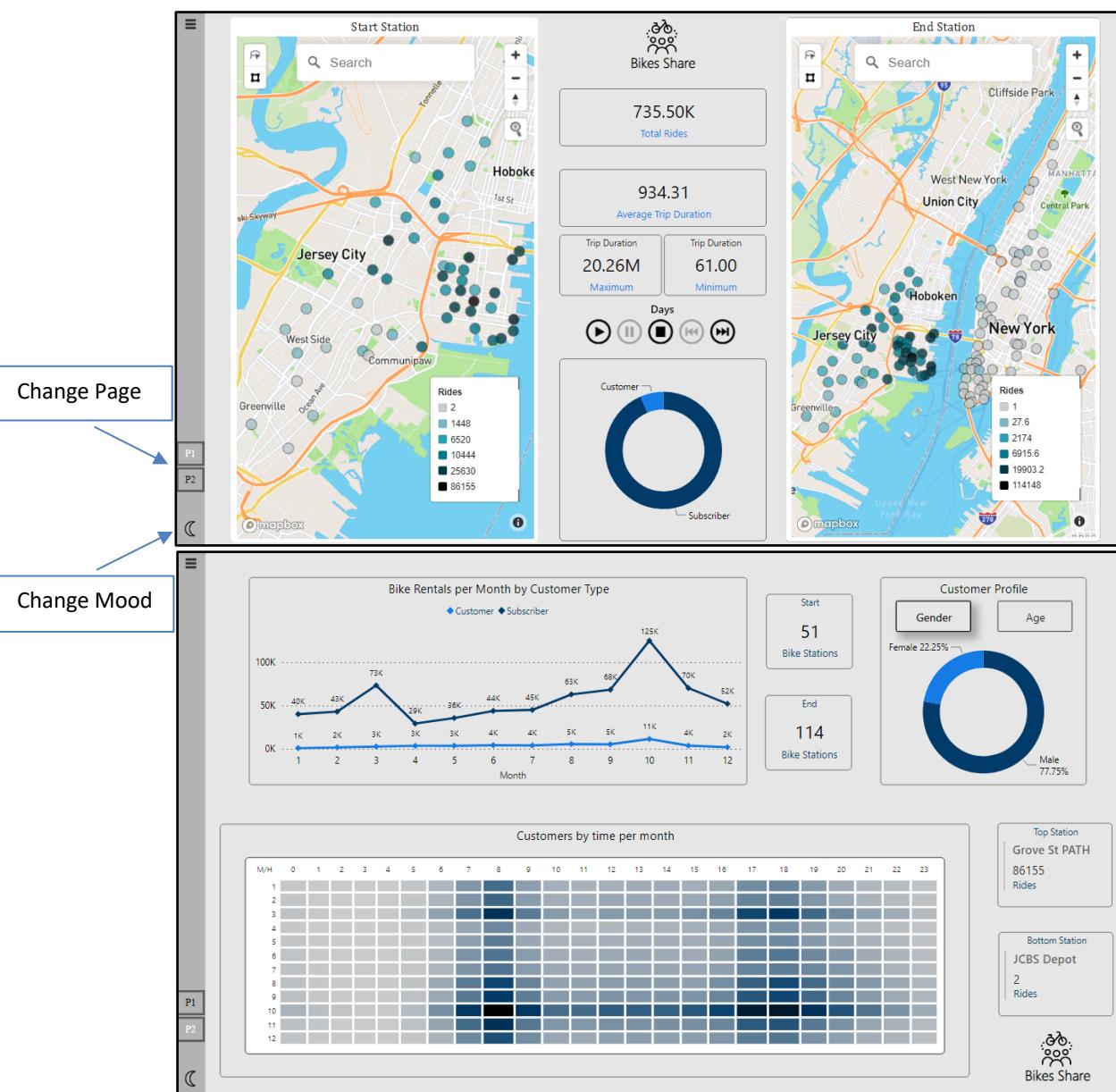




Visuals for the Gender and Age of the Customers.



The Report comes in two moods light and dark



## 9. Power Point Presentation

**New York City Bike Share**

Maria Zafeiropoulou & Marianna Konstantopoulou

**Business Problem**

Bicycle sharing is a service that allows users to rent a bike for a short-term use. These systems operate by allowing users to return the bike at any of the station's locations.

The City Bike Company based in New York need answers for the following questions:

- Which times are the busiest for the system?
- What's the most common ride duration?
- Which stations are the most popular?
- What is the age and gender of the riders?

New York City BikeShare

New York City Bike Share Dataset

Predict Gender of the riders

Natali Aksi • updated 4 years ago (version 1)

Data Code Discussion Activity Metadata Download (132 MB) New Notebook

Usability 71 License CC BY-NC-SA 4.0 Tags earth and nature, retail and shopping, cycling

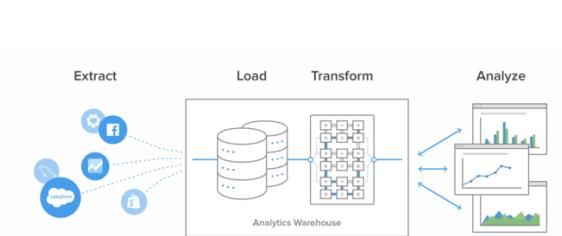
Description

This dataset enables quick, easy, and affordable bike trips around the New York City boroughs. They make regular open data releases (this dataset is a transformed version of the data from this link). The dataset contains 735022 anonymized trips information made from Jan 2013 to June 2017.

Acknowledgements -

This dataset is the property of NYC Bike Share, LLC and Jersey City Bike Share, LLC ("Bikeshare") operates New York City's Citi Bike bicycle sharing service for TSC, click here

New York City BikeShare



New York City BikeShare

**The initial dataset**

Trip	Duration	Start_Time	Stop_Time	Start_Station	Start_Station_ID	Start_Station_Name	Start_Station_Latitude	Start_Station_Longitude	ID	End_Station
0	376	01/10/2015 00:10:10	01/10/2015 00:26:16	3012 Christ Hospital	40.73478582	74.05044864	3307 Oakland Ave			
1	73	01/10/2015 00:17:07	01/10/2015 00:39	3207 Oakland Ave	40.73478582	74.05044864	3212 Christ Hospital			
2	2714	01/10/2015 00:32	01/10/2015 00:39	3191 Lincoln Park	40.7346051	74.07840595	3191 Lincoln Park			
3	275	01/10/2015 00:34	01/10/2015 00:39	1199 Newport Pkwy	40.7387448	-74.0321082	3187 Warren St			
4	561	01/10/2015 00:40	01/10/2015 00:49	3183 Exchange Place	40.7562469	-74.0334588	3192 Liberty Light Rail			
5	361	01/10/2015 00:41	01/10/2015 00:47	3198 Heights Elevator	40.7487195	-74.0443433	3215 Central Ave			
6	139	01/10/2015 00:43	01/10/2015 00:49	3200 Sip Ave	40.7352559	-74.0570439	3190 Sip Ave			
7	1200	01/10/2015 01:10	01/10/2015 01:31	3197 North St	40.7325159	-74.0447225	3215 Central Ave			
8	647	01/10/2015 01:20	01/10/2015 02:12	3213 Van Vorst Park	40.71648892	-74.0472663	3190 Sip Ave			
9	234	01/10/2015 04:43	01/10/2015 04:47	1194 McGinley Square	40.72533993	-74.06762213	3195 Sip Ave			
10	351	01/10/2015 04:43	01/10/2015 04:47	3215 Central Ave	40.7467299	-74.0492509	3197 North St			
11	1013	01/10/2015 05:31	01/10/2015 05:48	3193 Lincoln Park	40.7246051	74.07840595	3193 Lincoln Park			
12	374	01/10/2015 05:44	01/10/2015 05:51	3201 Brunswick Park	40.7241765	-74.0506564	3186 Grove St PATH			
13	308	01/10/2015 05:44	01/10/2015 05:49	3209 Brunswick St	40.72476037	-74.0524783	3185 City Hall			
14	667	01/10/2015 05:52	01/10/2015 06:03	3207 Oakland Ave	40.7376037	-74.0524783	3185 City Hall			
15	312	01/10/2015 06:25	01/10/2015 06:31	3214 Essex Light Rail	40.7327742	-74.0364857	3185 Grove St PATH			

New York City BikeShare

```
for file in glob.glob('*.csv'):
    df = pd.read_csv(file)
    df['start_time'] = pd.to_datetime(df['start_time'])
    df['end_time'] = pd.to_datetime(df['end_time'])

    # Fixing structural errors
    df['start_time'] = df['start_time'].dt.strftime('%Y-%m-%d %H:%M:%S')
    df['end_time'] = df['end_time'].dt.strftime('%Y-%m-%d %H:%M:%S')

    # Reordering columns
    df = df[['start_time', 'end_time', 'start_station_id', 'start_station_name', 'end_station_id', 'end_station_name', 'duration', 'trip_id', 'user_type', 'birth_year', 'gender']]

    # Dropped columns
    df = df.drop(['start_lat', 'start_lng', 'end_lat', 'end_lng'], axis=1)

    # Renamed categorical values
    df['user_type'] = df['user_type'].replace('Customer', 'Subscriber')
    df['user_type'] = df['user_type'].replace('Subscriber', 'Customer')

    # Fixed categorical values
    df['gender'] = df['gender'].replace('Male', 'Male')
    df['gender'] = df['gender'].replace('Female', 'Female')

    # Drop rows with missing data
    df = df.dropna()

    # Drop rows with invalid birth year
    df = df[df['birth_year'] > 1900]
```

New York City BikeShare

## Data cleaning process

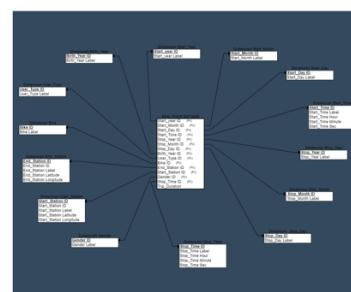
We used Python to manipulate the data.

- Fixed structural errors
- Reordered columns
- Dropped columns that contained the same information with others
- Renamed categorical values

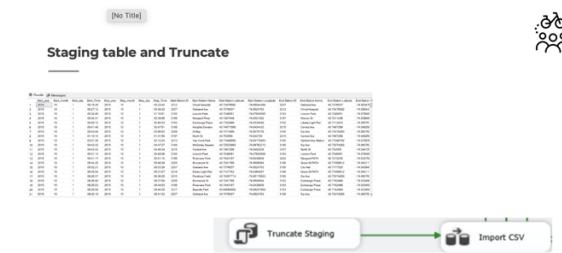
**The final dataset**

Start_Year	End_Year	Start_Month	Start_Day	Start_Hour	Start_Minute	Start_Sec	Start_Station	Start_Station_ID	Start_Station_Name	End_Station	End_Station_ID	End_Station_Name	
2015	2015	10	1	00:16:26	2015	50	1	00:22:42	3212 Christ Hospital	40.73478582	74.05044864	3307 Oakland Ave	
2015	2015	10	1	00:27:12	2015	50	1	00:39:32	3207 Oakland Ave	40.73478582	74.05044864	3212 Christ Hospital	
2015	2015	10	1	00:33:53	2015	50	1	00:49:53	3207 Oakland Ave	40.73478582	74.05044864	3307 Oakland Ave	
2015	2015	10	1	00:34:31	2015	50	1	00:49:06	1199 Newport Pkwy	40.7387448	-74.0321082	3187 Warren St	
2015	2015	10	1	00:40:12	2015	50	1	00:49:33	3183 Exchange Place	40.7562469	-74.0334588	3192 Liberty Light Rail	

New York City BikeShare



New York City BikeShare



New York City BikeShare

## Dimensions



Four small icons of people riding bicycles are positioned at the top right of the dimension section.

Results @ Messages	Results @ Messages	Results @ Messages	Results @ Messages
<code>d_Start_Year</code> , <code>d_End_Year</code>	<code>d_Start_Year</code> , <code>d_End_Year</code>	<code>d_Start_Year</code> , <code>d_End_Year</code>	<code>d_Start_Year</code> , <code>d_End_Year</code>
1	2	2	2
2015	2	1	1
2016	2	1	1
2017	2	1	1
2018	2	1	1

Results @ Messages	Results @ Messages	Results @ Messages	Results @ Messages
<code>d_Start_Month</code> , <code>d_End_Month</code>	<code>d_Start_Month</code> , <code>d_End_Month</code>	<code>d_Start_Month</code> , <code>d_End_Month</code>	<code>d_Start_Month</code> , <code>d_End_Month</code>
1	2	2	2
2	3	3	3
3	4	4	4
4	5	5	5
5	6	6	6
6	7	7	7
7	8	8	8
8	9	9	9
9	10	10	10
10	11	11	11
11	12	12	12
12	13	13	13

Results @ Messages	Results @ Messages	Results @ Messages	Results @ Messages
<code>d_Start_Hour</code> , <code>d_End_Hour</code>	<code>d_Start_Hour</code> , <code>d_End_Hour</code>	<code>d_Start_Hour</code> , <code>d_End_Hour</code>	<code>d_Start_Hour</code> , <code>d_End_Hour</code>
1	2	2	2
2	3	3	3
3	4	4	4
4	5	5	5
5	6	6	6
6	7	7	7
7	8	8	8
8	9	9	9
9	10	10	10
10	11	11	11
11	12	12	12
12	13	13	13
13	14	14	14
14	15	15	15
15	16	16	16
16	17	17	17
17	18	18	18
18	19	19	19
19	20	20	20
20	21	21	21
21	22	22	22
22	23	23	23
23	24	24	24
24	25	25	25
25	26	26	26
26	27	27	27
27	28	28	28
28	29	29	29
29	30	30	30
30	31	31	31
31	1	1	1

New York City Bikeshare

## Fact Table

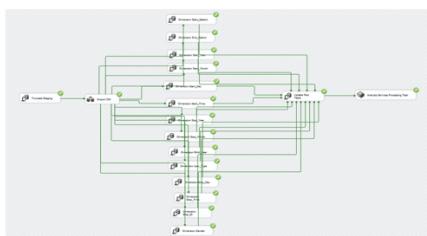


Two small icons of people riding bicycles are positioned at the top right of the fact table section.

Column Name	Date Type	Allow Nulls
Start_Hour	int	False
End_Hour	int	False
Start_Day	int	False
End_Day	int	False
Start_Month	int	False
End_Month	int	False
Start_Year	int	False
End_Year	int	False
Stop_Time	int	False
Drop_Time	int	False
Start_Station	int	False
End_Station	int	False
User_ID	int	False
User_Type	int	False
Birth_Year	int	False
Gender	int	False
Trip_Duration	int	False

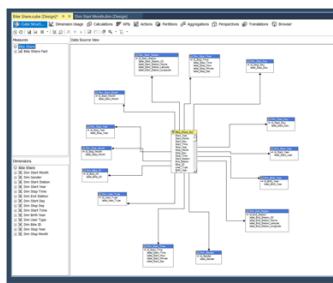
New York City Bikeshare

## Visual Studio

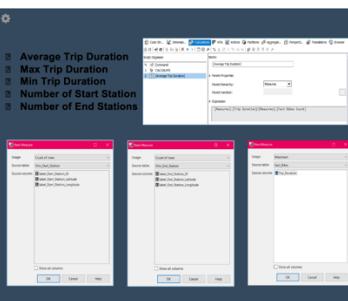


New York City Bikeshare

## Cube Calculations and Measures



New York City Bikeshare

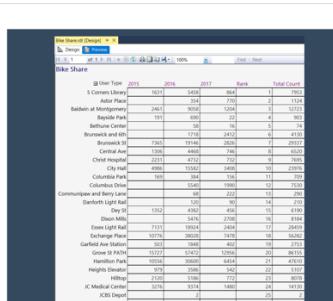


Three screenshots of an SSRS report interface. The first shows a report titled "Bike Share" with a single table containing data for "Average Trip Duration". The second and third screenshots show the report's preview and design tabs respectively, displaying the report's structure and data source.

## Cube Calculations and Measures

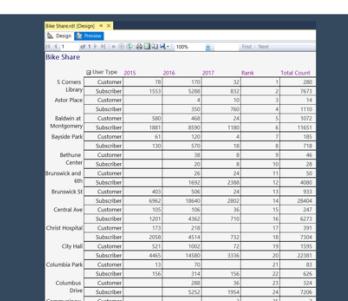
New York City Bikeshare

## Reporting Using SSRS



A screenshot of an SSRS report titled "Bike Share" showing a table of data for "Bike Share". The table includes columns for "User Type", "2015", "2016", "2017", "Rank", and "Total Count". The data shows the number of users for each category across three years.

New York City Bikeshare



A screenshot of an SSRS report titled "Bike Share" showing a table of data for "Bike Share". The table includes columns for "User Type", "2015", "2016", "2017", "Rank", and "Total Count". The data shows the number of users for each category across three years.

New York City Bikeshare

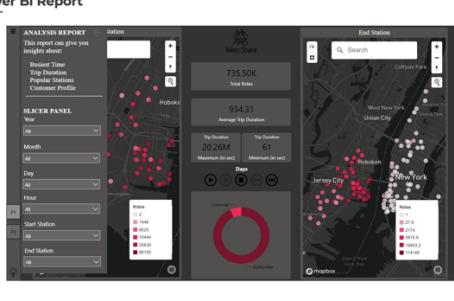
## Reporting Using SSRS

### Power BI Report



New York City Bikeshare

## Power BI Report



New York City Bikeshare

### Power BI Report

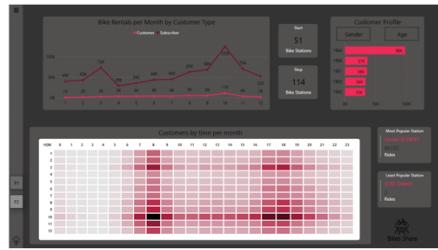


New York City Bikeshare

#### Power Bi Report



#### Power Bi Report



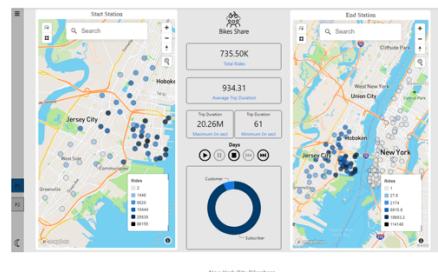
#### Power Bi Report



#### Power Bi Report



#### Power Bi Report



Thank  
You  
For Your  
Attention

