UK ACCIDENTS DATABASE

Data warehousing project under the guidance of Dr. Rathin Sarathy

Team:

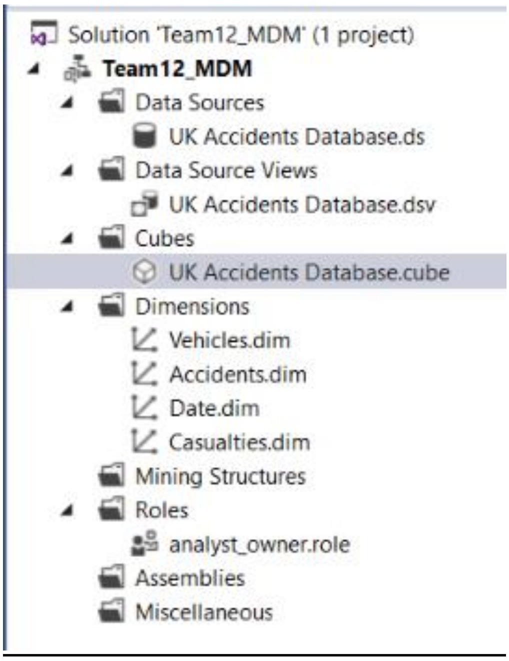
Ramya Kambadahalli Vannappa
Balapavan Kommareddy
Anup Kumar Chittimalla

Table of Contents:

Cube Creation

Step1: We must first create a connection to the data source in visual studio. i.e. we will connect to the relational database called UK Accidents Database.

The Team12_MDM is our new analysis database.
UK Accidents Relational Database will act as the source of data for our Team12_MDM analysis database.

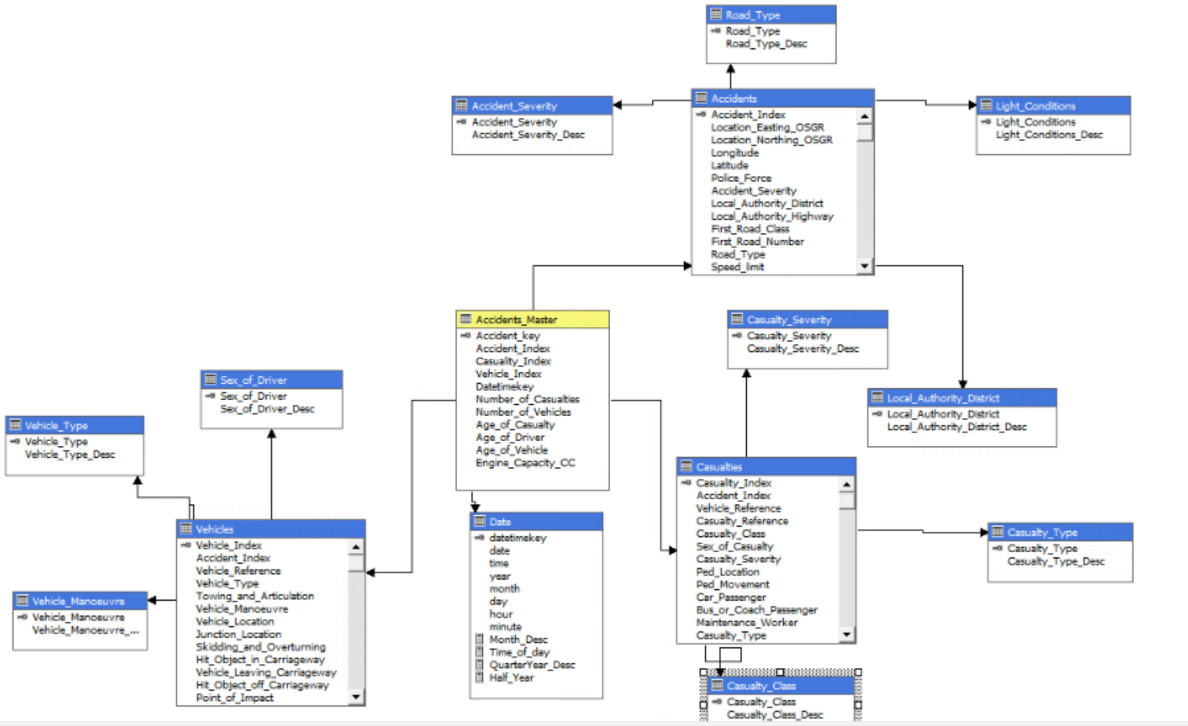


We see the new data source being connected to the UK Accidents relational database. Later, we deployed the data source to SQL Server Analysis Services.

Step 2: Create the data source view

We created the data source view which is the link between UK accidents relational database and the analysis services database Team12_MDM.

While creating the New Data Source View we had the option to include Accidents Master fact table. Later, we added the related tables. After the DSV is created within visual studio, it must be deployed to view the DSV within SQL Server Management Studio. This DSV becomes / acts as the relational database.



The above screenshot shows the DSV created within visual studio. It shows all the Dimension tables and Accident_Master Fact table.

We used the DimAccidents (renamed as Accidents) table within the DSV to create the analysis services Accidents Dimension.

Similarly, we used DimVehicles, DimDate, DimCasulaties tables within DSV (all renamed as shown in the above screenshot) and created the analysis services Vehicle, Date and Casualties Dimensions.

Accidents Dimension

- Accidents
  - Accident Index
  - Accident Severity
  - Accident Severity Desc
  - Carriageway Hazards
  - First Road Class
  - First Road Number
  - Junction Control
  - Junction Detail
  - Latitude
  - Light Conditions
  - Light Conditions Desc
  - Local Authority District
  - Local Authority District Desc
  - Local Authority Highway
  - Location Easting OSGR
  - Location Northing OSGR
  - Longitude
  - LSOA Of Accident Location
  - Ped Cross Human
  - Ped Cross Physical
  - Police Force
  - Police Officer Attend
  - Road Surface Conditions
  - Road Type
  - Road Type Desc
  - Second Road Class
  - Second Road Number
  - Special Conditions At Site
  - Speed Limit
  - Speed Limit Desc
  - Urban Rural
  - Urban Rural Desc
  - Weather Conditions

Casualties Dimension

- Casualties
  - Accident Index
  - Bus Or Coach Passenger
  - Car Passenger
  - Casuality Index
  - Casualty Class
  - Casualty Class Desc
  - Casualty Class Desc1
  - Casualty Home Area Type
  - Casualty Reference
  - Casualty Severity
  - Casualty Severity Desc
  - Casualty Severity Desc1
  - Casualty Type
  - Casualty Type Desc
  - Casualty Type Desc1
  - Maintenance Worker
  - Ped Location
  - Ped Movement
  - Sex Of Casualty
  - Vehicle Reference

Vehicle Dimension

- Vehicle_Index
- Accident_Index
- Vehicle_Reference
- Vehicle_Type
- Towing_and_Articulation
- Vehicle_Manoeuvre
- Vehicle_Location
- Junction_Location
- Skidding_and_Overturning
- Hit_Object_in_Carriageway
- Vehicle_Leaving_Carriage...
- Hit_Object_off_Carriage...
- Point_of_Impact
- Left_Hand_Drive
- Journey_Purpose
- Sex_of_Driver
- Propulsion_Code
- Driver_IMD_Decile

Date Dimension

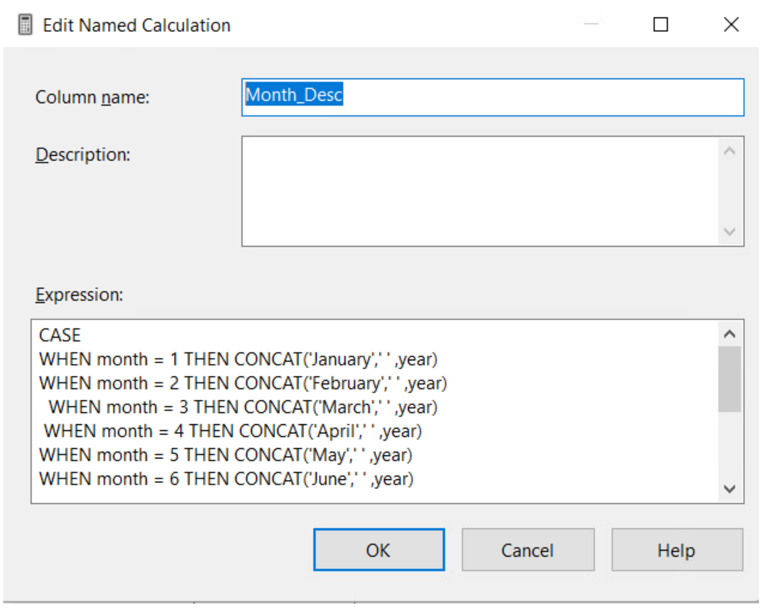
- Date
  - Date
  - Datetimekey
  - Day
  - Half Year
  - Hour
  - Minute
  - Month
  - Month Desc
  - Quarter Year Desc
  - Time Of Day
  - Year

Named Calculations Creation

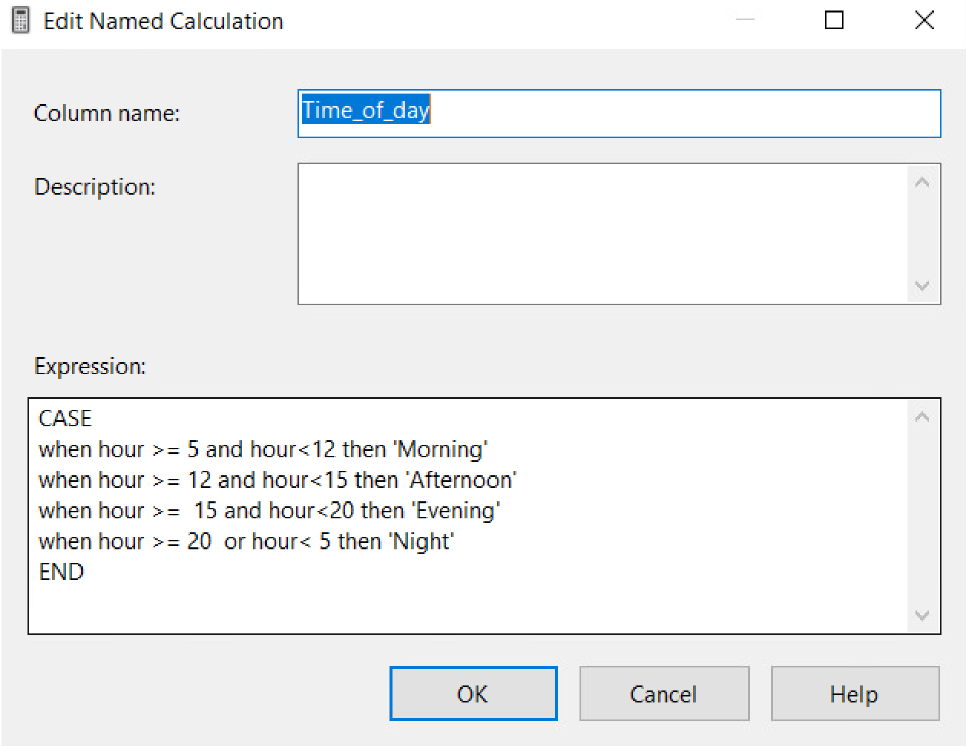
Named calculations refer to the new attributes created within the table. We use T-SQL since the DSV tables are relational databases.

- **Month_desc: is the concatenation of month and year**

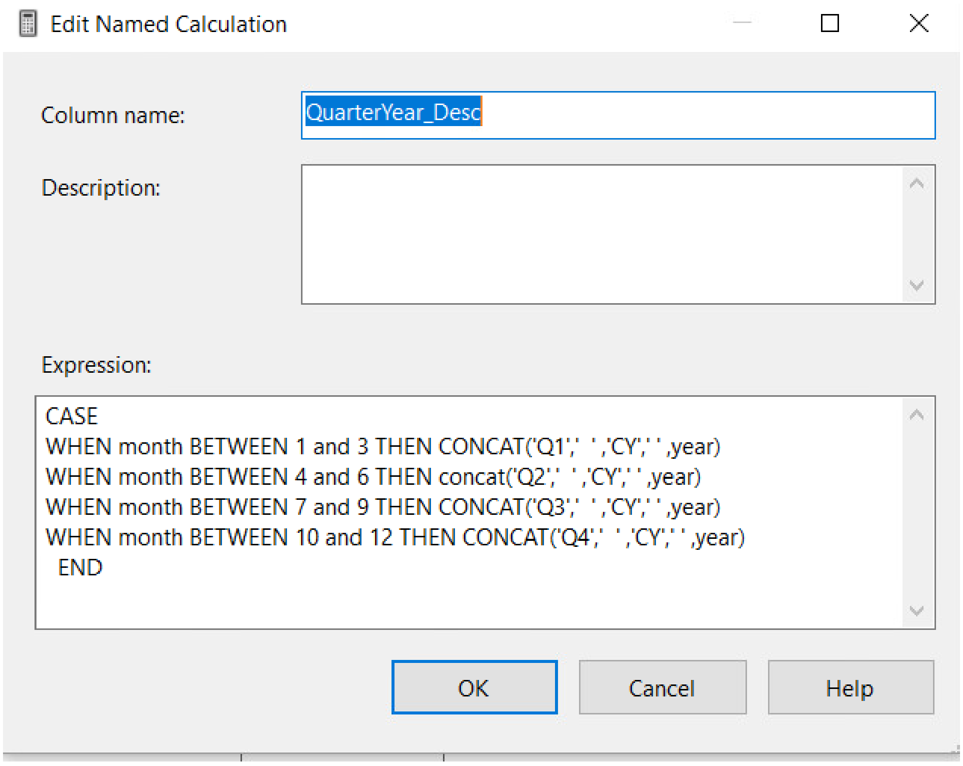
```
CASE
    WHEN month = 1 THEN CONCAT('January',' ' ,year)
    WHEN month = 2 THEN CONCAT('February',' ' ,year)
    WHEN month = 3 THEN CONCAT('March',' ' ,year)
    WHEN month = 4 THEN CONCAT('April',' ' ,year)
    WHEN month = 5 THEN CONCAT('May',' ' ,year)
    WHEN month = 6 THEN CONCAT('June',' ' ,year)
    WHEN month = 7 THEN CONCAT('July',' ' ,year)
    WHEN month = 8 THEN CONCAT('August',' ' ,year)
    WHEN month = 9 THEN CONCAT('September',' ' ,year)
    WHEN month = 10 THEN CONCAT('October',' ' ,year)
    WHEN month = 11 THEN CONCAT('November',' ' ,year)
    WHEN month = 12 THEN CONCAT('December',' ' ,year)
END
```



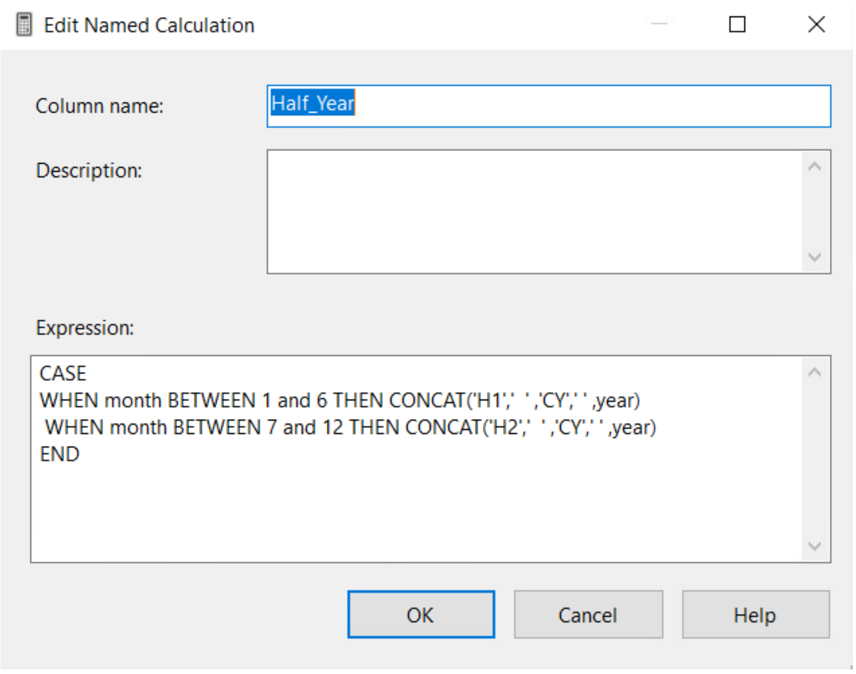
- **Time_of_day** : which explains different parts of the day



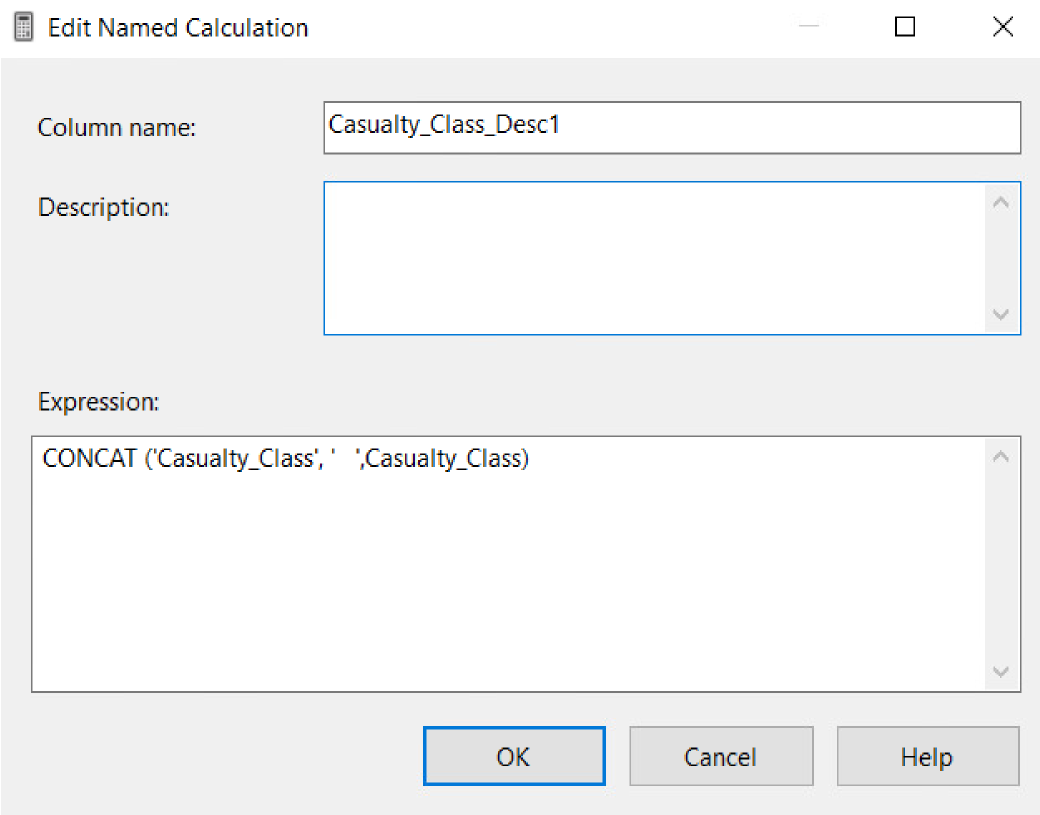
- **QuaterYear_Desc** : which categorizes months into each quarter



- **Half_Year**: which categories quarters into semesters

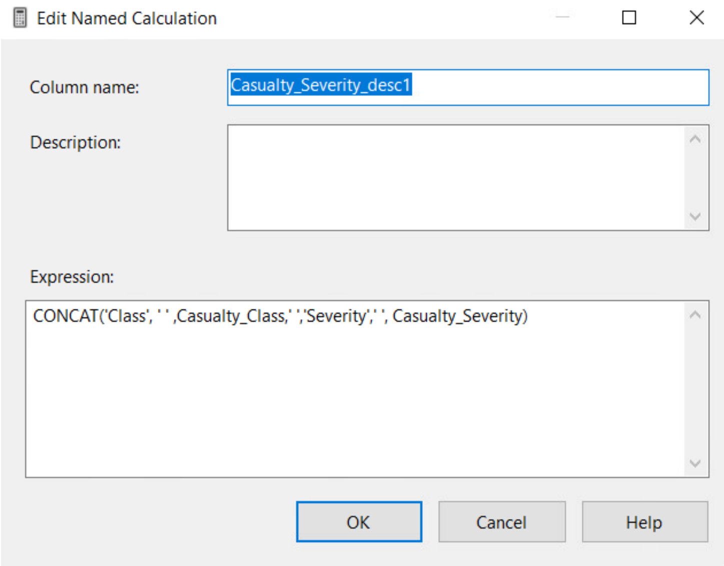


- **Casualty_Class_Desc1**: comprises of casualty class



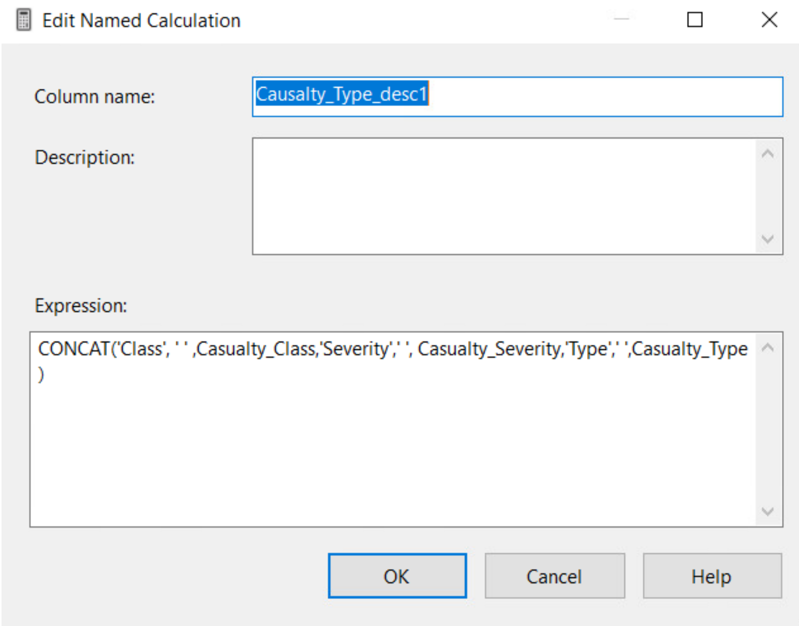
- **Casualty_Severity_desc1**:is the concatenation of casualty_class and casualty_severity

  i.e. Every casualty class has 3 casualty severities. In the below diagram, we are concatenating casualty class with casualty severity to get a unique record.



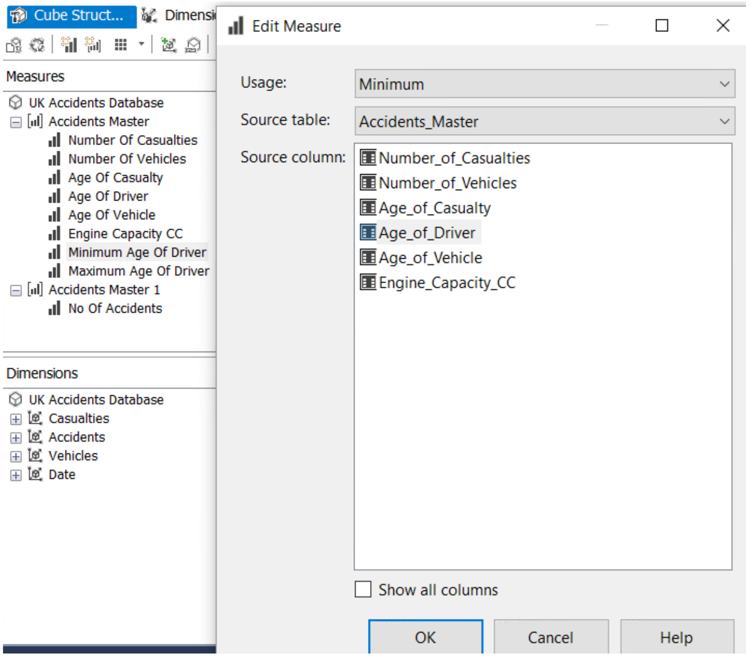
- **Casualty_Type_desc1** : is the concatenation of casualty_class, casualty_severity, and casualty_type

  i.e. Every casualty class has 3 casualty severities. Every casualty severity has further types. In the below diagram, we are concatenating casualty class with casualty severity and casualty type to get a unique record.

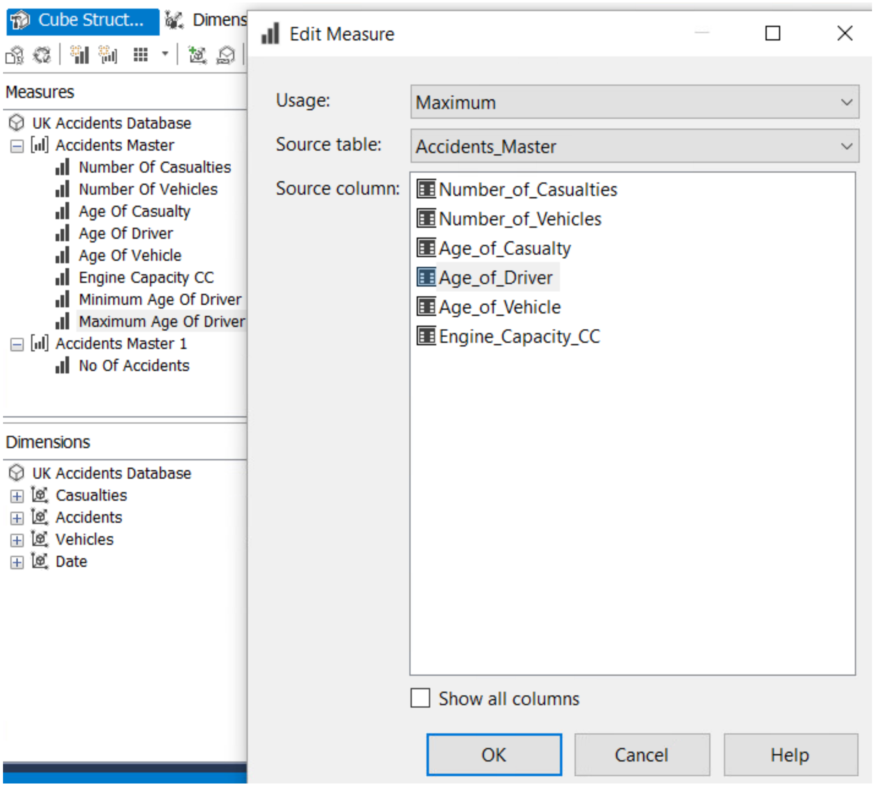


New Measures

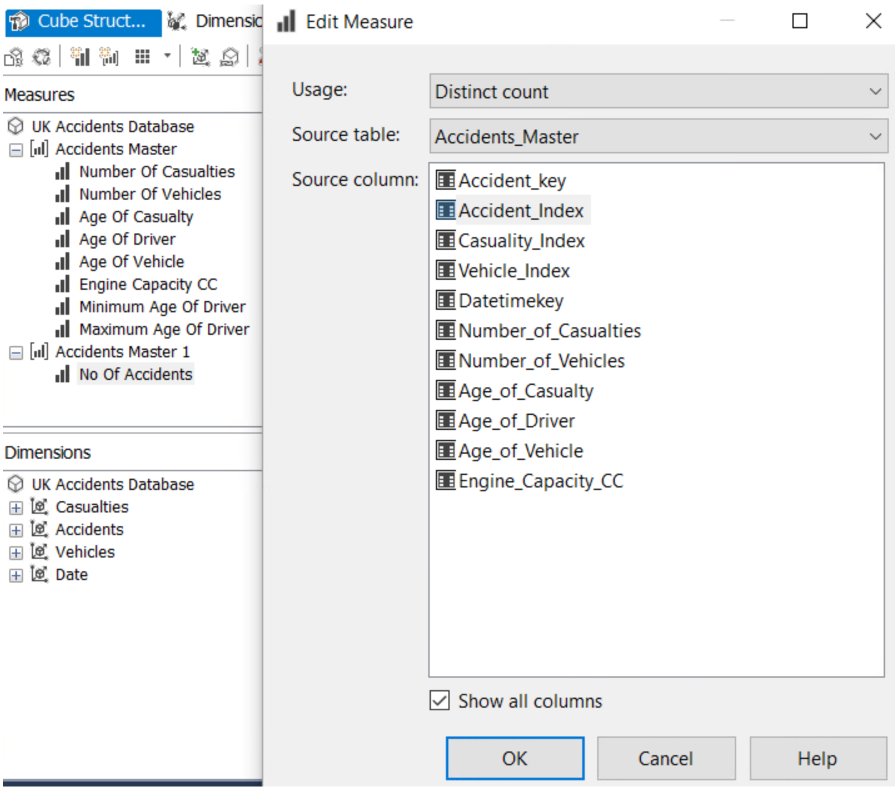
- **Minimum Age Of Driver** Measure

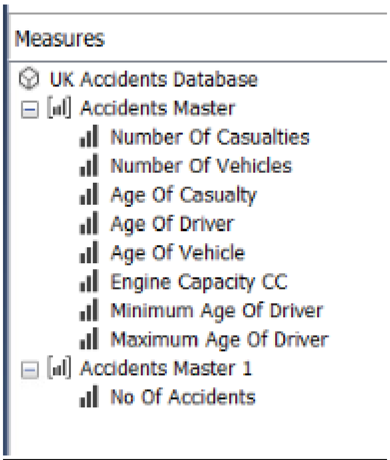


- **Maximum Age Of Driver** Measure



- **No of accidents Measure**





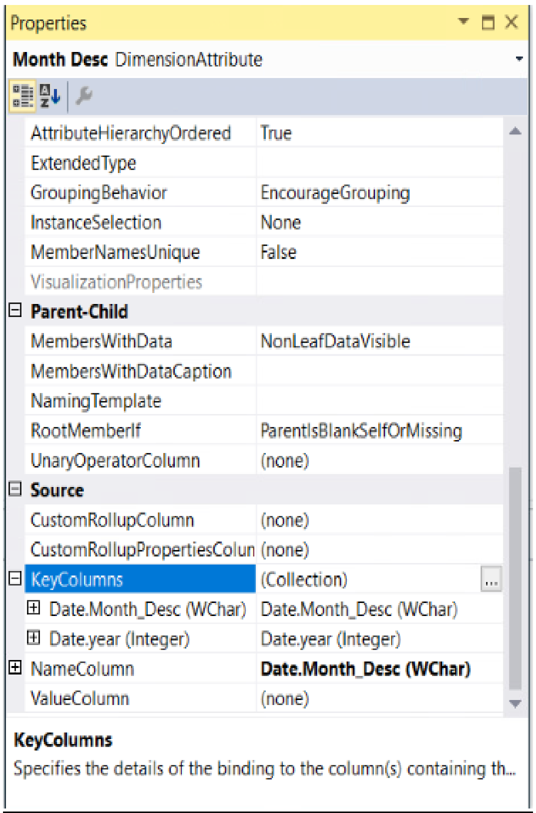
The above screenshot shows all the 3 measures created.

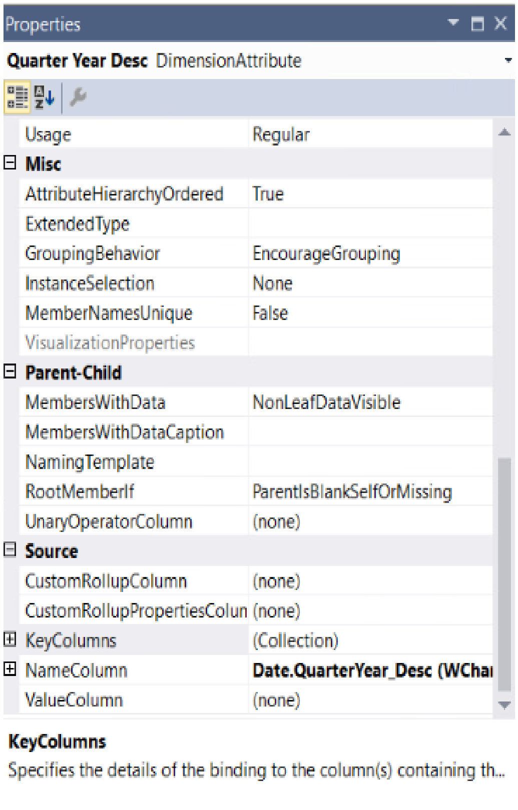
Hierarchy

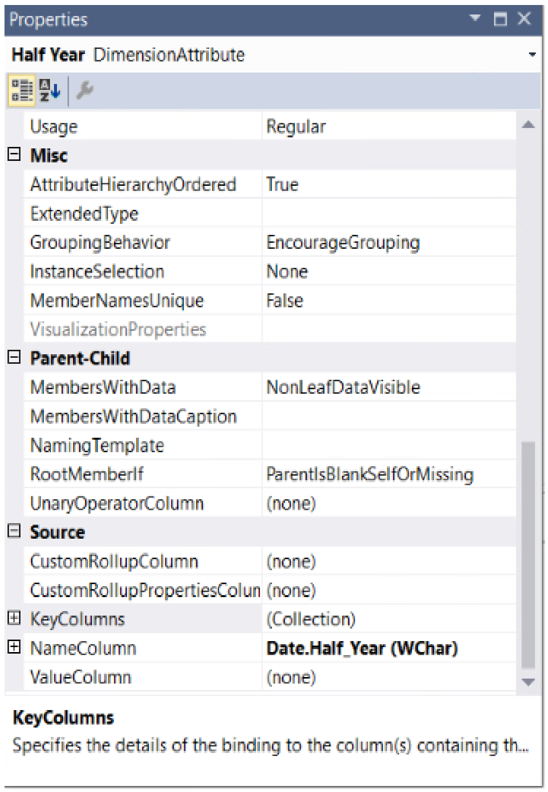
Hierarchies permit us to drill across, drill-down and roll up aggregates. We need to fix the keycolum and name column values found within the attributes of the dimension.

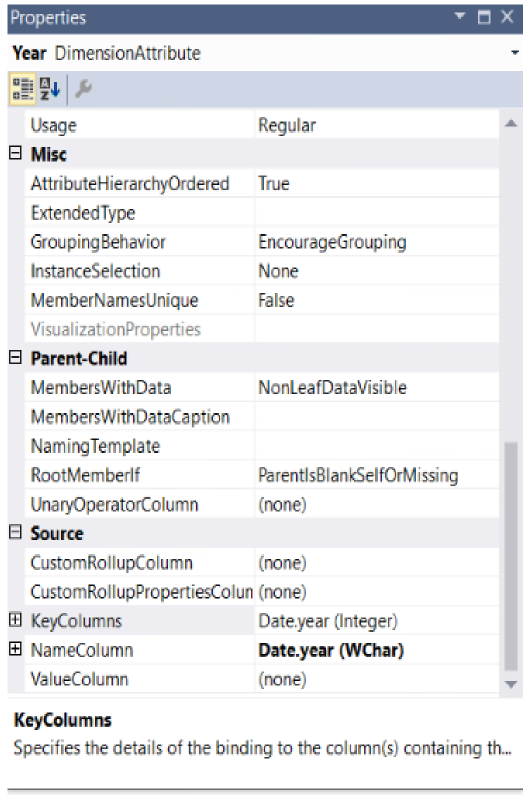
Date Hierarchy Key columns and named columns fixing

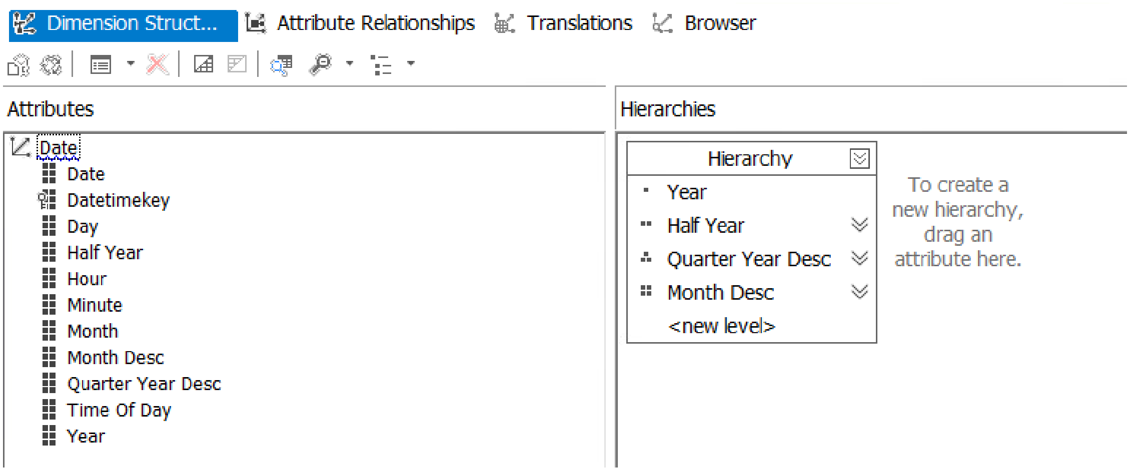
In-order to have unique values for a month description , we used a composite key which contains month within a given year. For instance, January month is available across all the years in the data source we have. To have a unique record of January, we concatenate Jan with CY and Year. i.e. the month name is not January but January CY 2015 . Similarly we have set composite keycolumns for Quarters and half-year as shown below



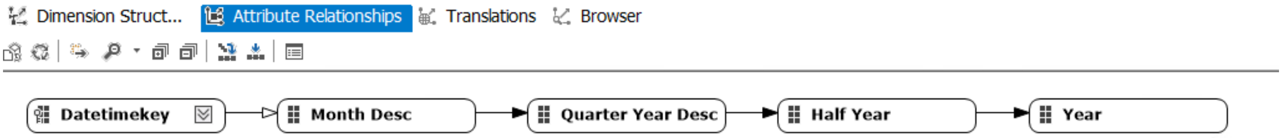



Properties
Half Year DimensionAttribute
Usage Regular
Misc
AttributeHierarchyOrdered True
ExtendedType
GroupingBehavior EncourageGrouping
InstanceSelection None
MemberNamesUnique False
VisualizationProperties
Parent-Child
MembersWithData NonLeafDataVisible
MembersWithDataCaption
NamingTemplate
RootMemberIf ParentIsBlankSelfOrMissing
UnaryOperatorColumn (none)
Source
CustomRollupColumn (none)
CustomRollupPropertiesColun (none)
KeyColumns (Collection)
NameColumn Date.Half_Year (WChar)
ValueColumn (none)
KeyColumns
Specifies the details of the binding to the column(s) containing th...

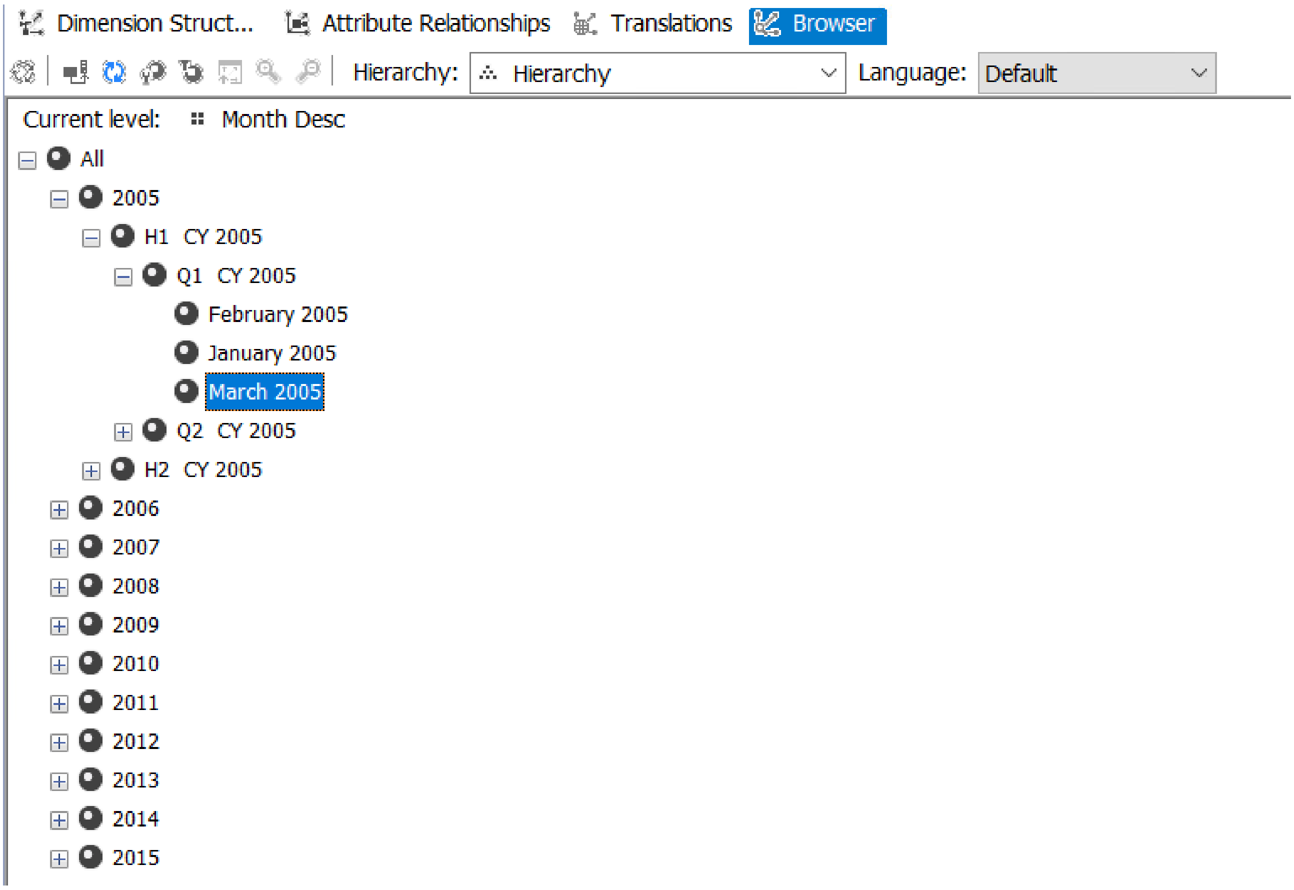

Properties
Year DimensionAttribute
Usage Regular
Misc
AttributeHierarchyOrdered True
ExtendedType
GroupingBehavior EncourageGrouping
InstanceSelection None
MemberNamesUnique False
VisualizationProperties
Parent-Child
MembersWithData NonLeafDataVisible
MembersWithDataCaption
NamingTemplate
RootMemberIf ParentIsBlankSelfOrMissing
UnaryOperatorColumn (none)
Source
CustomRollupColumn (none)
CustomRollupPropertiesColun (none)
KeyColumns Date.year (Integer)
NameColumn Date.year (WChar)
ValueColumn (none)
KeyColumns
Specifies the details of the binding to the column(s) containing th...


Dimension Struct...
Attribute Relationships
Translations
Browser
Attributes
Date
Date
Datetimekey
Day
Half Year
Hour
Minute
Month
Month Desc
Quarter Year Desc
Time Of Day
Year
Hierarchies
Hierarchy
Year
Half Year
Quarter Year Desc
Month Desc
<new level>
To create a new hierarchy, drag an attribute here.

The attribute relationships for date dimension hierarchy are shown below

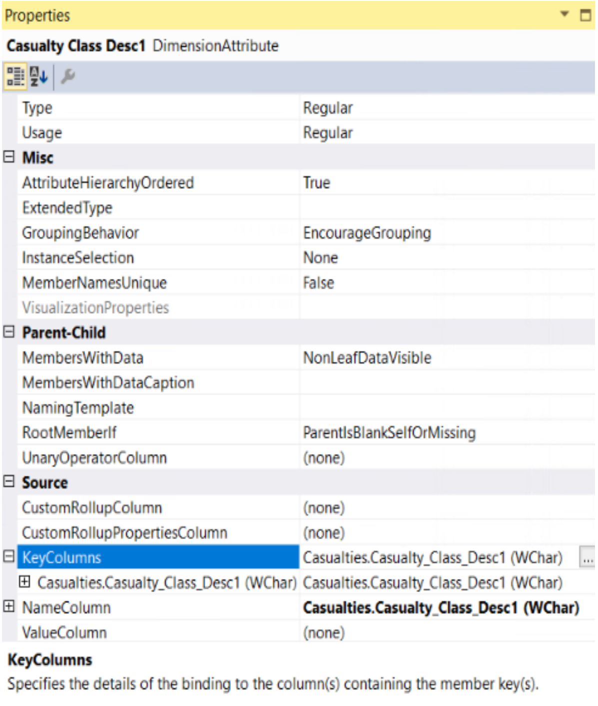


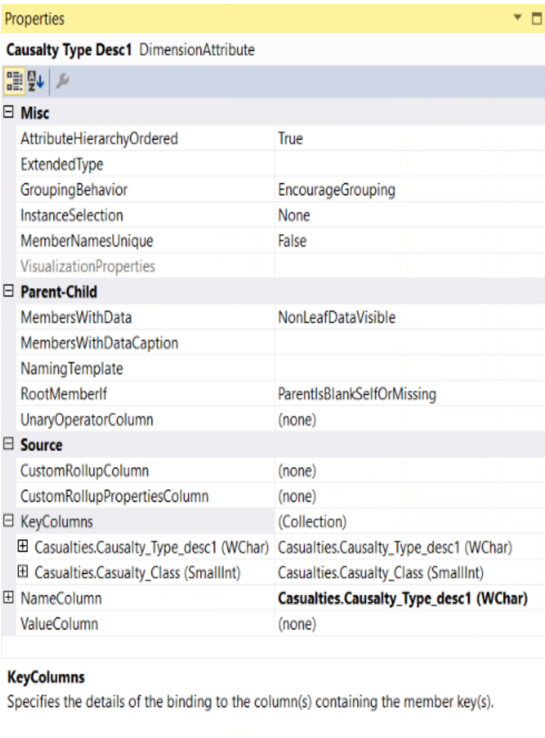
In the below date hierarchy screenshot, we see the individual years from 2005 to 2015. We can drill down within any given year to find the half yearly, quarterly and monthly records respectively.

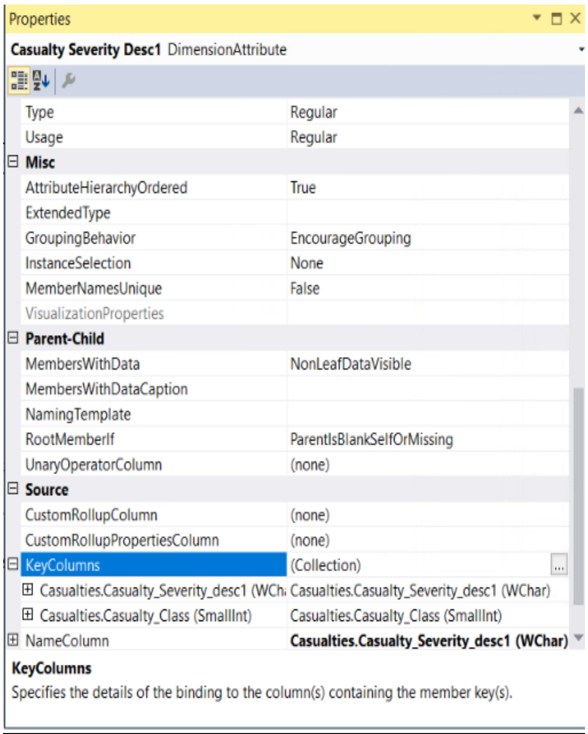


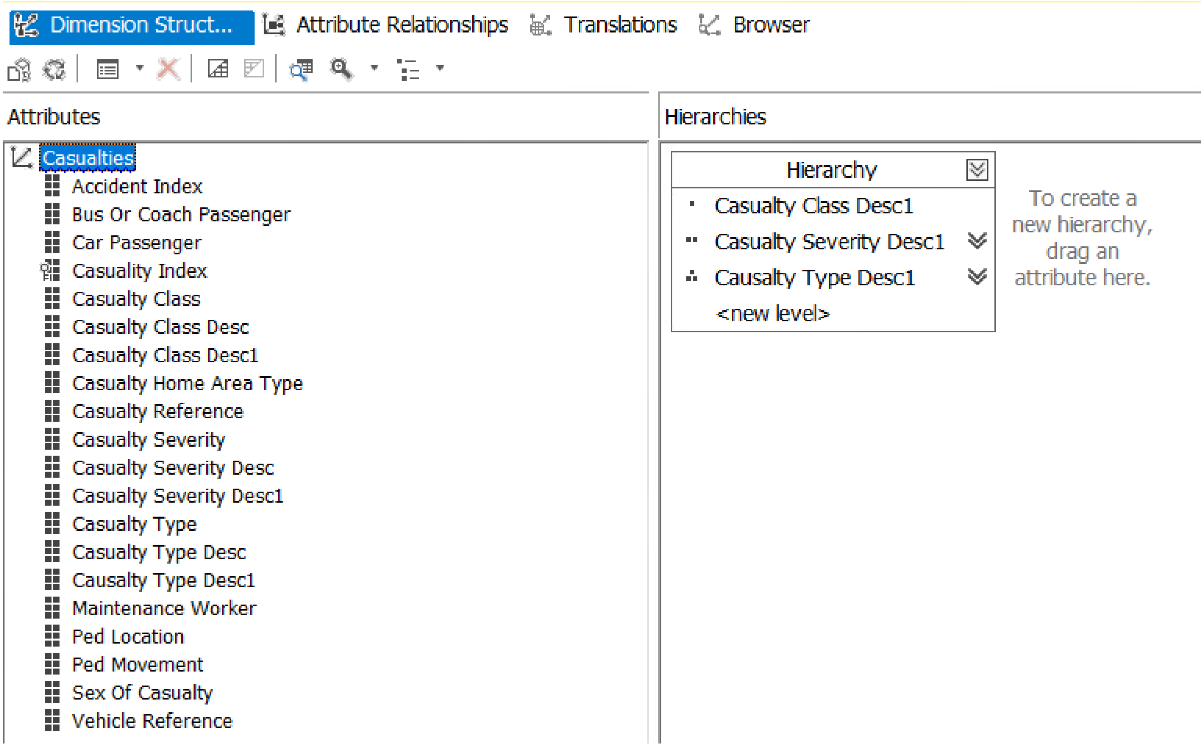
Casualties Dimension Hierarchy

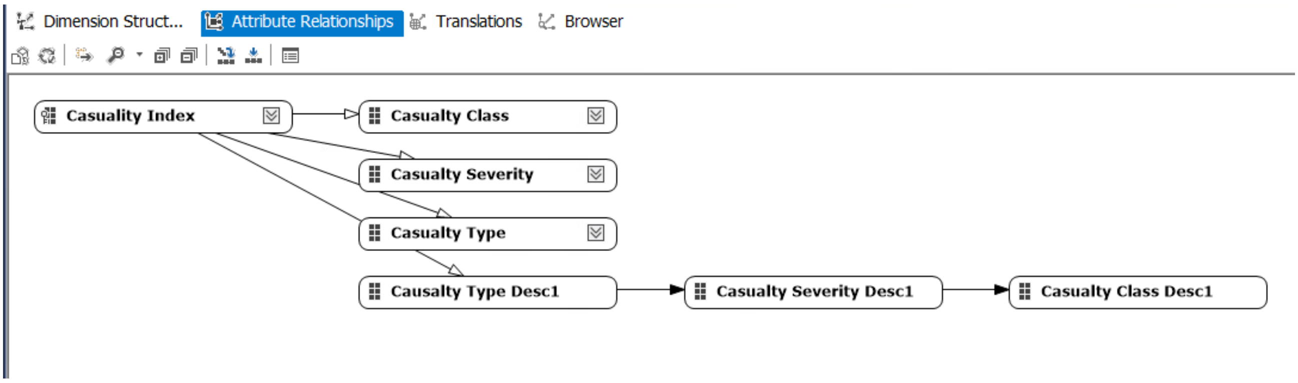
Every casualty class has 3 casualty severities. Every casualty severity has further casualty types. To get a unique record for each casualty class, severity and type we have a composite key structure as shown below.

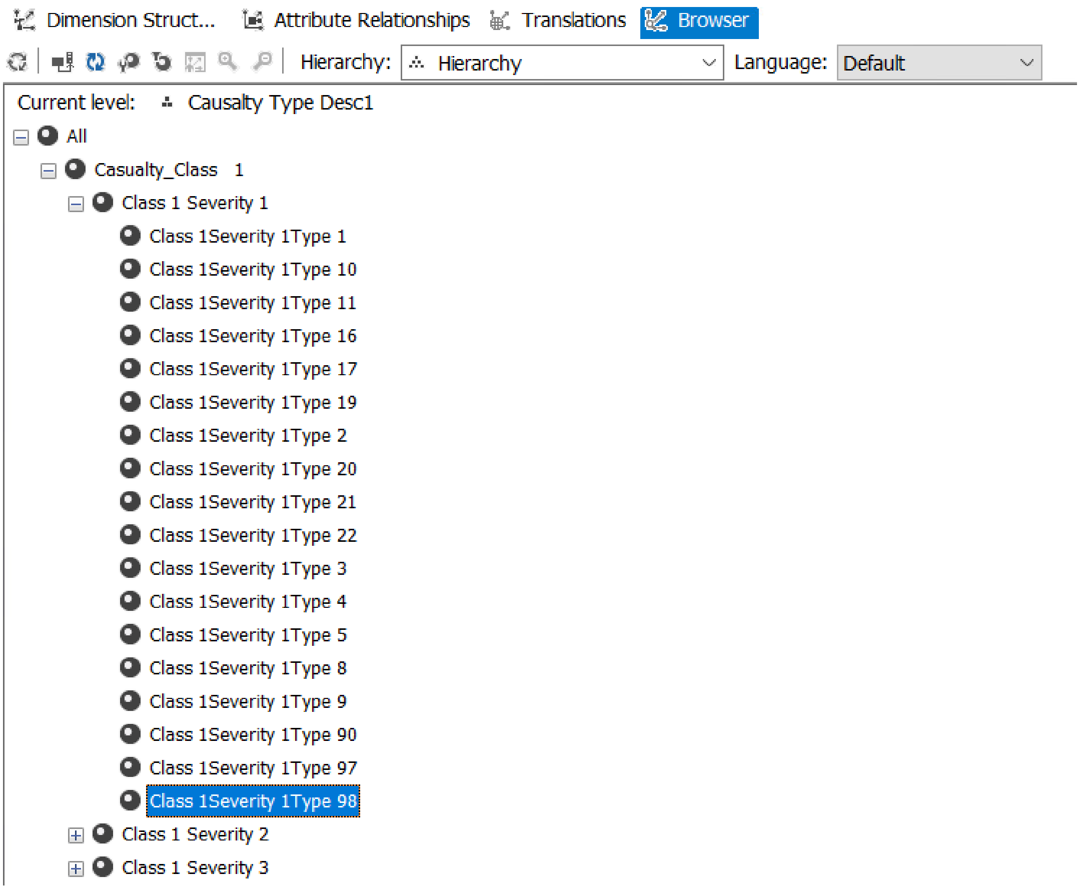






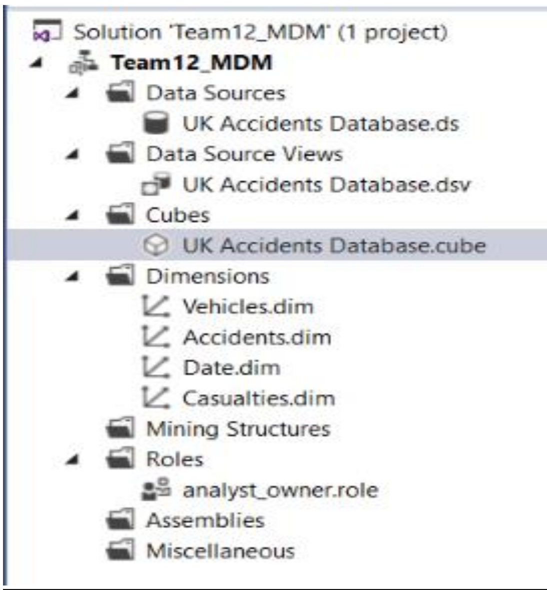

Dimension Struct...
Attribute Relationships
Translations
Browser
Attributes
Hierarchies
Casualties
Accident Index
Bus Or Coach Passenger
Car Passenger
Casuality Index
Casualty Class
Casualty Class Desc
Casualty Class Desc1
Casualty Home Area Type
Casualty Reference
Casualty Severity
Casualty Severity Desc
Casualty Severity Desc1
Casualty Type
Casualty Type Desc
Causalty Type Desc1
Maintenance Worker
Ped Location
Ped Movement
Sex Of Casualty
Vehicle Reference
Hierarchy
Casualty Class Desc1
Casualty Severity Desc1
Causalty Type Desc1
<new level>
To create a new hierarchy, drag an attribute here.


Dimension Struct...
Attribute Relationships
Translations
Browser
Casuality Index
Casualty Class
Casualty Severity
Casualty Type
Causalty Type Desc1
Casualty Severity Desc1
Casualty Class Desc1



UK Accidents Database Cube

We built the UK Accidents Database Cube using Vehicles, Accidents,date and Casualties dimensions.



Partitions and Aggregation

Partitions are used to manage and store data and aggregations for a measure group within a cube. When we process a partition, data is brought into the partition from the source. (Reference: Lecture 6 Part B slide 25)
We have created and deployed 2 partitions of the Accidents Masters Fact table(datetimekey – one up to year 2010 , the other after 2010).
The 1st partition includes any dates less than 2010 and the 2nd partition is based on the dates greater than 2010.

Accidents Master (2 Partitions)

	Partition Name	Source	Estimated Rows	Storage Mode	Aggregation Design
1	Accidents Master After 2010	SELECT [dbo].[Accidents_Master].[Accident_key],[dbo].[Accidents_Master...	0	MOLAP	AggregationDesign 30 percent
2	Accidents Master upto 2010	SELECT [dbo].[Accidents_Master].[Accident_key],[dbo].[Accidents_Master...	2779598	MOLAP	AggregationDesign 30 percent

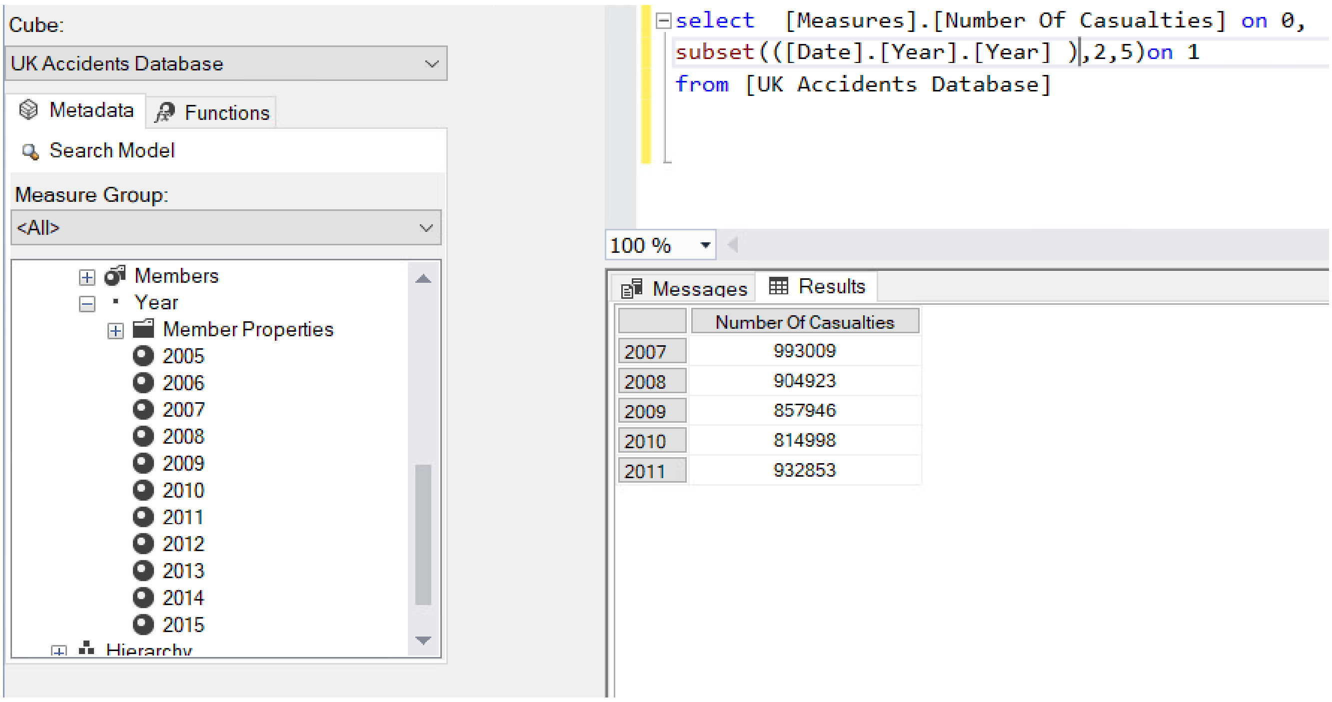
New Partition... Storage Settings...

We created Aggregations in each partition for 30% Performance.

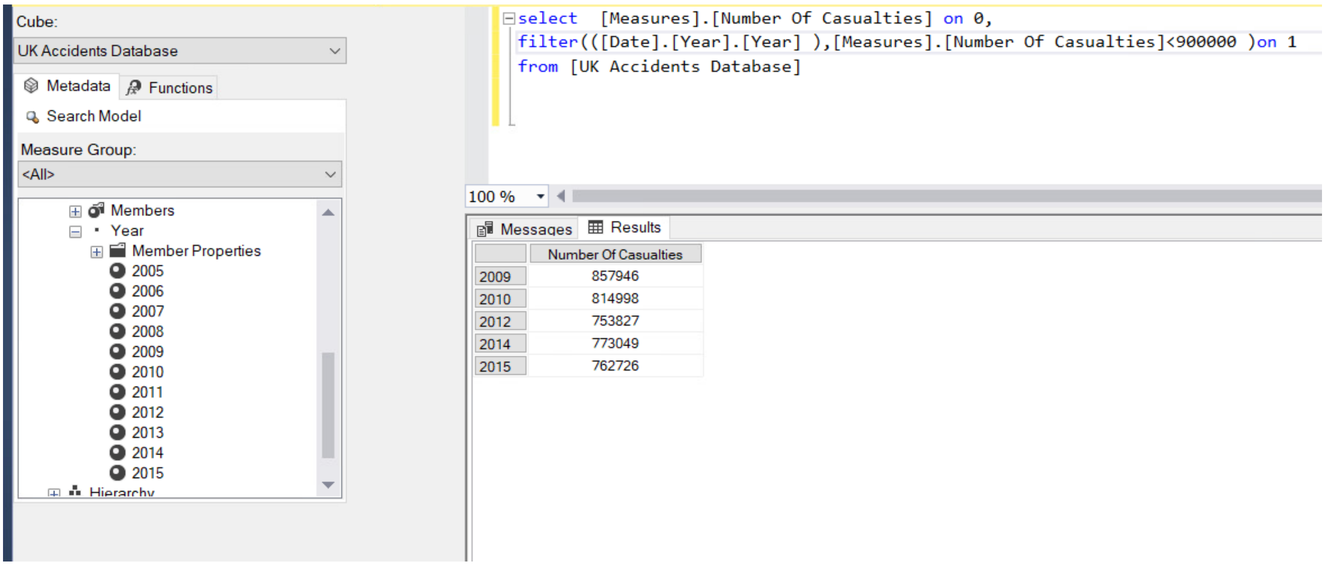
	Aggregations	Estimated Partition Size	Partitions
Accidents Master (1 Aggregation Design)			
AggregationDesign 30 percent	2	2779598	Accidents Master upto 2010, Accidents Master After 2010
Accidents Master 1 (0 Aggregation Designs)			

MDX Queries

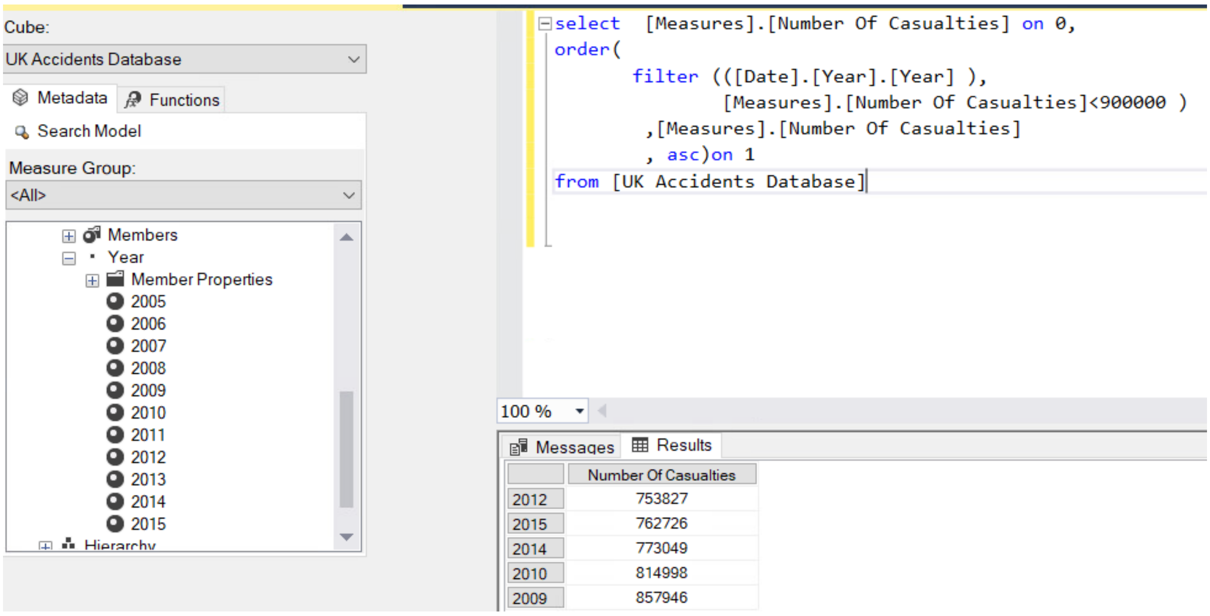
1 Display the number of casualties for the next 5 years starting from second year in the date hierarchy



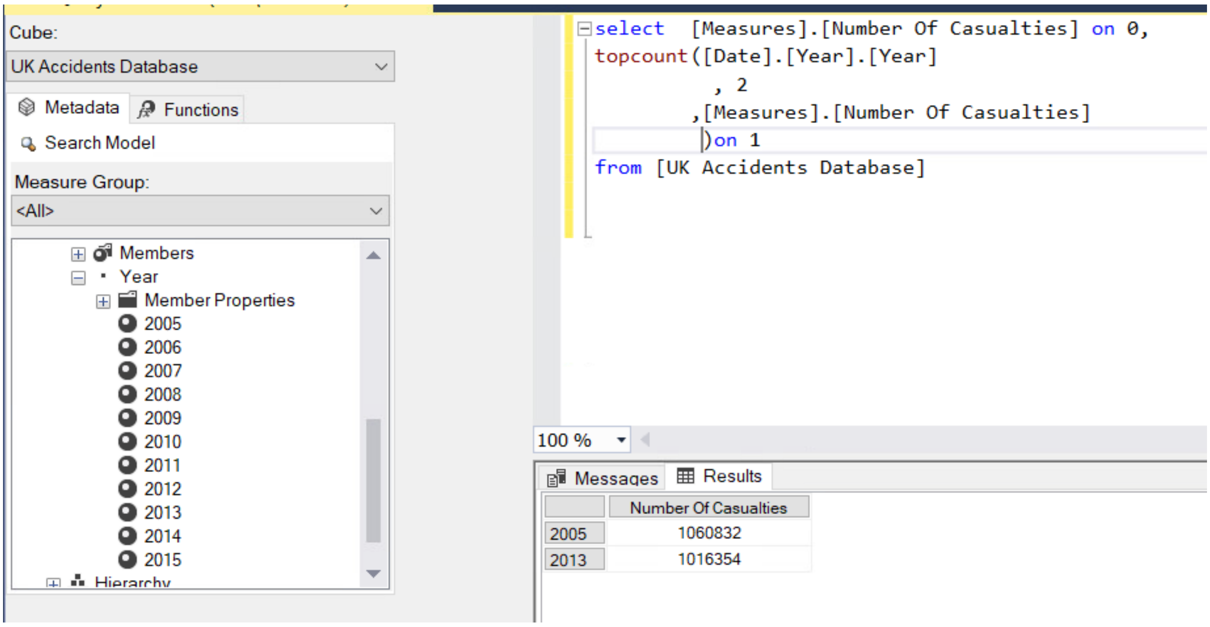
2. Display the number of Casualties for the years where number of casualties are less than 900000



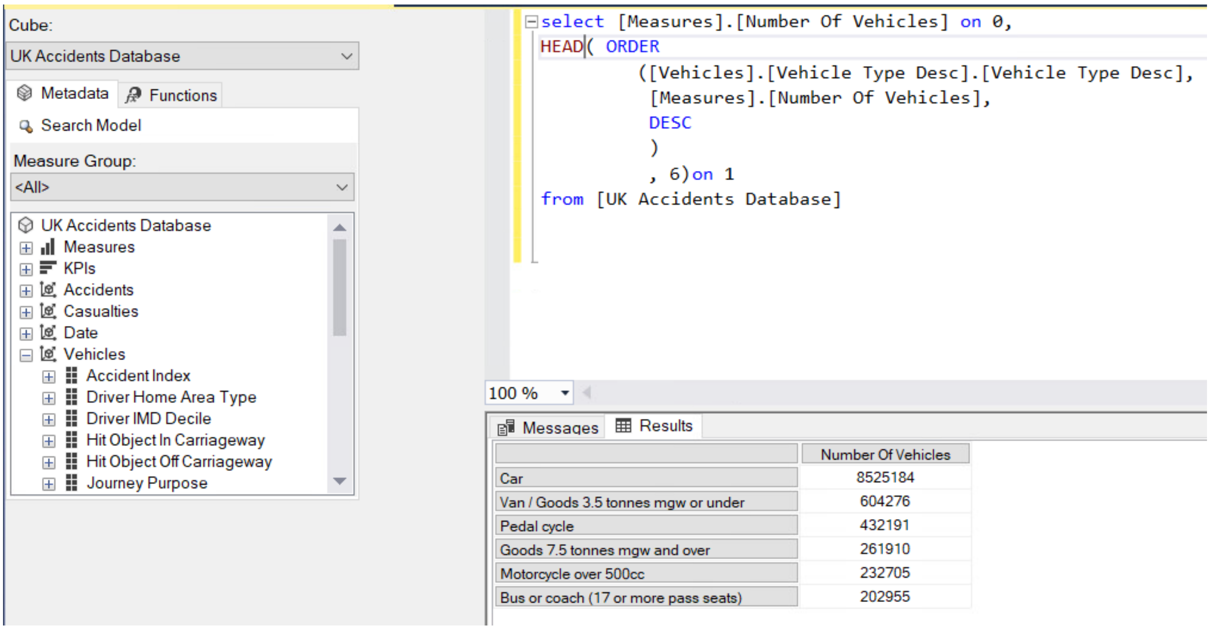
3. After displaying the casualties in the previous question, order by the number of casualties (ascending order)



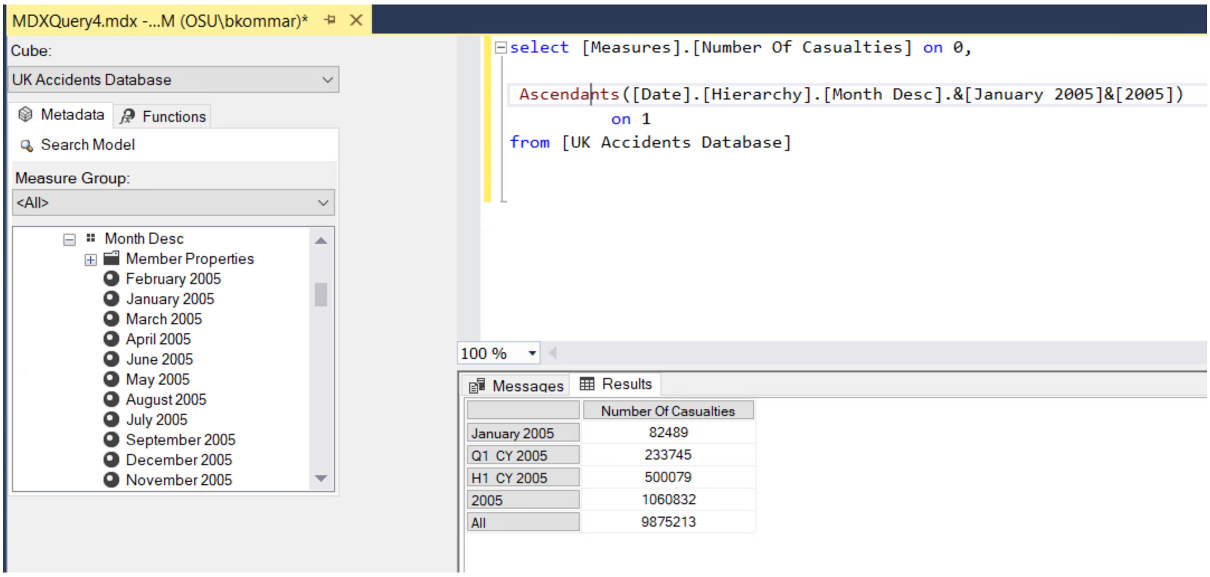
4. Display the top 2 years with the highest number of casualties



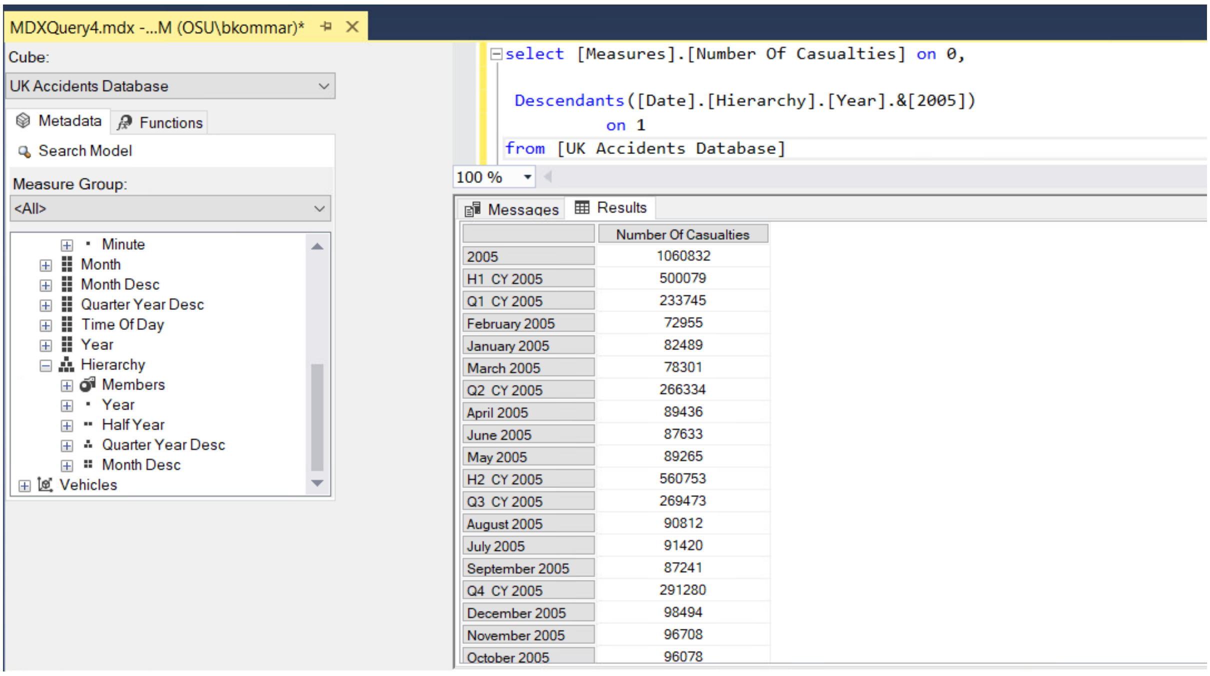
5. Display the top 6 vehicles and the vehicle description



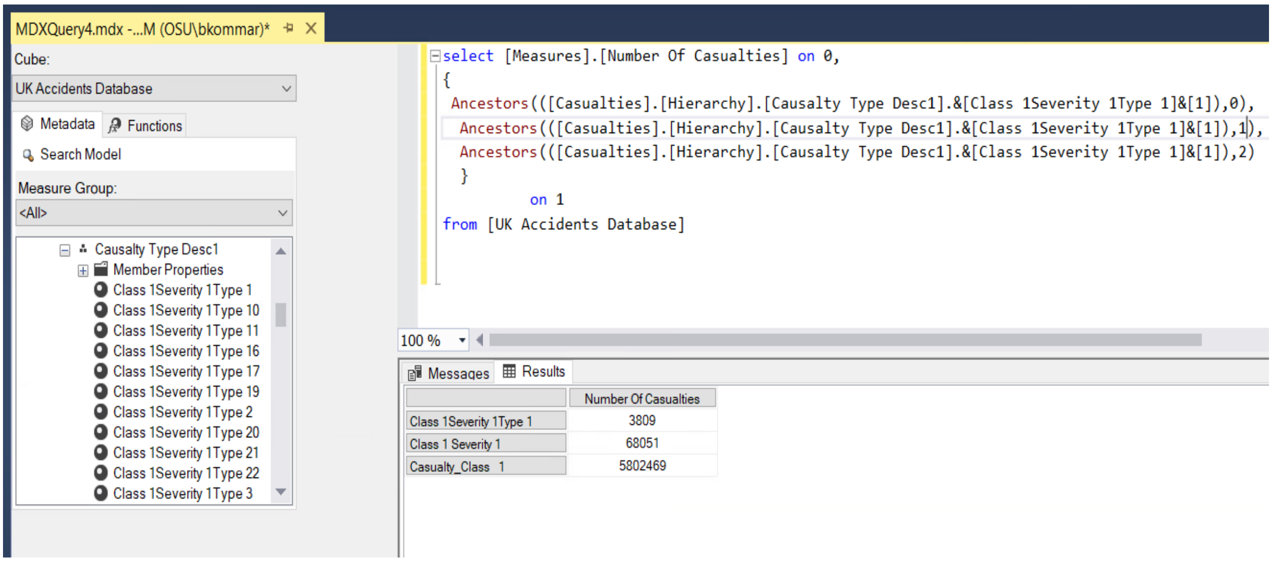
6. Display the number of casualties for the ascendants in the year 2005



7. Display the number of casualties for the descendants in the year 2005



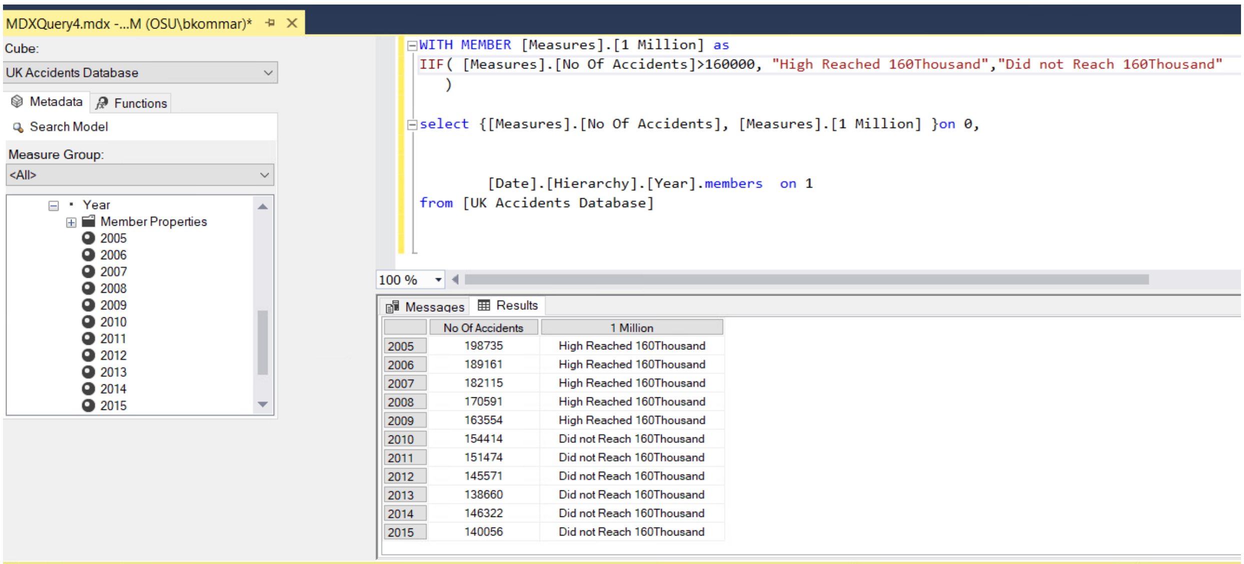
8. Display the number of casualties for the ancestors of Class1Severity1Type1 in the casualty hierarchy (user-defined) at level 0,1,2




```
select [Measures].[Number Of Casualties] on 0,
{
 Ancestors(([Casualties].[Hierarchy].[Causalty Type Desc1].&[Class 1Severity 1Type 1]&[1]),0),
  Ancestors(([Casualties].[Hierarchy].[Causalty Type Desc1].&[Class 1Severity 1Type 1]&[1]),1),
  Ancestors(([Casualties].[Hierarchy].[Causalty Type Desc1].&[Class 1Severity 1Type 1]&[1]),2)
  }
          on 1
from [UK Accidents Database]
```

	Number Of Casualties
Class 1Severity 1Type 1	3809
Class 1 Severity 1	68051
Casualty_Class 1	5802469

9. Display the number of accidents if it reached 160,000 or not for every year



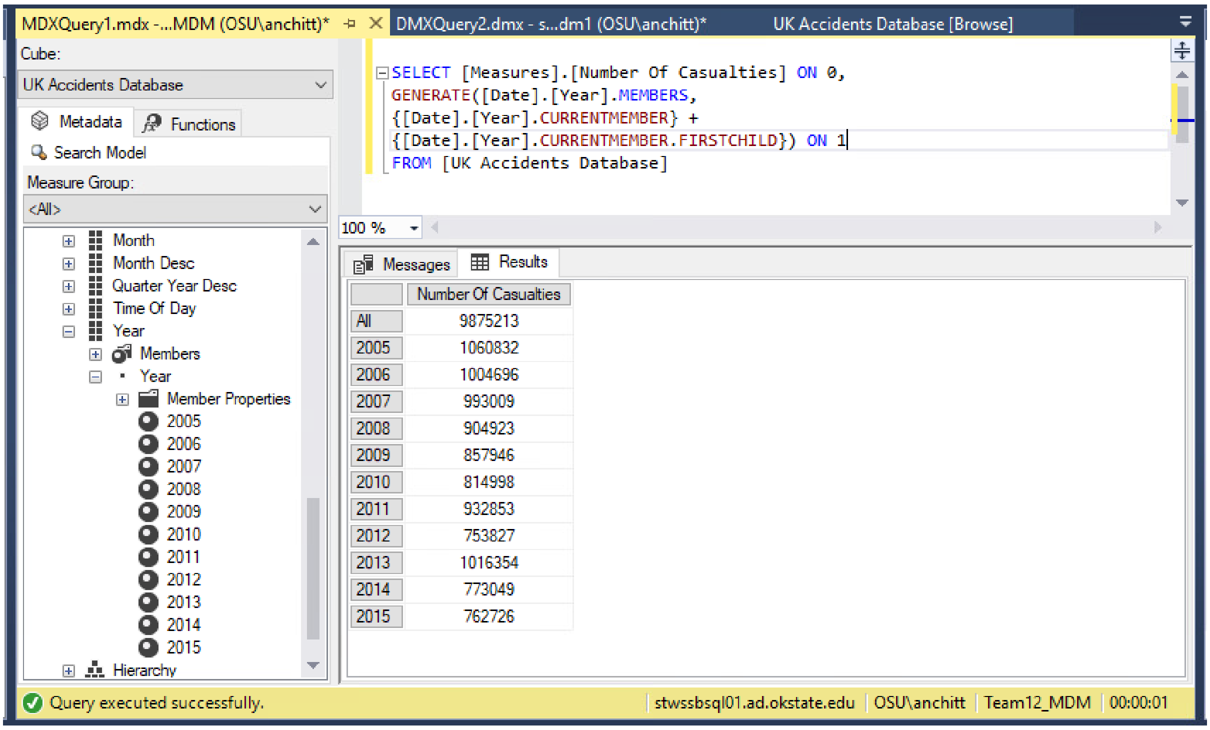

```
WITH MEMBER [Measures].[1 Million] as
IIF( [Measures].[No Of Accidents]>160000, "High Reached 160Thousand","Did not Reach 160Thousand"
  )

select {[Measures].[No Of Accidents], [Measures].[1 Million] }on 0,


         [Date].[Hierarchy].[Year].members  on 1
from [UK Accidents Database]
```

	No Of Accidents	1 Million
2005	198735	High Reached 160Thousand
2006	189161	High Reached 160Thousand
2007	182115	High Reached 160Thousand
2008	170591	High Reached 160Thousand
2009	163554	High Reached 160Thousand
2010	154414	Did not Reach 160Thousand
2011	151474	Did not Reach 160Thousand
2012	145571	Did not Reach 160Thousand
2013	138660	Did not Reach 160Thousand
2014	146322	Did not Reach 160Thousand
2015	140056	Did not Reach 160Thousand

10. Display the number of casualties for all years using GENERATE function.



Data Mining

Data mining involves exploring and analyzing large data to discover the hidden patterns and rules. It’s basically a technique used to predict future outcomes. (DataMining)

STWSSBSQL01.UK_A...dbo.DimAccidents

Column Name	Data Type
Accident_Index	varchar(50)
Location_Easting_OSGR	varchar(50)
Location_Northing_OSGR	varchar(50)
Longitude	varchar(50)
Latitude	varchar(50)
Police_Force	varchar(50)
Accident_Severity	smallint
Local_Authority_District	smallint
Local_Authority_Highway	varchar(50)
First_Road_Class	varchar(50)
First_Road_Number	varchar(50)
Road_Type	smallint
Speed_limit	varchar(50)
Junction_Detail	varchar(50)
Junction_Control	varchar(50)
Second_Road_Class	varchar(50)
Second_Road_Number	varchar(50)
Ped_Cross_Human	varchar(50)
Ped_Cross_Physical	varchar(50)
Light_Conditions	smallint
Weather_Conditions	varchar(50)
Road_Surface_Conditions	varchar(50)
Special_Conditions_at_Site	varchar(50)
Carriageway_Hazards	varchar(50)
Urban_Rural	varchar(50)
Police_Officer_Attend	varchar(50)
LSOA_of_Accident_Locat...	varchar(50)

STWSSBSQL01.UK_A...- dbo.DimVehicles

Column Name	Data Type
Vehicle_Index	numeric(20, 0)
Accident_Index	varchar(50)
Vehicle_Reference	varchar(50)
Vehicle_Type	smallint
Towing_and_Articulation	varchar(50)
Vehicle_Manoeuvre	smallint
Vehicle_Location	varchar(50)
Junction_Location	varchar(50)
Skidding_and_Overturning	varchar(50)
Hit_Object_in_Carriageway	varchar(50)
Vehicle_Leaving_Carriage...	varchar(50)
Hit_Object_off_Carriage...	varchar(50)
Point_of_Impact	varchar(50)
Left_Hand_Drive	varchar(50)
Journey_Purpose	varchar(50)
Sex_of_Driver	smallint
Propulsion_Code	varchar(50)
Driver_IMD_Decile	varchar(50)

The above pictures show the attributes of Accidents and Vehicles tables. We decided to predict accident severity. To decide on the contributing factors for accident severity, it was necessary to learn more about each attribute in both the above tables.

Accident Severity (we will be predicting this)

This is the target variable or the variable we would be predicting. As seen below, the accident severity which is 'Slight' has highest number of records within accidents table.

SQLQuery1.sql - st...(OSU\rkambad (64))*

```
SELECT *
FROM [dbo].[Accident_Severity]
```

	Accident_Severity	Accident_Severity_Desc
1	1	Fatal
2	2	Serious
3	3	Slight

SQLQuery1.sql - st...(OSU\rkambad (64))*

```
SELECT COUNT(*) , Accident_Severity
FROM [dbo].[DimAccidents]
GROUP BY Accident_Severity
```

	(No column name)	Accident_Severity
1	22998	1
2	242080	2
3	1515575	3

Road_Type

We understand that road type plays an important role in predicting the severity of an accident. To find out what each road type numerical value within the Accidents table meant, we queried the Road_Type table.

As seen below, we will consider all the road type values except road type = -1 or road type = 9 to predict the accident severity.

SQLQuery4.sql - st...(OSU\rkambad (55))*

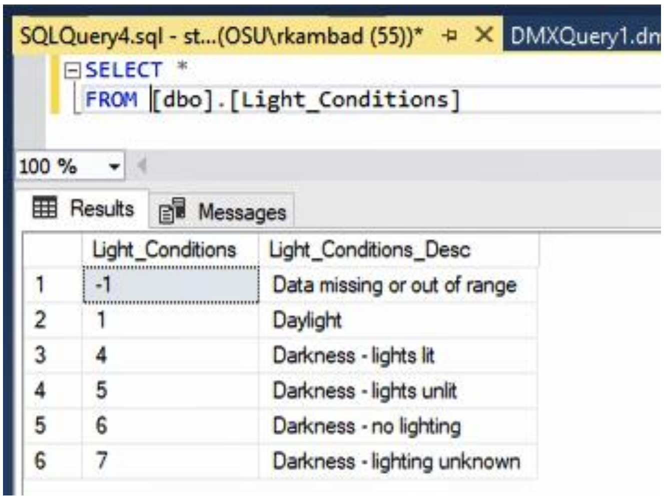
```
SELECT *
FROM [dbo].[Road_Type]
```

	Road_Type	Road_Type_Desc
1	-1	Data missing or out of range
2	1	Roundabout
3	2	One way street
4	3	Dual carriageway
5	6	Single carriageway
6	7	Slip road
7	9	Unknown
8	12	One way street/Slip road

Light condition

Light condition is another factor which greatly influences driving. Darkness while driving leads to accidents. To find out what each numerical value of light_conditions within the Accidents table meant, we queried the Light_conditions table.

As seen below, we will consider all the road type values except road type = -1 to predict the accident severity

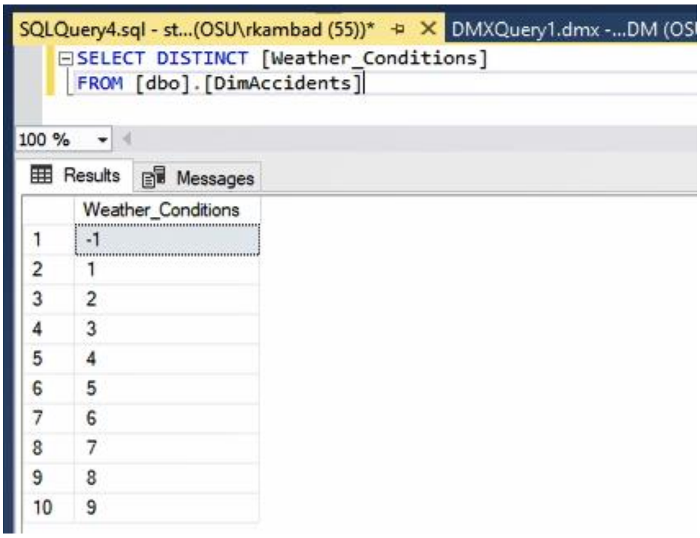



```sql
SELECT *
FROM [dbo].[Light_Conditions]
```

	Light_Conditions	Light_Conditions_Desc
1	-1	Data missing or out of range
2	1	Daylight
3	4	Darkness - lights lit
4	5	Darkness - lights unlit
5	6	Darkness - no lighting
6	7	Darkness - lighting unknown

Weather Conditions

Weather also helps in predicting accidents. Although we didn't have any table that specifically explains what each numerical value of weather conditions within the Accidents table meant, we decided to consider all numerical values except weather_condition =- 1 (believing it to be unknown or missing values) to predict accident severity.




```sql
SELECT DISTINCT [Weather_Conditions]
FROM [dbo].[DimAccidents]
```

	Weather_Conditions
1	-1
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9

Road Surface Condition

The condition of roads may also help in predicting accident severity. We decided to consider all the values except road_surface_conditions = -1 for predicting accident severity.

```sql
SELECT DISTINCT Road_Surface_Conditions
FROM [dbo].[DimAccidents]
```

	Road_Surface_Conditions
1	-1
2	1
3	2
4	3
5	4
6	5

Special Conditions at Site

We decided to eliminate special conditions at site = -1 while predicting accident severity. Although we do not know what special_conditions_at_site = 0 meant, based on the large number of records available, we decided to consider this value also for predicting accident severity.

```sql
SELECT COUNT(*) ,
       Special_Conditions_at_Site
FROM [dbo].[DimAccidents]
GROUP BY Special_Conditions_at_Site
```

	(No column name)	Special_Conditions_at_Site
1	124	-1
2	1736828	0
3	3263	1
4	901	2
5	2616	3
6	20741	4
7	4293	5
8	6222	6
9	5665	7

Urban Rural

We know that accidents happen more in urban areas compared to rural as the number of vehicles operating in urban areas are more. We think that the value of 1 indicates urban and a 2 indicates rural. Although there are only 143 records for urban_rural =3 within the accidents table, we decided to use all the three values of urban_rural to predict accident severity.

```sql
SELECT COUNT(*) ,
       Urban_Rural
FROM [dbo].[DimAccidents]
GROUP BY Urban_Rural
```

	(No column name)	Urban_Rural
1	1146421	1
2	634089	2
3	143	3

Junction detail

A junction is where two or more roads meet. It may also influence the accident severity. Hence, we are using this variable within our mining structure. We will not consider junction_detail = -1 for predicting accident severity.

```sql
SELECT COUNT(*) ,
       [Junction_Detail]
FROM [dbo].[DimAccidents]
GROUP BY [Junction_Detail]
```

	(No column name)	Junction_Detail
1	19	-1
2	716544	0
3	154723	1
4	19206	2
5	553692	3
6	26054	5
7	170738	6
8	23060	7
9	65219	8
10	51398	9

Junction Control

We will not consider junction_control=-1 for predicting the accident severity.

SQLQuery4.sql - st...(OSU\rkambad (55))* DMXQuer

```
SELECT COUNT(*) ,
       Junction_Control
FROM [dbo].[DimAccidents]
GROUP BY Junction_Control
```

100 %

Results | Messages

	(No column name)	Junction_Control
1	641392	-1
2	76916	0
3	2918	1
4	182829	2
5	10841	3
6	865757	4

Vehicle Type

The type of vehicle whether car, cycle and a motorcycle etc may also help in predicting accident severity.

SQLQuery4.sql - st...(OSU\rkambad (55))* DMXQuery1.dm

```
SELECT *
FROM [dbo].[Vehicle_Type]
```

100 %

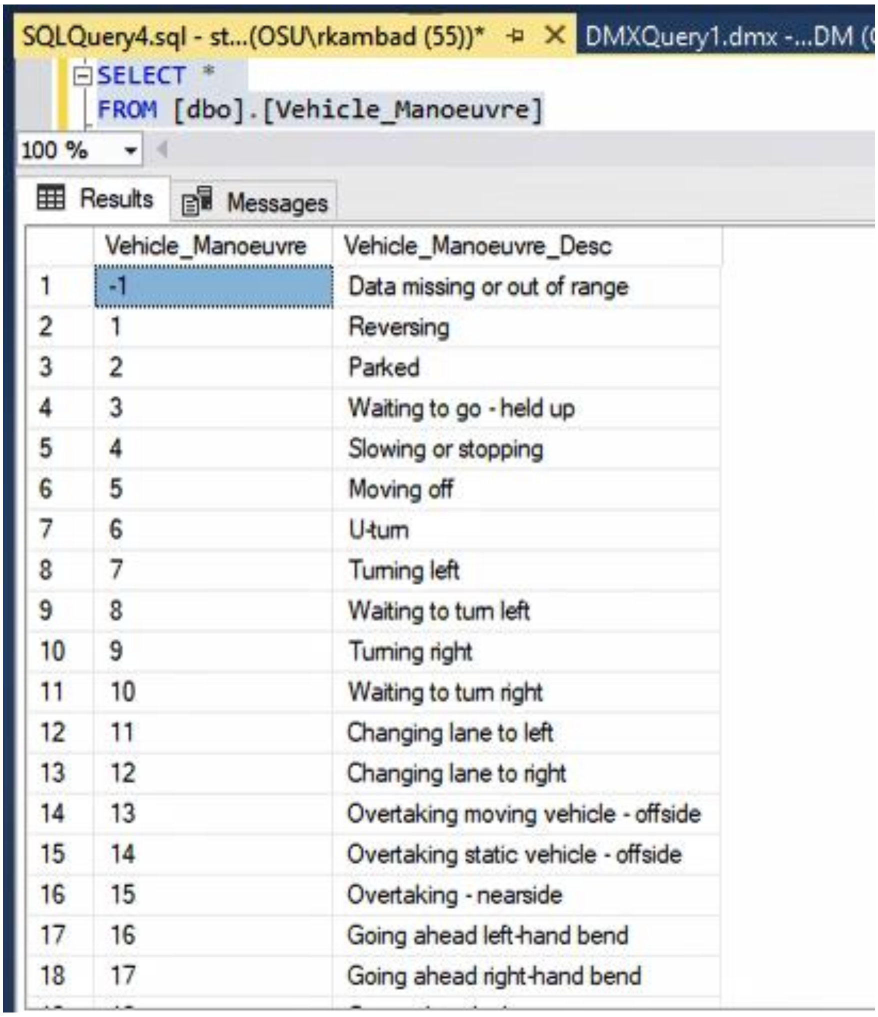
Results | Messages

	Vehicle_Type	Vehicle_Type_Desc
1	-1	Data missing or out of range
2	1	Pedal cycle
3	2	Motorcycle 50cc and under
4	3	Motorcycle 125cc and under
5	4	Motorcycle over 125cc and up to 500cc
6	5	Motorcycle over 500cc
7	8	Taxi/Private hire car
8	9	Car
9	10	Minibus (8 - 16 passenger seats)
10	11	Bus or coach (17 or more pass seats)
11	16	Ridden horse
12	17	Agricultural vehicle
13	18	Tram
14	19	Van / Goods 3.5 tonnes mgw or under
15	20	Goods over 3.5t. and under 7.5t
16	21	Goods 7.5 tonnes mgw and over
17	22	Mobility scooter
18	23	Electric motorcycle
19	90	Other vehicle
20	97	Motorcycle - unknown cc
21	98	Goods vehicle - unknown weight

Vehicle Manoeuvre

To understand what each numerical value of vehicle_manoeuvre meant within the accidents table,we queried the Vehicle Manoeuvre table. The description of the values can be seen in the below screenshot. Whether a vehicle was taking a U-turn, overtaking another or changing lanes may also help in predicting accident severity.

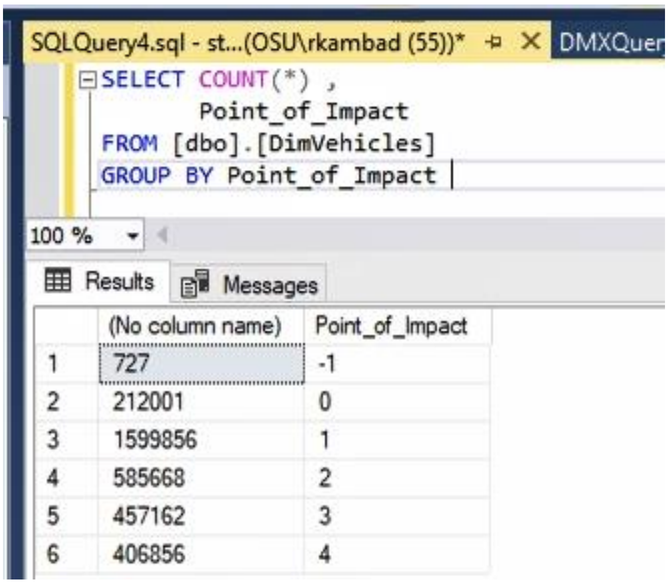
We won't consider vehicle_manoeuvre = -1 for our mining models.



	Vehicle_Manoeuvre	Vehicle_Manoeuvre_Desc
1	-1	Data missing or out of range
2	1	Reversing
3	2	Parked
4	3	Waiting to go - held up
5	4	Slowing or stopping
6	5	Moving off
7	6	U-turn
8	7	Turning left
9	8	Waiting to turn left
10	9	Turning right
11	10	Waiting to turn right
12	11	Changing lane to left
13	12	Changing lane to right
14	13	Overtaking moving vehicle - offside
15	14	Overtaking static vehicle - offside
16	15	Overtaking - nearside
17	16	Going ahead left-hand bend
18	17	Going ahead right-hand bend

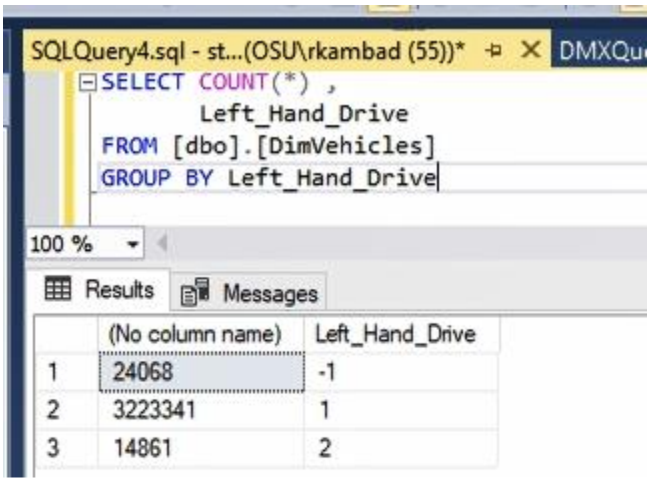
Point of Impact

We think that the point of impact refers to whether the impact was on the left, right, front or rear of the vehicle etc. We will consider all the values for point of impact except -1



Left Hand Drive

We will not consider the left_hand_drive = -1 for our mining models.



Sex of Driver

The sex of the driver may also help in predicting accident severity. We have considered only the values sex of driver = 1 or 2 to predict the severity of accidents.

SQLQuery4.sql - st...(OSU\rkambad (55))*

```
SELECT COUNT(*) ,
        Sex_of_Driver
FROM [dbo].[DimVehicles]
GROUP BY Sex_of_Driver
```

	(No column name)	Sex_of_Driver
1	52	-1
2	2147401	1
3	924565	2
4	190252	3

SQLQuery2.sql - st...(OSU\rkambad (61))*

```
SELECT *
FROM [dbo].[Sex_of_Driver]
```

	Sex_of_Driver	Sex_of_Driver_Desc
1	-1	Data missing or out of range
2	1	Male
3	2	Female
4	3	Not known

Journey Purpose

Journey purpose may also help in predicting the severity of an accident. We will not consider journey_purpose = -1 for our mining model.

SQLQuery4.sql - st...(OSU\rkambad (55))*

```
SELECT COUNT(*) ,
        Journey_Purpose
FROM [dbo].[DimVehicles]
GROUP BY Journey_Purpose
```

	(No column name)	Journey_Purpose
1	44945	-1
2	545046	1
3	1362699	15
4	311840	2
5	31786	3
6	11118	4
7	22462	5
8	932374	6

Based on the above analysis of the attributes within Accidents and Vehicle table, we used the following attributes to predict accident severity.

Attributes from accidents table

- Accident_Index - Key column which uniquely identifies an entity
- Road_Type
- Speed_limit
- Light_Conditions
- Weather_Conditions
- Road_Surface_Conditions
- Special_Conditions_at_Site
- Urban_Rural
- Junction_Detail
- Junction_Control

Attributes from Vehicles table

- Vehicle_Type
- Vehicle_Manoeuvre
- Point_of_Impact
- Left_Hand_Drive
- Sex_of_Driver
- Journey Purpose

Various mining models could be built using the above UK Accidents Database. We decided to use the following data mining techniques based on Microsoft data mining algorithms.

- Decision Tree
- Logistic Regression
- Neural Networks

Creating the Mining Structure

We will use the CREATE MINING STRUCTURE DMX statement to create the mining structure. Since we will be creating many mining models, we will use ALTER MINING STRUCTURE statement to add mining models to the structure.

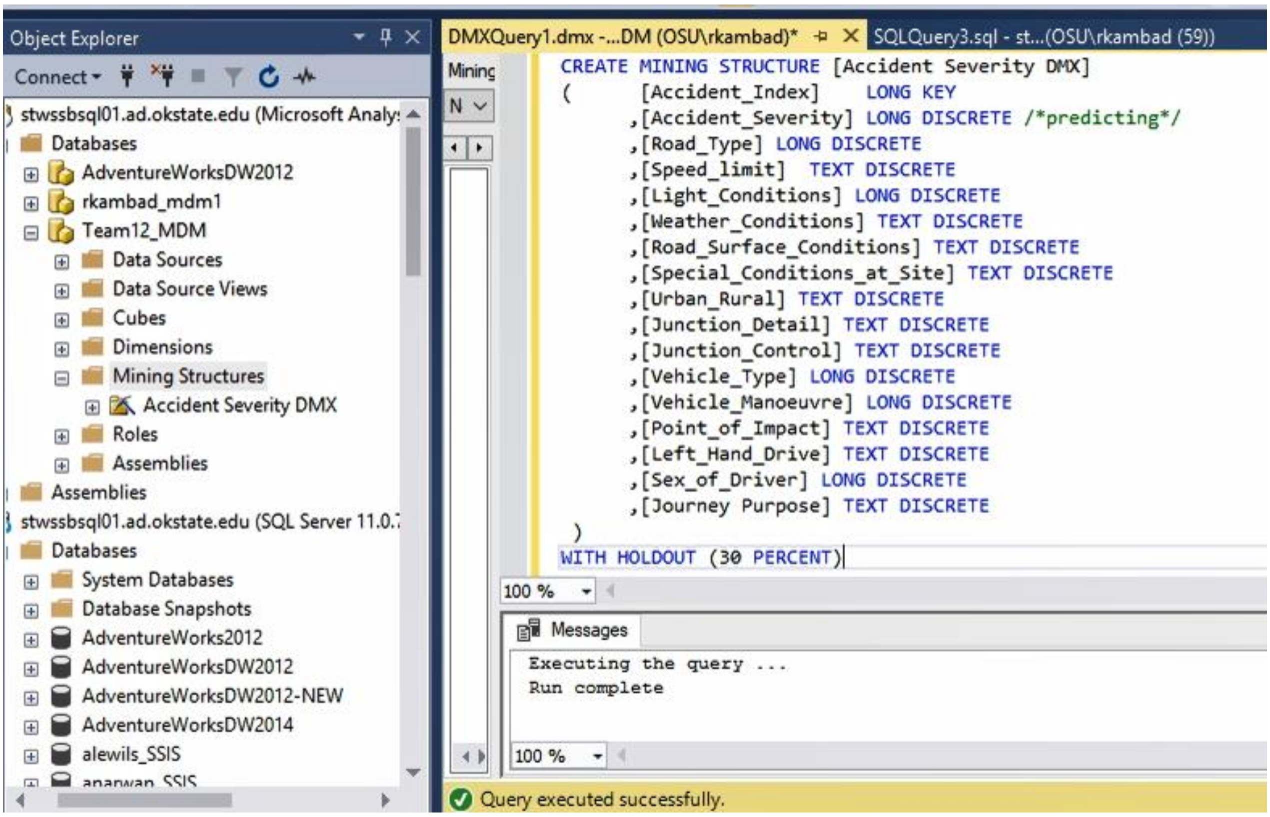
The name of the mining structure we created is : Accident Severity DMX

```
CREATE MINING STRUCTURE [Accident Severity DMX]
(    [Accident_Index]    LONG KEY
    ,[Accident_Severity] LONG DISCRETE /*predicting*/
    ,[Road_Type] LONG DISCRETE
    ,[Speed_limit]  TEXT DISCRETE
    ,[Light_Conditions] LONG DISCRETE
    ,[Weather_Conditions] TEXT DISCRETE
    ,[Road_Surface_Conditions] TEXT DISCRETE
    ,[Special_Conditions_at_Site] TEXT DISCRETE
    ,[Urban_Rural] TEXT DISCRETE
    ,[Junction_Detail] TEXT DISCRETE
    ,[Junction_Control] TEXT DISCRETE
    ,[Vehicle_Type] LONG DISCRETE
    ,[Vehicle_Manoeuvre] LONG DISCRETE
    ,[Point_of_Impact] TEXT DISCRETE
    ,[Left_Hand_Drive] TEXT DISCRETE
    ,[Sex_of_Driver] LONG DISCRETE
    ,[Journey Purpose] TEXT DISCRETE
 )
WITH HOLDOUT (30 PERCENT)
```

The key column for the mining structure uniquely identifies an entity in the source data. We have also defined the mining columns. Additionally, we specified what portion of the data is used for testing mining models. The remaining data is used for training the models.

By default, analysis services will create a test data set which contains 30% of all the data. We can also add a specification that the test data set should contain 30% of the cases up to a maximum of 1000 cases. (reference lecture 7, slide no. 17)

In the below screenshot, we see the Accident Severity DMX mining structure created under the Mining Structure folder.



Creating Mining Models

After identifying the mining structure, we will use the ALTER MINING STRUCTURE STATEMENT to alter the mining structure and add mining models.

It is necessary to define the predictable and the input columns. Additionally, we must determine which algorithm to use.

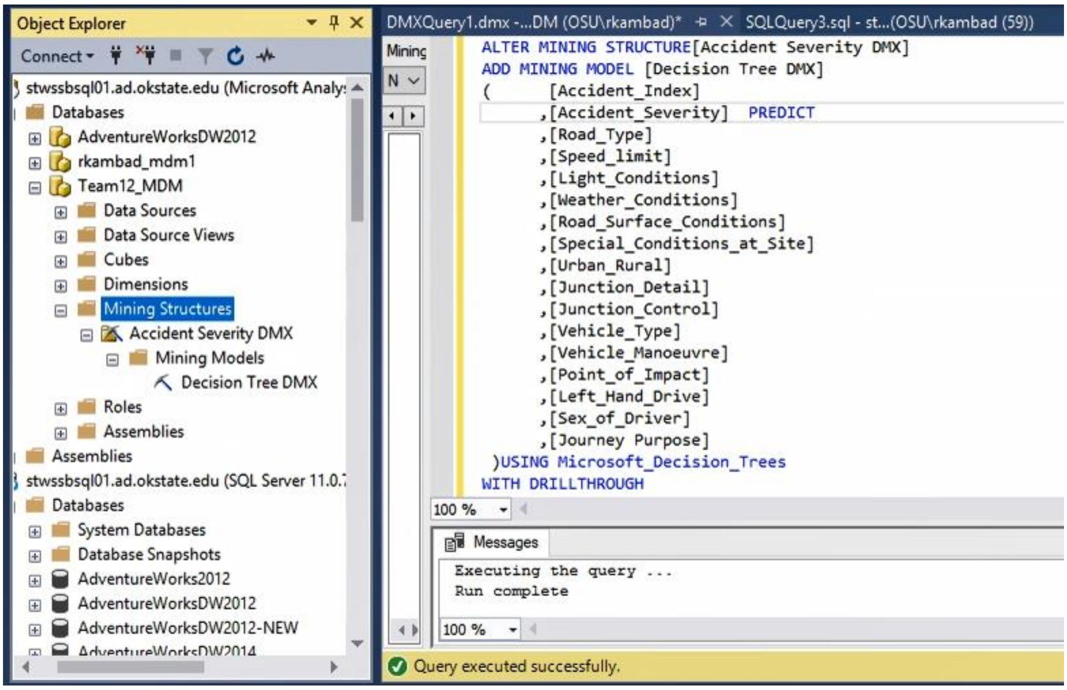
As stated earlier, we will use the following data mining techniques based on Microsoft data mining algorithms.

- Decision Tree
- Logistic Regression
- Neural Networks

Model1: Decision Tree Mining Model

```
ALTER MINING STRUCTURE[Accident Severity DMX]
ADD MINING MODEL [Decision Tree DMX]
(    [Accident_Index]
    ,[Accident_Severity]  PREDICT
    ,[Road_Type]
    ,[Speed_limit]
    ,[Light_Conditions]
    ,[Weather_Conditions]
    ,[Road_Surface_Conditions]
    ,[Special_Conditions_at_Site]
    ,[Urban_Rural]
    ,[Junction_Detail]
    ,[Junction_Control]
    ,[Vehicle_Type]
    ,[Vehicle_Manoeuvre]
    ,[Point_of_Impact]
    ,[Left_Hand_Drive]
    ,[Sex_of_Driver]
    ,[Journey Purpose]
 )USING Microsoft_Decision_Trees
WITH DRILLTHROUGH
```

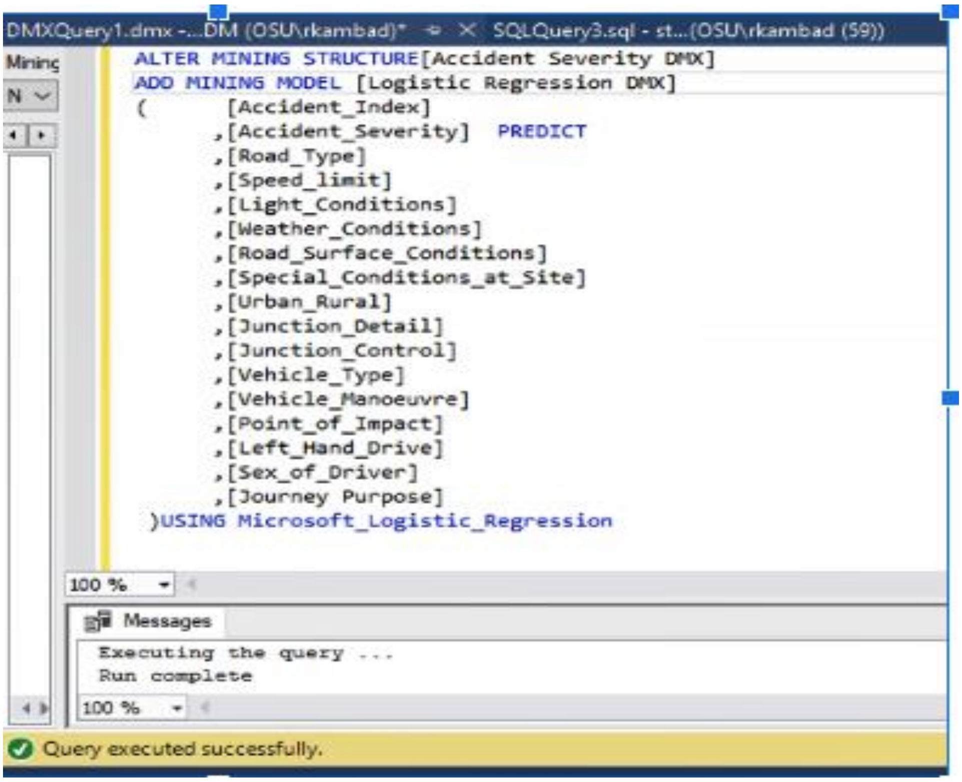
In the below screenshot, we see the Decision Tree DMX mining model created under Accident Severity DMX mining structure.



Model2: Logistic Regression Mining Model

```
ALTER MINING STRUCTURE[Accident Severity DMX]
ADD MINING MODEL [Logistic Regression DMX]
(     [Accident_Index]
      ,[Accident_Severity]  PREDICT
      ,[Road_Type]
      ,[Speed_limit]
      ,[Light_Conditions]
      ,[Weather_Conditions]
      ,[Road_Surface_Conditions]
      ,[Special_Conditions_at_Site]
      ,[Urban_Rural]
      ,[Junction_Detail]
      ,[Junction_Control]
      ,[Vehicle_Type]
      ,[Vehicle_Manoeuvre]
      ,[Point_of_Impact]
      ,[Left_Hand_Drive]
      ,[Sex_of_Driver]
      ,[Journey Purpose]
 )USING Microsoft_Logistic_Regression
```

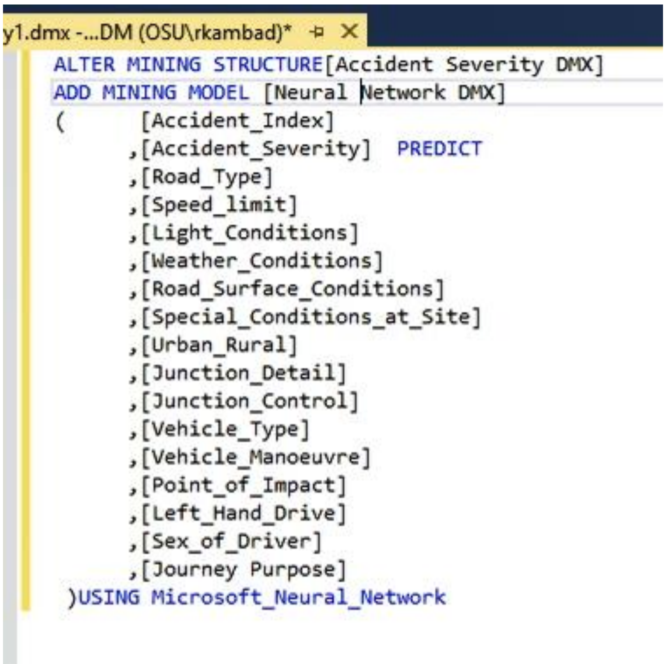
In the below screenshot, we see the Logistic Regression DMX mining model created under Accident Severity DMX mining structure.



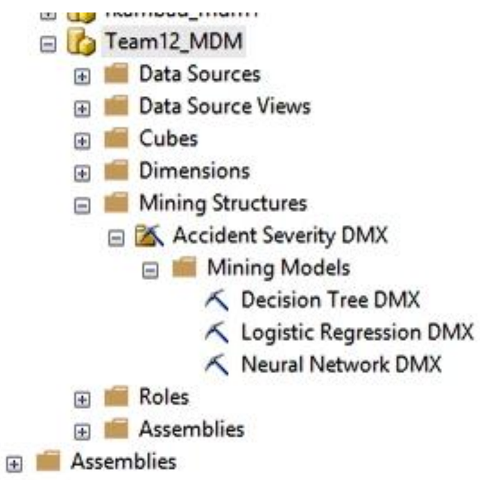
Model3: Neural Networks Mining Model

```
ALTER MINING STRUCTURE[Accident Severity DMX]
ADD MINING MODEL [Neural Network DMX]
(   [Accident_Index]
   ,[Accident_Severity]  PREDICT

     ,[Road_Type]
     ,[Speed_limit]
     ,[Light_Conditions]
     ,[Weather_Conditions]
     ,[Road_Surface_Conditions]
     ,[Special_Conditions_at_Site]
     ,[Urban_Rural]
     ,[Junction_Detail]
     ,[Junction_Control]
     ,[Vehicle_Type]
     ,[Vehicle_Manoeuvre]
     ,[Point_of_Impact]
     ,[Left_Hand_Drive]
     ,[Sex_of_Driver]
     ,[Journey Purpose]
    )USING Microsoft_Neural_Network
```



The below screenshot shows all the three mining models we created under the Accident Severity mining structure.




```
INSERT INTO MINING STRUCTURE [Accident Severity DMX]
(     [Accident_Index]
     ,[Accident_Severity]
     ,[Road_Type]
     ,[Speed_limit]
     ,[Light_Conditions]
     ,[Weather_Conditions]
     ,[Road_Surface_Conditions]
     ,[Special_Conditions_at_Site]
     ,[Urban_Rural]
    ,[Junction_Detail]
     ,[Junction_Control]
     ,[Vehicle_Type]
     ,[Vehicle_Manoeuvre]
     ,[Point_of_Impact]
     ,[Left_Hand_Drive]
     ,[Sex_of_Driver]
     ,[Journey Purpose]
) OPENQUERY ([UK Accidents Database],

'SELECT TOP 100000
     DimAccidents.Accident_Index
     ,Accident_Severity
     ,Road_Type
     ,Speed_limit
     ,Light_Conditions
     ,Weather_Conditions
```

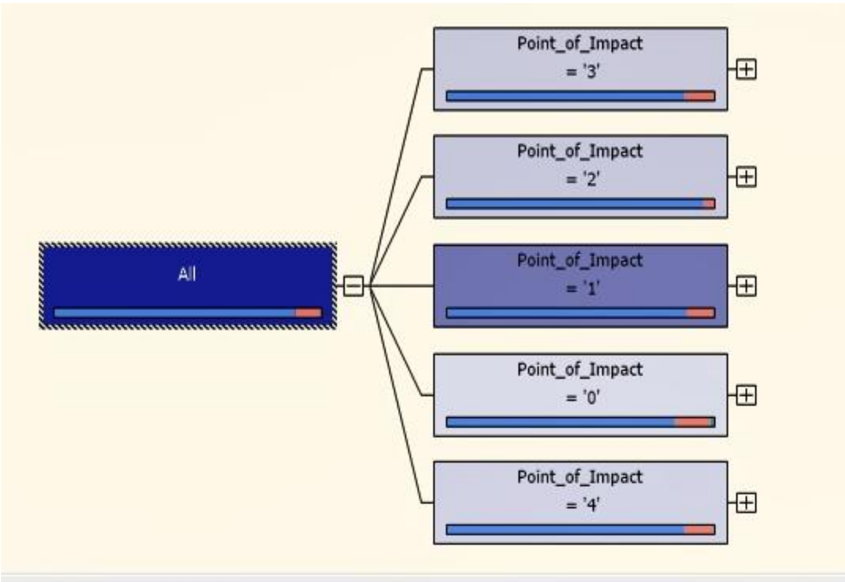
```
    ,Road_Surface_Conditions
    ,Special_Conditions_at_Site
    ,Urban_Rural
   ,Junction_Detail
    ,Junction_Control
    ,Vehicle_Type
    ,Vehicle_Manoeuvre
    ,Point_of_Impact
    ,Left_Hand_Drive
    ,Sex_of_Driver
   ,Journey_Purpose
 FROM DimAccidents ,
          DimVehicles
WHERE DimAccidents.Accident_Index = DimVehicles.Accident_Index
  AND (Road_Type <> -1 OR Road_Type != 9)
  AND Light_Conditions != -1
  AND Weather_Conditions != -1
  AND Road_Surface_Conditions != -1
  AND Special_Conditions_at_Site != -1
  AND Junction_Detail ! = -1
  AND Junction_Control ! = -1
  AND Vehicle_Type != -1
  AND Vehicle_Manoeuvre!=-1
  AND Point_of_Impact !=-1
  AND Left_Hand_Drive!=-1
  AND (Sex_of_Driver !=-1 OR Sex_of_Driver!=3)
  AND Journey_Purpose!=-1')
```

```
INSERT INTO MINING STRUCTURE [Accident Severity DMX]
(       [Accident_Index]
       ,[Accident_Severity]
       ,[Road_Type]
       ,[Speed_limit]
       ,[Light_Conditions]
       ,[Weather_Conditions]
       ,[Road_Surface_Conditions]
       ,[Special_Conditions_at_Site]
       ,[Urban_Rural]
      ,[Junction_Detail]
       ,[Junction_Control]
       ,[Vehicle_Type]
       ,[Vehicle_Manoeuvre]
       ,[Point_of_Impact]
       ,[Left_Hand_Drive]
       ,[Sex_of_Driver]
       ,[Journey Purpose]
) OPENQUERY ([UK Accidents Database],
'SELECT TOP 100000
       DimAccidents.Accident_Index
       ,Accident_Severity
       ,Road_Type
       ,Speed_limit
       ,Light_Conditions
       ,Weather_Conditions
```

```
.dmx -...DM (OSU\rkambad)*
        ,Road_Surface_Conditions
        ,Special_Conditions_at_Site
        ,Urban_Rural
        ,Junction_Detail
        ,Junction_Control
        ,Vehicle_Type
        ,Vehicle_Manoeuvre
        ,Point_of_Impact
        ,Left_Hand_Drive
        ,Sex_of_Driver
        ,Journey_Purpose
    FROM DimAccidents ,
         DimVehicles
   WHERE DimAccidents.Accident_Index = DimVehicles.Accident_Index
     AND (Road_Type <> -1 OR Road_Type != 9)
     AND Light_Conditions != -1
     AND Weather_Conditions != -1
     AND Road_Surface_Conditions != -1
     AND Special_Conditions_at_Site != -1
     AND Junction_Detail ! = -1
     AND Junction_Control ! = -1
     AND Vehicle_Type != -1
     AND Vehicle_Manoeuvre!=-1
     AND Point_of_Impact !=-1
     AND Left_Hand_Drive!=-1
     AND (Sex_of_Driver !=-1 OR Sex_of_Driver!=3)
     AND Journey_Purpose!=-1')
100 %
Messages
Executing the query ...
Run complete
100 %
Query executed successfully.
```

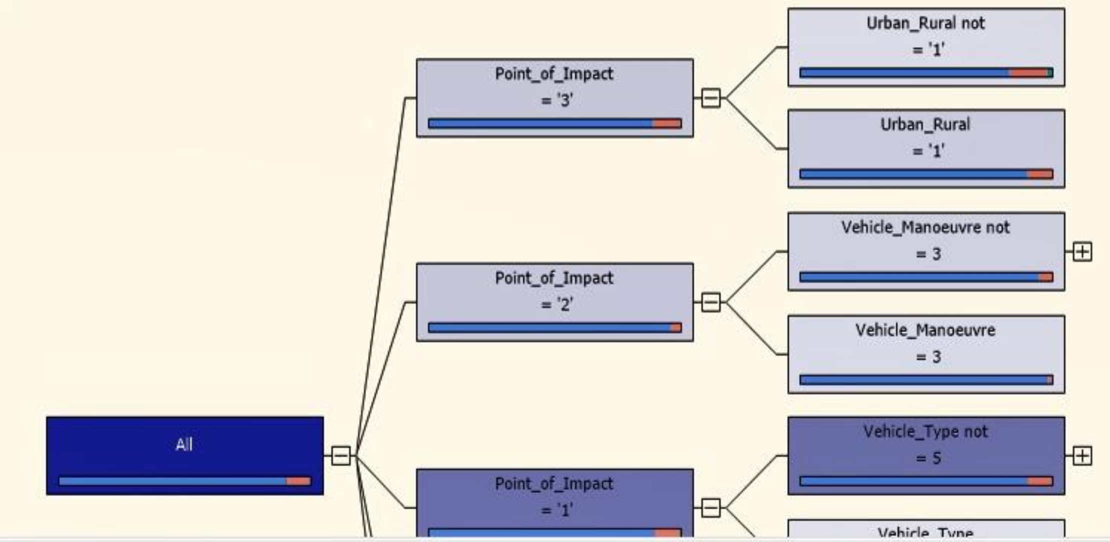
Model 1: Decision Tree

1st Level



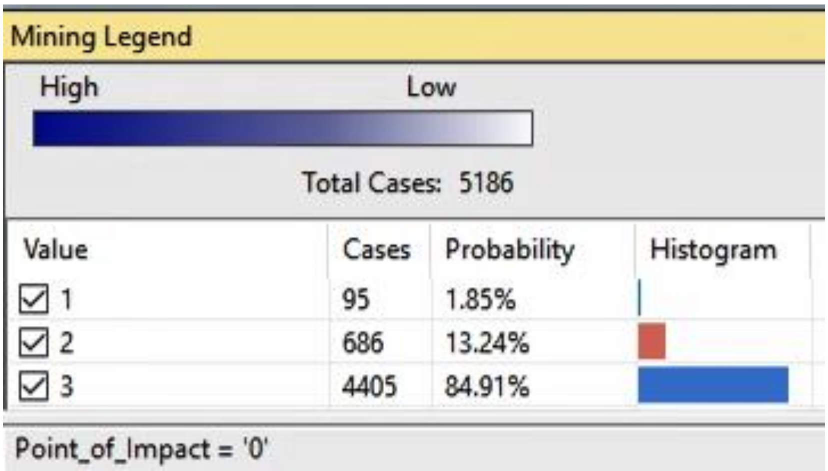
From the above decision tree output, we see that whether the point of impact is front, rear, left or right, it is the most important factor in predicting accident severity.

2nd Level



Depending upon the point of impact,

- If the point of impact is 3, the accident severity depends on location if it is urban or rural.
- If the point of impact is 2, the accident severity depends on the vehicle manoeuvre or not.
- If the point of impact is 1, the accident severity depends on the vehicle type.


Mining Legend

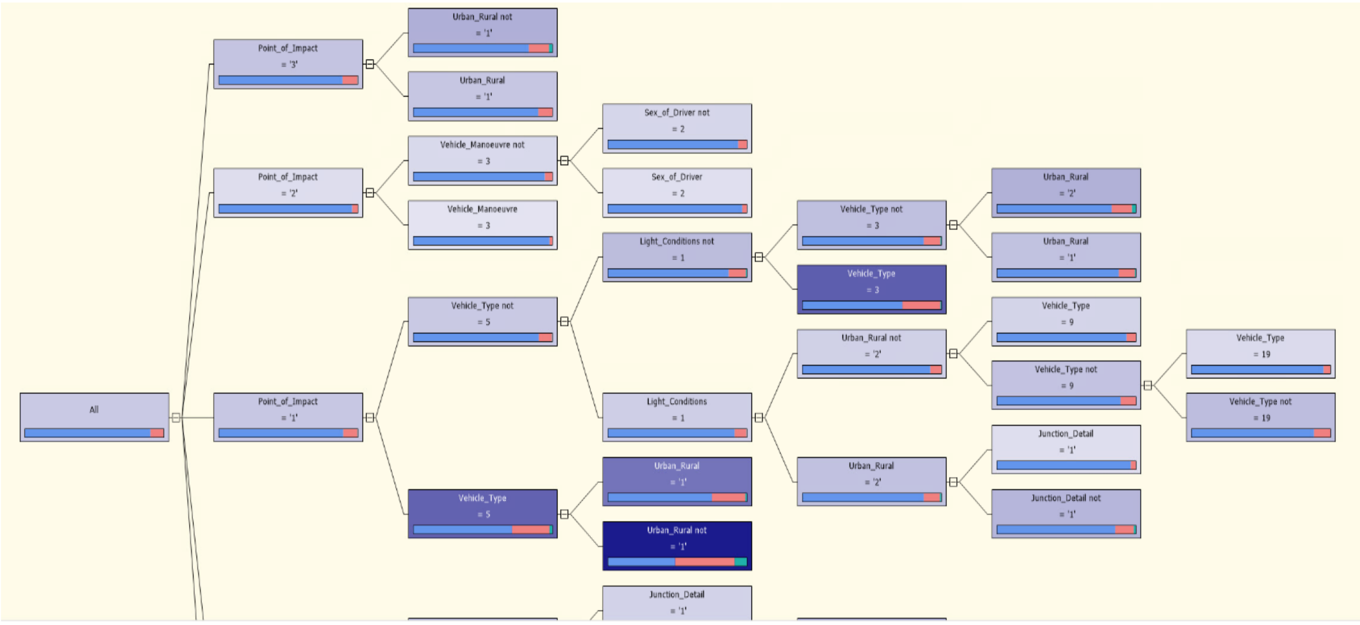
High Low

Total Cases: 5186

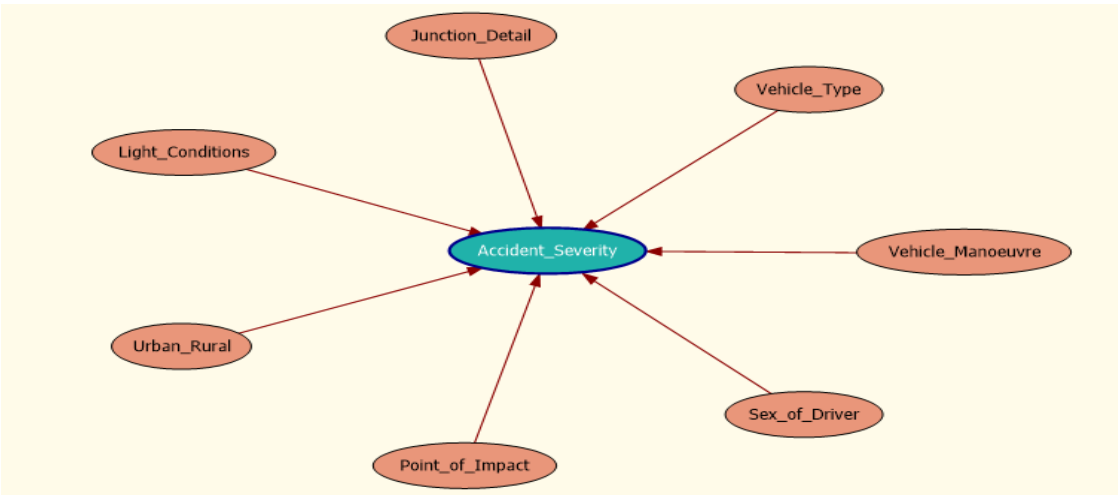
Value	Cases	Probability	Histogram
1	95	1.85%	
2	686	13.24%	
3	4405	84.91%	

Point_of_Impact = '0'

Complete Decision Tree

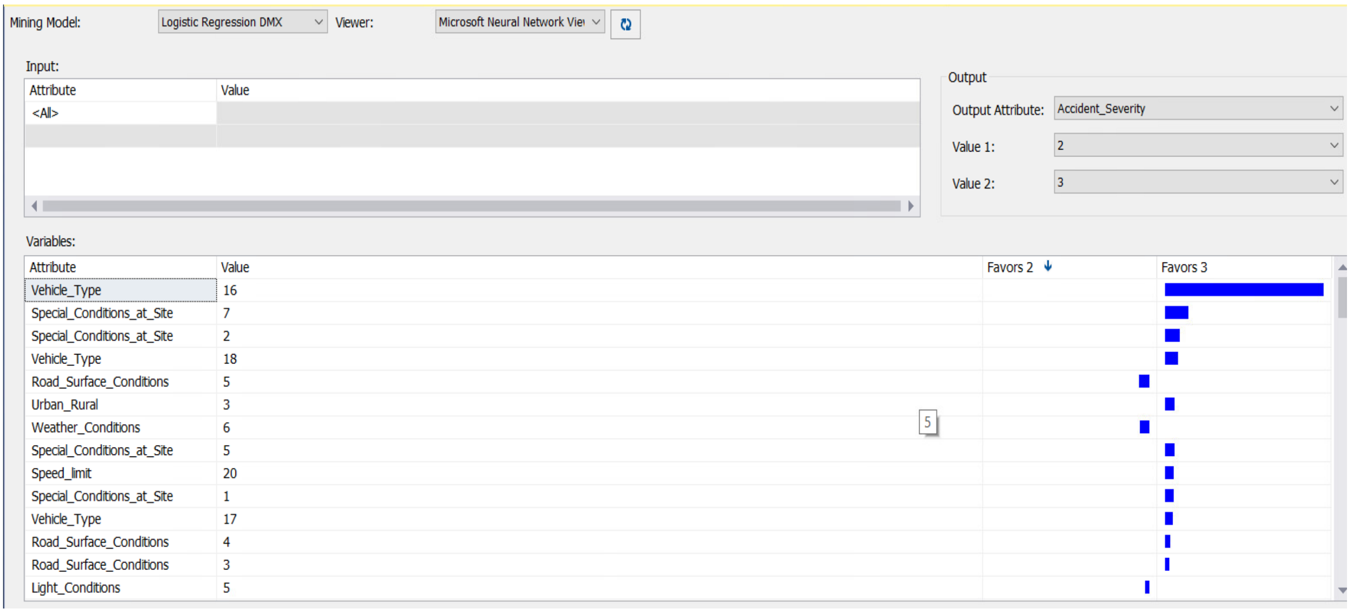


Dependency Network



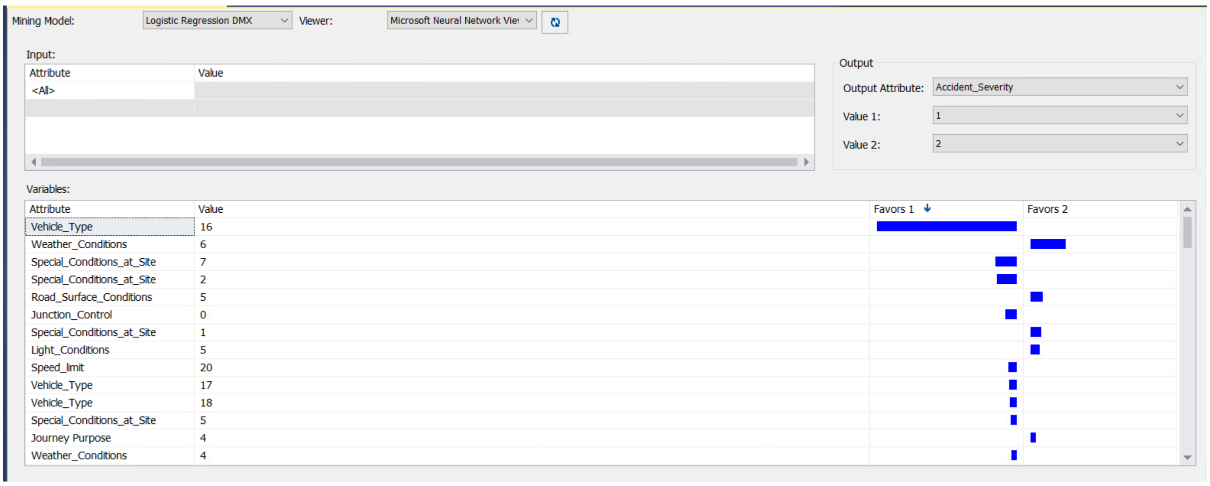
From the above screenshots of the complete decision tree and dependency network, we conclude that point of impact, location whether urban_rural ,light conditions, junction detail, vehicle type, vehicle manoeuvre, sex of the driver play good roles in predicting accident severity.

Model 2: Logistic Regression



From the above logistic regression output we can say that, when we are considering Accident Severity 2 and 3, Vehicle_Type value =16 favours Accident Severity= 3. Special_conditions_at_site=7 is the next important variable which favours Accident Severity 3 the most.

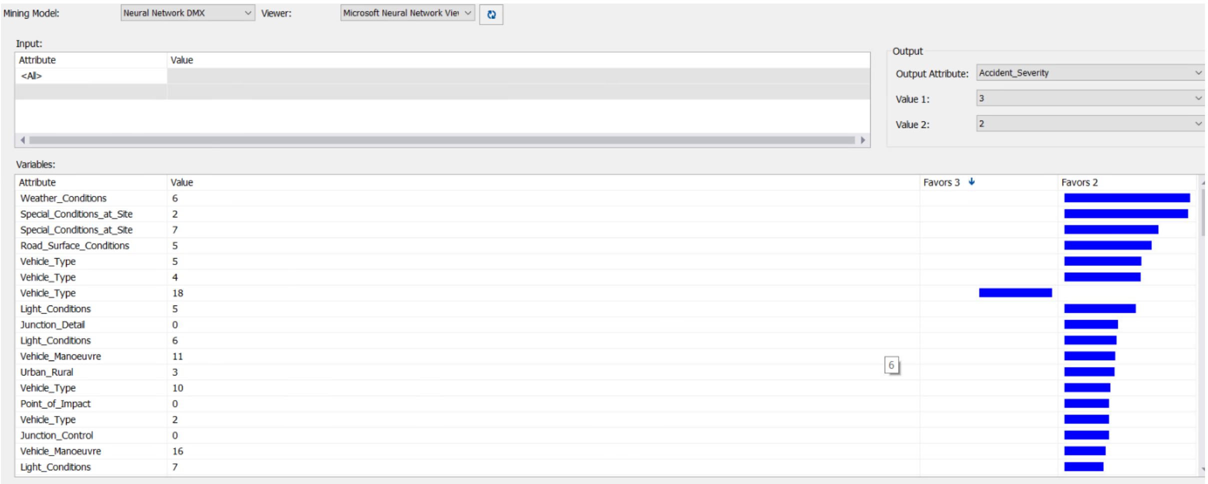
In similar manner when we are considering Accident Severity 1 and 2, vehicle_Type value 16 favors Accident Severity 1 and Weather_Conditions = 6 favours Accident Severity 2 the most.



Model3: Neural Network

The Microsoft Neural Network algorithm combines each possible state of the input attribute with each possible state of the predictable attribute and uses the training data to calculate probabilities.

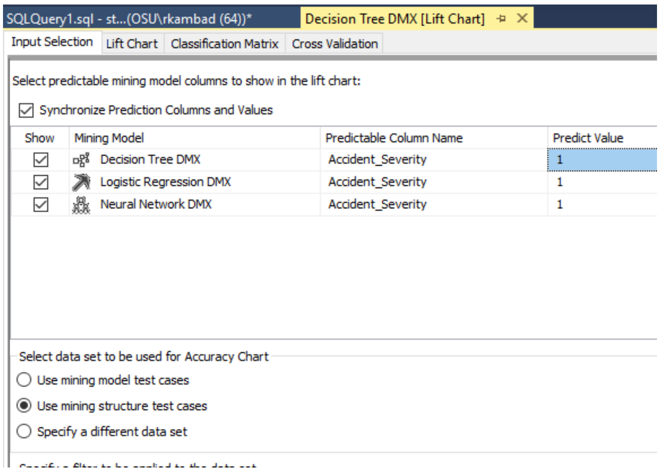
From the below neural network output we can say that weather condition = 6 favors the Accident_Severity= 2 the most. Also, 2nd most important value is special conditions at site = 2, it favors accident severity 2.

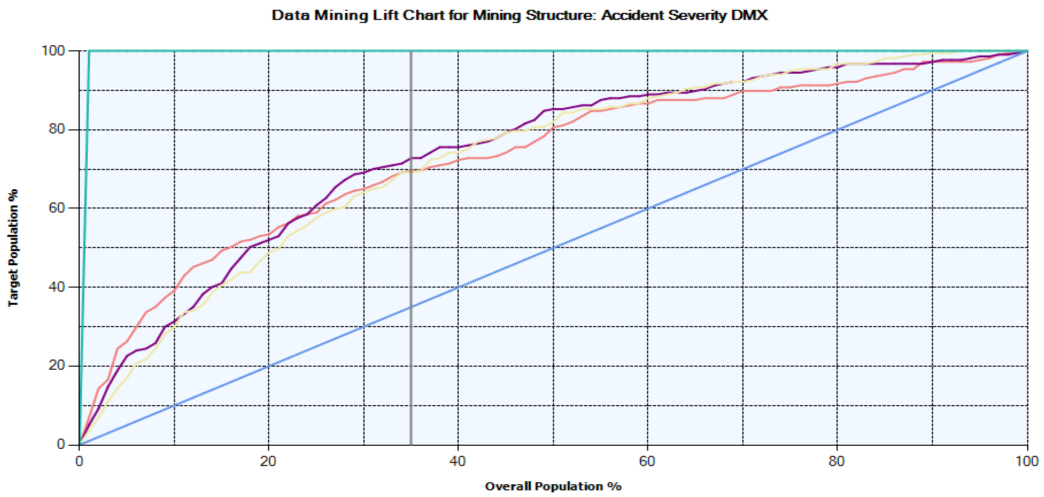


Comparison of the results

Lift Chart

Lift chart helps us to visualize the improvement we get when we use a data mining model when compared to the random guess model. Using the mining structure test cases, we constructed the lift chart for all the 3 mining models for accident severity = 1 (Fatal)





The y-axis is the accuracy measure for the corresponding population percentage we targeted which is 35%(vertical grey line in the above screenshot).

Mining Legend

Population percentage: 35.00%

Series, Model	Score	Target population	Predict probability
Decision Tree DMX	0.74	69.59%	0.40%
Logistic Regressio...	0.75	72.81%	2.41%
Neural Network D...	0.73	69.12%	1.17%
Random Guess M...		35.00%	
Ideal Model for: D...		100.00%	

From the above charts, we can say that if we target 35% of the population,

- The Random Guess Model will correctly identify 35% of all the accident severity =1(fatal) within the population
- The ideal line/model for decision tree will correctly identify 100% of all the accident severity =1(fatal) within the population. (slope=1)
- The Decision Tree model will correctly identify 69.6% of all the accident severity =1(fatal) within the population
- The Logistic Regression model will correctly identify 72.81% of all the accident severity =1(fatal) within the population
- The Neural Network model will correctly identify 69.12% of all the accident severity =1(fatal) within the population

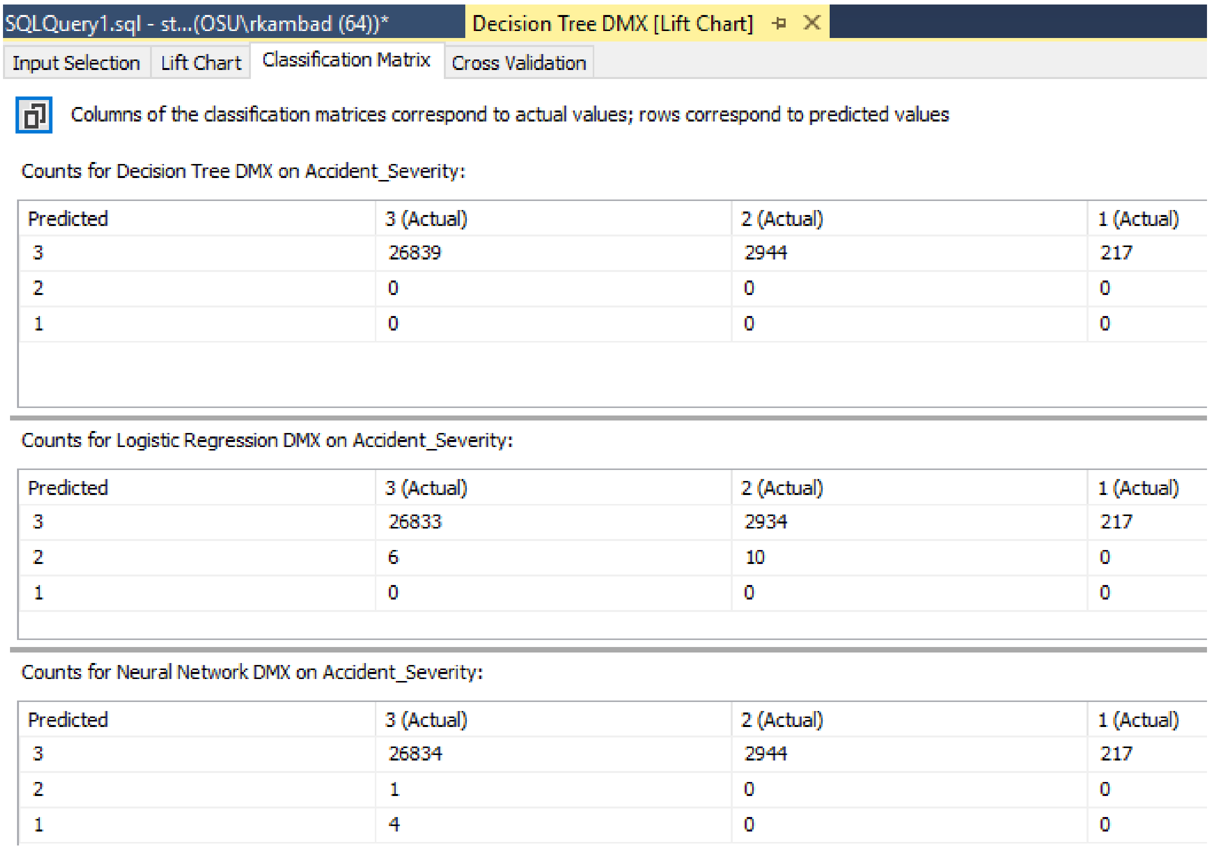
The lift score is highest for Logistic Regression model. (Although, there is not much difference between all the 3 models).

Interpreting Predict probability

- To identify the accidents from the Logistic Regression model which have accident severity = 1(fatal), we need to use a query to retrieve cases with a predict probability of at least 2.41%
- To identify the accidents from Decision Tree model which have accident severity = 1(fatal), we need to use a query to retrieve cases with a predict probability of at least 0.40%
- To identify the accidents from Neural Network model which have accident severity = 1(fatal), we need to use a query to retrieve cases with a predict probability of at least 1.71%

(Reference for lift score, chart, predict probability: Lecture 7 slide 39)

Classification Matrix


SQLQuery1.sql - st...(OSU\rkambad (64))* Decision Tree DMX [Lift Chart]

Input Selection | Lift Chart | Classification Matrix | Cross Validation

Columns of the classification matrices correspond to actual values; rows correspond to predicted values

Counts for Decision Tree DMX on Accident_Severity:

Predicted	3 (Actual)	2 (Actual)	1 (Actual)
3	26839	2944	217
2	0	0	0
1	0	0	0

Counts for Logistic Regression DMX on Accident_Severity:

Predicted	3 (Actual)	2 (Actual)	1 (Actual)
3	26833	2934	217
2	6	10	0
1	0	0	0

Counts for Neural Network DMX on Accident_Severity:

Predicted	3 (Actual)	2 (Actual)	1 (Actual)
3	26834	2944	217
2	1	0	0
1	4	0	0

Classification interpretations

Decision tree model results predicted that 26,839 accidents would have had accident severity level 3 correctly. 2,944 accidents who has severity level 3, predicted wrongly that it has severity level 2. And 217 accidents who has severity level 3, predicted wrongly that it has severity level 1.

Logistic regression model results predicted that 26,833 accidents would have had accident severity level 3 correctly. 2,934 accidents who has severity level 3, predicted wrongly that it has severity level 2. And 217 accidents who has severity level 3, predicted wrongly that it has severity level 1.

Neural network model results predicted that 26,834 accidents would have had accident severity level 3 correctly. 2,944 accidents who has severity level 3, predicted wrongly that it has severity level 2. And 217 accidents who has severity level 3, predicted wrongly that it has severity level 1.

Cross Validation

We have specified target state = 1 (accident severity = 1 or fatal) and a target threshold =0.2

Fold Count: 5 Max Cases: 0 Get Results

Target Attribute: Accident_Severity Target State: 1 Target Threshold: 0.2

Decision Tree DMX

Partition Index	Partition Size	Test	Measure	Value
1	14000	Classification	True Positive	0.000e+000
2	14000	Classification	True Positive	0.000e+000
3	14000	Classification	True Positive	0.000e+000
4	14000	Classification	True Positive	0.000e+000
5	14000	Classification	True Positive	0.000e+000
			Average	0.000e+000
			Standard Deviation	0.000e+000
1	14000	Classification	False Positive	0.000e+000
2	14000	Classification	False Positive	0.000e+000
3	14000	Classification	False Positive	0.000e+000
4	14000	Classification	False Positive	0.000e+000
5	14000	Classification	False Positive	0.000e+000
			Average	0.000e+000
			Standard Deviation	0.000e+000
1	14000	Classification	True Negative	13904
2	14000	Classification	True Negative	13904
3	14000	Classification	True Negative	13904

4	14000	Classification	True Positive	0.000e+000
5	14000	Classification	True Positive	0.000e+000
			Average	0.000e+000
			Standard Deviation	0.000e+000
1	14000	Classification	False Positive	0.000e+000
2	14000	Classification	False Positive	0.000e+000
3	14000	Classification	False Positive	0.000e+000
4	14000	Classification	False Positive	0.000e+000
5	14000	Classification	False Positive	0.000e+000
			Average	0.000e+000
			Standard Deviation	0.000e+000
1	14000	Classification	True Negative	13904
2	14000	Classification	True Negative	13904
3	14000	Classification	True Negative	13904
4	14000	Classification	True Negative	13904
5	14000	Classification	True Negative	13904
			Average	13904
			Standard Deviation	0.000e+000
1	14000	Classification	False Negative	96
2	14000	Classification	False Negative	96

3	14000	Classification	False Negative	96
4	14000	Classification	False Negative	96
5	14000	Classification	False Negative	96
			Average Standard Deviation	96 0.000e+000
1	14000	Likelihood	Log Score	-0.3475
2	14000	Likelihood	Log Score	-0.3488
3	14000	Likelihood	Log Score	-0.3473
4	14000	Likelihood	Log Score	-0.3489
5	14000	Likelihood	Log Score	-0.35
			Average Standard Deviation	-0.3485 0.001
1	14000	Likelihood	Lift	0.0136
2	14000	Likelihood	Lift	0.0123
3	14000	Likelihood	Lift	0.0138
4	14000	Likelihood	Lift	0.0122
5	14000	Likelihood	Lift	0.0111
			Average Standard Deviation	0.0126 0.001
1	14000	Likelihood	Root Mean Square Error	0.1167
2	14000	Likelihood	Root Mean Square Error	0.1173
3	14000	Likelihood	Root Mean Square Error	0.1161
4	14000	Likelihood	Root Mean Square Error	0.1153
5	14000	Likelihood	Root Mean Square Error	0.1161
			Average Standard Deviation	0.1163 0.0007

Logistic Regression DMX

Partition Index	Partition Size	Test	Measure	Value
1	14000	Classification	True Positive	2
2	14000	Classification	True Positive	2
3	14000	Classification	True Positive	3
4	14000	Classification	True Positive	5
5	14000	Classification	True Positive	2
			Average Standard Deviation	2.8 1.1662
1	14000	Classification	False Positive	71
2	14000	Classification	False Positive	96
3	14000	Classification	False Positive	83
4	14000	Classification	False Positive	68
5	14000	Classification	False Positive	63
			Average Standard Deviation	76.2 11.8895
1	14000	Classification	True Negative	13833
2	14000	Classification	True Negative	13808
3	14000	Classification	True Negative	13821
4	14000	Classification	True Negative	13836
5	14000	Classification	True Negative	13841
			Average Standard Deviation	13827.8 11.8895
1	14000	Classification	False Negative	94
2	14000	Classification	False Negative	94
3	14000	Classification	False Negative	93
4	14000	Classification	False Negative	91
5	14000	Classification	False Negative	94

			Average	93.2
			Standard Deviation	1.1662
1	14000	Likelihood	Log Score	-0.3843
2	14000	Likelihood	Log Score	-0.3775
3	14000	Likelihood	Log Score	-0.3831
4	14000	Likelihood	Log Score	-0.3811
5	14000	Likelihood	Log Score	-0.381
			Average	-0.3814
			Standard Deviation	0.0023
1	14000	Likelihood	Lift	-0.0231
2	14000	Likelihood	Lift	-0.0164
3	14000	Likelihood	Lift	-0.022
4	14000	Likelihood	Lift	-0.02
5	14000	Likelihood	Lift	-0.0199
			Average	-0.0203
			Standard Deviation	0.0023
1	14000	Likelihood	Root Mean Square Error	0.1374
2	14000	Likelihood	Root Mean Square Error	0.1414
3	14000	Likelihood	Root Mean Square Error	0.1358
4	14000	Likelihood	Root Mean Square Error	0.1325
5	14000	Likelihood	Root Mean Square Error	0.1299
			Average	0.1354
			Standard Deviation	0.004

Neural Network DMX

Partition Index	**Partition Size**	**Test**	**Measure**	**Value**
1	14000	Classification	True Positive	3
2	14000	Classification	True Positive	1
3	14000	Classification	True Positive	0.000e+000
4	14000	Classification	True Positive	0.000e+000
5	14000	Classification	True Positive	2
			Average	1.2
			Standard Deviation	1.1662
1	14000	Classification	False Positive	21
2	14000	Classification	False Positive	32
3	14000	Classification	False Positive	26
4	14000	Classification	False Positive	21
5	14000	Classification	False Positive	26
			Average	25.2
			Standard Deviation	4.0694
1	14000	Classification	True Negative	13883
2	14000	Classification	True Negative	13872
3	14000	Classification	True Negative	13878
4	14000	Classification	True Negative	13883
5	14000	Classification	True Negative	13878
			Average	13878.8
			Standard Deviation	4.0694

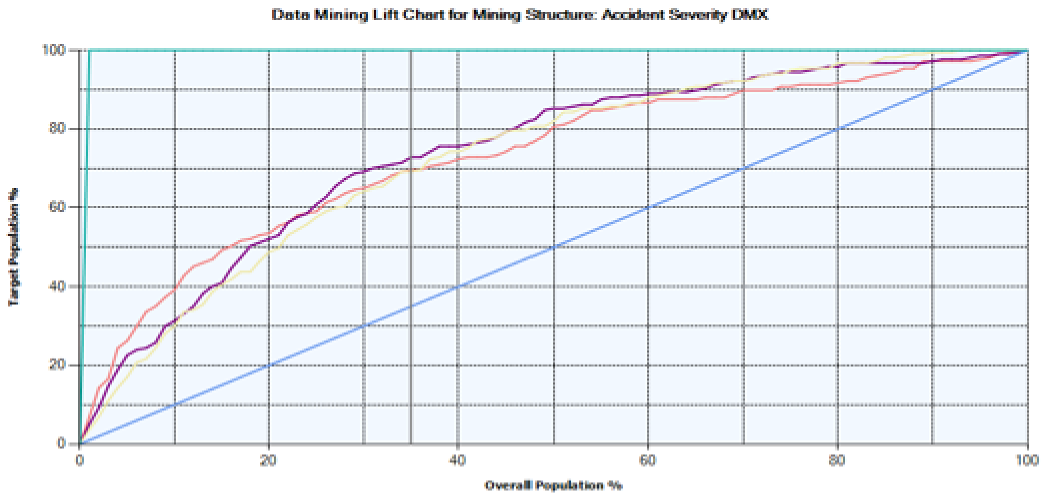
1	14000	Classification	False Negative	93
2	14000	Classification	False Negative	95
3	14000	Classification	False Negative	96
4	14000	Classification	False Negative	96
5	14000	Classification	False Negative	94
			Average	94.8
			Standard Deviation	1.1662
1	14000	Likelihood	Log Score	-0.3564
2	14000	Likelihood	Log Score	-0.3538
3	14000	Likelihood	Log Score	-0.3557
4	14000	Likelihood	Log Score	-0.353
5	14000	Likelihood	Log Score	-0.3567
			Average	-0.3551
			Standard Deviation	0.0015
1	14000	Likelihood	Lift	0.0047
2	14000	Likelihood	Lift	0.0073
3	14000	Likelihood	Lift	0.0054
4	14000	Likelihood	Lift	0.0081
5	14000	Likelihood	Lift	0.0044
			Average	0.006
			Standard Deviation	0.0015
1	14000	Likelihood	Root Mean Square Error	0.136
2	14000	Likelihood	Root Mean Square Error	0.1404
3	14000	Likelihood	Root Mean Square Error	0.1345
4	14000	Likelihood	Root Mean Square Error	0.1321
5	14000	Likelihood	Root Mean Square Error	0.1315
			Average	0.1349
			Standard Deviation	0.0032

Sl.No.	Model Name	Lift Score	Log Score	RMSE
1	Decision Tree Model	0.0126	-0.348	0.116
2	Logistic Regression Model	-0.0203	-0.381	0.135
3	Neural Networks Model	0.006	-0.3551	0.135

Summary of the results

Based on our above analysis of all the models, we conclude that decision tree is the best model. Here are the reasons to conclude so.

Lift Chart for accident severity =1



Mining Legend			
Population percentage: 35.00%			
Series, Model	Score	Target population	Predict probability
Decision Tree DMX	0.74	69.59%	0.40%
Logistic Regressio...	0.75	72.81%	2.41%
Neural Network D...	0.73	69.12%	1.17%
Random Guess M...		35.00%	
Ideal Model for: D...		100.00%	

Based on the above screenshot after lift chart generation, we see that though the lift score for the logistic regression is good, there is not much great difference in the lift score for decision tree model.

Classification matrix

Counts for Decision Tree DMX on Accident_Severity:

Predicted	3 (Actual)	2 (Actual)	1 (Actual)
3	26839	2944	217
2	0	0	0
1	0	0	0

Decision tree model results predicted that 26,839 accidents would have had accident severity level 3 correctly. 2,944 accidents who has severity level 3, predicted wrongly that it has severity level 2. And 217 accidents who has severity level 3, predicted wrongly that it has severity level 1.

The prediction of decision model is slightly better than the other two models.

Cross validation

We have specified target state = 1 (accident severity = 1 or fatal) and a target threshold =0.2

Sl.No.	Model Name	Lift Score	Log Score	RMSE
1	Decision Tree Model	0.0126	-0.348	0.116

- The log score for the decision tree model is also closest to 0. The average prediction probability for the same model is e^(-.348) = 0.7
- The lift score indicates that there is a 1.26% improvement in the probability of the target outcome when decision tree model is used.

Since the log and lift score for decision tree is slightly good compared to the other two models, decision tree model seems to be a good model built

Conclusion

1. The decision tree model is not only more accurate than other models, but the factors explained by it are also makes sense in real world.

2. Based on whether the point of impact is front of the vehicle, rear, sides etc., we can say how severe an accident is. Also, there will be ideally a greater number of accidents in urban_rural = 1 or 2.

3. The kind of vehicle we travel whether motorcycle 125 cc and under, light conditions and sex of the driver plays important roles in deciding accident severity.

4. Compared to the vehicle which is waiting to go (held up), the vehicles which are turning left or right or changing the lanes, they are prone to accidents.