DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY

MASTER OF SCIENCE IN BUSINESS ANALYTICS

Lesson: Data Management and Business Intelligence

Professor: D. Chatziantoniou

Assignment: 2

Editors: Vogiatzis George, Zourou Myrsini

AM: P2821827, P2821828

Date: Monday 17 December 2018

Table of Contents

1. Data Description 3
2. Description of the fact table and the dimensions 9
2.1 Deletions and alterations of our dataset. 10
2.2 Creation of fact and dimension tables in sql-server management studio. 13
2.3 Visual Studio – Cube Creation 18
2.4 Olap Reports 27
3. TABLEAU 29

1. Data Description

Road traffic accidents (RTAs) have emerged as an important public health issue, which needs to be tackled by a multi-disciplinary approach. The trend in RTA injuries and death is becoming alarming in many countries around the world. The number of fatal and disabling road accident is increasing day by day. Therefore, it is a real public health challenge for all the concerned agencies to prevent it. The approach to implement the rules and regulations available to prevent road accidents is often ineffective and half-hearted. Awareness creation, strict implementation of traffic rules, and scientific engineering measures can prevent this public health catastrophe. The dataset, which selected as part for this assignment, is relative to this serious social issue. Because contains informations about the accidents which take place in New York City about the last 6 six years (2012-2018). Specifically contains informations about the date, the time and the location of each accident, the number of the vehicles, which was collided, the numbers of the injured or death people per accident and the type of vehicle that take part in the accident. Also gives information about the reasons that led in the accident. The original form of the dataset was in NYOpenData and it is shown on the table below:

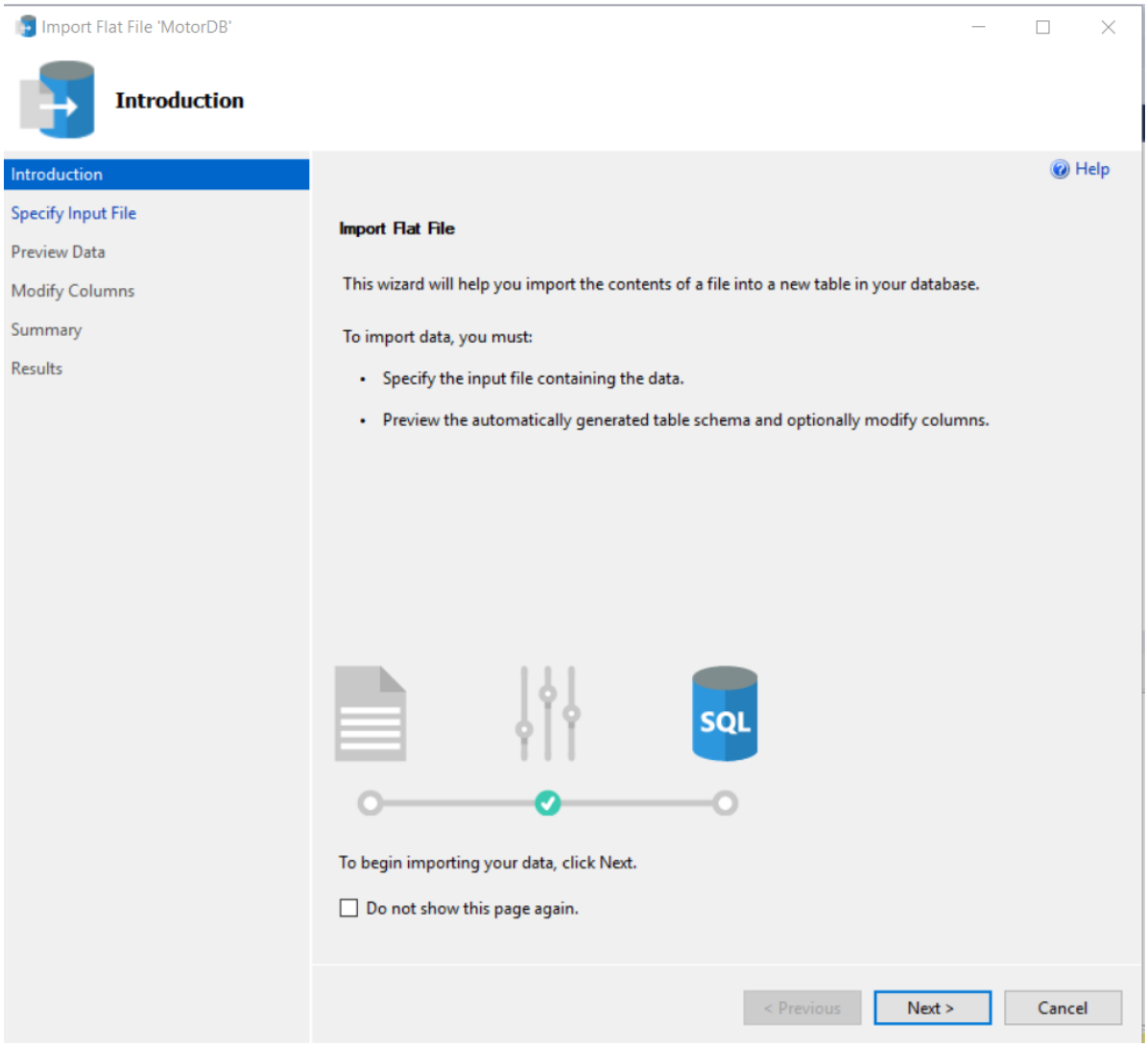
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITU...	LOCATION	ON STREET NAME	CROSS STREET NAME
11/16/2018	0:10	MANHATTAN	10010	40.742275	-73.988914	(40.742275°, -73.988...	5 AVENUE	BROADWAY
11/16/2018	0:45	BROOKLYN	11211	40.710197	-73.95843	(40.710197°, -73.958...	BORINQUEN PLACE	HAVEMEYER STREET
11/16/2018	0:40			40.76272	-73.72817	(40.76272°, -73.7281...	LONG ISLAND EXPRESSWAY	
11/16/2018	1:00	BRONX	10454	40.803555	-73.91184	(40.803555°, -73.911...	EAST 137 STREET	WILLOW AVENUE
11/16/2018	1:00	BROOKLYN	11221	40.694923	-73.915565	(40.694923°, -73.915...	WILSON AVENUE	PALMETTO STREET
11/16/2018	1:20			40.8047	-73.91243	(40.8047°, -73.91243°)	EAST 138 STREET	BRUCKNER BOULEVARD
11/16/2018	1:41			40.737682	-73.85206	(40.737682°, -73.852...	108 STREET	HORACE HARDING EXPRESSWAY
11/16/2018	2:50	BRONX	10451	40.819057	-73.92923	(40.819057°, -73.929...	EAST 149 STREET	GERARD AVENUE
11/16/2018	4:22			40.81175	-73.93144	(40.81175°, -73.9314...	MAJOR DEEGAN EXPRESSWAY	
11/16/2018	5:30	QUEENS	11417	40.67887	-73.83419	(40.67887°, -73.8341...	ROCKAWAY BOULEVARD	CENTREVILLE AVENUE

Through this analysis, we want to identify the most often reason which leads to an accident, the type of the cars which take part in an accident and to show the variation of the number of deaths or injuries by car accidents change through the years. Furthermore, we will specify which area of the following Bronx, Brooklyn, Manhattan, Queens or Staten Island has the more accidents. The results of our analysis could be used from the local authorities in order to define the tax for the violations of the road traffic code and the days, which happens the most accidents in order to be on hand.

The initial entries of our dataset were 1,385,920 lines and the dataset consisted of 29 columns in total, but after the ETL processes, the dataset ended up having 918,846 entries. We load the dataset into a local database ('ny_acc') through the Import Flat File option from the Microsoft SQL Server Management Studio. We chose datatype nvarchar(100) null for all the columns except for the metric ones which were imported as int, to avoid any insertion errors and we proceeded with the alteration of some types later. The steps, which follow to insert the data, were the following:

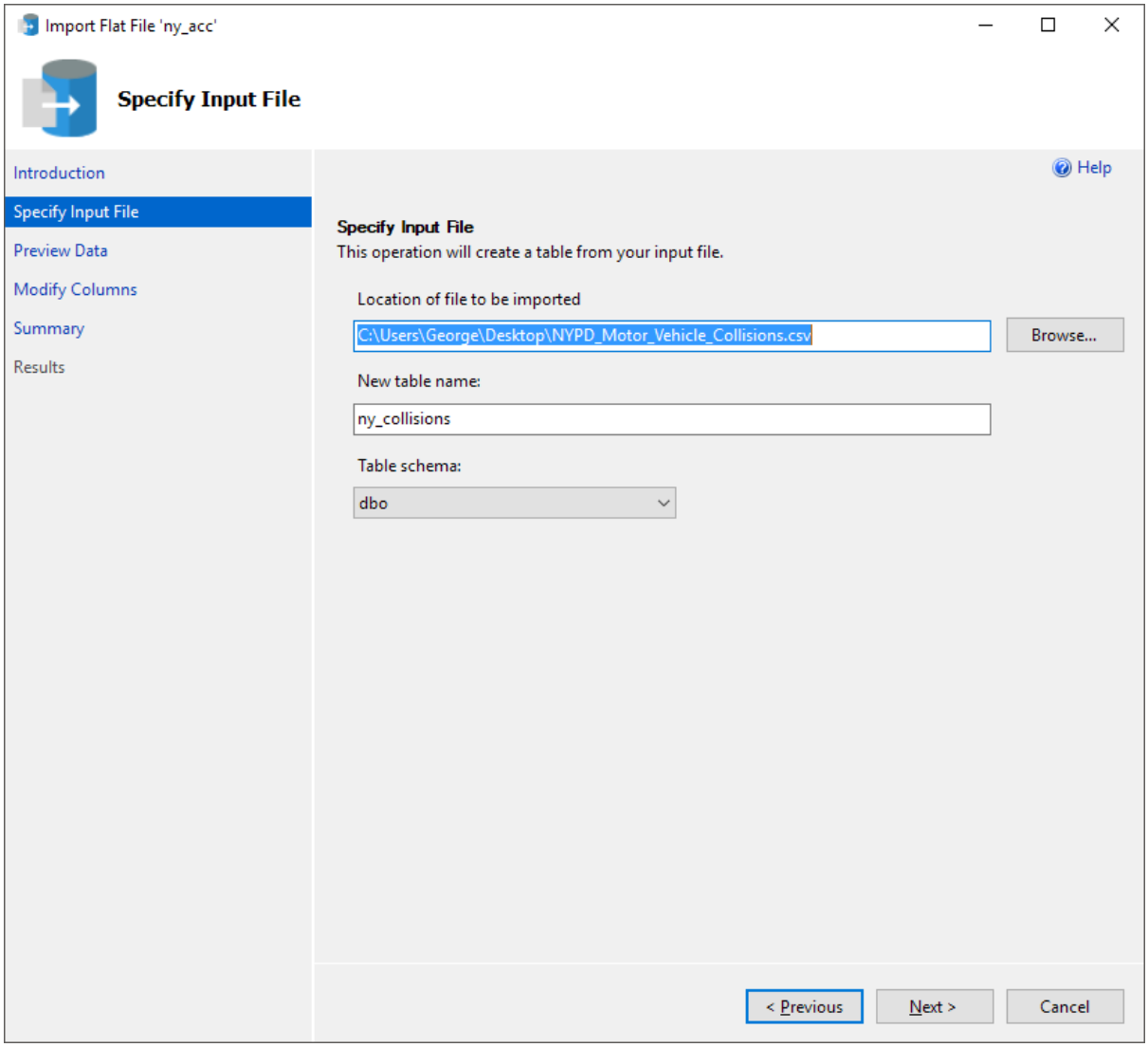
Step 1

The first page of the wizard is the welcome page



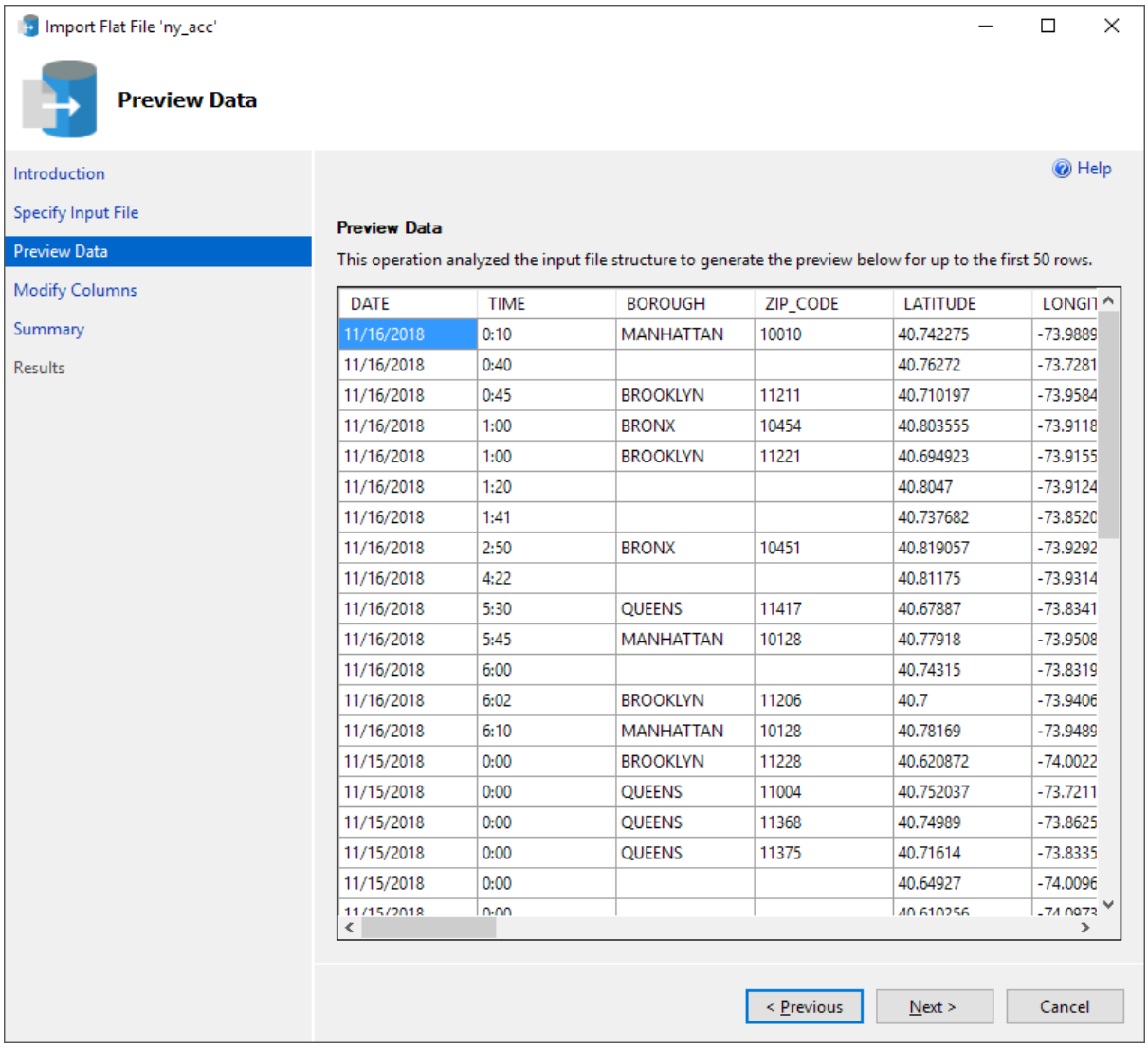
Step 2

We click the browse button to select the file of our dataset and after we give the new table name, which will contain our dataset.



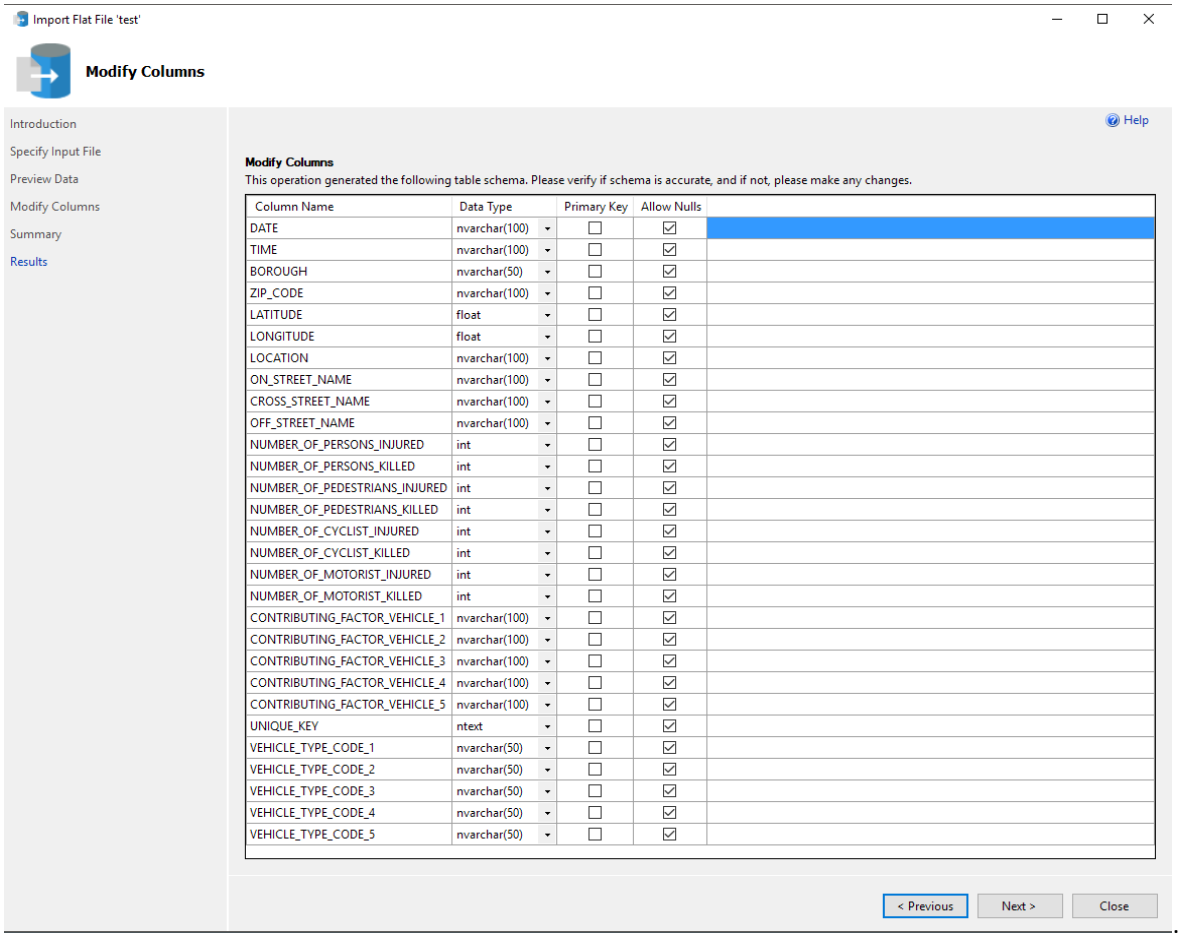
Step 3

The wizard generates a preview where you can view for the first 50 rows of the dataset.



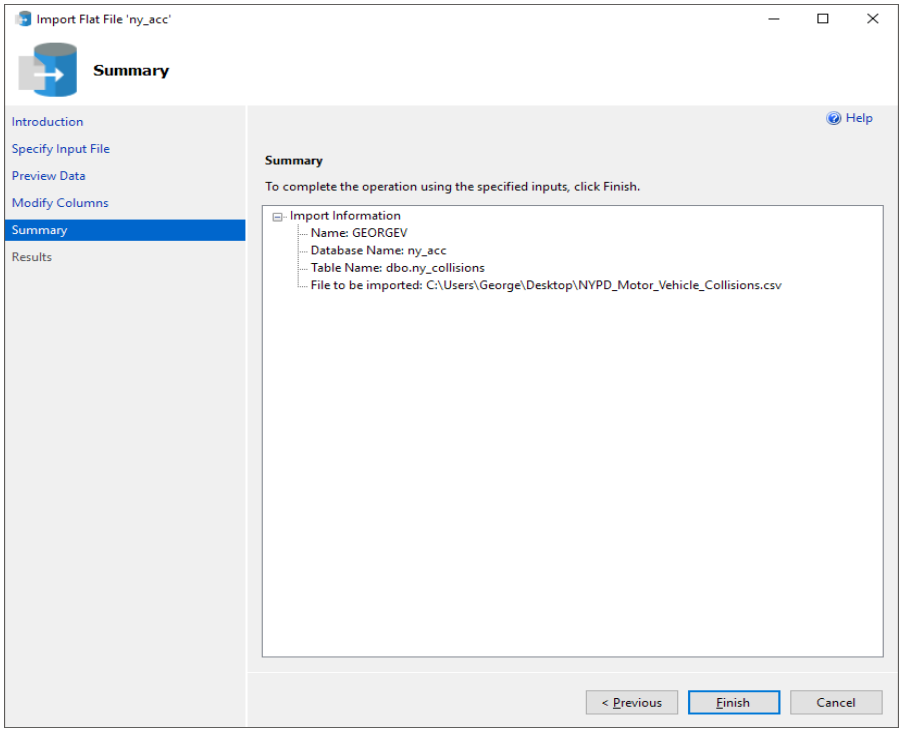
Step 4

In this step, the wizard identifies what data type believes that describe better our variables. However, we proceeded with some alterations in the proposed data types in order to insert the data with the right variables types, names, data types, etc



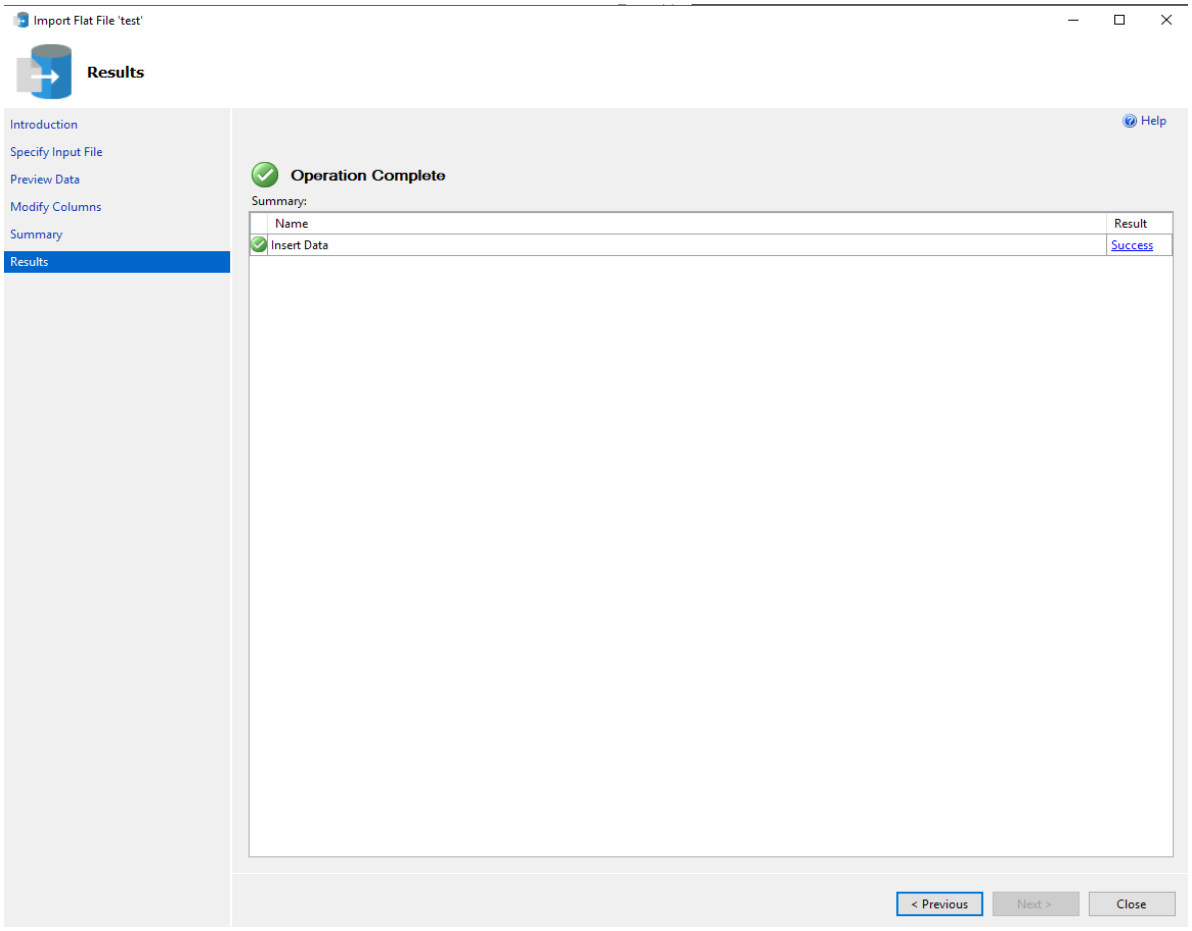
Step 5

This is simply a summary page displaying our current configuration. If there are issues, we should go back to previous sections to alter the problems, which appears. Otherwise, we click the finish bottom to attempt the import process.



Step 6

This page indicates whether the import was successful. If a green check mark appears, it was a success, otherwise you may need to review our configuration or input file for any errors.



2. Description of the fact table and the dimensions

After the successful import of the dataset, we continued with the determination of the fact table and the dimensions tables, which are connected. Our dataset as we already mentioned, refers to traffic accidents occurred in New York City the last 6 years. Fact table will represent all the information about each accident and the dimensions table will give further information about some variables. Our dimensions are the following tables:

- Time of accident
- Date of accident
- Location of accident
- Factors of accident
- Type of vehicles

In the next section, we will analyze in a more descriptive way the configuration of the fact table and of the dimensions. Just before, we created the schema of our database we had to clean the data in order to be in a desired form and organize them properly. In the next section, we explain how we do the cleansing of our data and their configuration.

2.1 Deletions and alterations of our dataset.

When we inserted the flat file on the server we had to define almost all datatypes as nvarchar or integer (if they were measures) to avoid getting insertion errors. Therefore, after this procedure we had to change some datatypes in order to represent better their fields.

We have successfully load the dataset in the database with the name Collisions. After that, we searched if our data had some unreasonable values. Specifically every line of our dataset should describe one vehicle collision.

Firstly, we change the empty cells of the variables with NULL and we continue with the deletion of NULLS for the following variable values: Longitude, Latitude, Zip Code, contributing factor vehicle and type of the vehicle because we want to keep the lines, which have all the information. Secondly, we delete some observations that do not make sense and maybe are wrong entries. For example, latitude and longitude equal to zero should be deleted because we know that our data refers to New York City and this coordination represent a position in Atlantic Ocean. In addition, we delete the observations where the contributing factor vehicle was equal to 1 and 80 because these observations did not reveal any reason why the accident happened. Below we give the query, which we use to do the aforementioned deletions.

Query1:

```
alter table [ny_acc].[dbo].[ny_collisions] alter column UNIQUE_KEY nvarchar(50);

update [ny_acc].[dbo].[ny_collisions] set ZIP_CODE = null  where  ZIP_CODE = ' ';
update [ny_acc].[dbo].[ny_collisions] set ON_STREET_NAME = null  where  ON_STREET_NAME = ' ';

delete from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_1 is Null;

delete from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_1 is Null
and VEHICLE_TYPE_CODE_2 is Null
and VEHICLE_TYPE_CODE_3 is Null
and VEHICLE_TYPE_CODE_4 is Null
and VEHICLE_TYPE_CODE_5 is Null

delete from [ny_acc].[dbo].[ny_collisions] where LOCATION is Null;
delete from [ny_acc].[dbo].[ny_collisions]  where ZIP_CODE is null;
delete from [ny_acc].[dbo].[ny_collisions] where LATITUDE = 0 and LONGITUDE = 0;

delete from [ny_acc].[dbo].[ny_collisions]
where CONTRIBUTING_FACTOR_VEHICLE_1 = '80'
or CONTRIBUTING_FACTOR_VEHICLE_2 = '80'
or CONTRIBUTING_FACTOR_VEHICLE_3 = '80'
or CONTRIBUTING_FACTOR_VEHICLE_4 = '80'
or CONTRIBUTING_FACTOR_VEHICLE_5 = '80'
or CONTRIBUTING_FACTOR_VEHICLE_1 = '1'
or CONTRIBUTING_FACTOR_VEHICLE_2 = '1'
or CONTRIBUTING_FACTOR_VEHICLE_3 = '1'
or CONTRIBUTING_FACTOR_VEHICLE_4 = '1'
or CONTRIBUTING_FACTOR_VEHICLE_5 = '1'

alter table [ny_acc].[dbo].[ny_collisions] drop column OFF_STREET_NAME;
alter table [ny_acc].[dbo].[ny_collisions] drop column LOCATION;
alter table [ny_acc].[dbo].[ny_collisions] drop column CROSS_STREET_NAME;
```

Secondly, we delete the column Location of our dataset because it contains the same information with the columns Longitude and Latitude. Furthermore, we delete the columns with the name 'OFF_STREET_NAME' and 'CROSS_STREET_NAME', because most of the rows were empty and generally the information provided by

these fields is unnecessary for our analysis. Below we can see the query we used to do the aforementioned deletions.

Query 2:

```
alter table [collisions].[dbo].[Collisions] drop column OFF_STREET_NAME;
alter table [collisions].[dbo].[Collisions] drop column LOCATION;
alter table [collisions].[dbo].[Collisions] drop column CROSS_STREET_NAME;
```

Furthermore, we altered the type of the 'TIME' variable, which we inserted as varchar. We transform it in time type because it will suit better our analysis tools. The last step of the deletion process was to alter the entries in the columns CONTRIBUTING_FACTOR_VEHICLE and VEHICLE_TYPE_CODE because there were categories which described the same entity but with different literal so we altered the names in order to group in the same category. For example, the VEHICLE_TYPE_CODE in the beginning had 380 unique categories and after the process, we end up with 46 unique categories. We alter the names which descried the same car type but with the abbreviation and not with the total name. In addition, we added many vehicle types in the category unknown because it was not clear in which vehicle type they refer to. Below we can see the queries, we executed concerning the CONTRIBUTING_FACTOR_VEHICLE records and the first column of vehicle type from a total of five columns. However, we executed the same query for the rest vehicles' contributing factor records.

Query 4:

CONTRIBUTING_FACTOR_VEHICLE

```
update [collisions].[dbo].[Collisions] set CONTRIBUTING_FACTOR_VEHICLE_1 = 'Illness'
where  CONTRIBUTING_FACTOR_VEHICLE_1 = 'Illnes'
update [collisions].[dbo].[Collisions] set CONTRIBUTING_FACTOR_VEHICLE_2 = 'Illness'
where  CONTRIBUTING_FACTOR_VEHICLE_2 = 'Illnes'

UPDATE [collisions].[dbo].[Collisions] SET CONTRIBUTING_FACTOR_VEHICLE_1 = 'Reaction to Uninvolved Vehicle'
where CONTRIBUTING_FACTOR_VEHICLE_1 in ('Reaction to Other Uninvolved Vehicle');
UPDATE [collisions].[dbo].[Collisions] SET CONTRIBUTING_FACTOR_VEHICLE_2 = 'Reaction to Uninvolved Vehicle'
where CONTRIBUTING_FACTOR_VEHICLE_2 in ('Reaction to Other Uninvolved Vehicle');
UPDATE [collisions].[dbo].[Collisions] SET CONTRIBUTING_FACTOR_VEHICLE_3 = 'Reaction to Uninvolved Vehicle'
where CONTRIBUTING_FACTOR_VEHICLE_3 in ('Reaction to Other Uninvolved Vehicle');
UPDATE [collisions].[dbo].[Collisions] SET CONTRIBUTING_FACTOR_VEHICLE_4 = 'Reaction to Uninvolved Vehicle'
where CONTRIBUTING_FACTOR_VEHICLE_4 in ('IReaction to Other Uninvolved Vehicle');
UPDATE [collisions].[dbo].[Collisions] SET CONTRIBUTING_FACTOR_VEHICLE_5 = 'Reaction to Uninvolved Vehicle'
where CONTRIBUTING_FACTOR_VEHICLE_5 in ('Reaction to Other Uninvolved Vehicle');
```

VEHICLE_TYPE_CODE

```
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'AMBULANCE'
where VEHICLE_TYPE_CODE_1 in ('Ambulance','AMBUL','AM','Ambul','Ambul','ambul','AMB','AMBU','ambu','AMBULANCEANCE','ABULA');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'DELIV'
where VEHICLE_TYPE_CODE_1 in ('deliv','Deliv','delv','DELV','DELV.');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'ARMORED_TRUCK'
where VEHICLE_TYPE_CODE_1 in ('armor','ARMOR','AR','Armored Truck');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'EBIKE'
where VEHICLE_TYPE_CODE_1 in ('E BIK','E-BIK','EBIKE','e-bik','E SCO','e sco','elect','ELECT');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'VAN' where VEHICLE_TYPE_CODE_1 in
('VAN','van','REFRI','Refrigerated Van','RF','MINIV','GMC V','Van Camper','VAN T','VAN/B','VAN/T','Vanete','VN','Vanette');
update [collisions].[dbo].[Collisions] set VEHICLE_TYPE_CODE_1 = 'UTILITY VEHICLES'
where VEHICLE_TYPE_CODE_1 in ('UTILI','Utili','utili','UTIL','UT','Util','BOB C','bobca','uliti','BR','BROOM','Bucke','bulld','BULLD',
'CRANE','UTLL','cemen','CMIX','Concrete Mixer','CONST','Hopper','JCB40','JLG B','Lift','Lift Boom','Lunch Wagon','CAT','cat 3','CATER',
'Pallet','scaff','utlit','FORK','Forkl','FORK-','forkl','FORKL','FORKLIFT','Well Driller');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'UNKNOWN' where VEHICLE_TYPE_CODE_1 in ('Unkno','unkno',
'unk','UKN','OTHER','N/A','NA','ND','Enclosed Body - Removable Enclosure','stree','(ceme','00','013','1',
'15 Pa','18 WH','1S','315 e','994','ACCES','B5-44','BA','BACK','BACKH','BS','CASE','CB','CHERR','DUNBA','e com','E ONE','E PAS','E-MOT','EMRGN','EMS',
'EMS H','EN','ENGIN','f550','FORTL','FREE','FREIG','frieg','FRONT','G Sem','Glass Rack','GN','GR','H/WH','Hand','HELP','HIGHL','HI-LO','HINO','HO',
'HOTDO','HWY C','Inter','IP','JOHN','LIGHT','LL','LP','RYDER','ROAD','RGS','REP','Unk,','Unkn','UNKOW','Comix', 'CHART', 'Carri', 'east', 'GAS S',
'GE/SC', 'LF', 'LW', 'MACK', 'MAN L', 'MARK', 'Marke', 'MB', 'mcy', 'MD', 'Mecha', 'MH', 'MILLI', 'MK', 'NEW Y', 'NS AM', 'NYC A', 'nyc d', 'NYC F',
'OBJEC', 'OMR', 'OP', 'Open Body','PL', 'PM', 'Porta', 'POWER', 'PSD', 'PUMP', 'SC', 'RD/S', 'red', 'refg','RENTA', 'RESCU',  'seagr', 'SELF', 'SELF-',
'small', 'SP', 'SPC', 'spc p', 'spec', 'Sprin', 'ST', 'ST150', 'STAK', 'Subn', 'SUBN/', 'SUBUR', 'SWEEP', 'SWT', 'TL TR', 'Tlr', 'TRAFF', 'TRL', 'TRLPM',
'TRLR', 'TT', 'U.S', 'Uber', 'UD', 'U-HAU', 'UHUAL', 'VC', 'VERIZ', 'WC', 'WD', 'WHEEL', 'WORK','WORKH','work','p/sh','P/SH','nyc a','NYC a','NYC A','EB',
'TN', 'red,','U.S.' );
```

```
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'PASSENGER VEHICLE' where
VEHICLE_TYPE_CODE_1 in ('PAS','pas','passa','Passe','Pavin','2 DOO','2 dr sedan','3D','3DC-','3-Door','4 dr sedan','4DSD','4whee','Chevr','City','CONV',
'Convertible','ford','GOLF','GOVER','jeep','nissa','Sedan','Smart','WHITE','YELLO','SPORT UtilTY / STATION WAGON','Station Wagon/Sport Utility Vehicle',
'SPORT UTILITY / STATION WAGON','SPORT_UTILITY/STATION_WAGON''COUPE','GRAY','Humme','KENWO','MINI','Mini');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'TRUCK'
where VEHICLE_TYPE_CODE_1 in ('Truck','truck','uhaul','BOOM','BOOML','BOX','Beverage Truck','trk','tk','Pick-up TRUCK','Flat Rack','Flat Bed',
'Pickup with mounted Camper','pick','PK','PICKU','Pick-','PICK-UP TRUCK','FLAT','flat','FLATB','ice c','icecr','box t','BOX T','boxtr','COM.',
'COMER','COMME','COMMM','Food','COM T','COM','C0MME','CO','CM','HEAVY','LARGE COM VEH(6 OR MORE TIRES)','WAGON','TOW T','Tow T','Tow t','tow t',
'TOW','tow','SEMI-','SEMI','Semi','FB','CARGO','Cargo','Tow Truck,','Tow Truck / Wrecker','TOW_TRUCK','TR','2- to','2 TON','Box Truck', 'OIL T',
 'Tract','TRACT','tract','TRAC','Tractor Truck Diesel','Tractor Truck Gasoline','Tow Truck','TRANS');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'TRAILER' where VEHICLE_TYPE_CODE_1 in ('TRAI','trailL','rv','TrailL','trail','trail');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'BICYCLE' where VEHICLE_TYPE_CODE_1 in ('mta b','MTA B','Bike','BK','Minibike','Minicycle');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'TAXI' where VEHICLE_TYPE_CODE_1 in ('Taxi','CAB');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'Tanker' where VEHICLE_TYPE_CODE_1 in ('TANK');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'Subn' where VEHICLE_TYPE_CODE_1 in ('SUBN','subn');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'SPC' where VEHICLE_TYPE_CODE_1 in ('spc p|Spc');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'SNOW PLOW' where VEHICLE_TYPE_CODE_1 in ('SNOW','Snow Plow');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'SCOOTER'
where VEHICLE_TYPE_CODE_1 in ('SCOOT','Scoot','scoot','SCOOTERER','Motorscooter','MS','mot s');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'BUS'
where VEHICLE_TYPE_CODE_1 in ('bus','BUS','SCHOO','Schoo','schoo','School Busl Bus','School Busl Bus','BU','School Bus','omni', 'omnib');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'GARBAGE_TRUCK'
 where VEHICLE_TYPE_CODE_1 in ('garba','GARBA','Garba','Garbage or Refuse','SANIT','sanit','Sanit','DS','dsny','Dump','DUMPT','dump','DUMP','Dumps');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'FIRE_TRUCK'
where VEHICLE_TYPE_CODE_1 in ('FIRE','Firet','fire','Fd fi','FIRET','Fire','FIRE TRUCK TRUCK','fdny','FDNY');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'MINI' where VEHICLE_TYPE_CODE_1 in ('Mini');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'FORD' where VEHICLE_TYPE_CODE_1 in ('ford');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'FORKLIFT' where VEHICLE_TYPE_CODE_1 in ('FORK','Forkl','FORK-','forkl','FORKL');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'MOTORCYCLE' where VEHICLE_TYPE_CODE_1 in ('MOTORCYCLE','motor','MOTOR','Motorbike');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'FIRE TRUCK'
where VEHICLE_TYPE_CODE_1 in ('FIRE TRUCK TRUCK','fd tr','FD tr','FIRE','FIRET','FR','FD LA');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'LIMO' where VEHICLE_TYPE_CODE_1 in ('LIMO','Limou');
```

```
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'MAIL'
where VEHICLE_TYPE_CODE_1 in ('mail','MAIL','UPS t','usps','US PO','POSTA','POSTO','posta','FEDEX','DP');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'MOPED' where VEHICLE_TYPE_CODE_1 in ('Mo pa','mopad','MOPED');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'PEDICAB' where VEHICLE_TYPE_CODE_1 in ('Pedicab','PEDIC');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'GOVERNMENT' where VEHICLE_TYPE_CODE_1 in ('GG','FED E');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'FARM VEHICLE' where VEHICLE_TYPE_CODE_1 in ('FARM','Bulk Agriculture');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'LIVERY VEHICLE' where VEHICLE_TYPE_CODE_1 in ('LIVER');
UPDATE [collisions].[dbo].[Collisions] SET VEHICLE_TYPE_CODE_1 = 'CAMPING VEHICLE' where VEHICLE_TYPE_CODE_1 in ('CABIN','CAMP','CAMPE');
```

We also altered the data type of column Time from nvarchar(100) to time(7) because the SSAS tool could handle this column in its native form, which is time.

```
alter table [ny_acc].[dbo].[ny_collisions] add  coltime time;
UPDATE [ny_acc].[dbo].[ny_collisions]  SET coltime =  convert (time, time,8);
alter table [ny_acc].[dbo].[ny_collisions] alter column time nvarchar(100) NULL;
UPDATE [ny_acc].[dbo].[ny_collisions]  SET TIME = null;
UPDATE [ny_acc].[dbo].[ny_collisions]  SET time = coltime;
alter table [ny_acc].[dbo].[ny_collisions] alter column time time(7) NOT NULL;
alter table [ny_acc].[dbo].[ny_collisions] drop column coltime;
```

Apart from Time column we altered the type of ‘Latitude’ and ‘Longitude’ columns from float to Decimal(9,6) so that the bi tool could understand these two columns as geographical.

```
UPDATE [ny_acc].[dbo].[ny_collisions]
SET LATITUDE = CAST(LATITUDE as decimal(9, 6))
ALTER TABLE [dbo].[ny_collisions] ALTER COLUMN LATITUDE DECIMAL(9,6);

UPDATE [ny_acc].[dbo].[ny_collisions]
SET LONGITUDE = CAST(LONGITUDE as decimal(9, 6))
ALTER TABLE [dbo].[ny_collisions] ALTER COLUMN LONGITUDE DECIMAL(9,6);
```

2.2 Creation of fact and dimension tables in sql-server management studio.

We decided to use the star schema for the creation of our tables. In this section, we describe the creation of the dimension tables along with the steps followed for the creation of the fact table.

Location:

We used the below queries to create the table with all the necessary features of Location dimension. The columns of our original dataset that could be included in the Location are the borough of the accident, the street name, the zip code and the latitude and longitude variables. But we observe that there is no unique key that can be set as primary for this dimension, so we used a group-by for all these characteristics that together make up a complete address and we transferred the results of this group-by in 'loc_table' and created the address_id. Finally, we assign the content of 'loc_table' table in our Location dimension table.

```
--LOCATION

create table loc_table(  borough nvarchar(100), on_street_name nvarchar(100), cross_street_name nvarchar(100),
zip_code nvarchar(100),longitude decimal(9,6),latitude decimal(9,6));

insert into loc_table (borough,on_street_name,zip_code,longitude,latitude)
select BOROUGH, ON_STREET_NAME, ZIP_CODE,LONGITUDE, LATITUDE FROM [ny_acc].[dbo].[ny_collisions]
group by  BOROUGH, ON_STREET_NAME, ZIP_CODE,LONGITUDE, LATITUDE;

alter table loc_table
add location_id int identity(1,1);

select * from loc_table order by location_id;

create table Location(
location_id nvarchar(100),
borough nvarchar(100),
on_street_name nvarchar(100),
zip_code  nvarchar(100),
longitude decimal(9,6) ,
latitude decimal(9,6)
PRIMARY KEY(location_id));

insert into Location(location_id,borough,on_street_name,zip_code,longitude,latitude)
select location_id, borough,on_street_name,zip_code,longitude,latitude from loc_table;
```

Date:

For the Date dimension we used the date column of our original dataset. To achieve this we created a temp table where we inserted the distinct values of dates present in the dataset and afterwards we add the Date_id to represent the primary key. Finally, we assigned the content of the temp table in our Date dimension table where we set the Date_id as a primary key.

```
--DATE

create table Accident_Date1(Acc_Date nvarchar(100));

insert into Accident_Date1(Acc_Date) select distinct(DATE) from [ny_acc].[dbo].[ny_collisions];
alter table Accident_Date1  add Date_id  int identity(1,1);

create table Accident_Date (
Date_id nvarchar(100),
Acc_Date nvarchar(100)
primary key(Date_id)
);

insert into Accident_Date(Date_id,Acc_Date)select Date_id, Acc_Date from Accident_Date1;
```

Time:

Regarding the time dimension table, we created it in the same manner with Date table, simply by inserting the distinct values of time we found in the original dataset, and then assigning a time id as primary key.

```
--TIME

create table time1(Acc_Time time(7));

insert into time1(Acc_Time) select distinct(TIME) from [ny_acc].[dbo].[ny_collisions];
alter table time1  add Time_id  int identity(1,1);

create table Accident_Time (
Time_id nvarchar(100),
Acc_Time time(7)
primary key(Time_id)
);

insert into Accident_Time(Time_id,Acc_Time)select Time_id, Acc_Time from time1;
```

Accident_factor:

This table is about the factor that caused the accident. To populate this table we had to select the distinct values from the 5 different contributing factor columns of our original dataset.

```sql
--Accident Factor

create table factor( factor nvarchar(100));

insert into factor(factor)
 select factor from (
    select CONTRIBUTING_FACTOR_VEHICLE_1 as factor from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_1 is not null
    union all select CONTRIBUTING_FACTOR_VEHICLE_2 from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_2 is not null
    union all select CONTRIBUTING_FACTOR_VEHICLE_3 from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_3 is not null
    union all select CONTRIBUTING_FACTOR_VEHICLE_4 from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_4 is not null
    union all select CONTRIBUTING_FACTOR_VEHICLE_5 from [ny_acc].[dbo].[ny_collisions] where CONTRIBUTING_FACTOR_VEHICLE_5 is not null
) t1 group by factor order by factor;

alter table factor add factor_id int identity(1,1);

create table  Accident_factor(
factor_id nvarchar(100),
factor nvarchar(100)
primary key(factor_id));

insert into Accident_factor (factor_id, factor)
select factor_id, factor from factor;
```

Vehicle_type:

The specific table is about the vehicle types observed in all these accidents. It was created in the same manner as the accident_factor. Therefore, we had to perform a union on all vehicle type codes to extract the distinct values, which will be in the vehicle_type dimension table.

```sql
--vehicle type code
create table vtype( vtype nvarchar(100));

 insert into vtype(vtype)
 select vtype from (
    select VEHICLE_TYPE_CODE_1 as vtype from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_1 is not null
    union all select VEHICLE_TYPE_CODE_2 from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_2 is not null
    union all select VEHICLE_TYPE_CODE_3 from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_3 is not null
    union all select VEHICLE_TYPE_CODE_4 from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_4 is not null
    union all select VEHICLE_TYPE_CODE_5 from [ny_acc].[dbo].[ny_collisions] where VEHICLE_TYPE_CODE_5 is not null
) t1 group by vtype order by vtype;

alter table vtype add vtype_id int identity(1,1);

create table  Vehicle_type(
vtype_id nvarchar(100),
vtype nvarchar(100)
primary key(vtype_id));

insert into Vehicle_type (vtype_id, vtype)
select vtype_id, vtype from vtype;
```

Fact_table:

Finally, to complete the star schema, we have to create our facttable, which is about accidents and has as foreign keys the primary keys of our dimension tables. As metrics, we defined the persons_injured, persons_killed, cyclists_injured, cyclists_killed, pedestrians_injured, pedestrians_killed, motorists_injured and motorists_killed, which represent the populations of victims in these accidents. To create our fact table we had to alter out original table containing the dataset (ny_collisions) and add the id columns of our dimensions.

```
alter table [ny_acc].[dbo].[ny_collisions]
add Date_id nvarchar(100);

alter table [ny_acc].[dbo].[ny_collisions]
add Time_id nvarchar(100);

alter table [ny_acc].[dbo].[ny_collisions]
add Factor_id1 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Factor_id2 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Factor_id3 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Factor_id4 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Factor_id5 nvarchar(100);
```

```
alter table [ny_acc].[dbo].[ny_collisions]
add Vtype_id1 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Vtype_id2 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Vtype_id3 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Vtype_id4 nvarchar(100);
alter table [ny_acc].[dbo].[ny_collisions]
add Vtype_id5 nvarchar(100);

alter table [ny_acc].[dbo].[ny_collisions]
add Location_id nvarchar(100);
```

The next step was to insert the corresponding values in the id columns:

```
update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Date_id = [ny_acc].[dbo].[Accident_Date].Date_id
from [ny_acc].[dbo].[Accident_Date]
where [ny_acc].[dbo].[ny_collisions].DATE = [ny_acc].[dbo].[Accident_Date].Acc_Date

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Time_id = [ny_acc].[dbo].[Accident_Time].Time_id
from [ny_acc].[dbo].[Accident_Time]
where [ny_acc].[dbo].[ny_collisions].TIME = [ny_acc].[dbo].[Accident_Time].Acc_Time

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Factor_id1 = [ny_acc].[dbo].[Accident_factor].factor_id
from [ny_acc].[dbo].[Accident_factor]
where [ny_acc].[dbo].[ny_collisions].CONTRIBUTING_FACTOR_VEHICLE_1 = [ny_acc].[dbo].[Accident_factor].factor

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Factor_id2 = [ny_acc].[dbo].[Accident_factor].factor_id
from [ny_acc].[dbo].[Accident_factor]
where [ny_acc].[dbo].[ny_collisions].CONTRIBUTING_FACTOR_VEHICLE_2 = [ny_acc].[dbo].[Accident_factor].factor

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Factor_id3 = [ny_acc].[dbo].[Accident_factor].factor_id
from [ny_acc].[dbo].[Accident_factor]
where [ny_acc].[dbo].[ny_collisions].CONTRIBUTING_FACTOR_VEHICLE_3 = [ny_acc].[dbo].[Accident_factor].factor

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Factor_id4 = [ny_acc].[dbo].[Accident_factor].factor_id
from [ny_acc].[dbo].[Accident_factor]
where [ny_acc].[dbo].[ny_collisions].CONTRIBUTING_FACTOR_VEHICLE_4 = [ny_acc].[dbo].[Accident_factor].factor

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Factor_id5 = [ny_acc].[dbo].[Accident_factor].factor_id
from [ny_acc].[dbo].[Accident_factor]
where [ny_acc].[dbo].[ny_collisions].CONTRIBUTING_FACTOR_VEHICLE_5 = [ny_acc].[dbo].[Accident_factor].factor
```

```
update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Vtype_id1 = [ny_acc].[dbo].[Vehicle_type].vtype_id
from [ny_acc].[dbo].[Vehicle_type]
where [ny_acc].[dbo].[ny_collisions].VEHICLE_TYPE_CODE_1 = Vehicle_type.vtype

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Vtype_id2 = Vehicle_type.vtype_id
from [ny_acc].[dbo].[Vehicle_type]
where [ny_acc].[dbo].[ny_collisions].VEHICLE_TYPE_CODE_2 = Vehicle_type.vtype

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Vtype_id3 = [ny_acc].[dbo].[Vehicle_type].vtype_id
from [ny_acc].[dbo].[Vehicle_type]
where [ny_acc].[dbo].[ny_collisions].VEHICLE_TYPE_CODE_3 = Vehicle_type.vtype

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Vtype_id4 = [ny_acc].[dbo].[Vehicle_type].vtype_id
from [ny_acc].[dbo].[Vehicle_type]
where [ny_acc].[dbo].[ny_collisions].VEHICLE_TYPE_CODE_4 = Vehicle_type.vtype

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Vtype_id5 = Vehicle_type.vtype_id
from [ny_acc].[dbo].[Vehicle_type]
where [ny_acc].[dbo].[ny_collisions].VEHICLE_TYPE_CODE_5 = Vehicle_type.vtype

update [ny_acc].[dbo].[ny_collisions]
set [ny_acc].[dbo].[ny_collisions].Location_id = Location.location_id
from [ny_acc].[dbo].[Location]
where [ny_acc].[dbo].[ny_collisions].BOROUGH = Location.borough and
[ny_acc].[dbo].[ny_collisions].ZIP_CODE = Location.zip_code and
[ny_acc].[dbo].[ny_collisions].LONGITUDE = Location.longitude and
[ny_acc].[dbo].[ny_collisions].LATITUDE = Location.latitude;
```

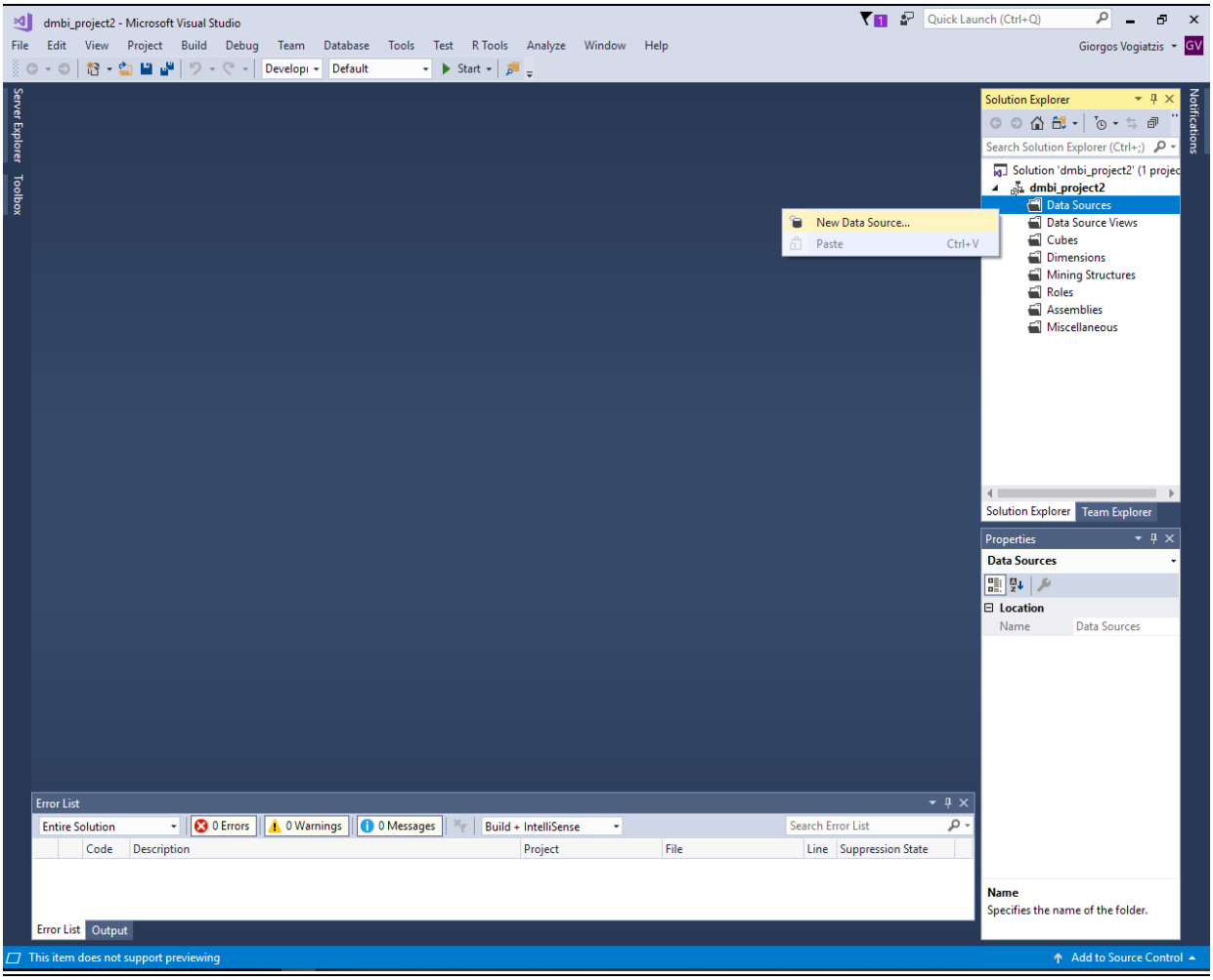
Final we conclude the procedure of the creation of the fact table and the insertion of the data we run the following queries.

```
create table fact_table
(
unique_key nvarchar(50),
time_id nvarchar(100),
date_id nvarchar(100),
location_id nvarchar(100),
persons_injured int,
persons_killed int,
pedestrians_injured int,
pedestrians_killed int,
cyclists_injured int,
cyclists_killed int,
motorists_injured int,
motorists_killed int,
factor_vehicle1_id nvarchar(100),
factor_vehicle2_id nvarchar(100),
factor_vehicle3_id nvarchar(100),
factor_vehicle4_id nvarchar(100),
factor_vehicle5_id nvarchar(100),
vtype1_id nvarchar(100),
vtype2_id nvarchar(100),
vtype3_id nvarchar(100),
vtype4_id nvarchar(100),
vtype5_id nvarchar(100),
foreign key(time_id) references [ny_acc].[dbo].[Accident_Time](Time_id),
foreign key(date_id) references [ny_acc].[dbo].[Accident_Date](Date_id),
foreign key(location_id) references [ny_acc].[dbo].[Location](location_id),
foreign key(factor_vehicle1_id) references [ny_acc].[dbo].[Accident_factor](factor_id),
foreign key(factor_vehicle2_id) references [ny_acc].[dbo].[Accident_factor](factor_id),
foreign key(factor_vehicle3_id) references [ny_acc].[dbo].[Accident_factor](factor_id),
foreign key(factor_vehicle4_id) references [ny_acc].[dbo].[Accident_factor](factor_id),
foreign key(factor_vehicle5_id) references [ny_acc].[dbo].[Accident_factor](factor_id),
foreign key(vtype1_id) references [ny_acc].[dbo].Vehicle_type(vtype_id),
foreign key(vtype2_id) references [ny_acc].[dbo].Vehicle_type(vtype_id),
foreign key(vtype3_id) references [ny_acc].[dbo].Vehicle_type(vtype_id),
foreign key(vtype4_id) references [ny_acc].[dbo].Vehicle_type(vtype_id),
foreign key(vtype5_id) references [ny_acc].[dbo].Vehicle_type(vtype_id)
)
```

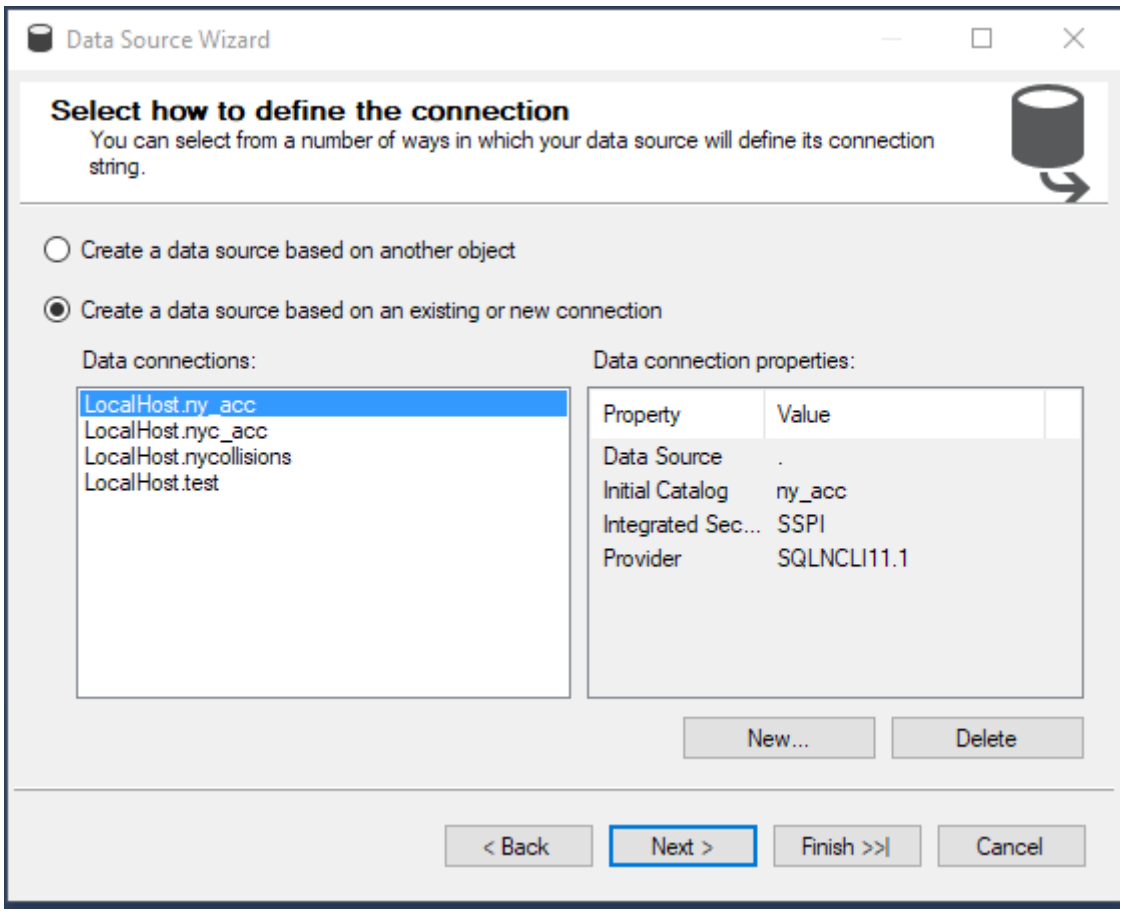
```
insert into fact_table (unique_key,time_id,date_id,location_id,persons_injured,persons_killed,pedestrians_injured,pedestrians_killed,
cyclists_injured,cyclists_killed,motorists_injured,motorists_killed,factor_vehicle1_id,factor_vehicle2_id,factor_vehicle3_id,factor_vehicle4_id,factor_vehicle5_id,
vtype1_id,vtype2_id,vtype3_id,vtype4_id,vtype5_id )
select UNIQUE_KEY,Time_id,Date_id,Location_id,NUMBER_OF_PERSONS_INJURED,NUMBER_OF_PERSONS_KILLED,NUMBER_OF_PEDESTRIANS_INJURED,NUMBER_OF_PEDESTRIANS_KILLED,
NUMBER_OF_CYCLIST_INJURED,NUMBER_OF_CYCLIST_KILLED,NUMBER_OF_MOTORIST_INJURED,NUMBER_OF_MOTORIST_KILLED,
Factor_id1,Factor_id2,Factor_id3,Factor_id4,Factor_id5,Vtype_id1,Vtype_id2,Vtype_id3,Vtype_id4,Vtype_id5
from [ny_acc].[dbo].[ny_collisions]
```

2.3 Visual Studio – Cube Creation

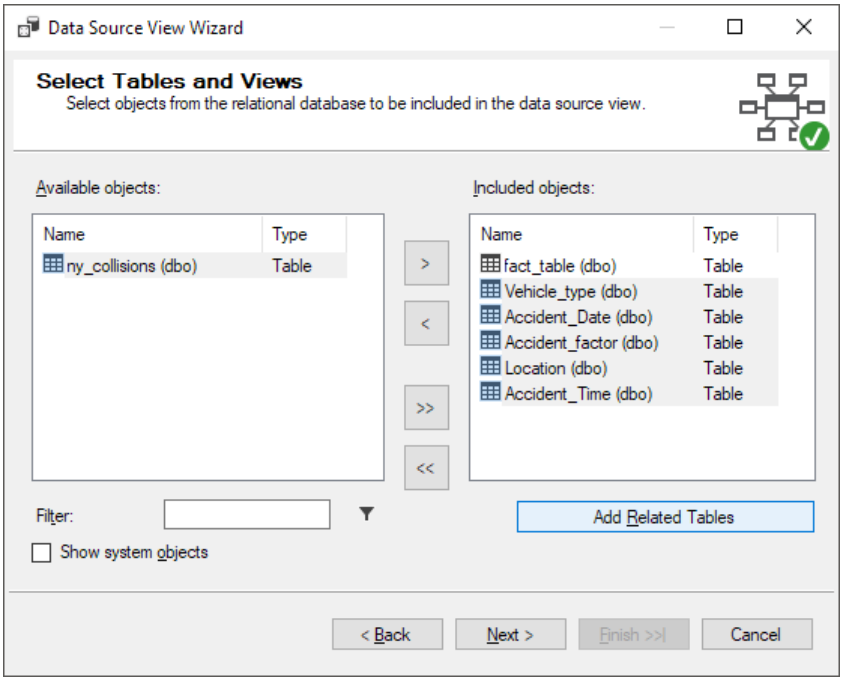
Initially, we created an analysis services multidimensional project. Screenshots provided show the steps followed to create the cube. Firstly, we choose the data source:

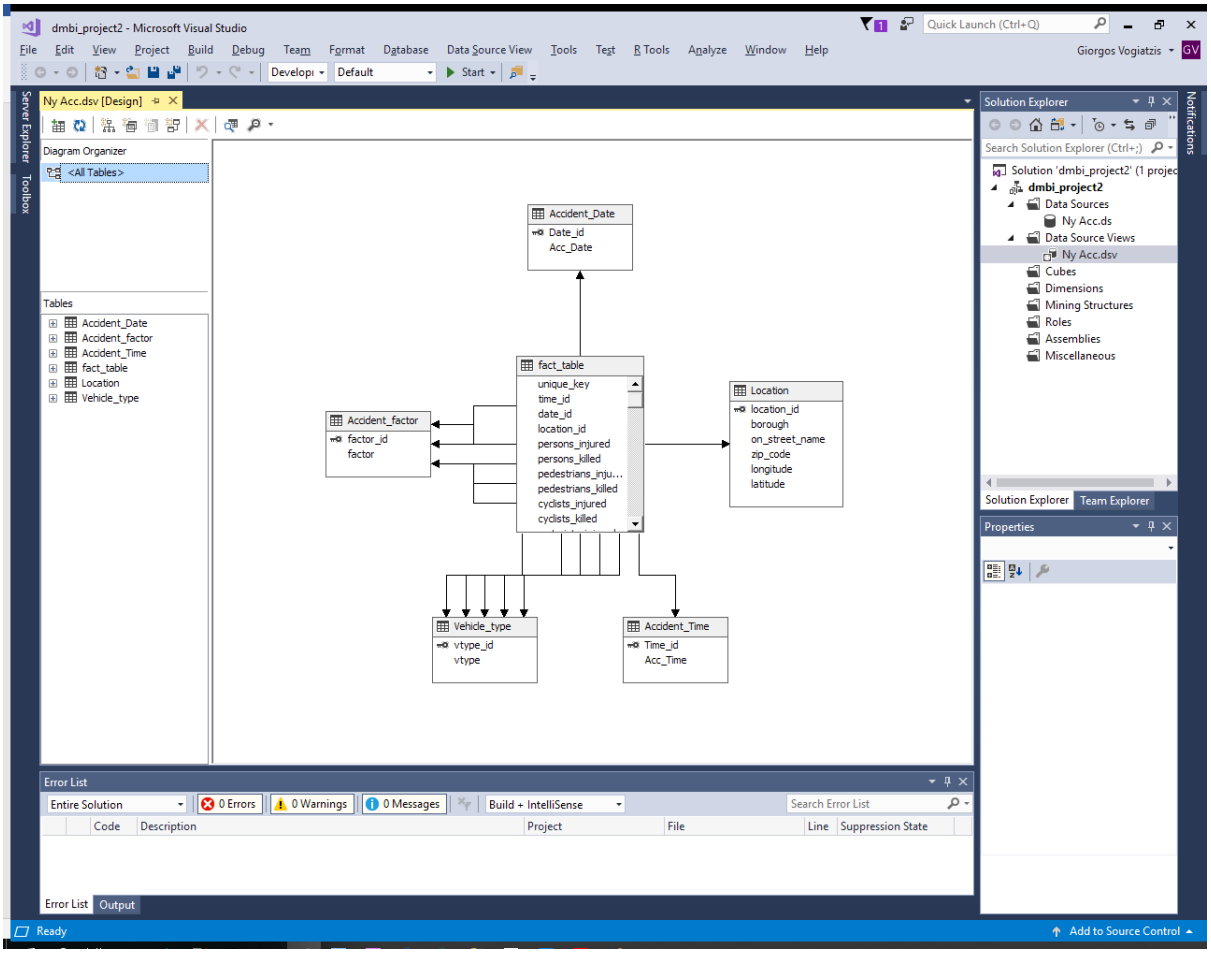

dmbi_project2 - Microsoft Visual Studio
File Edit View Project Build Debug Team Database Tools Test R Tools Analyze Window Help
Quick Launch (Ctrl+Q)
Giorgos Vogiatzis
Start
Server Explorer
Toolbox
Notifications
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'dmbi_project2' (1 projec
dmbi_project2
Data Sources
Data Source Views
Cubes
Dimensions
Mining Structures
Roles
Assemblies
Miscellaneous
New Data Source...
Paste Ctrl+V
Solution Explorer
Team Explorer
Properties
Data Sources
Location
Name
Data Sources
Name
Specifies the name of the folder.
Error List
Entire Solution
0 Errors
0 Warnings
0 Messages
Build + IntelliSense
Search Error List
Code
Description
Project
File
Line
Suppression State
Error List
Output
This item does not support previewing
Add to Source Control

Then, we prompted to choose our data connection

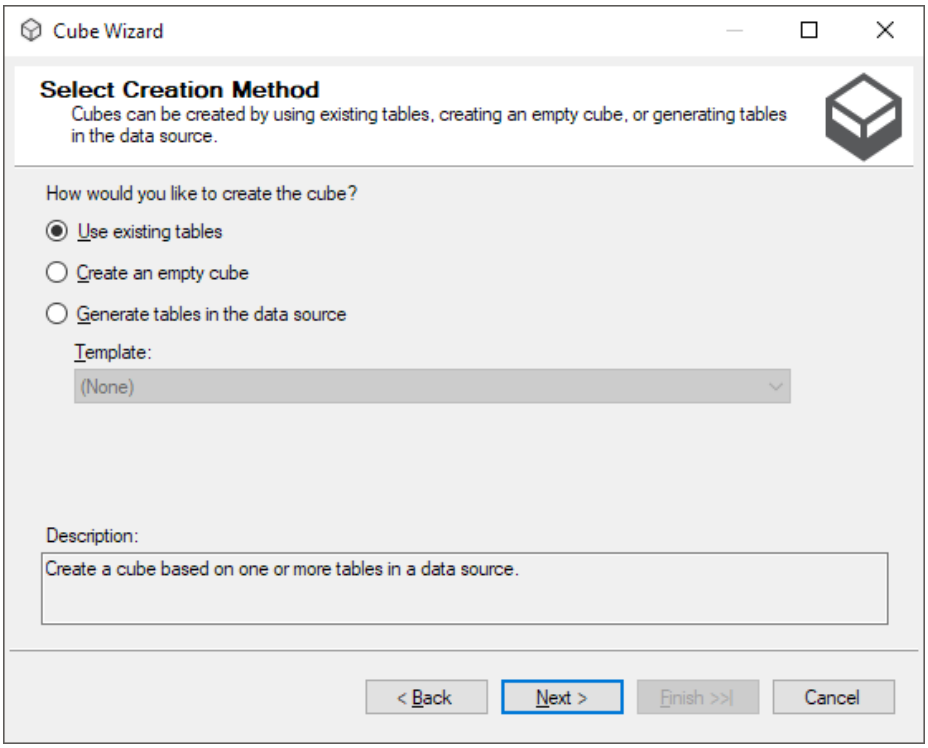


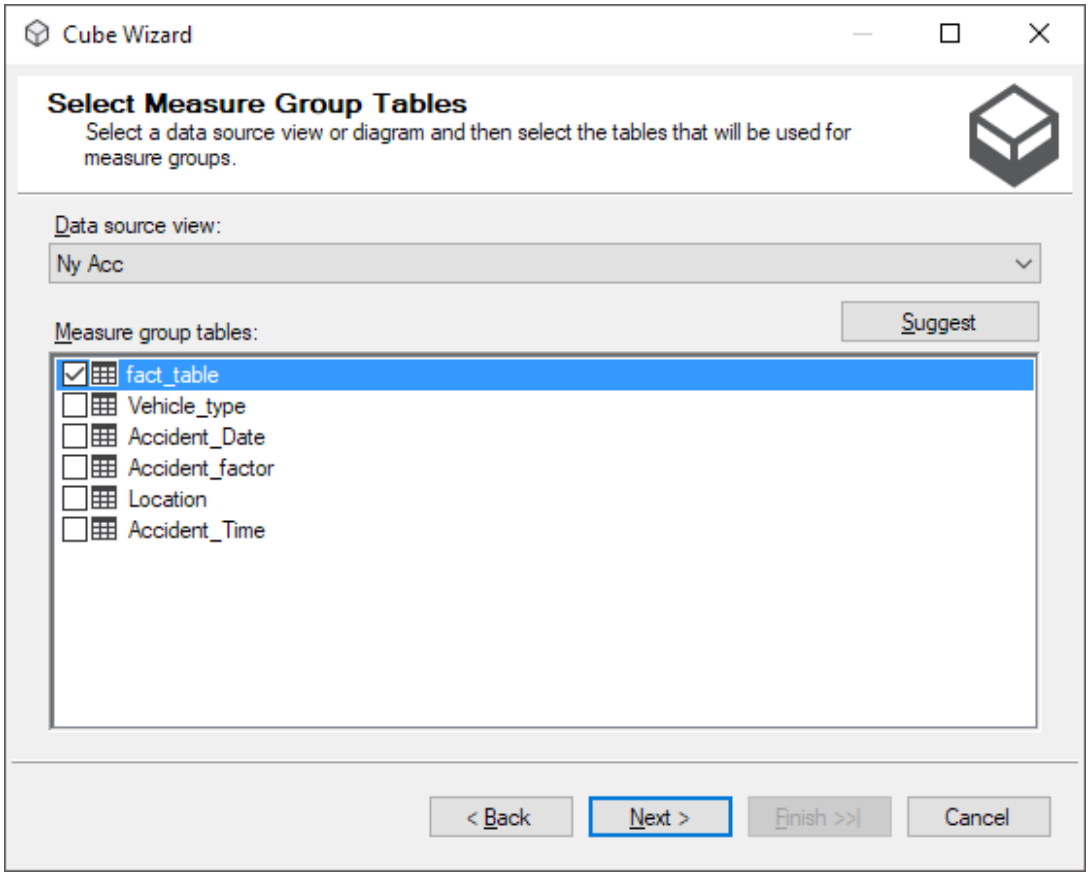
After, he had to create a new data source view and transfer our fact table and its related dimensions in the data source views.

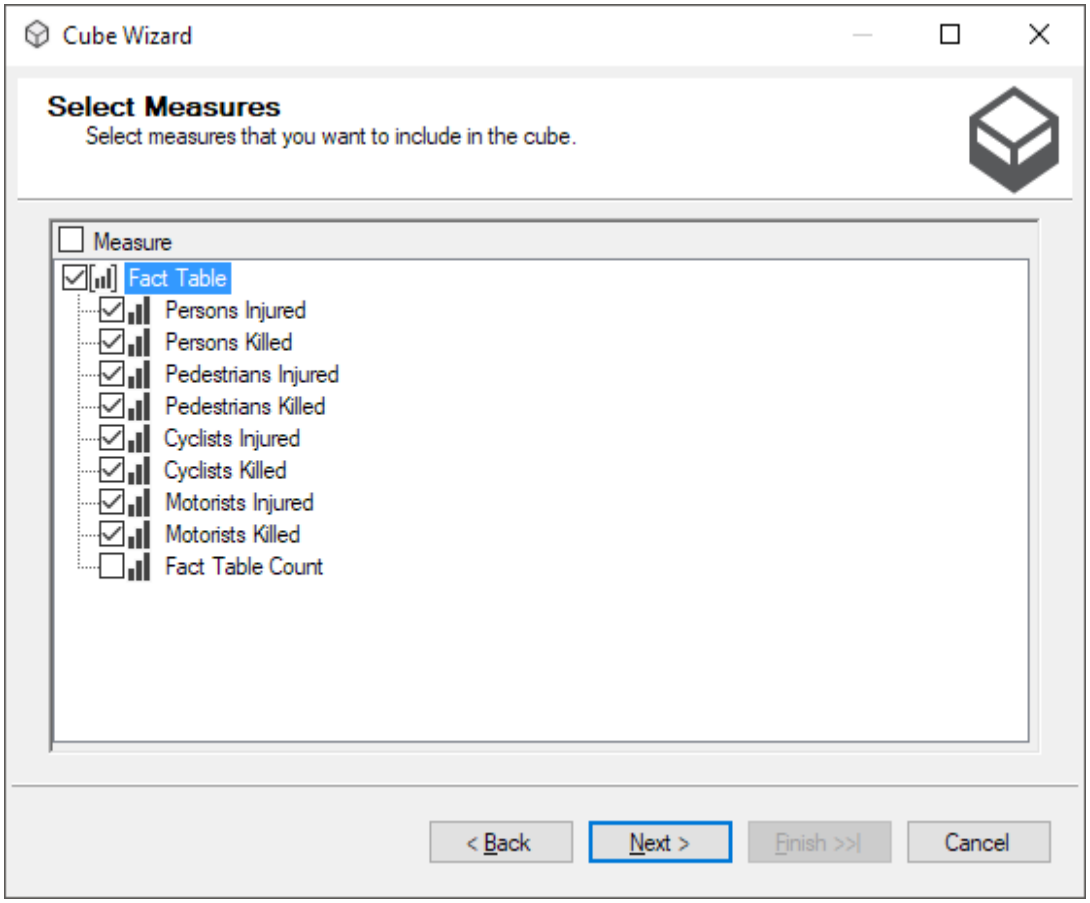




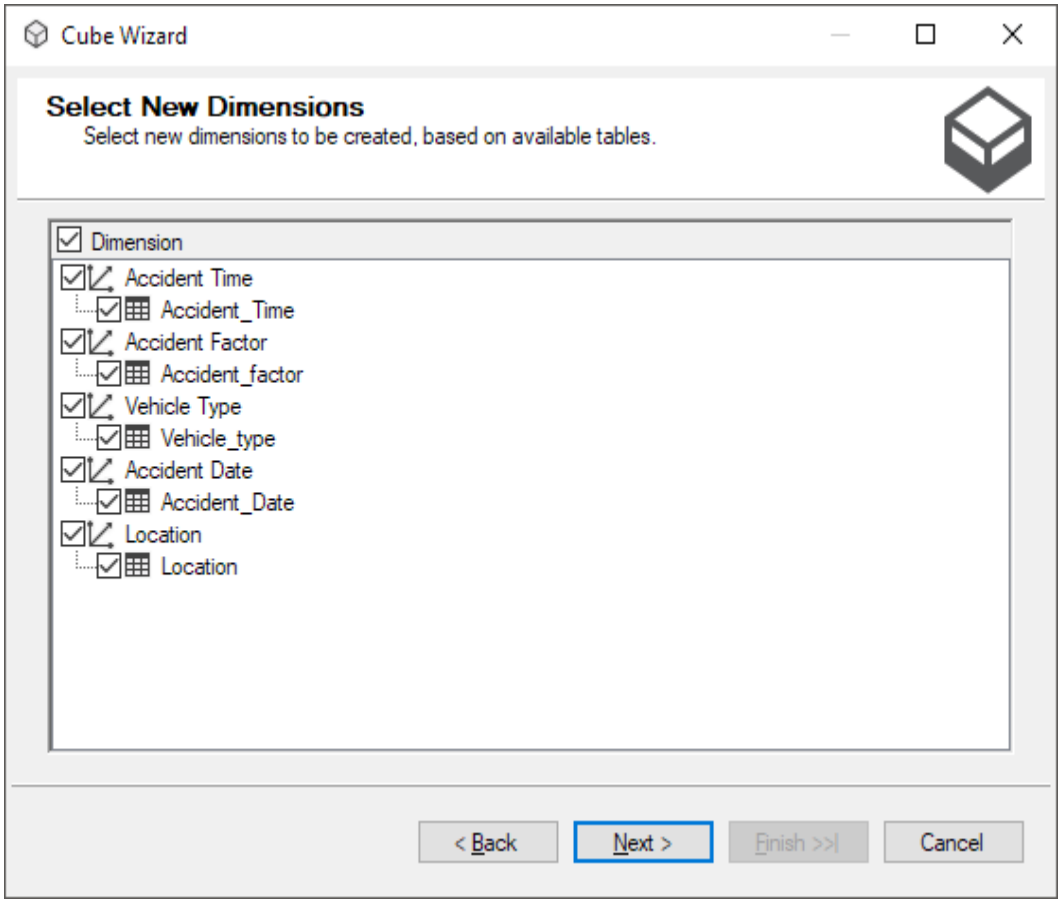
Moreover, we continue with the creation of the cube and its dimensions with the following steps:

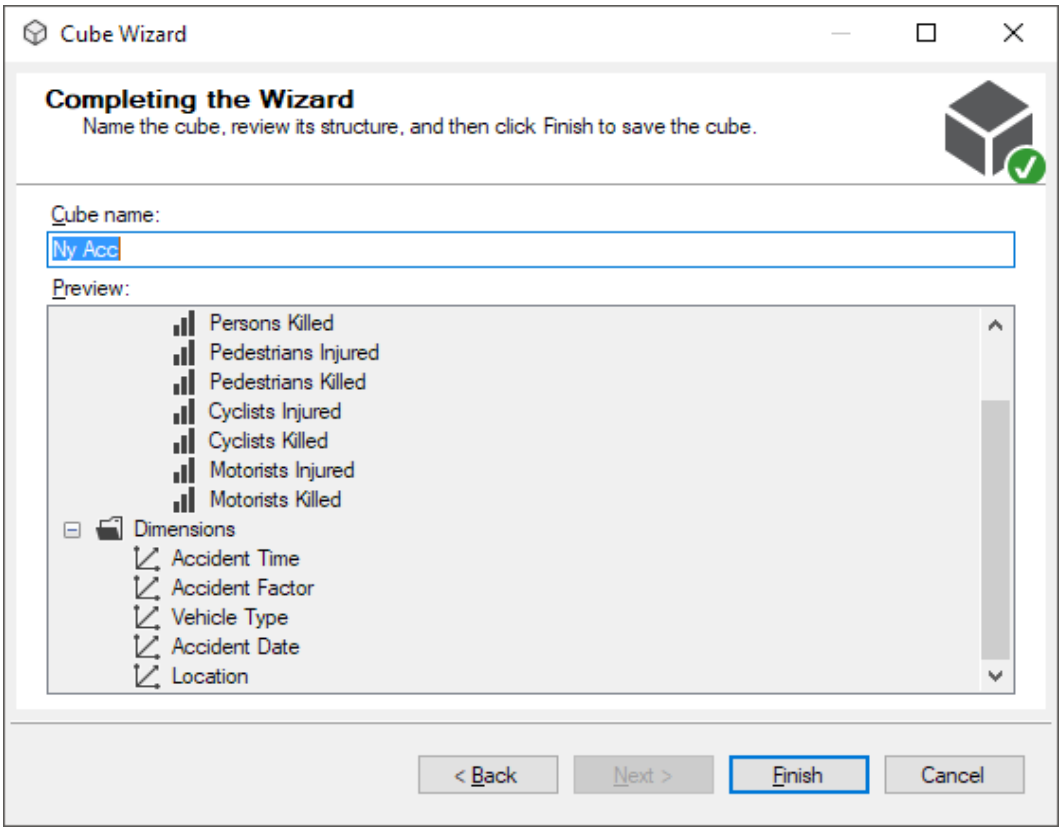


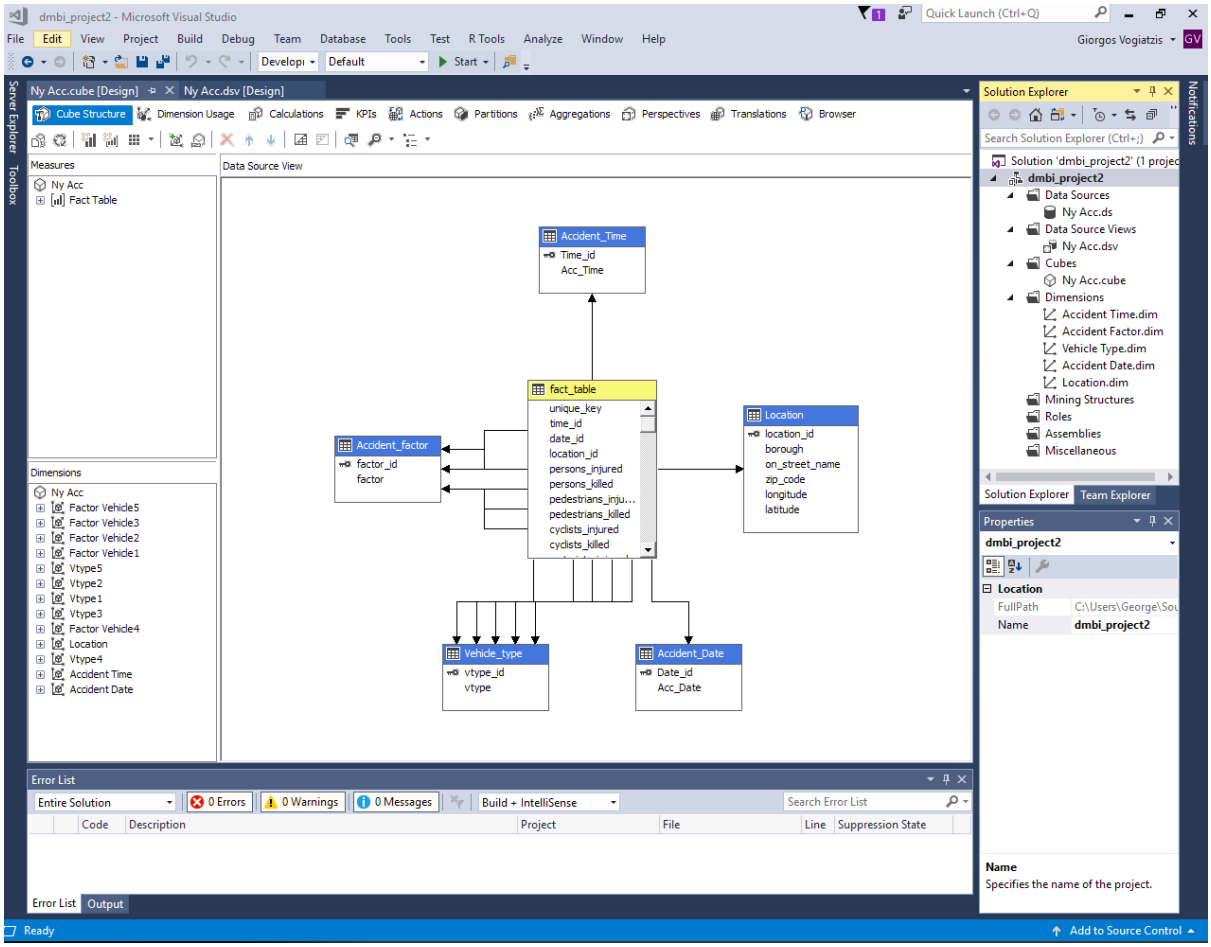

Cube Wizard
Select Measure Group Tables
Select a data source view or diagram and then select the tables that will be used for measure groups.
Data source view:
Ny Acc
Measure group tables:
Suggest
fact_table
Vehicle_type
Accident_Date
Accident_factor
Location
Accident_Time
< Back
Next >
Finish >>|
Cancel


Cube Wizard
Select Measures
Select measures that you want to include in the cube.
Measure
Fact Table
Persons Injured
Persons Killed
Pedestrians Injured
Pedestrians Killed
Cyclists Injured
Cyclists Killed
Motorists Injured
Motorists Killed
Fact Table Count
< Back
Next >
Finish >>|
Cancel

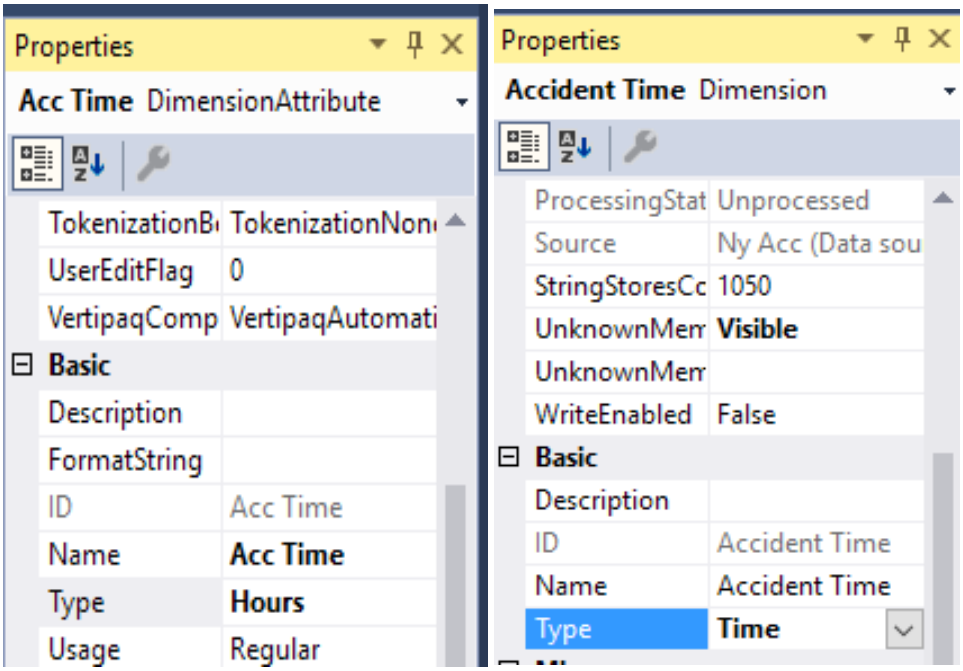
After the choice of the fact table, we continue with the dimensions:

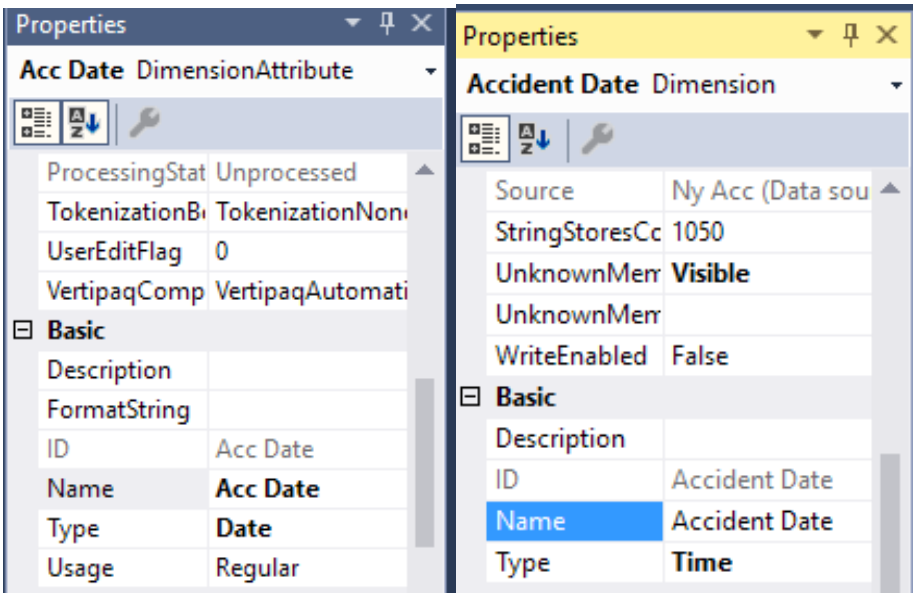




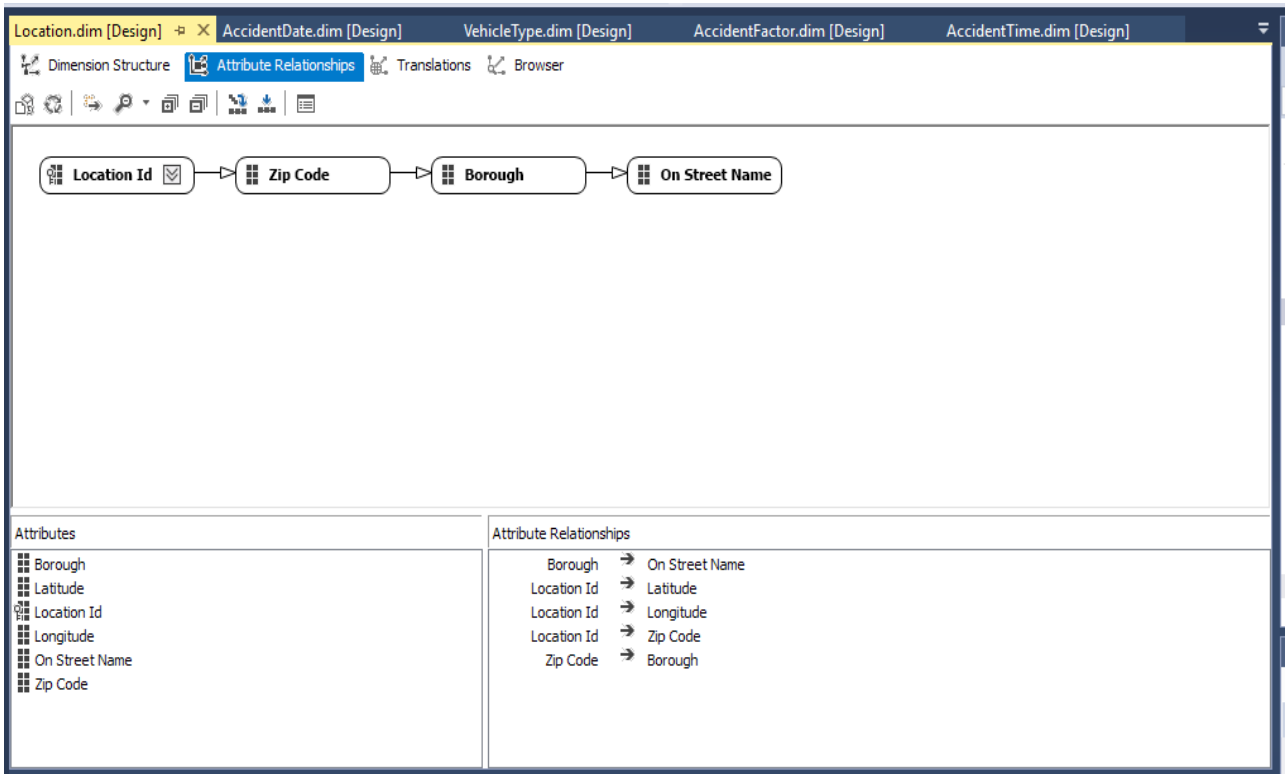


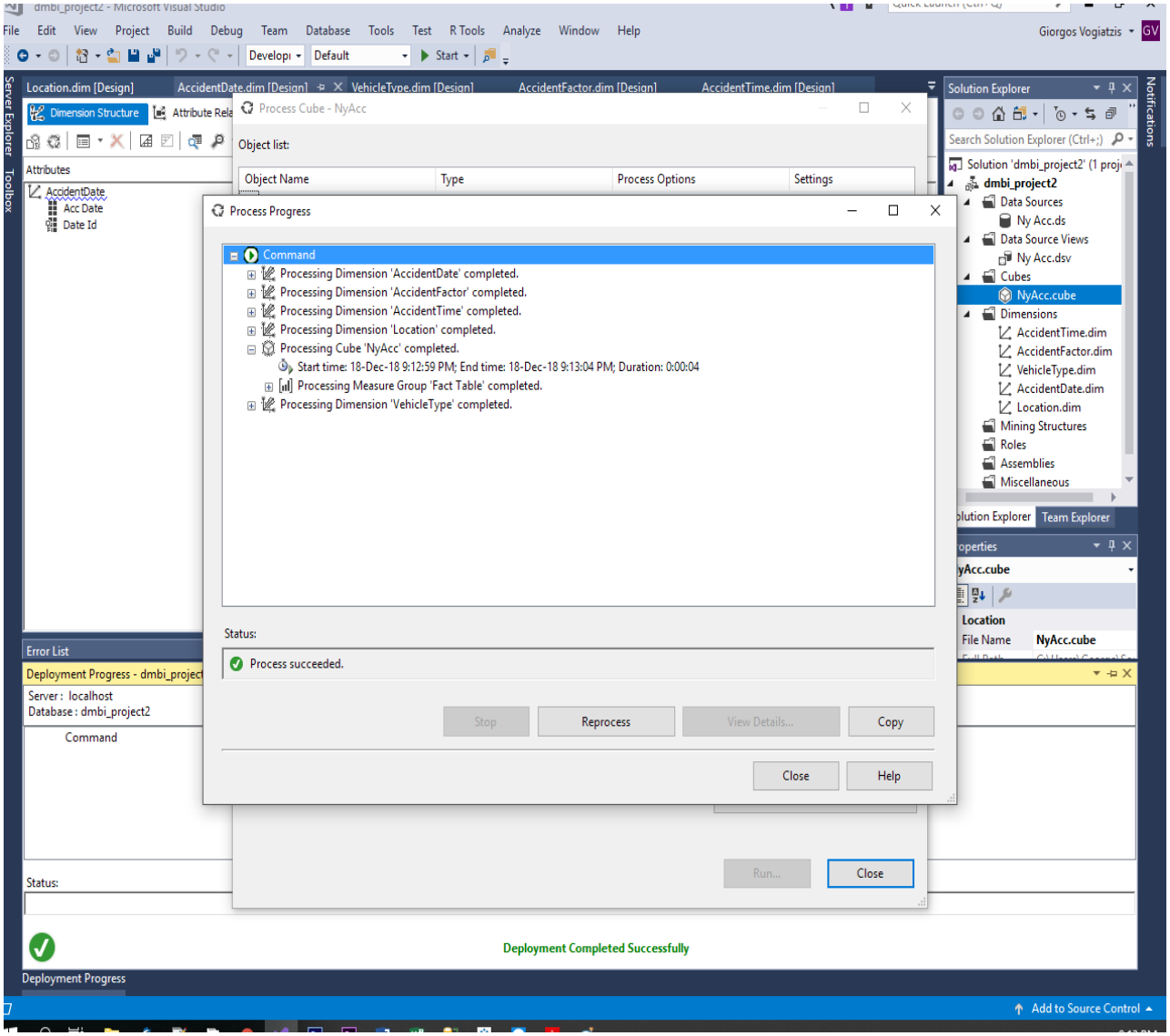
Then we had to change the data type of ‘acc_time’ and ‘acc_date’ attributes dimensions to Hours and Date respectively, and change the dimensions time and date to Time type.





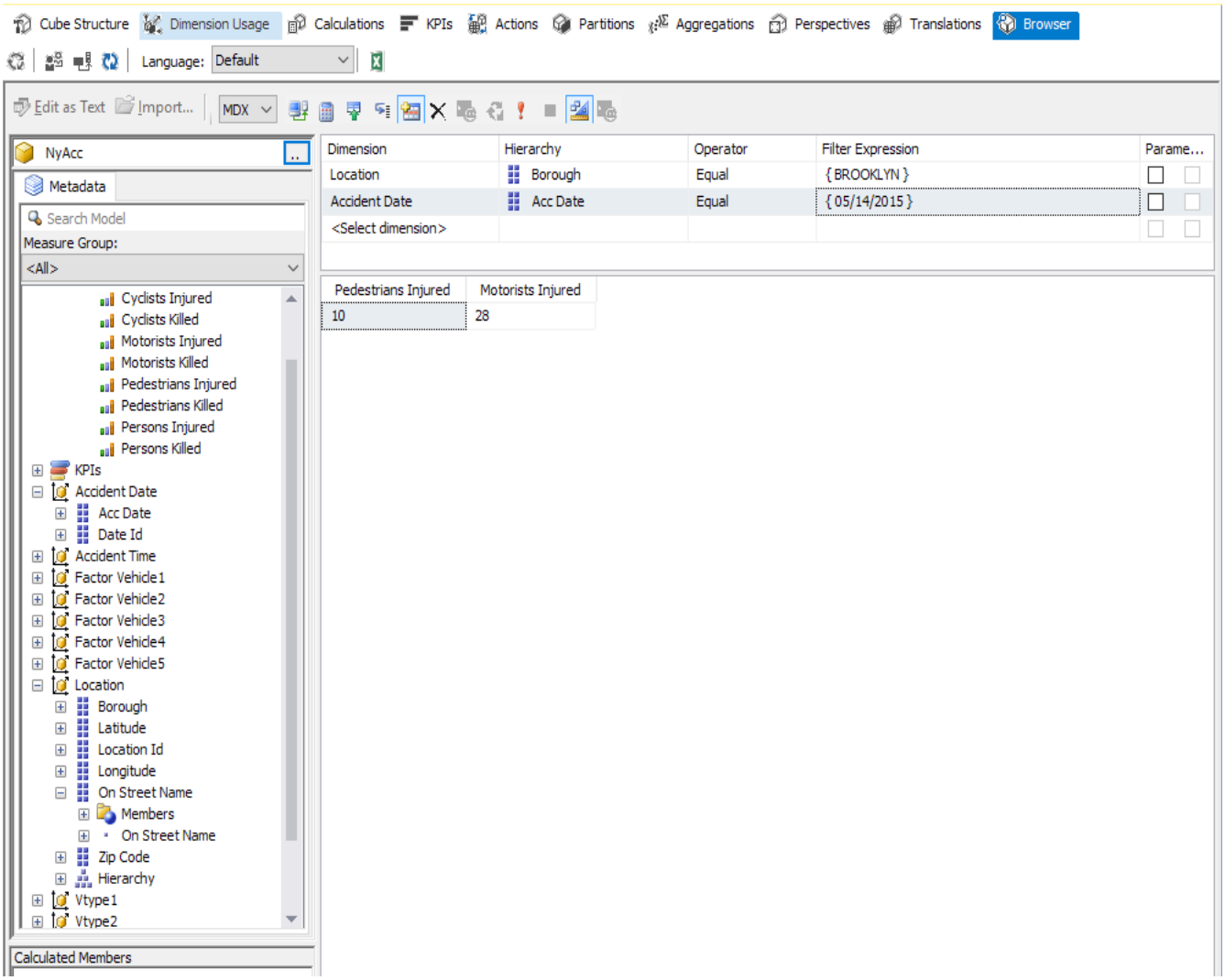
After, we defined the hierarchies in the dimension location table. So, we proccessed and deployed the cube successfully.



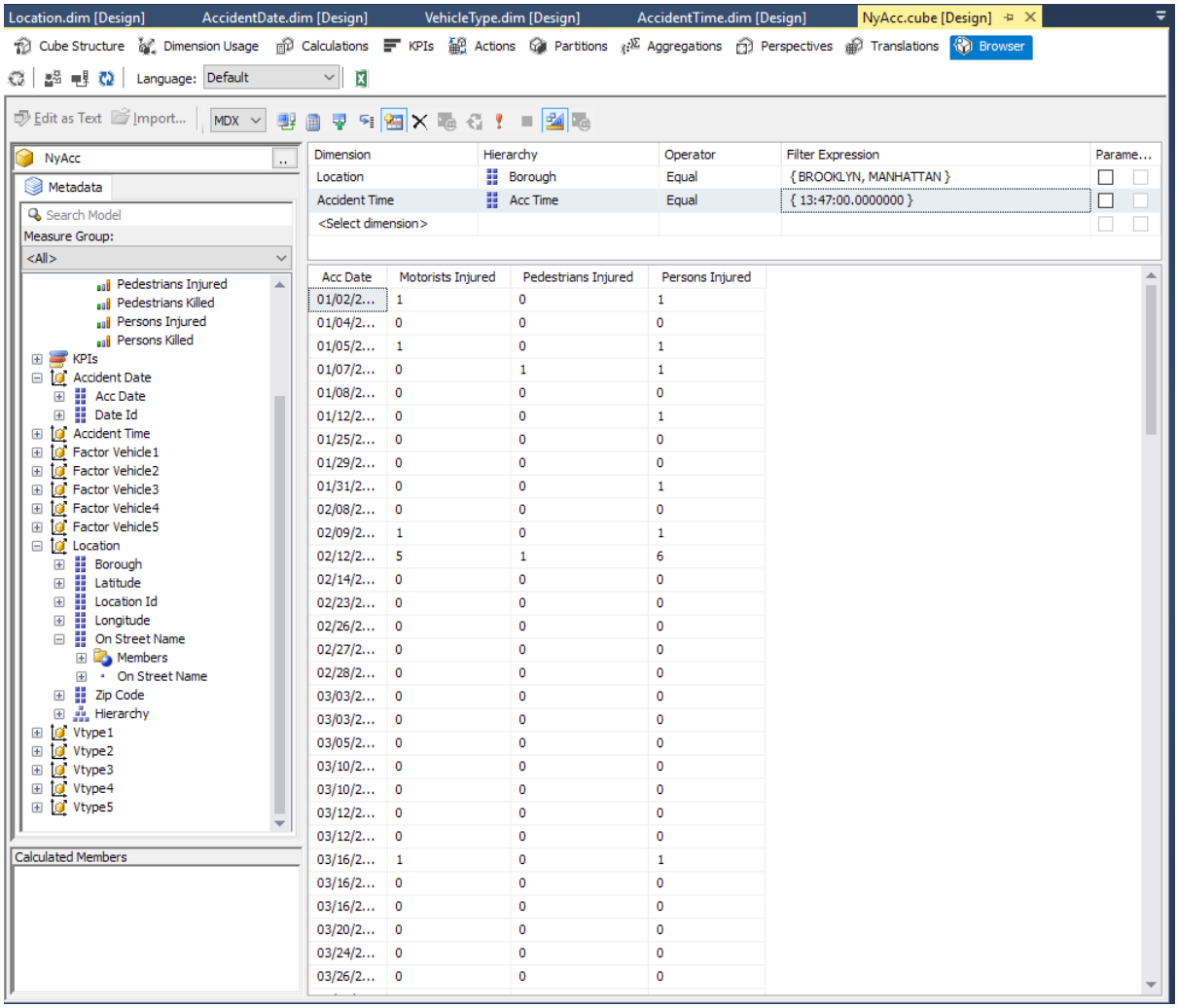

Process Progress
Command
Processing Dimension 'AccidentDate' completed.
Processing Dimension 'AccidentFactor' completed.
Processing Dimension 'AccidentTime' completed.
Processing Dimension 'Location' completed.
Processing Cube 'NyAcc' completed.
Start time: 18-Dec-18 9:12:59 PM; End time: 18-Dec-18 9:13:04 PM; Duration: 0:00:04
Processing Measure Group 'Fact Table' completed.
Processing Dimension 'VehicleType' completed.
Status:
Process succeeded.
Stop
Reprocess
View Details...
Copy
Close
Help
Process Cube - NyAcc
Object list:
Object Name
Type
Process Options
Settings
Run...
Close
Deployment Completed Successfully
Deployment Progress
Solution Explorer
dmbi_project2
Data Sources
Ny Acc.ds
Data Source Views
Ny Acc.dsv
Cubes
NyAcc.cube
Dimensions
AccidentTime.dim
AccidentFactor.dim
VehicleType.dim
AccidentDate.dim
Location.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Location
File Name
NyAcc.cube
Server : localhost
Database : dmbi_project2
Command
Status:
Add to Source Control

2.4 Olap Reports

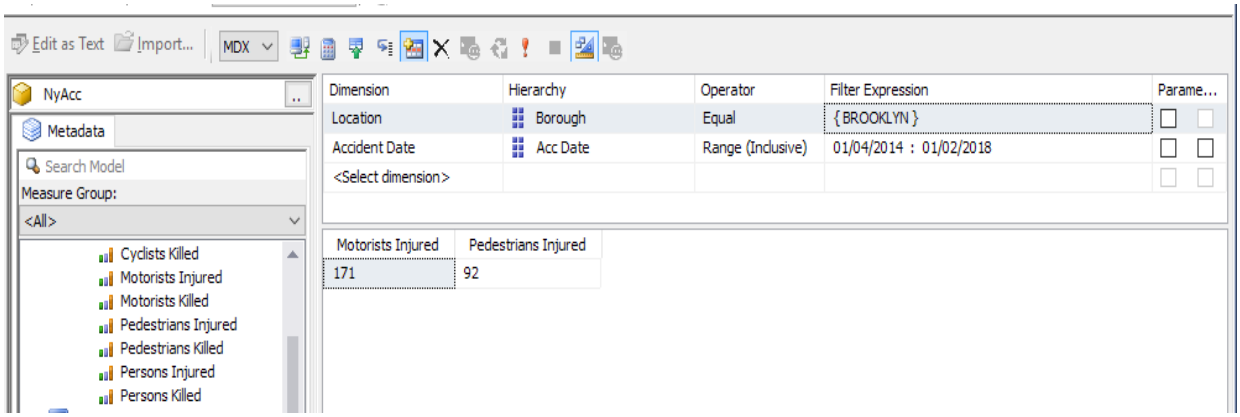
In the following report we can see the number of pedestrians and motorists that got injured in the borough of Brooklyn on the date 14th of may 2015



Here we can see the result of our rollup for all the injuries occurred in Brooklyn and Manhattan at 13:47 o'clock for all the days in our calendar.

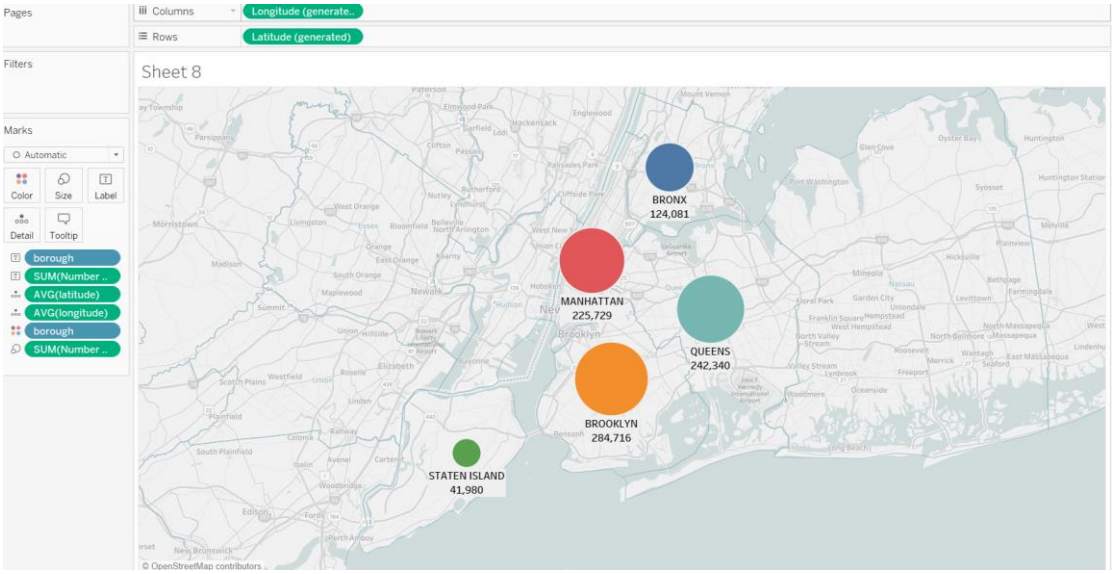


And finally a slice and dice example which gives us the number of pedestrians and motorists killed in Brooklyn between the 4th of January 2014 and the 2nd of january 2018.



3. TABLEAU

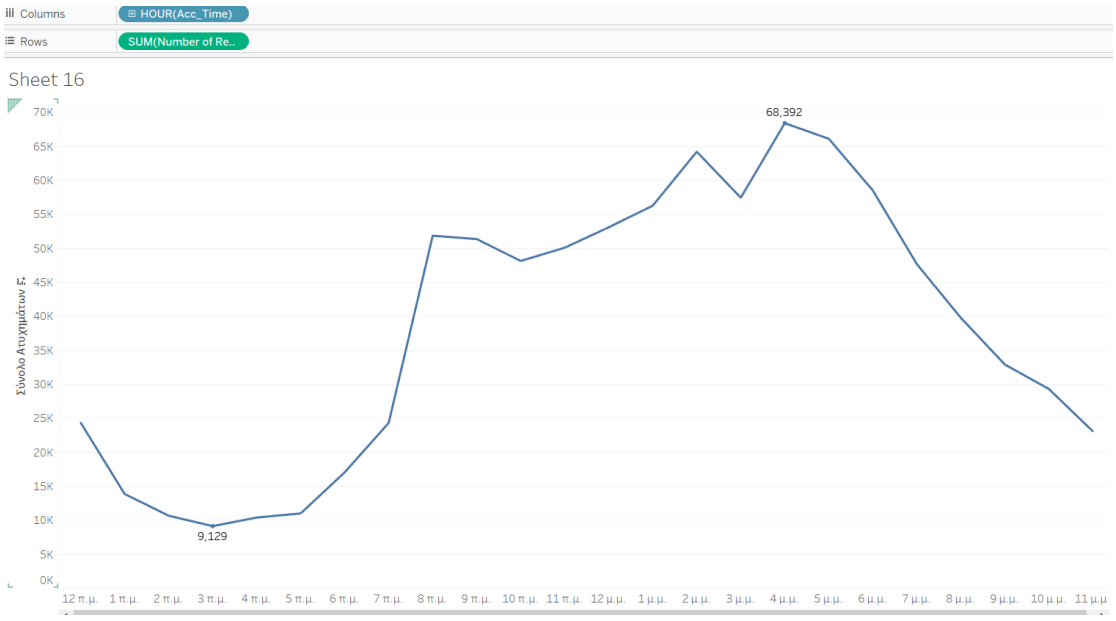
We use the tableau program in order to make the visualization analysis of our data. In the first diagram, we observe that the Brooklyn have the more accidents and the Staten Islands the lowest collisions in New Work City.



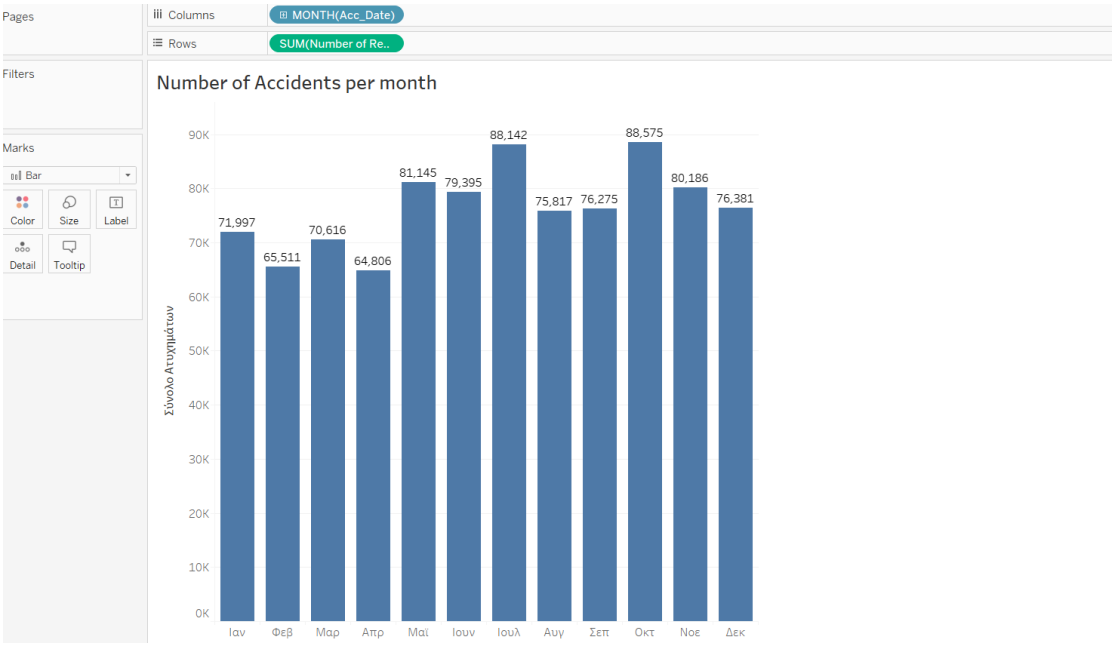
In the second diagram, we observe that the Brooklyn have the more accident in all the examed years and the Staten Islands has the less collisions in all the examed years. Moreover, we observed that from 2012 until 2014 Manhattan have more accidents than Queens has but that change in 2016 when Queens has more accidents. That pattern continues until today.

From the below diagram we see that the most collisions happened in 2015 and the less in the 2012.

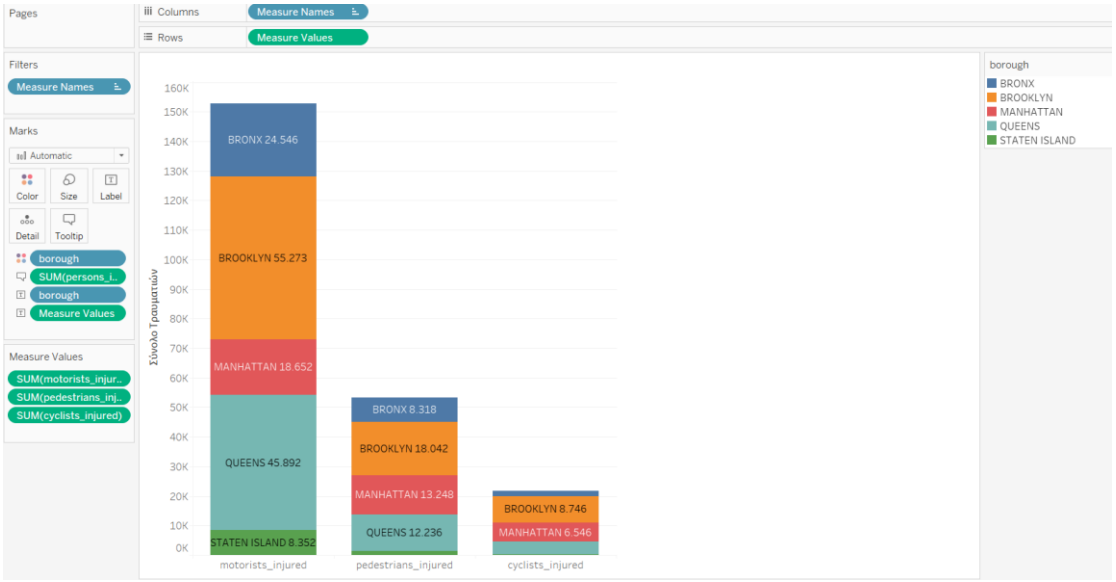
Except from the year, we want to observe how the number of collisions alter through the time. We observe that the most collisions happened in 4 a.m. and the less in 3 p.m...



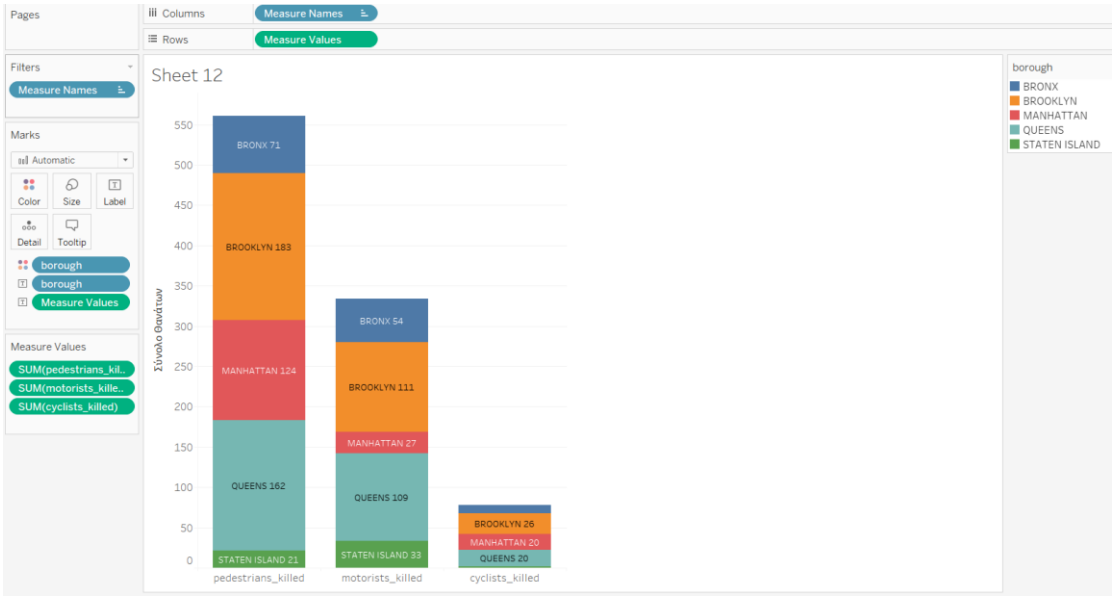
Furthermore, with the above diagram we see that the most collisions happened in October and the less in April.



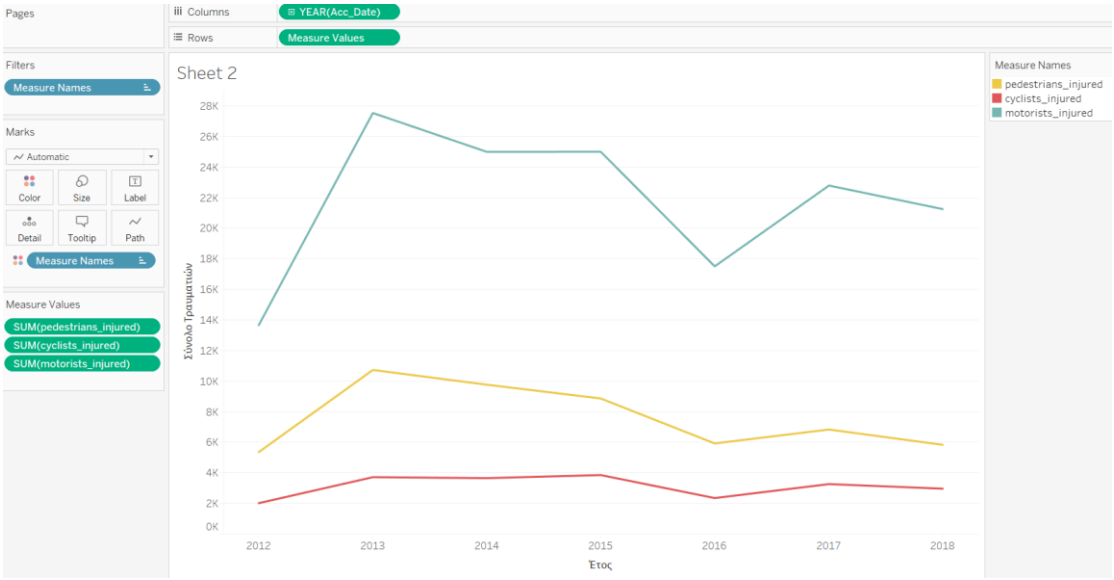
From the above diagram we see that from the total of the injured people after a collision are the motorists are the most and the cyclist the less. In addition, we see how this number alter in towns. We observe that the Brooklyn have the most injures.



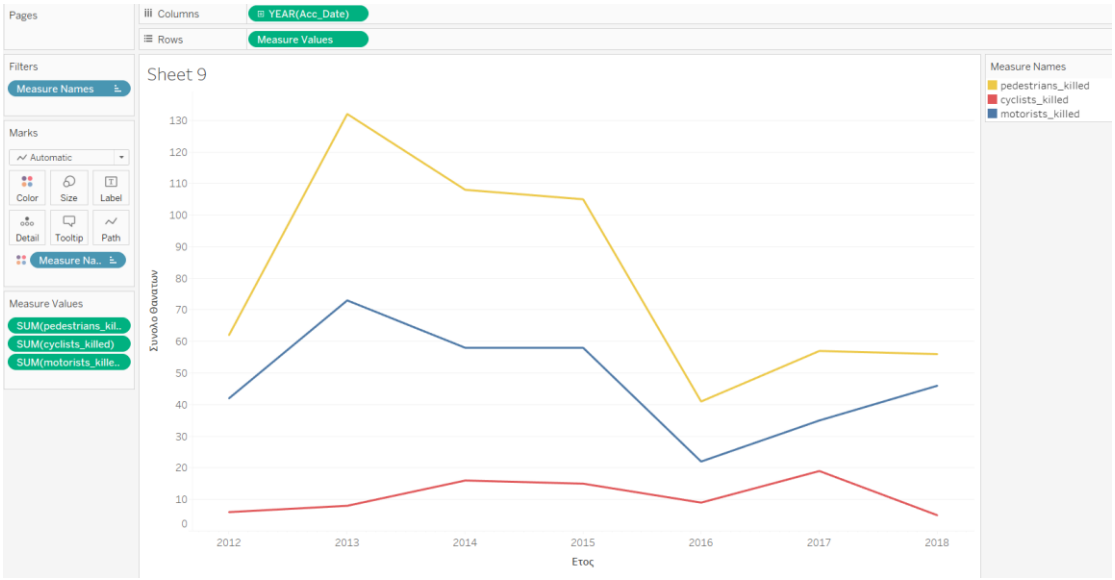
From the above diagram we see that the Brooklyn have the most deaths which affect most the pedestrians.



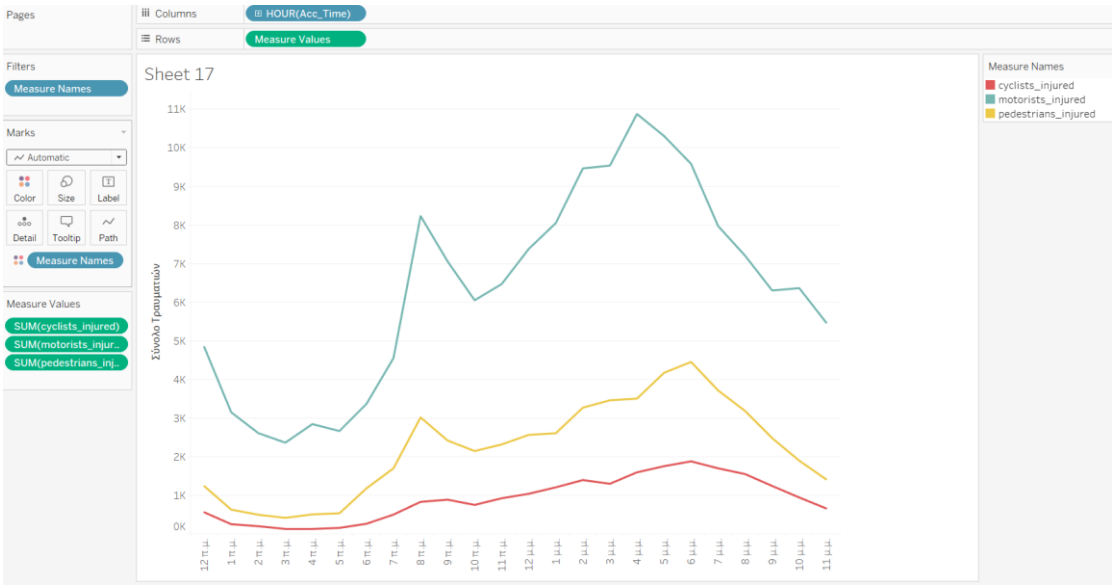
From the above plot, we see that the most injuries have the motorist in 2013. In 2016 we observe a sharp reduce in all the categories but the in the motorists was more observed.



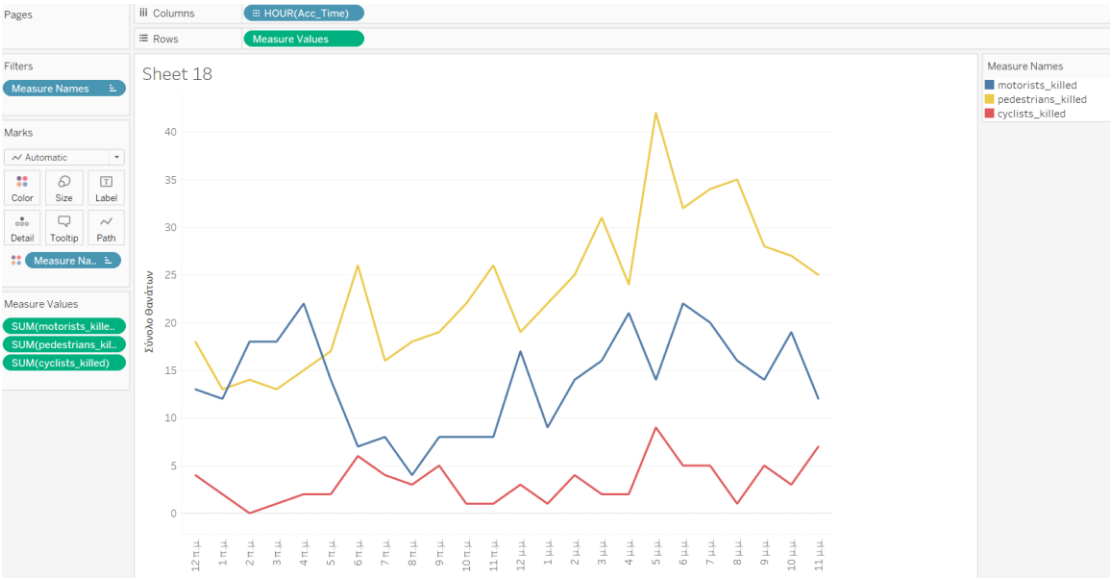
From the above diagram, we see that the most deaths affect the pedestrians and this is stable through the years. In addition, the numbers of deaths in cyclist are the lowest and the only one who reduced in 2018 in contrast to other categories, which increased.



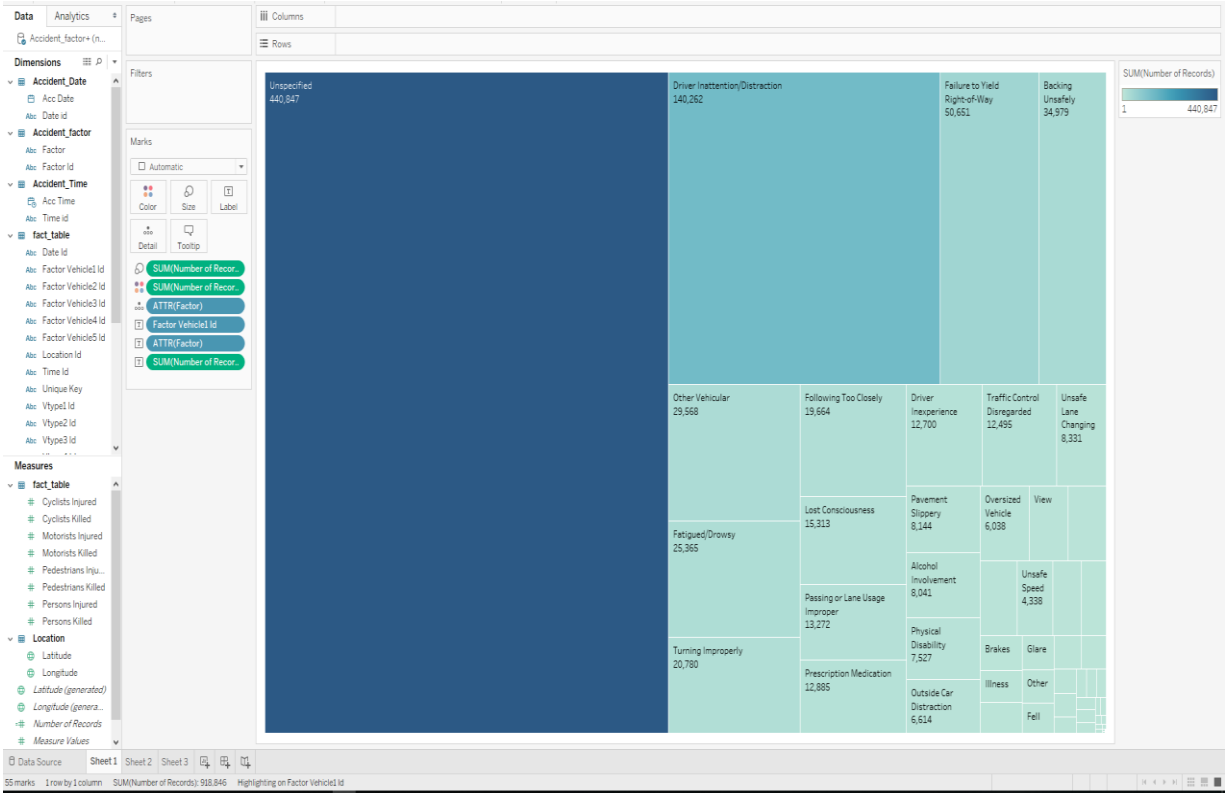
The most injures happen among 8 a.m. and 6 p.m. seems all the categories follow the same pattern with time.



However the number of people who dead in the collision do not seem to follow the same pattern through time. The most deaths, which refers in pedestrians and cyclist, happen in 6 p.m. and the most deaths of motorists happened in 3a.m.



The reason, which affects a collision in the most cases, are unspecified. The second most often reason is the distraction of the driver and the third more often reason is the failure to yield right of way. For this diagram, we use the variable of our dataset 'factor vehicle_1' because we think that the first vehicle is responsible for the accident.



In corrspondence that we use the ‘factor vehicle_1’ we use the variable about the ‘vehicle type_1’. In order to see what type of vehicle causes a collision. Final we see that the type of vehicle which take part in the most collisions are the passenger vehicle, the second category are the taxi and third the category with unknown vehicle type.