PROJECT ASSIGNMENT - (PART II)

This part of the project includes creating a data cube, performing MDX queries and Power BI visualization.

Assignment 1:

Build a data cube from the data of the tables in your database, defining the appropriate hierarchies. Create the needed measures based on the queries you need to answer.

Before creating the cubes, We tried to understand the task so we could filtered out which data required to create the cube by using the database table from part 1 of the project.

Configuring the Data Source:

The first step is to configure the data source. The data source for the cube is the database created in the previous assignments on the database server, *lds.di.unipi.it*

Configuring Data Source View:

As the name shows, this is the view of the data source. We created the following new attributes in our data source view to be used in the cube:

• *Primary_Cause_ID:* which is calculated based on Cause_Description from DimCause.

• *Secondary_Cause_ID:* which is calculated based on Cause_Description from DimCause.

we have divided cause description into portions of Human Nature and Environmental or supporting causes. Filtered out the ids and wrote expressions. Based on cause description, we added Human Nature for Primary Cause IDs and Environmental or supporting causes as Secondary Cause IDs in Fact Crash directly because fast retrieval and add measures on it.

• *Combined_Date:* which is calculated by concatenating Year, Quarter, Month and Day from DimDate.

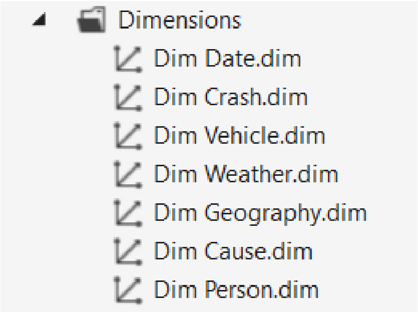
• *Street_Name, Street_Type, Zip_Code* which is calculated based on City from DimGeography.

By using OpenStreetMap API (Nominatim) and open Datasets we fetched Street Name, Street Type and Zip Code as we have City, State, Latitude and Longitude.

• *Intersection:* It is combination of Street name and street type. we performed concatenation on it. Street_Name + ' & ' + Street_Type.

We create all the above new attributes to be used in hierarchies and to maintain relevant attribute relationships and functional dependencies within the dimensions to be created.

Creating Dimensions and Hierarchies:

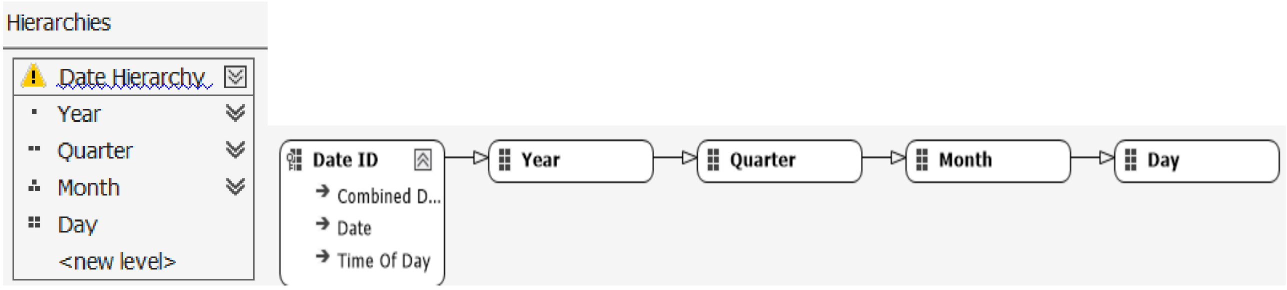


Dimensions
- Dim Date.dim
- Dim Crash.dim
- Dim Vehicle.dim
- Dim Weather.dim
- Dim Geography.dim
- Dim Cause.dim
- Dim Person.dim

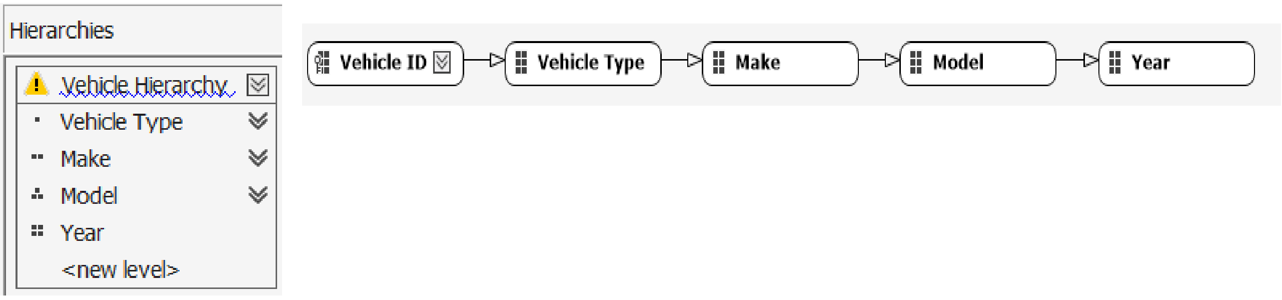
The Following dimensions were created. To create these dimensions because we have used all of them in different assignments.

Based on the assignments, we have created required hierarchies.

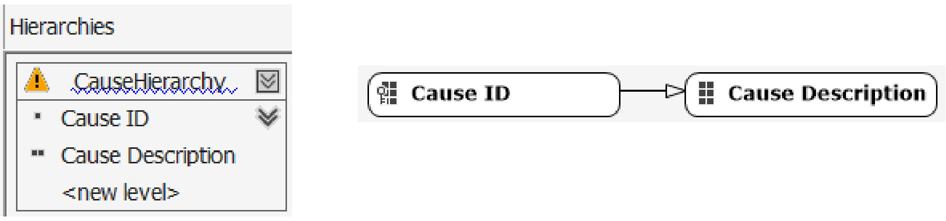
1. Date Hierarchy



2. Vehicle Hierarchy



3. Cause Hierarchy



Similarly, we created other required hierarchies as well.

Geography Hierarchy: State → City → Zip Code

Weather Hierarchy: Weather Condition → Visibility Condition

Crash Hierarchy: Crash Type → Traffic Control Device → Lighting Condition

Person Hierarchy: Age Group → Role → Person_ID

The central table (Fact Crash) connects to each dimension table using foreign key relationships (Date_ID, Vehicle_ID, etc.). Each dimension contains hierarchies that organize the data into levels, enabling aggregation and filtering for MDX queries.

Application in Assignments

Assignment	Dimension	Hierarchy	Level(s)
Assignment 2	Dim Date, Dim Geography	Date Hierarchy, Location	Year, Month, State, City
Assignment 3	Dim Date, Fact Crash	Date Hierarchy	Year
Assignment 4	Dim Date, Dim Geography	Date Hierarchy, Location	Year, State, City
Assignment 5	Dim Date, Fact Crash	Date Hierarchy, Location	Quarter, Year
Assignment 6	Dim Vehicle, Dim Person	Vehicle Hierarchy, Person	Vehicle Type, Year, Person_ID
Assignment 7	Dim Weather, Fact Crash	Weather Hierarchy	Weather Condition, Damage Amount

Creating Measures/Calculated Measures:

After the creating of relevant dimensions, we are ready to create the cube and its relevant measures.

The structure of the cube is given in the figure.

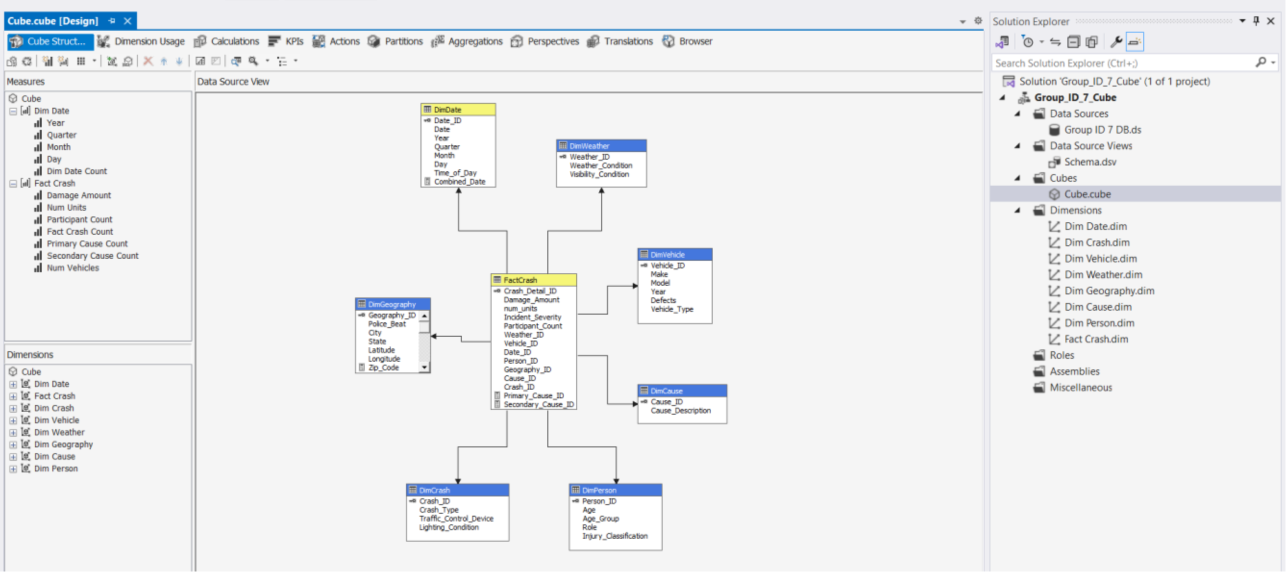
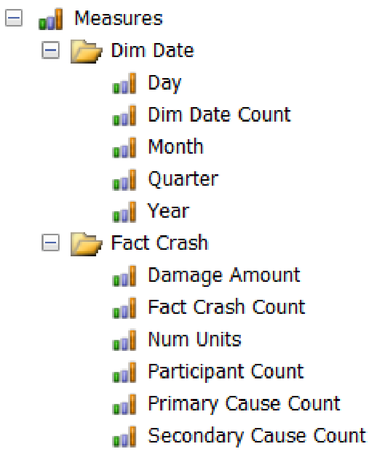


Figure 15: Data Cube Structure

The measures in the cube are:

- [Year, Quarter, Month, Day]: This measure is the in the *Dim Date*.
- [Dim Date Count]: This measure counts total dates.
- [Damage Amount, Num Units, Participant Count]: This measures the sum of the respective columns from Fact Crash.
- [Fact Crash Count]: This count the Crash from Fact Crash using Crash_ID.
- [Primary_Cause_Count & Secondary_Cause_COunt]: This counts the Crashes from Fact Crash.

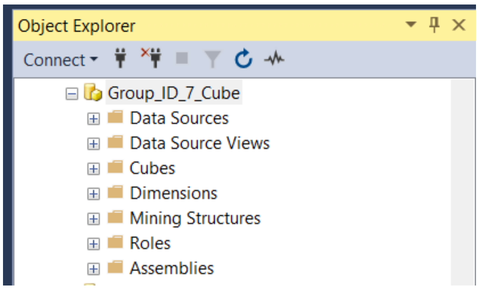


- Measures
  - Dim Date
    - Day
    - Dim Date Count
    - Month
    - Quarter
    - Year
  - Fact Crash
    - Damage Amount
    - Fact Crash Count
    - Num Units
    - Participant Count
    - Primary Cause Count
    - Secondary Cause Count

Figure 16: Measures

Deploying the Cube:

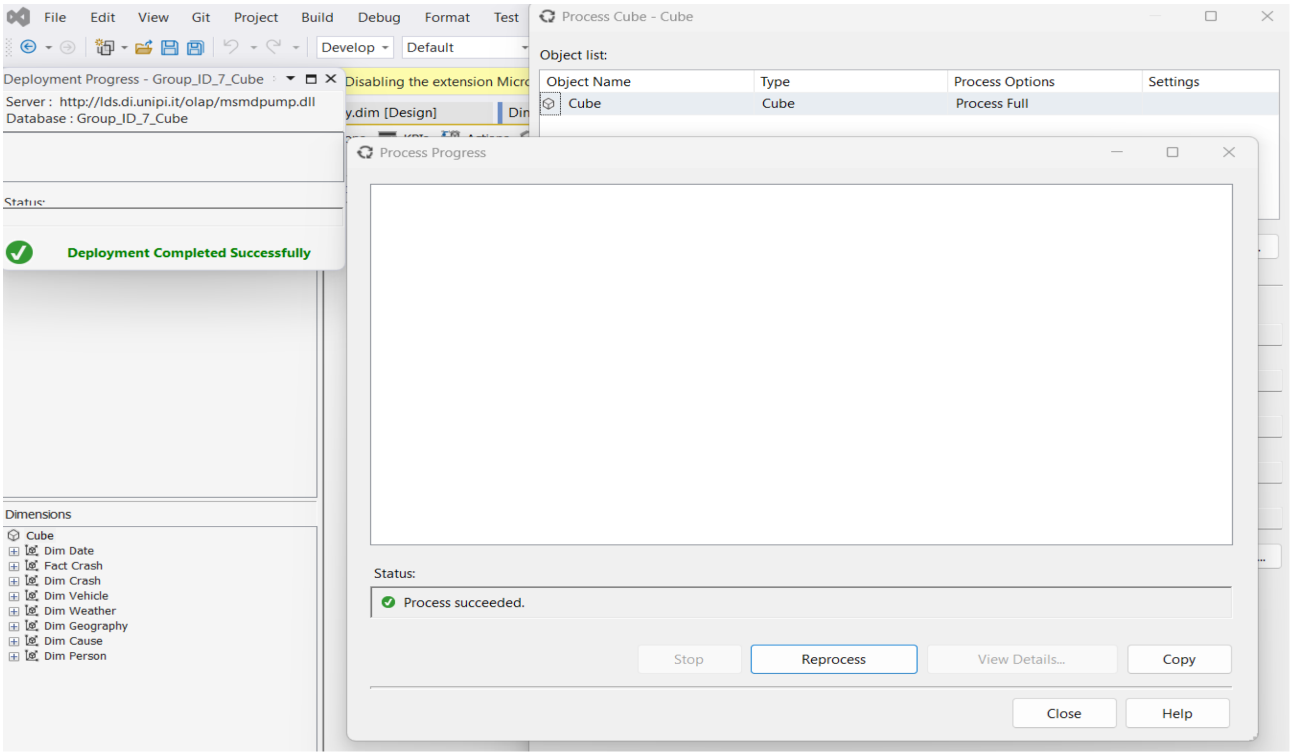
In order to deploy the created cube, we use the following connection string to process and deploy it on the Analysis Server in the database, *Group_ID_7_Cube that could be seen right side image of SASS SSMS* using server name *http://lds.di.unipi.it/olap/msmdpump.dll*.



- Group_ID_7_Cube
  - Data Sources
  - Data Source Views
  - Cubes
  - Dimensions
  - Mining Structures
  - Roles
  - Assemblies

Figure 17: SASS Cube View

Cube Deployed Successfully



Deployment Progress - Group_ID_7_Cube

Server : http://lds.di.unipi.it/olap/msmdpump.dll
Database : Group_ID_7_Cube

Status:

✓ Deployment Completed Successfully

Dimensions

- Cube
  - Dim Date
  - Fact Crash
  - Dim Crash
  - Dim Vehicle
  - Dim Weather
  - Dim Geography
  - Dim Cause
  - Dim Person

Process Cube - Cube

Object list:

Object Name	Type	Process Options	Settings
Cube	Cube	Process Full	

Process Progress

Status:

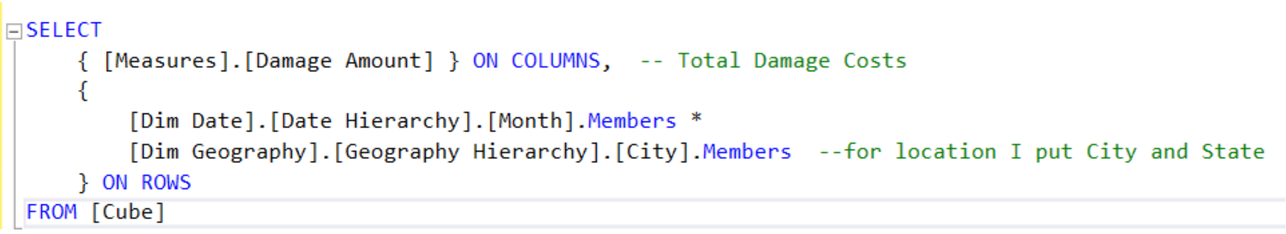
✓ Process succeeded.

Figure 18: Cubes Deployed Successfully

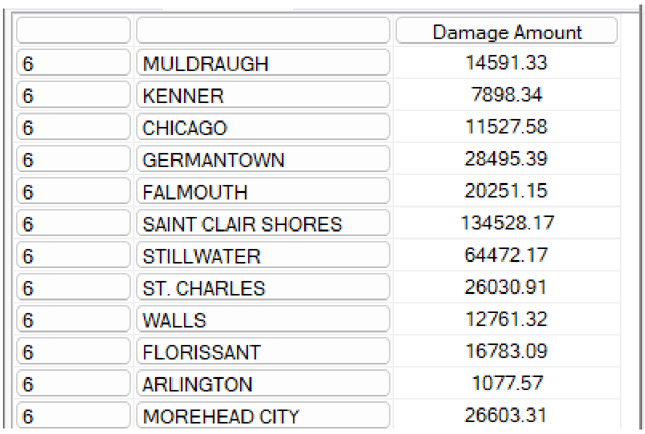
The cube is ready to be queried on the Analysis Server.

Assignment 2*

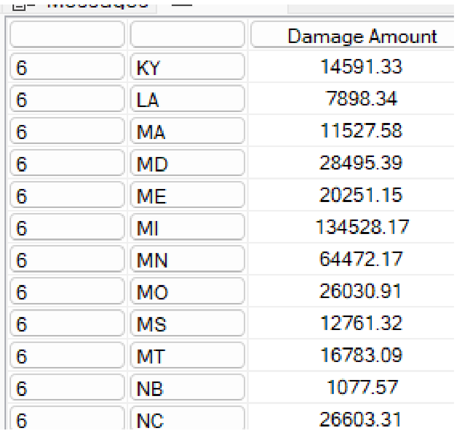
For each month, show the total damage costs for each location and the grand total with respect to the location.



```
SELECT
    { [Measures].[Damage Amount] } ON COLUMNS,  -- Total Damage Costs
    {
        [Dim Date].[Date Hierarchy].[Month].Members *
        [Dim Geography].[Geography Hierarchy].[City].Members  --for location I put City and State
    } ON ROWS
FROM [Cube]
```



		Damage Amount
6	MULDRAUGH	14591.33
6	KENNER	7898.34
6	CHICAGO	11527.58
6	GERMANTOWN	28495.39
6	FALMOUTH	20251.15
6	SAINT CLAIR SHORES	134528.17
6	STILLWATER	64472.17
6	ST. CHARLES	26030.91
6	WALLS	12761.32
6	FLORISSANT	16783.09
6	ARLINGTON	1077.57
6	MOREHEAD CITY	26603.31



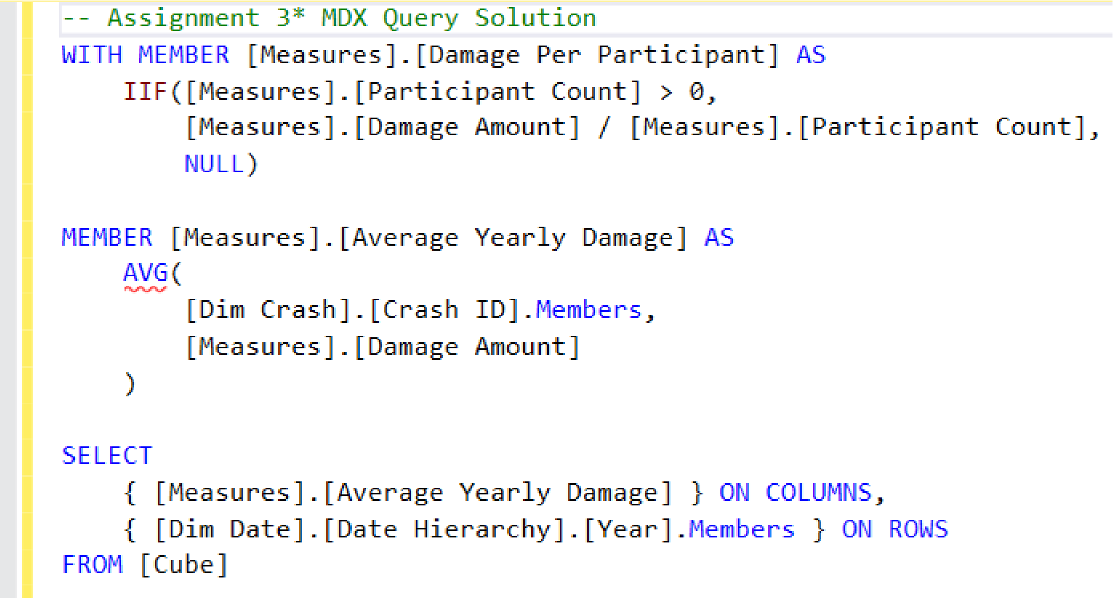
		Damage Amount
6	KY	14591.33
6	LA	7898.34
6	MA	11527.58
6	MD	28495.39
6	ME	20251.15
6	MI	134528.17
6	MN	64472.17
6	MO	26030.91
6	MS	12761.32
6	MT	16783.09
6	NB	1077.57
6	NC	26603.31

Figure 19: Output 1- Versus City

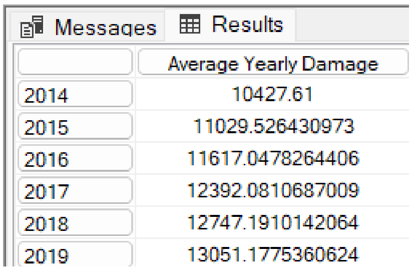
Figure 20: Output-2 Versus State

Assignment 3*

Compute the average yearly damage costs as follows: for each crash, calculate the total damage to the user divided by the number of distinct people involved in the crash. Then, compute the average of these values across all crashes in a year.



```
-- Assignment 3* MDX Query Solution
WITH MEMBER [Measures].[Damage Per Participant] AS
    IIF([Measures].[Participant Count] > 0,
        [Measures].[Damage Amount] / [Measures].[Participant Count],
        NULL)

MEMBER [Measures].[Average Yearly Damage] AS
    AVG(
        [Dim Crash].[Crash ID].Members,
        [Measures].[Damage Amount]
    )

SELECT
    { [Measures].[Average Yearly Damage] } ON COLUMNS,
    { [Dim Date].[Date Hierarchy].[Year].Members } ON ROWS
FROM [Cube]
```

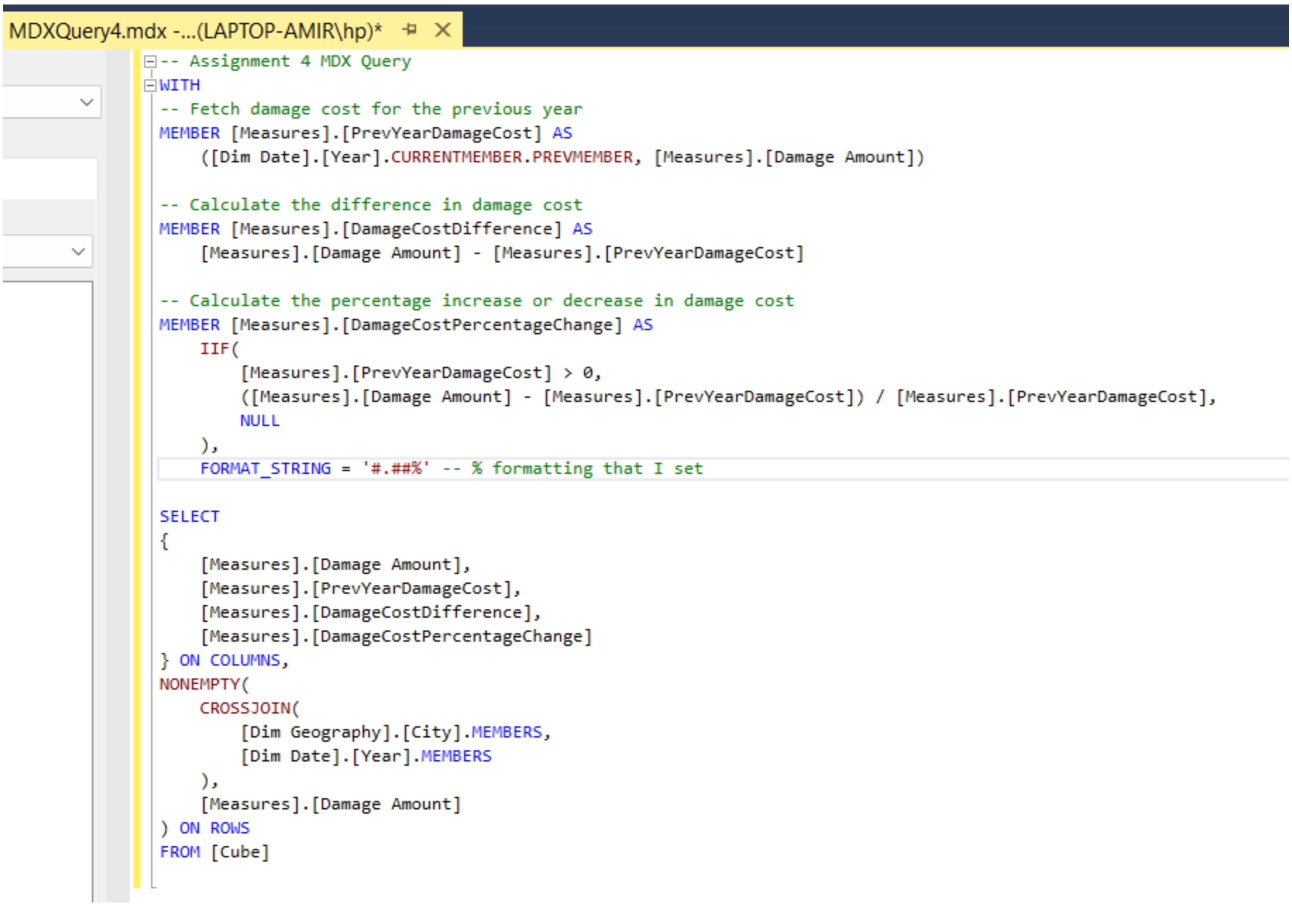


	Average Yearly Damage
2014	10427.61
2015	11029.526430973
2016	11617.0478264406
2017	12392.0810687009
2018	12747.1910142064
2019	13051.1775360624

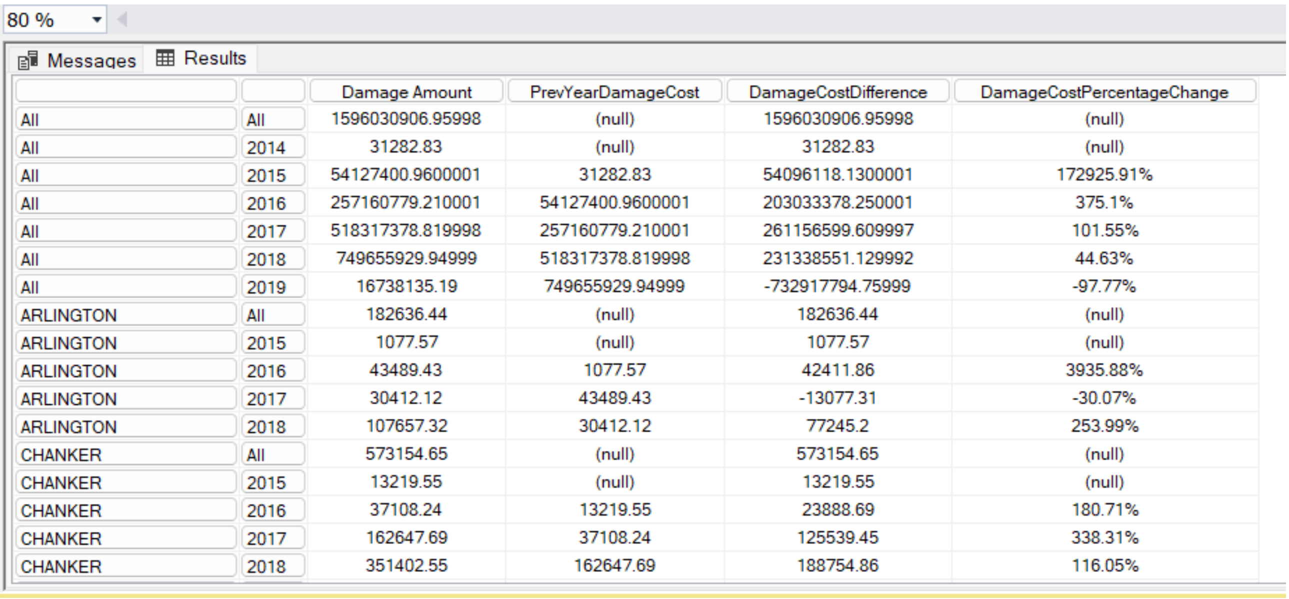
Figure 21: Average Yearly Damage

Assignment 4*

For each location, show the damage costs increase or decrease, in percentage, with respect to the previous year.



```
-- Assignment 4 MDX Query
WITH
-- Fetch damage cost for the previous year
MEMBER [Measures].[PrevYearDamageCost] AS
    ([Dim Date].[Year].CURRENTMEMBER.PREVMEMBER, [Measures].[Damage Amount])

-- Calculate the difference in damage cost
MEMBER [Measures].[DamageCostDifference] AS
    [Measures].[Damage Amount] - [Measures].[PrevYearDamageCost]

-- Calculate the percentage increase or decrease in damage cost
MEMBER [Measures].[DamageCostPercentageChange] AS
    IIF(
        [Measures].[PrevYearDamageCost] > 0,
        ([Measures].[Damage Amount] - [Measures].[PrevYearDamageCost]) / [Measures].[PrevYearDamageCost],
        NULL
    ),
    FORMAT_STRING = '#.##%' -- % formatting that I set

SELECT
{
    [Measures].[Damage Amount],
    [Measures].[PrevYearDamageCost],
    [Measures].[DamageCostDifference],
    [Measures].[DamageCostPercentageChange]
} ON COLUMNS,
NONEMPTY(
    CROSSJOIN(
        [Dim Geography].[City].MEMBERS,
        [Dim Date].[Year].MEMBERS
    ),
    [Measures].[Damage Amount]
) ON ROWS
FROM [Cube]
```



		Damage Amount	PrevYearDamageCost	DamageCostDifference	DamageCostPercentageChange
All	All	1596030906.95998	(null)	1596030906.95998	(null)
All	2014	31282.83	(null)	31282.83	(null)
All	2015	54127400.9600001	31282.83	54096118.1300001	172925.91%
All	2016	257160779.210001	54127400.9600001	203033378.250001	375.1%
All	2017	518317378.819998	257160779.210001	261156599.609997	101.55%
All	2018	749655929.94999	518317378.819998	231338551.129992	44.63%
All	2019	16738135.19	749655929.94999	-732917794.75999	-97.77%
ARLINGTON	All	182636.44	(null)	182636.44	(null)
ARLINGTON	2015	1077.57	(null)	1077.57	(null)
ARLINGTON	2016	43489.43	1077.57	42411.86	3935.88%
ARLINGTON	2017	30412.12	43489.43	-13077.31	-30.07%
ARLINGTON	2018	107657.32	30412.12	77245.2	253.99%
CHANKER	All	573154.65	(null)	573154.65	(null)
CHANKER	2015	13219.55	(null)	13219.55	(null)
CHANKER	2016	37108.24	13219.55	23888.69	180.71%
CHANKER	2017	162647.69	37108.24	125539.45	338.31%
CHANKER	2018	351402.55	162647.69	188754.86	116.05%

Manual Calculation:

1. "All" Level, 2015

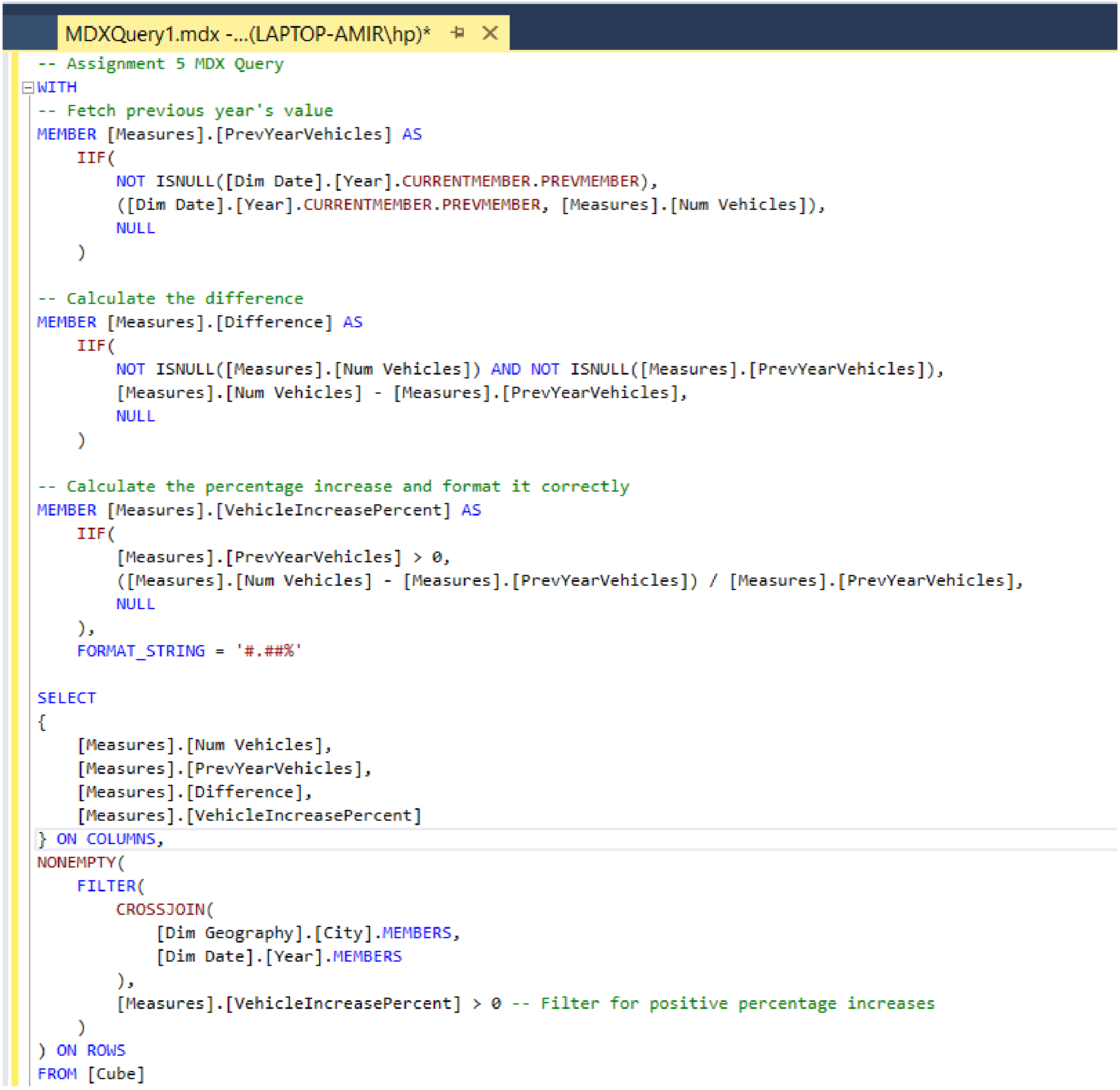
- **Damage Amount**: 54,127,400.96
- **PrevYearDamageCost**: 31,282.83
- **DamageCostDifference**: 54,127,400.96−31,282.83 = 54,096,118.13
- **DamageCostPercentageChange**: 54,096,118.13 / 31,282.83×100 ≈ 172925.91%

2. CHANKER, 2018

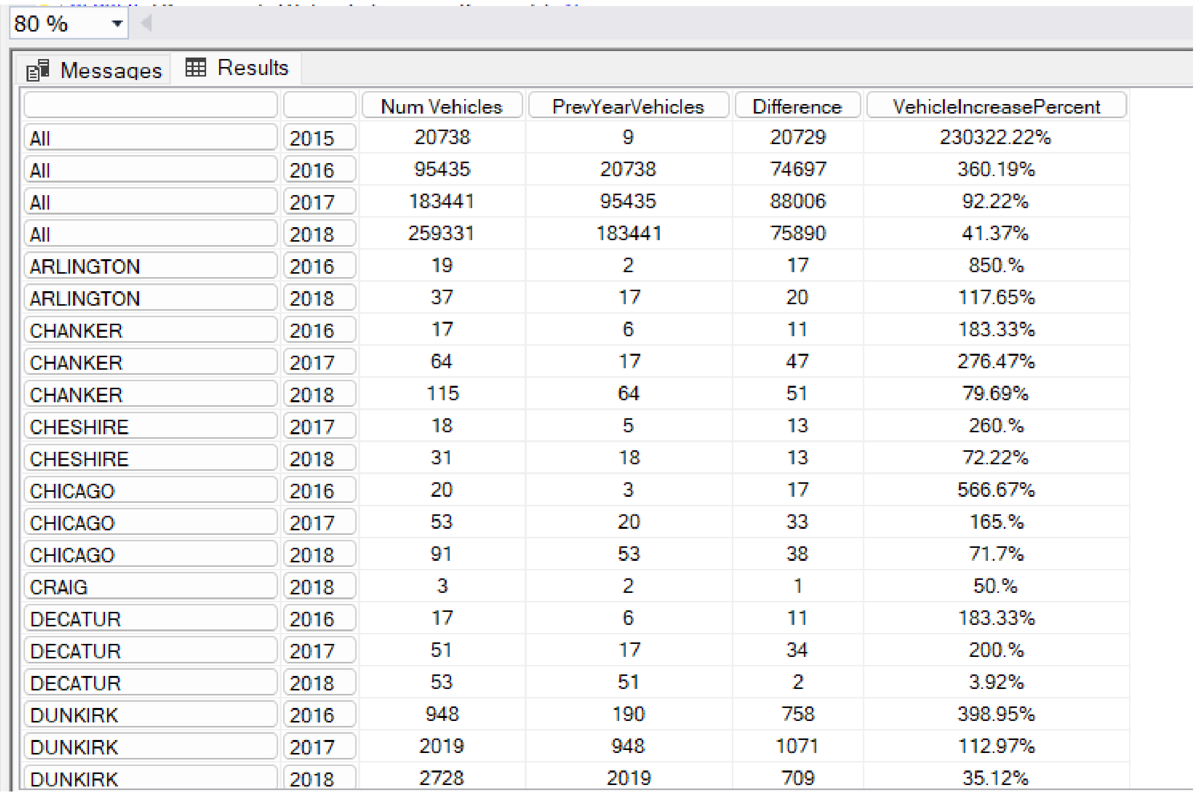
- **Damage Amount**: 351,402.55
- **PrevYearDamageCost**: 162,647.69
- **DamageCostDifference**: 351,402.55−162,647.69 =188,754.86
- **DamageCostPercentageChange**: 188,754.86 / 162,647.69×100≈116.05%

Assignment 5* *(updated version)*

For each quarter, show all the locations where the number of vehicles involved exceeds the average number of vehicles involved in the corresponding quarter of the previous year. Also, report the increase as percentage.



```
-- Assignment 5 MDX Query
WITH
-- Fetch previous year's value
MEMBER [Measures].[PrevYearVehicles] AS
    IIF(
        NOT ISNULL([Dim Date].[Year].CURRENTMEMBER.PREVMEMBER),
        ([Dim Date].[Year].CURRENTMEMBER.PREVMEMBER, [Measures].[Num Vehicles]),
        NULL
    )

-- Calculate the difference
MEMBER [Measures].[Difference] AS
    IIF(
        NOT ISNULL([Measures].[Num Vehicles]) AND NOT ISNULL([Measures].[PrevYearVehicles]),
        [Measures].[Num Vehicles] - [Measures].[PrevYearVehicles],
        NULL
    )

-- Calculate the percentage increase and format it correctly
MEMBER [Measures].[VehicleIncreasePercent] AS
    IIF(
        [Measures].[PrevYearVehicles] > 0,
        ([Measures].[Num Vehicles] - [Measures].[PrevYearVehicles]) / [Measures].[PrevYearVehicles],
        NULL
    ),
    FORMAT_STRING = '#.##%'

SELECT
{
    [Measures].[Num Vehicles],
    [Measures].[PrevYearVehicles],
    [Measures].[Difference],
    [Measures].[VehicleIncreasePercent]
} ON COLUMNS,
NONEMPTY(
    FILTER(
        CROSSJOIN(
            [Dim Geography].[City].MEMBERS,
            [Dim Date].[Year].MEMBERS
        ),
        [Measures].[VehicleIncreasePercent] > 0 -- Filter for positive percentage increases
    )
) ON ROWS
FROM [Cube]
```



		Num Vehicles	PrevYearVehicles	Difference	VehicleIncreasePercent
All	2015	20738	9	20729	230322.22%
All	2016	95435	20738	74697	360.19%
All	2017	183441	95435	88006	92.22%
All	2018	259331	183441	75890	41.37%
ARLINGTON	2016	19	2	17	850.%
ARLINGTON	2018	37	17	20	117.65%
CHANKER	2016	17	6	11	183.33%
CHANKER	2017	64	17	47	276.47%
CHANKER	2018	115	64	51	79.69%
CHESHIRE	2017	18	5	13	260.%
CHESHIRE	2018	31	18	13	72.22%
CHICAGO	2016	20	3	17	566.67%
CHICAGO	2017	53	20	33	165.%
CHICAGO	2018	91	53	38	71.7%
CRAIG	2018	3	2	1	50.%
DECATUR	2016	17	6	11	183.33%
DECATUR	2017	51	17	34	200.%
DECATUR	2018	53	51	2	3.92%
DUNKIRK	2016	948	190	758	398.95%
DUNKIRK	2017	2019	948	1071	112.97%
DUNKIRK	2018	2728	2019	709	35.12%

Manual Calculation

Example 1: CHICAGO, 2017

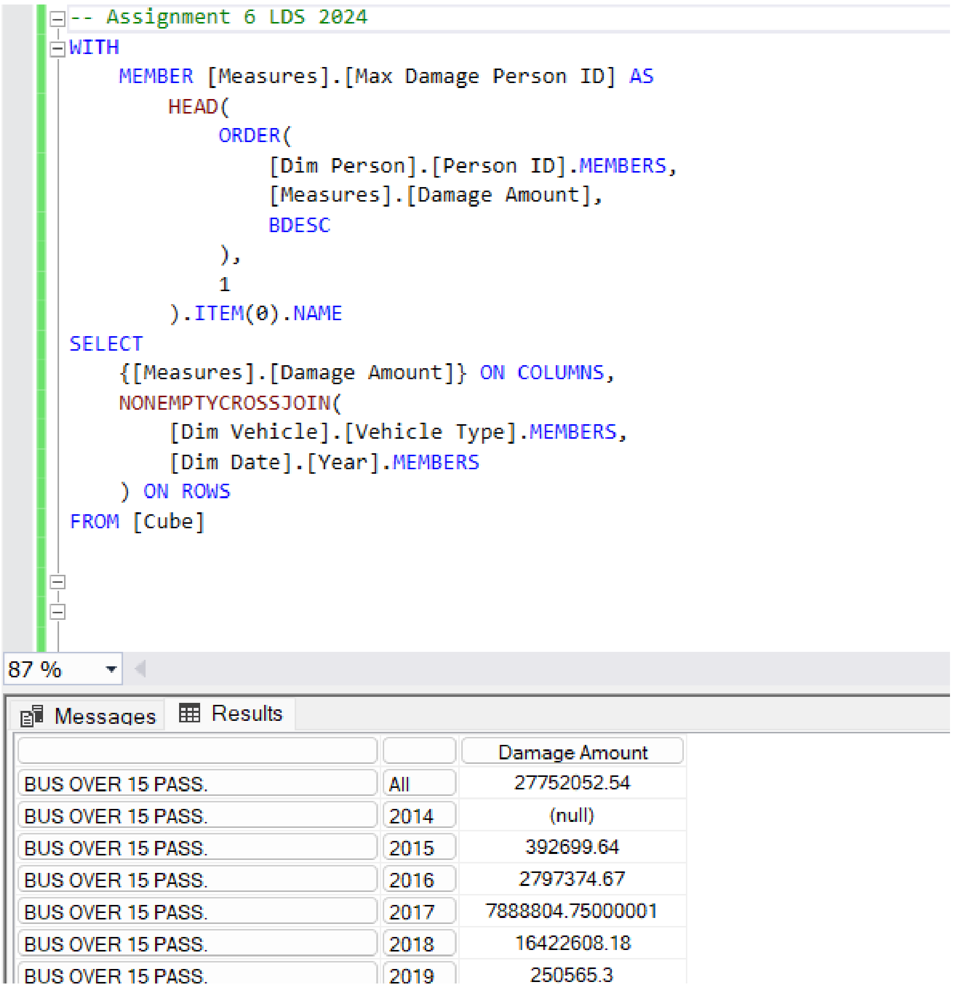
- **Num Vehicles**: 53
- **PrevYearVehicles**: 20
- **Difference**: 53−20 = 33
- **VehicleIncreasePercent**: 33/20×100 = 165%
-

Example 2: "All", 2017

- **Num Vehicles**: 183,441
- **PrevYearVehicles**: 95,435
- **Difference**: 183441−95435 = 88006
- **VehicleIncreasePercent**: 88006 / 95435×100 ≈ 92.22%

*Assignment 6**

For each vehicle type and each year, show the information and the (total) damage costs of the person with the highest reported damage



```
-- Assignment 6 LDS 2024
WITH
    MEMBER [Measures].[Max Damage Person ID] AS
        HEAD(
            ORDER(
                [Dim Person].[Person ID].MEMBERS,
                [Measures].[Damage Amount],
                BDESC
            ),
            1
        ).ITEM(0).NAME
SELECT
    {[Measures].[Damage Amount]} ON COLUMNS,
    NONEMPTYCROSSJOIN(
        [Dim Vehicle].[Vehicle Type].MEMBERS,
        [Dim Date].[Year].MEMBERS
    ) ON ROWS
FROM [Cube]
```

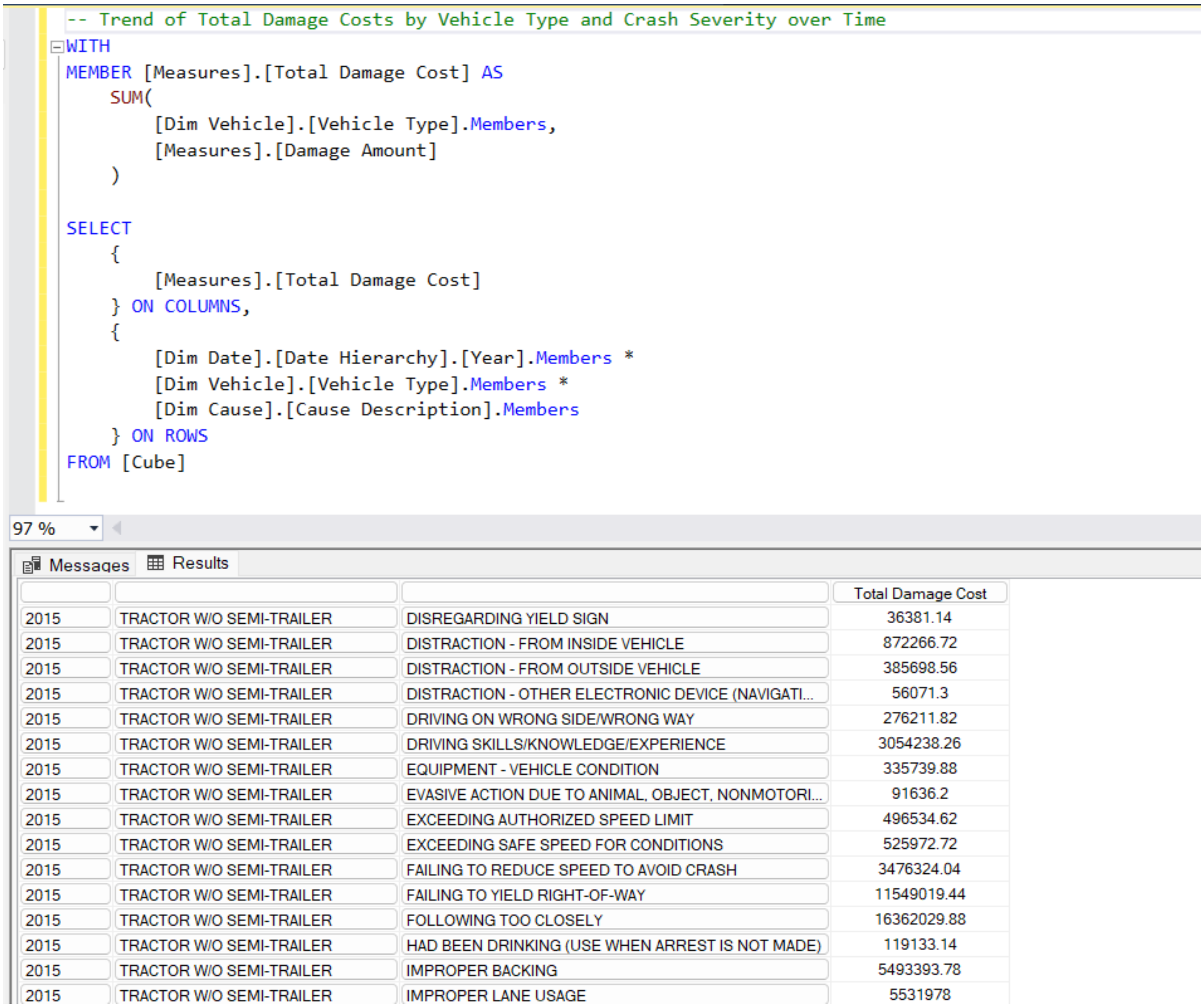
		Damage Amount
BUS OVER 15 PASS.	All	27752052.54
BUS OVER 15 PASS.	2014	(null)
BUS OVER 15 PASS.	2015	392699.64
BUS OVER 15 PASS.	2016	2797374.67
BUS OVER 15 PASS.	2017	7888804.75000001
BUS OVER 15 PASS.	2018	16422608.18
BUS OVER 15 PASS.	2019	250565.3

*Assignment 7**

Propose and solve a query showing some interesting and **non-trivial** facts you discover during the first part of the project

Query 1: **Trend of Total Damage Costs by Vehicle Type and Crash Severity over Time**

This query looks at how **total damage costs** change for each **vehicle type** and **crash severity level** over time (using **Year** from the `Dim Date` dimension). This could help us see how certain vehicle types contribute to the overall crash costs in each year.

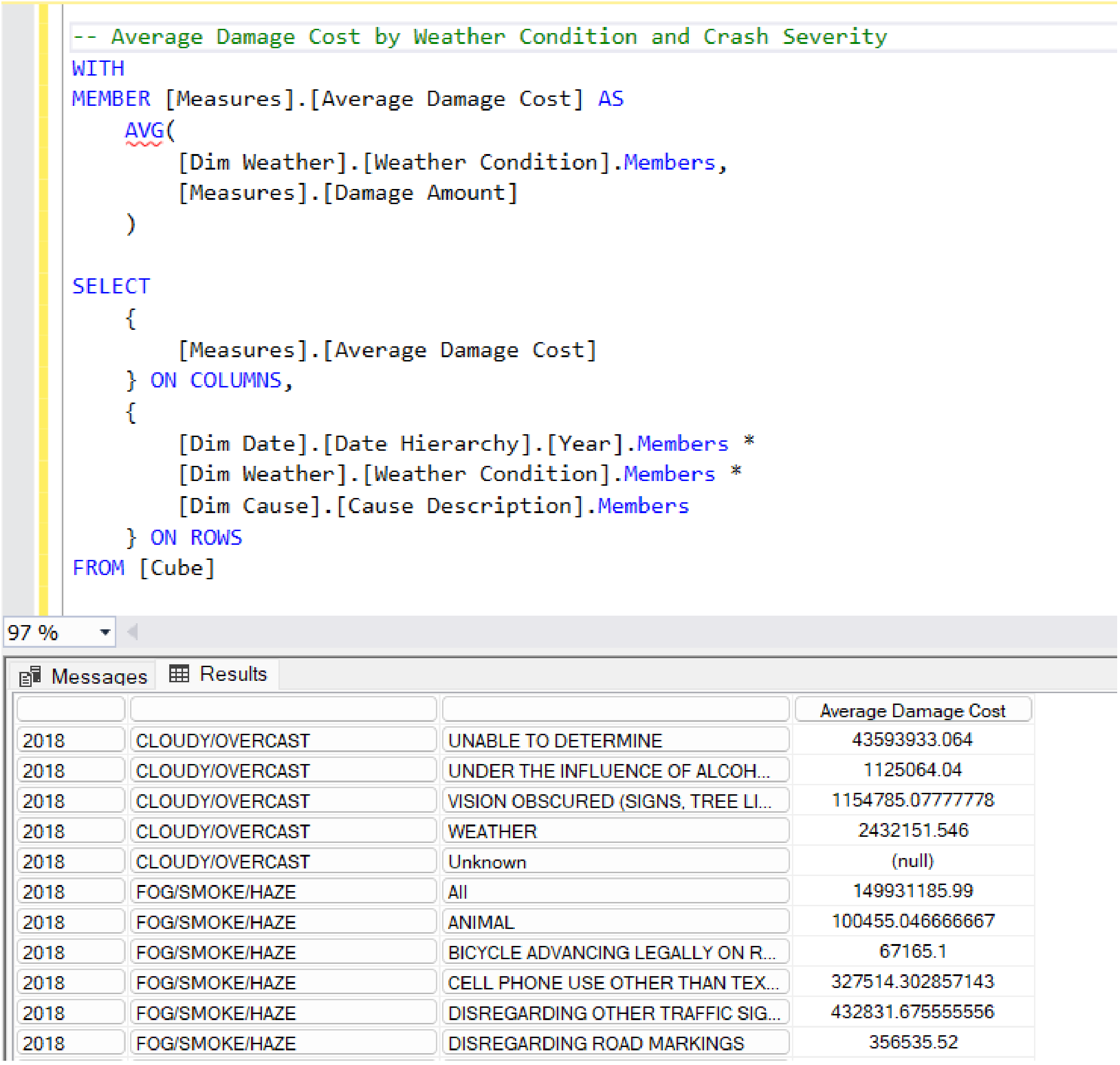


```
-- Trend of Total Damage Costs by Vehicle Type and Crash Severity over Time
WITH
MEMBER [Measures].[Total Damage Cost] AS
    SUM(
        [Dim Vehicle].[Vehicle Type].Members,
        [Measures].[Damage Amount]
    )

SELECT
    {
        [Measures].[Total Damage Cost]
    } ON COLUMNS,
    {
        [Dim Date].[Date Hierarchy].[Year].Members *
        [Dim Vehicle].[Vehicle Type].Members *
        [Dim Cause].[Cause Description].Members
    } ON ROWS
FROM [Cube]
```

			Total Damage Cost
2015	TRACTOR W/O SEMI-TRAILER	DISREGARDING YIELD SIGN	36381.14
2015	TRACTOR W/O SEMI-TRAILER	DISTRACTION - FROM INSIDE VEHICLE	872266.72
2015	TRACTOR W/O SEMI-TRAILER	DISTRACTION - FROM OUTSIDE VEHICLE	385698.56
2015	TRACTOR W/O SEMI-TRAILER	DISTRACTION - OTHER ELECTRONIC DEVICE (NAVIGATI...	56071.3
2015	TRACTOR W/O SEMI-TRAILER	DRIVING ON WRONG SIDE/WRONG WAY	276211.82
2015	TRACTOR W/O SEMI-TRAILER	DRIVING SKILLS/KNOWLEDGE/EXPERIENCE	3054238.26
2015	TRACTOR W/O SEMI-TRAILER	EQUIPMENT - VEHICLE CONDITION	335739.88
2015	TRACTOR W/O SEMI-TRAILER	EVASIVE ACTION DUE TO ANIMAL, OBJECT, NONMOTORI...	91636.2
2015	TRACTOR W/O SEMI-TRAILER	EXCEEDING AUTHORIZED SPEED LIMIT	496534.62
2015	TRACTOR W/O SEMI-TRAILER	EXCEEDING SAFE SPEED FOR CONDITIONS	525972.72
2015	TRACTOR W/O SEMI-TRAILER	FAILING TO REDUCE SPEED TO AVOID CRASH	3476324.04
2015	TRACTOR W/O SEMI-TRAILER	FAILING TO YIELD RIGHT-OF-WAY	11549019.44
2015	TRACTOR W/O SEMI-TRAILER	FOLLOWING TOO CLOSELY	16362029.88
2015	TRACTOR W/O SEMI-TRAILER	HAD BEEN DRINKING (USE WHEN ARREST IS NOT MADE)	119133.14
2015	TRACTOR W/O SEMI-TRAILER	IMPROPER BACKING	5493393.78
2015	TRACTOR W/O SEMI-TRAILER	IMPROPER LANE USAGE	5531978

Query 2: **Average Damage Cost by Weather Condition and Crash Severity**

This query can give you insights into how **weather conditions** correlate with **crash severity** in terms of the average damage cost.

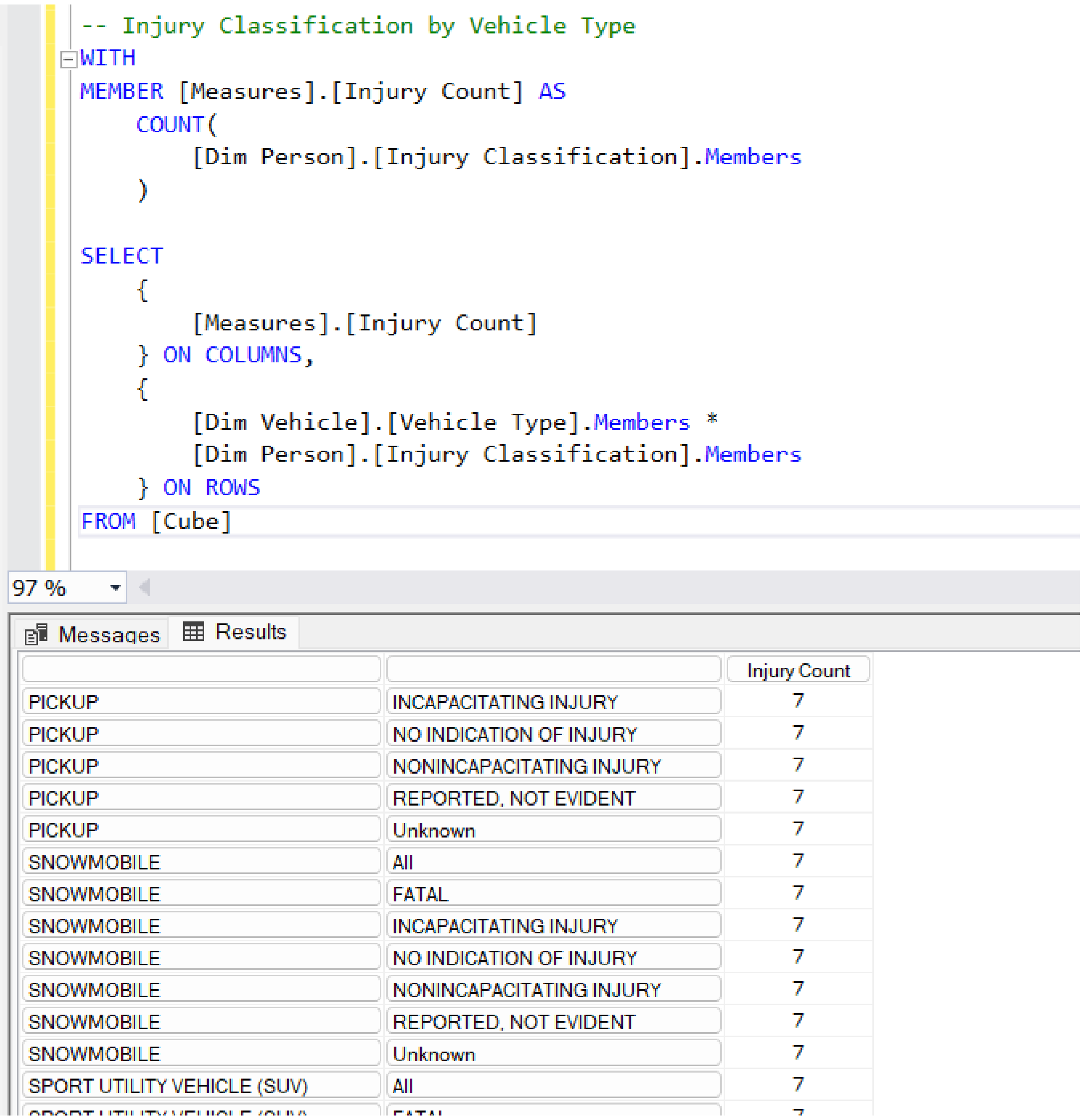


```
-- Average Damage Cost by Weather Condition and Crash Severity
WITH
MEMBER [Measures].[Average Damage Cost] AS
    AVG(
        [Dim Weather].[Weather Condition].Members,
        [Measures].[Damage Amount]
    )

SELECT
    {
        [Measures].[Average Damage Cost]
    } ON COLUMNS,
    {
        [Dim Date].[Date Hierarchy].[Year].Members *
        [Dim Weather].[Weather Condition].Members *
        [Dim Cause].[Cause Description].Members
    } ON ROWS
FROM [Cube]
```

			Average Damage Cost
2018	CLOUDY/OVERCAST	UNABLE TO DETERMINE	43593933.064
2018	CLOUDY/OVERCAST	UNDER THE INFLUENCE OF ALCOH...	1125064.04
2018	CLOUDY/OVERCAST	VISION OBSCURED (SIGNS, TREE LI...	1154785.07777778
2018	CLOUDY/OVERCAST	WEATHER	2432151.546
2018	CLOUDY/OVERCAST	Unknown	(null)
2018	FOG/SMOKE/HAZE	All	149931185.99
2018	FOG/SMOKE/HAZE	ANIMAL	100455.046666667
2018	FOG/SMOKE/HAZE	BICYCLE ADVANCING LEGALLY ON R...	67165.1
2018	FOG/SMOKE/HAZE	CELL PHONE USE OTHER THAN TEX...	327514.302857143
2018	FOG/SMOKE/HAZE	DISREGARDING OTHER TRAFFIC SIG...	432831.675555556
2018	FOG/SMOKE/HAZE	DISREGARDING ROAD MARKINGS	356535.52

Query 3: Injury Classification by Vehicle Type

This query helps explore how different vehicle types are associated with **injury classifications** during crashes, which can provide interesting insights for safety and vehicle type performance.

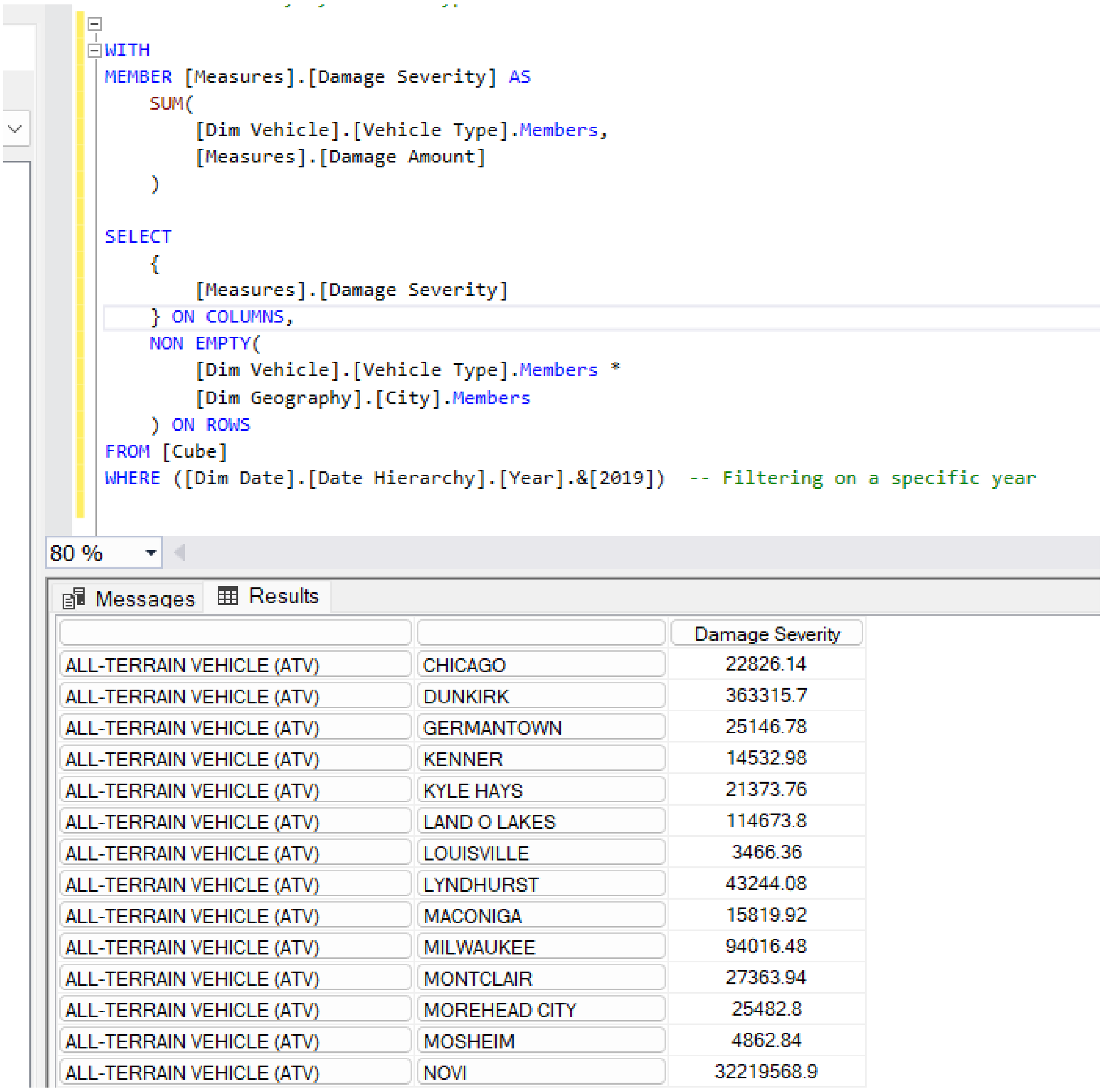


```
-- Injury Classification by Vehicle Type
WITH
MEMBER [Measures].[Injury Count] AS
    COUNT(
        [Dim Person].[Injury Classification].Members
    )

SELECT
    {
        [Measures].[Injury Count]
    } ON COLUMNS,
    {
        [Dim Vehicle].[Vehicle Type].Members *
        [Dim Person].[Injury Classification].Members
    } ON ROWS
FROM [Cube]
```

		Injury Count
PICKUP	INCAPACITATING INJURY	7
PICKUP	NO INDICATION OF INJURY	7
PICKUP	NONINCAPACITATING INJURY	7
PICKUP	REPORTED, NOT EVIDENT	7
PICKUP	Unknown	7
SNOWMOBILE	All	7
SNOWMOBILE	FATAL	7
SNOWMOBILE	INCAPACITATING INJURY	7
SNOWMOBILE	NO INDICATION OF INJURY	7
SNOWMOBILE	NONINCAPACITATING INJURY	7
SNOWMOBILE	REPORTED, NOT EVIDENT	7
SNOWMOBILE	Unknown	7
SPORT UTILITY VEHICLE (SUV)	All	7
SPORT UTILITY VEHICLE (SUV)	FATAL	7

Query 4: Crash Severity by Vehicle Type

A breakdown of **damage costs** and **severity of crashes** by **vehicle type**



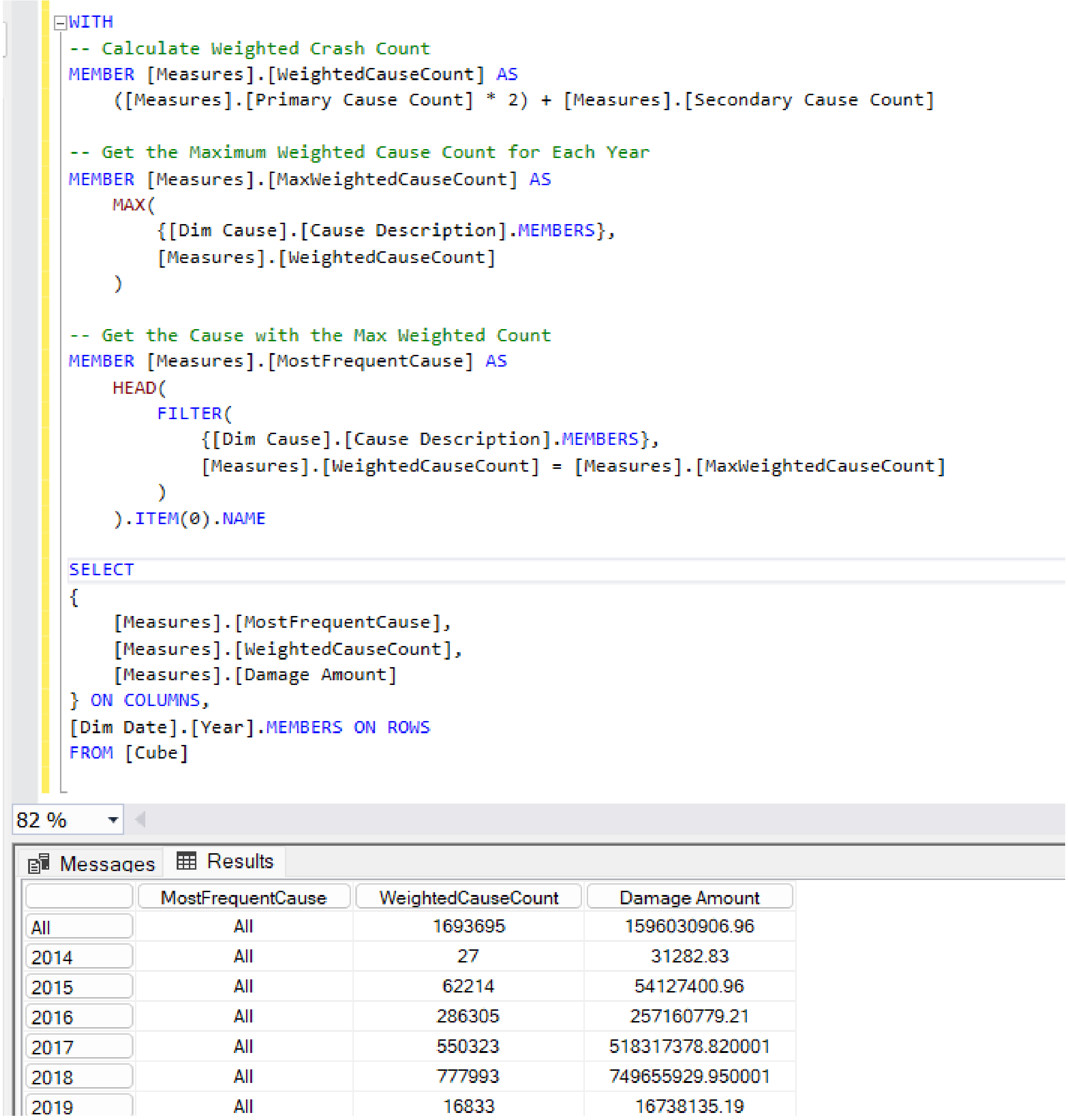
```
WITH
MEMBER [Measures].[Damage Severity] AS
    SUM(
        [Dim Vehicle].[Vehicle Type].Members,
        [Measures].[Damage Amount]
    )

SELECT
    {
        [Measures].[Damage Severity]
    } ON COLUMNS,
    NON EMPTY(
        [Dim Vehicle].[Vehicle Type].Members *
        [Dim Geography].[City].Members
    ) ON ROWS
FROM [Cube]
WHERE ([Dim Date].[Date Hierarchy].[Year].&[2019])  -- Filtering on a specific year
```

		Damage Severity
ALL-TERRAIN VEHICLE (ATV)	CHICAGO	22826.14
ALL-TERRAIN VEHICLE (ATV)	DUNKIRK	363315.7
ALL-TERRAIN VEHICLE (ATV)	GERMANTOWN	25146.78
ALL-TERRAIN VEHICLE (ATV)	KENNER	14532.98
ALL-TERRAIN VEHICLE (ATV)	KYLE HAYS	21373.76
ALL-TERRAIN VEHICLE (ATV)	LAND O LAKES	114673.8
ALL-TERRAIN VEHICLE (ATV)	LOUISVILLE	3466.36
ALL-TERRAIN VEHICLE (ATV)	LYNDHURST	43244.08
ALL-TERRAIN VEHICLE (ATV)	MACONIGA	15819.92
ALL-TERRAIN VEHICLE (ATV)	MILWAUKEE	94016.48
ALL-TERRAIN VEHICLE (ATV)	MONTCLAIR	27363.94
ALL-TERRAIN VEHICLE (ATV)	MOREHEAD CITY	25482.8
ALL-TERRAIN VEHICLE (ATV)	MOSHEIM	4862.84
ALL-TERRAIN VEHICLE (ATV)	NOVI	32219568.9

Assignment 8:

1- For each year, show the most frequent cause of crashes and the corresponding total damage costs. The primary crash contributing factor is given twice the weight of the secondary factor in the analysis. Additionally, show the overall most frequent crash cause across all years.

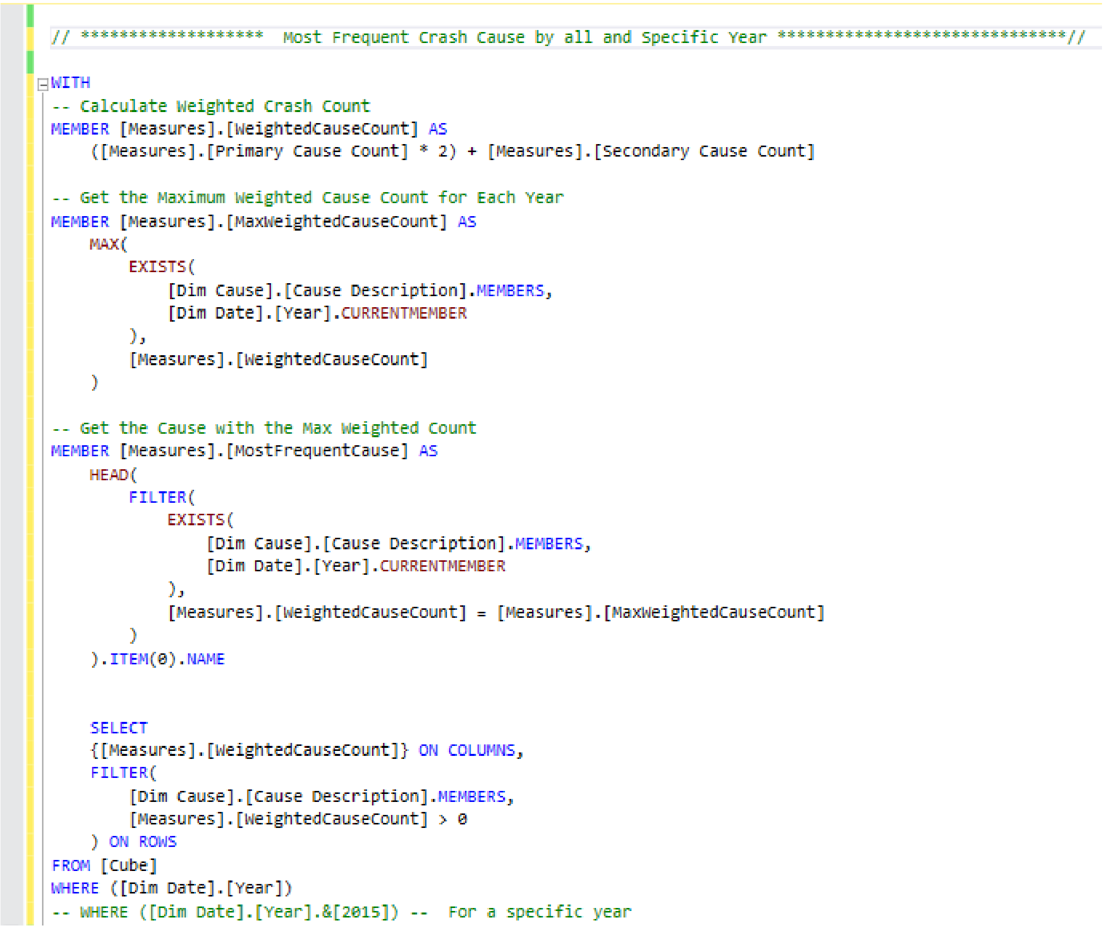
MDX Query for Per-Year Analysis



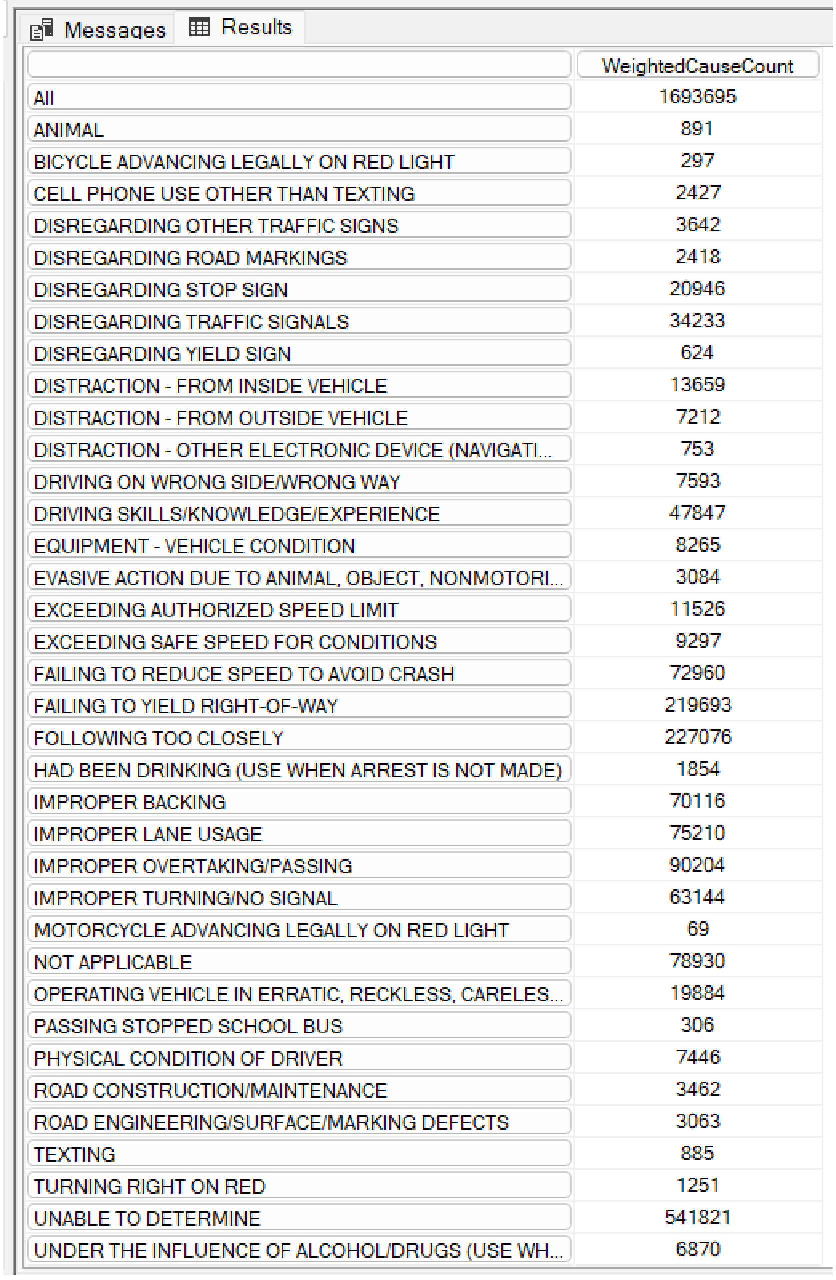
```
WITH
-- Calculate Weighted Crash Count
MEMBER [Measures].[WeightedCauseCount] AS
    ([Measures].[Primary Cause Count] * 2) + [Measures].[Secondary Cause Count]

-- Get the Maximum Weighted Cause Count for Each Year
MEMBER [Measures].[MaxWeightedCauseCount] AS
    MAX(
        {[Dim Cause].[Cause Description].MEMBERS},
        [Measures].[WeightedCauseCount]
    )

-- Get the Cause with the Max Weighted Count
MEMBER [Measures].[MostFrequentCause] AS
    HEAD(
        FILTER(
            {[Dim Cause].[Cause Description].MEMBERS},
            [Measures].[WeightedCauseCount] = [Measures].[MaxWeightedCauseCount]
        )
    ).ITEM(0).NAME

SELECT
{
    [Measures].[MostFrequentCause],
    [Measures].[WeightedCauseCount],
    [Measures].[Damage Amount]
} ON COLUMNS,
[Dim Date].[Year].MEMBERS ON ROWS
FROM [Cube]
```

	MostFrequentCause	WeightedCauseCount	Damage Amount
All	All	1693695	1596030906.96
2014	All	27	31282.83
2015	All	62214	54127400.96
2016	All	286305	257160779.21
2017	All	550323	518317378.820001
2018	All	777993	749655929.950001
2019	All	16833	16738135.19

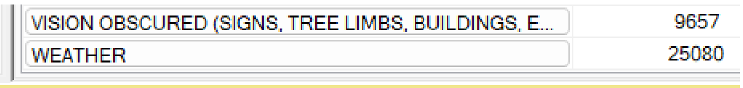
MDX Most Frequent Cause Name and Count



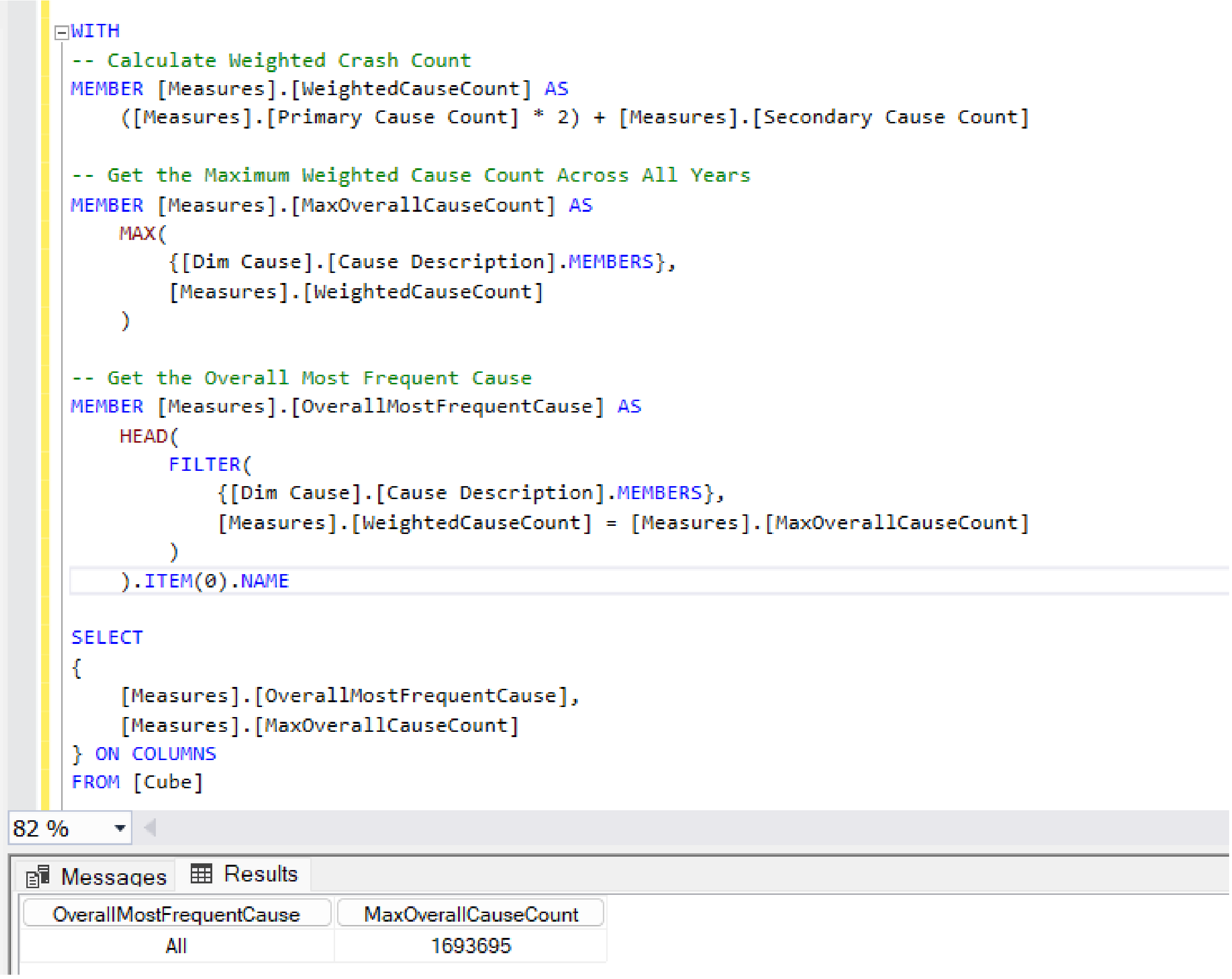
```
// *******************  Most Frequent Crash Cause by all and Specific Year ******************************//

WITH
-- Calculate Weighted Crash Count
MEMBER [Measures].[WeightedCauseCount] AS
    ([Measures].[Primary Cause Count] * 2) + [Measures].[Secondary Cause Count]

-- Get the Maximum Weighted Cause Count for Each Year
MEMBER [Measures].[MaxWeightedCauseCount] AS
    MAX(
        EXISTS(
            [Dim Cause].[Cause Description].MEMBERS,
            [Dim Date].[Year].CURRENTMEMBER
        ),
        [Measures].[WeightedCauseCount]
    )

-- Get the Cause with the Max Weighted Count
MEMBER [Measures].[MostFrequentCause] AS
    HEAD(
        FILTER(
            EXISTS(
                [Dim Cause].[Cause Description].MEMBERS,
                [Dim Date].[Year].CURRENTMEMBER
            ),
            [Measures].[WeightedCauseCount] = [Measures].[MaxWeightedCauseCount]
        )
    ).ITEM(0).NAME


    SELECT
    {[Measures].[WeightedCauseCount]} ON COLUMNS,
    FILTER(
        [Dim Cause].[Cause Description].MEMBERS,
        [Measures].[WeightedCauseCount] > 0
    ) ON ROWS
FROM [Cube]
WHERE ([Dim Date].[Year])
-- WHERE ([Dim Date].[Year].&[2015]) --  For a specific year
```



	WeightedCauseCount
All	1693695
ANIMAL	891
BICYCLE ADVANCING LEGALLY ON RED LIGHT	297
CELL PHONE USE OTHER THAN TEXTING	2427
DISREGARDING OTHER TRAFFIC SIGNS	3642
DISREGARDING ROAD MARKINGS	2418
DISREGARDING STOP SIGN	20946
DISREGARDING TRAFFIC SIGNALS	34233
DISREGARDING YIELD SIGN	624
DISTRACTION - FROM INSIDE VEHICLE	13659
DISTRACTION - FROM OUTSIDE VEHICLE	7212
DISTRACTION - OTHER ELECTRONIC DEVICE (NAVIGATI...	753
DRIVING ON WRONG SIDE/WRONG WAY	7593
DRIVING SKILLS/KNOWLEDGE/EXPERIENCE	47847
EQUIPMENT - VEHICLE CONDITION	8265
EVASIVE ACTION DUE TO ANIMAL, OBJECT, NONMOTORI...	3084
EXCEEDING AUTHORIZED SPEED LIMIT	11526
EXCEEDING SAFE SPEED FOR CONDITIONS	9297
FAILING TO REDUCE SPEED TO AVOID CRASH	72960
FAILING TO YIELD RIGHT-OF-WAY	219693
FOLLOWING TOO CLOSELY	227076
HAD BEEN DRINKING (USE WHEN ARREST IS NOT MADE)	1854
IMPROPER BACKING	70116
IMPROPER LANE USAGE	75210
IMPROPER OVERTAKING/PASSING	90204
IMPROPER TURNING/NO SIGNAL	63144
MOTORCYCLE ADVANCING LEGALLY ON RED LIGHT	69
NOT APPLICABLE	78930
OPERATING VEHICLE IN ERRATIC, RECKLESS, CARELES...	19884
PASSING STOPPED SCHOOL BUS	306
PHYSICAL CONDITION OF DRIVER	7446
ROAD CONSTRUCTION/MAINTENANCE	3462
ROAD ENGINEERING/SURFACE/MARKING DEFECTS	3063
TEXTING	885
TURNING RIGHT ON RED	1251
UNABLE TO DETERMINE	541821
UNDER THE INFLUENCE OF ALCOHOL/DRUGS (USE WH...	6870
VISION OBSCURED (SIGNS, TREE LIMBS, BUILDINGS, E...	9657
WEATHER	25080



MDX Query for Overall Analysis Across All Years

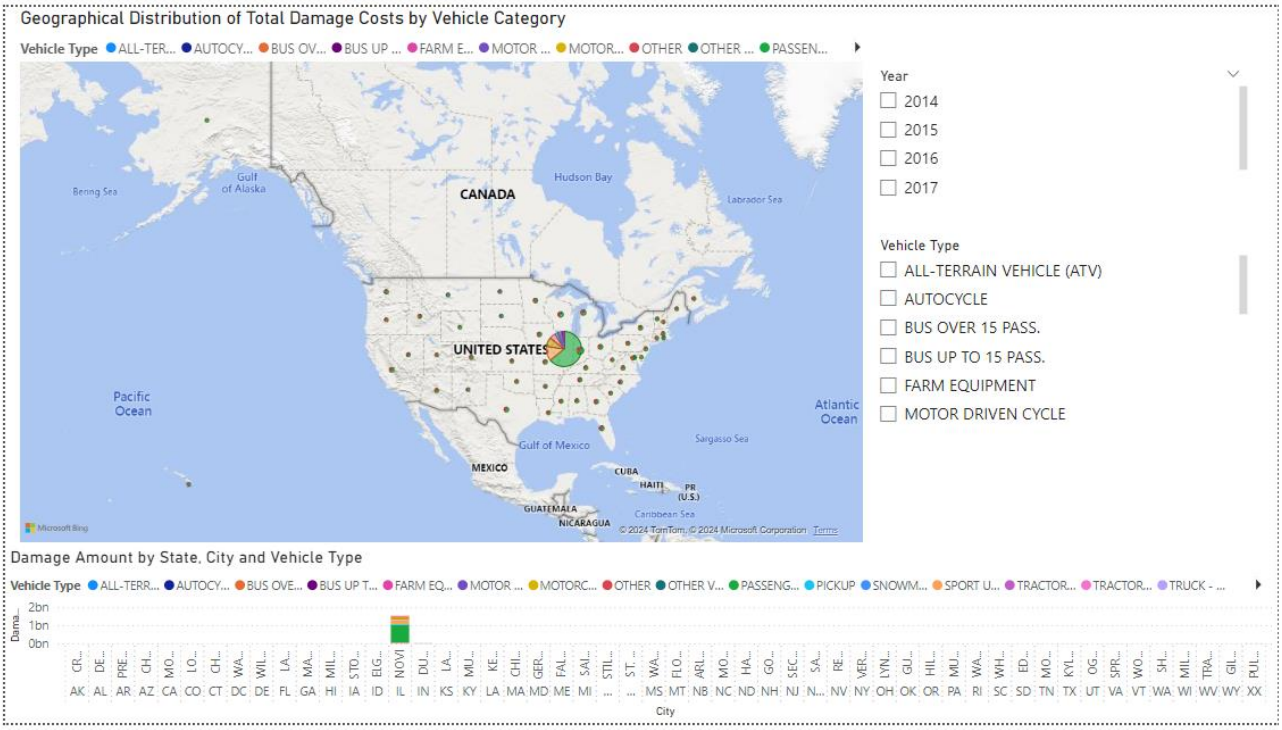


```
WITH
-- Calculate Weighted Crash Count
MEMBER [Measures].[WeightedCauseCount] AS
    ([Measures].[Primary Cause Count] * 2) + [Measures].[Secondary Cause Count]

-- Get the Maximum Weighted Cause Count Across All Years
MEMBER [Measures].[MaxOverallCauseCount] AS
    MAX(
        {[Dim Cause].[Cause Description].MEMBERS},
        [Measures].[WeightedCauseCount]
    )

-- Get the Overall Most Frequent Cause
MEMBER [Measures].[OverallMostFrequentCause] AS
    HEAD(
        FILTER(
            {[Dim Cause].[Cause Description].MEMBERS},
            [Measures].[WeightedCauseCount] = [Measures].[MaxOverallCauseCount]
        )
    ).ITEM(0).NAME

SELECT
{
    [Measures].[OverallMostFrequentCause],
    [Measures].[MaxOverallCauseCount]
} ON COLUMNS
FROM [Cube]
```

OverallMostFrequentCause	MaxOverallCauseCount
All	1693695

Assignment 9:

Create a dashboard that shows the geographical distribution of the total damage costs for each vehicle category


Geographical Distribution of Total Damage Costs by Vehicle Category
Vehicle Type ● ALL-TER... ● AUTOCY... ● BUS OV... ● BUS UP ... ● FARM E... ● MOTOR ... ● MOTOR... ● OTHER ● OTHER ... ● PASSEN...
CANADA
UNITED STATES
MEXICO
Pacific Ocean
Atlantic Ocean
Hudson Bay
Gulf of Mexico
Damage Amount by State, City and Vehicle Type
Vehicle Type ● ALL-TERR... ● AUTOCY... ● BUS OVE... ● BUS UP T... ● FARM EQ... ● MOTOR ... ● MOTORC... ● OTHER ● OTHER V... ● PASSENG... ● PICKUP ● SNOWM... ● SPORT U... ● TRACTOR... ● TRACTOR... ● TRUCK - ...
2bn
1bn
0bn
Dama...
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI ... MS MT NB NC ND NH NJ N... NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY XX
City

Year

□ 2014

□ 2015

□ 2016

□ 2017

Vehicle Type

□ ALL-TERRAIN VEHICLE (ATV)

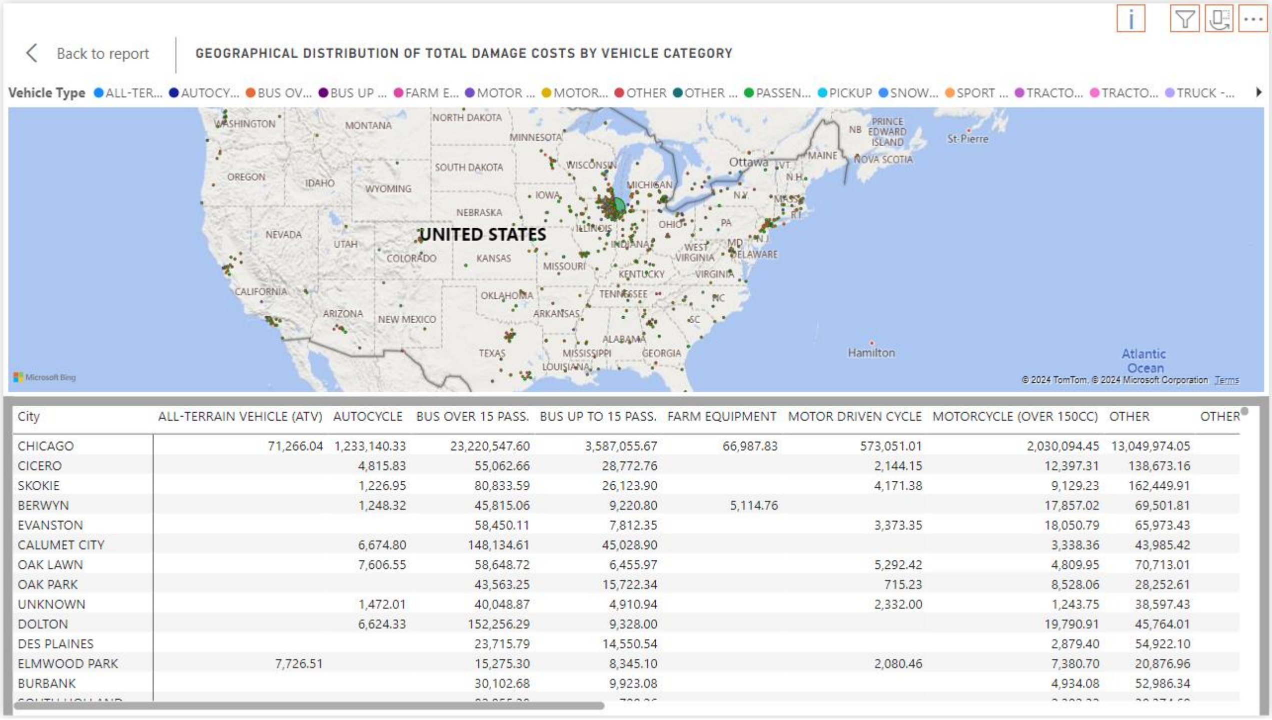
□ AUTOCYCLE

□ BUS OVER 15 PASS.

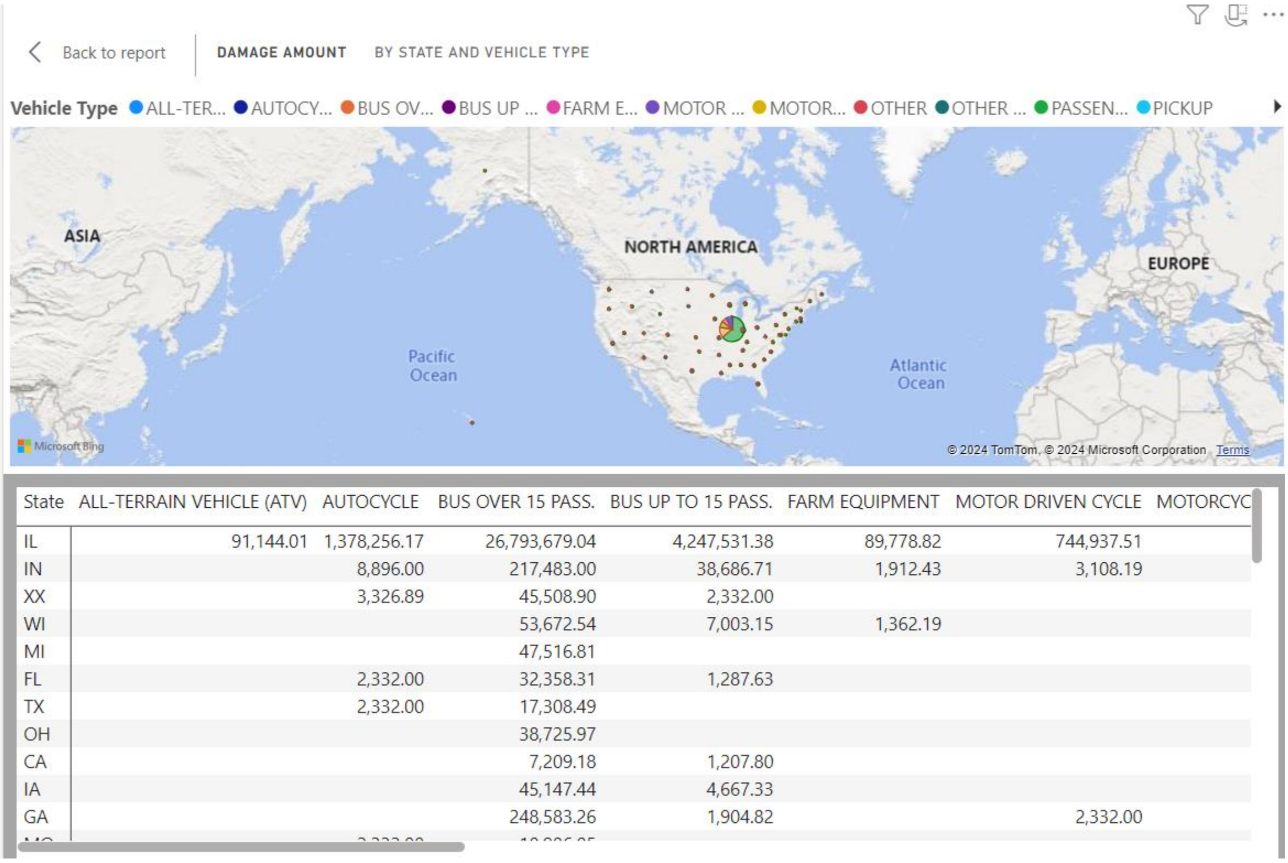
□ BUS UP TO 15 PASS.

□ FARM EQUIPMENT

□ MOTOR DRIVEN CYCLE

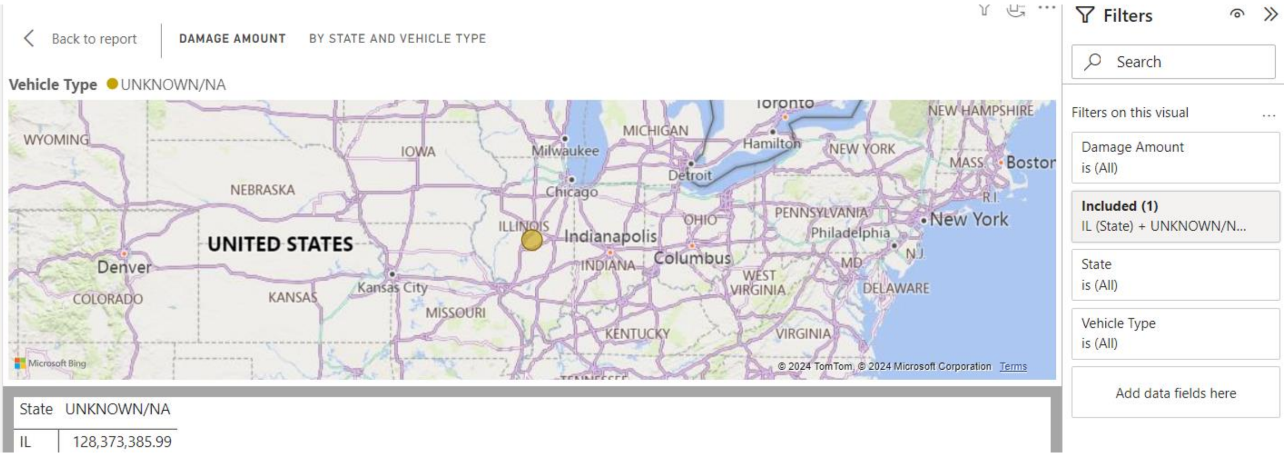


City	ALL-TERRAIN VEHICLE (ATV)	AUTOCYCLE	BUS OVER 15 PASS.	BUS UP TO 15 PASS.	FARM EQUIPMENT	MOTOR DRIVEN CYCLE	MOTORCYCLE (OVER 150CC)	OTHER	OTHER
CHICAGO	71,266.04	1,233,140.33	23,220,547.60	3,587,055.67	66,987.83	573,051.01	2,030,094.45	13,049,974.05	
CICERO		4,815.83	55,062.66	28,772.76		2,144.15	12,397.31	138,673.16	
SKOKIE		1,226.95	80,833.59	26,123.90		4,171.38	9,129.23	162,449.91	
BERWYN		1,248.32	45,815.06	9,220.80	5,114.76		17,857.02	69,501.81	
EVANSTON			58,450.11	7,812.35		3,373.35	18,050.79	65,973.43	
CALUMET CITY		6,674.80	148,134.61	45,028.90			3,338.36	43,985.42	
OAK LAWN		7,606.55	58,648.72	6,455.97		5,292.42	4,809.95	70,713.01	
OAK PARK			43,563.25	15,722.34		715.23	8,528.06	28,252.61	
UNKNOWN		1,472.01	40,048.87	4,910.94		2,332.00	1,243.75	38,597.43	
DOLTON		6,624.33	152,256.29	9,328.00			19,790.91	45,764.01	
DES PLAINES			23,715.79	14,550.54			2,879.40	54,922.10	
ELMWOOD PARK	7,726.51		15,275.30	8,345.10		2,080.46	7,380.70	20,876.96	
BURBANK			30,102.68	9,923.08			4,934.08	52,986.34	



Vehicle Type ● ALL-TER... ● AUTOCY... ● BUS OV... ● BUS UP ... ● FARM E... ● MOTOR ... ● MOTOR... ● OTHER ● OTHER ... ● PASSEN... ● PICKUP

State	ALL-TERRAIN VEHICLE (ATV)	AUTOCYCLE	BUS OVER 15 PASS.	BUS UP TO 15 PASS.	FARM EQUIPMENT	MOTOR DRIVEN CYCLE	MOTORCYC
IL	91,144.01	1,378,256.17	26,793,679.04	4,247,531.38	89,778.82	744,937.51	
IN		8,896.00	217,483.00	38,686.71	1,912.43	3,108.19	
XX		3,326.89	45,508.90	2,332.00			
WI			53,672.54	7,003.15	1,362.19		
MI			47,516.81				
FL		2,332.00	32,358.31	1,287.63			
TX		2,332.00	17,308.49				
OH			38,725.97				
CA			7,209.18	1,207.80			
IA			45,147.44	4,667.33			
GA			248,583.26	1,904.82		2,332.00	



Vehicle Type ● UNKNOWN/NA

State	UNKNOWN/NA
IL	128,373,385.99

Filters

Search

Filters on this visual

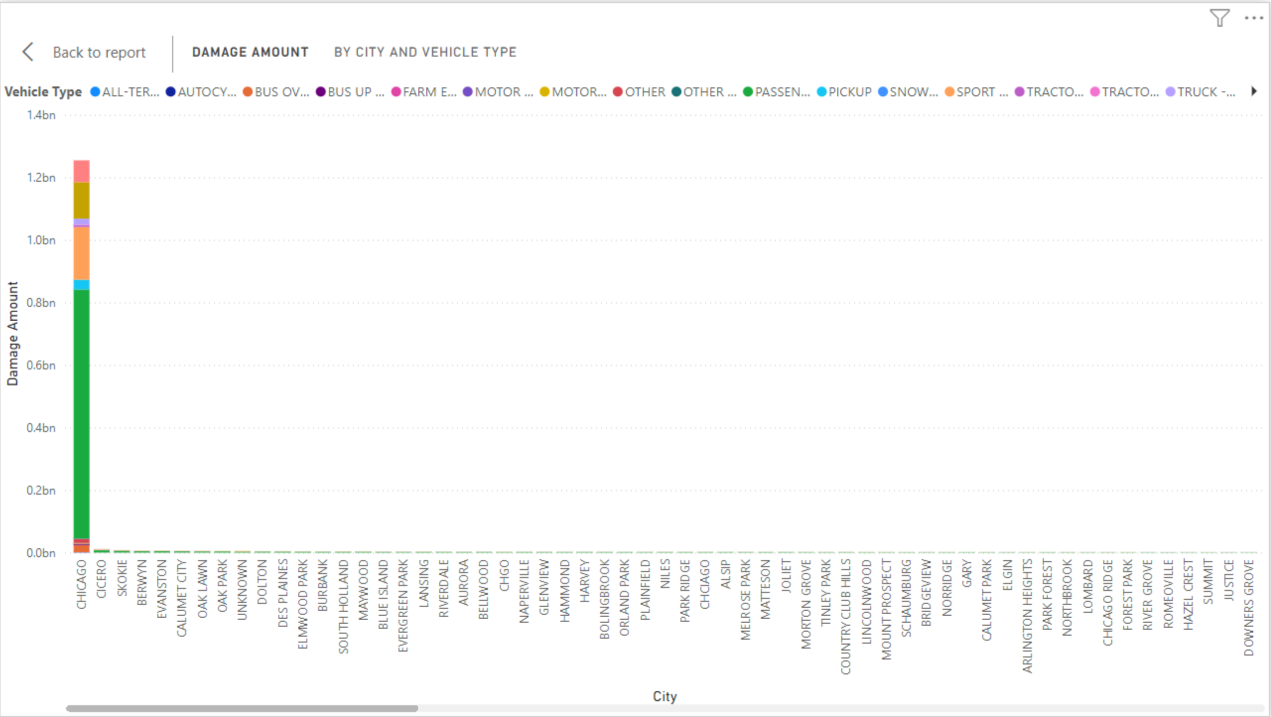
Damage Amount
is (All)

Included (1)
IL (State) + UNKNOWN/N...

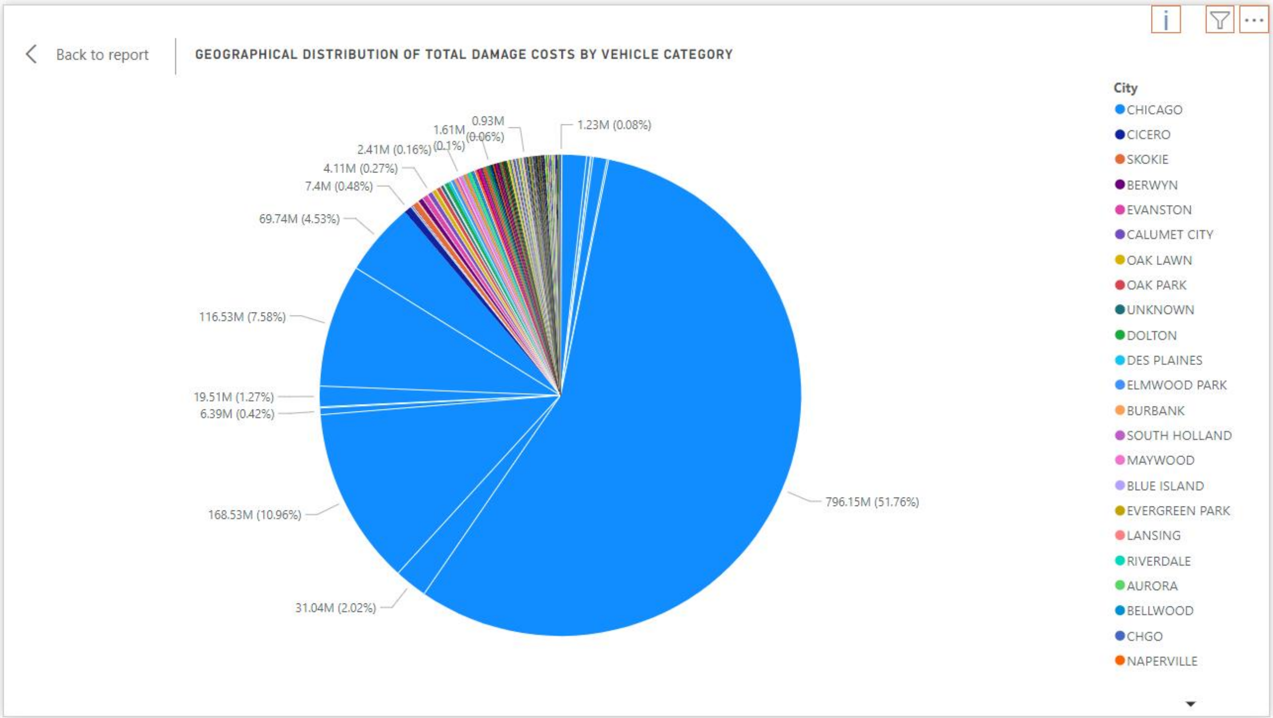
State
is (All)

Vehicle Type
is (All)

Add data fields here



DAMAGE AMOUNT BY CITY AND VEHICLE TYPE



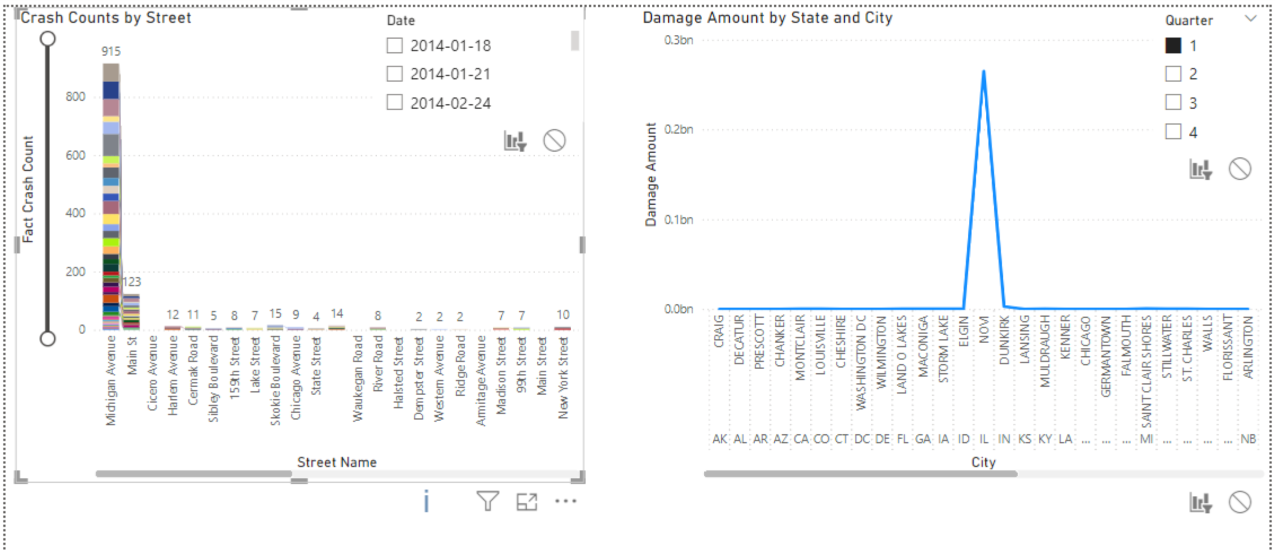
GEOGRAPHICAL DISTRIBUTION OF TOTAL DAMAGE COSTS BY VEHICLE CATEGORY

Assignment 10:

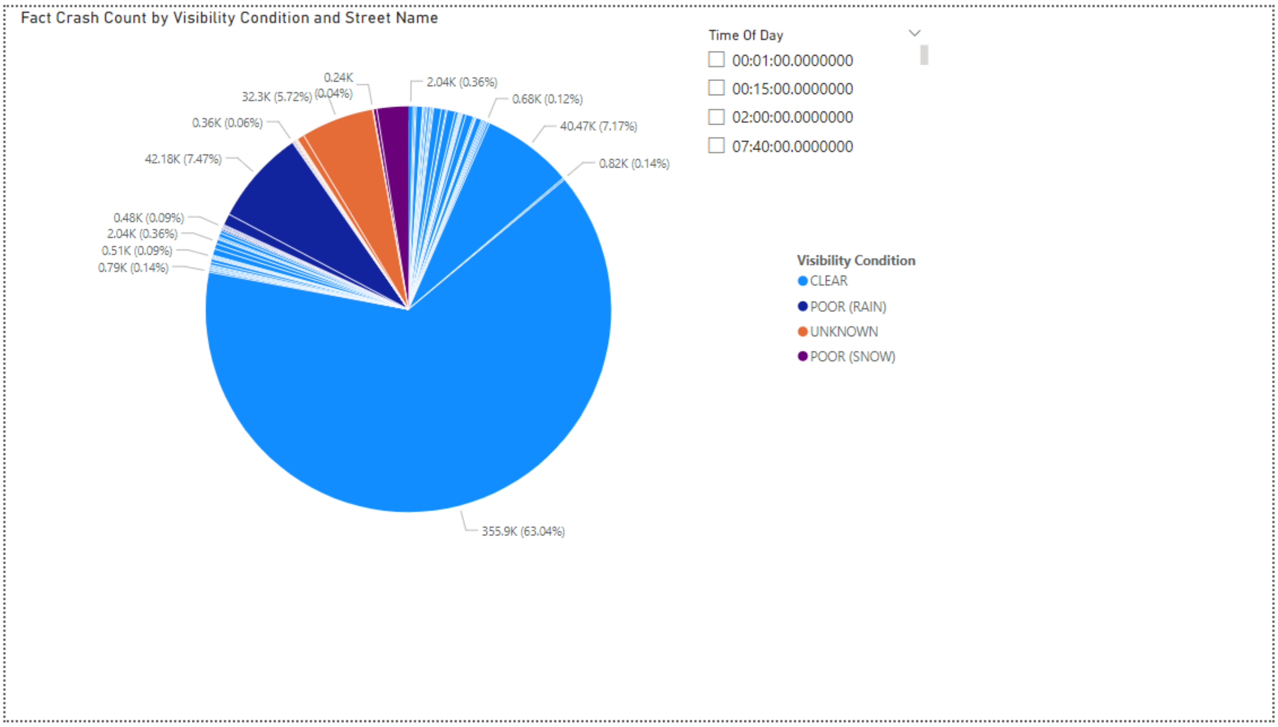
Create a plot/dashboard that you deem interesting w.r.t. the data available in your cube, focusing on data about the street.

A. Crash Count by Street with Data Filter and damage amount by State and City by with Quarter filter. In the power BI menu bar, we used format to enable and disable edit interactions.

It helped me to apply filters on specific graphs.

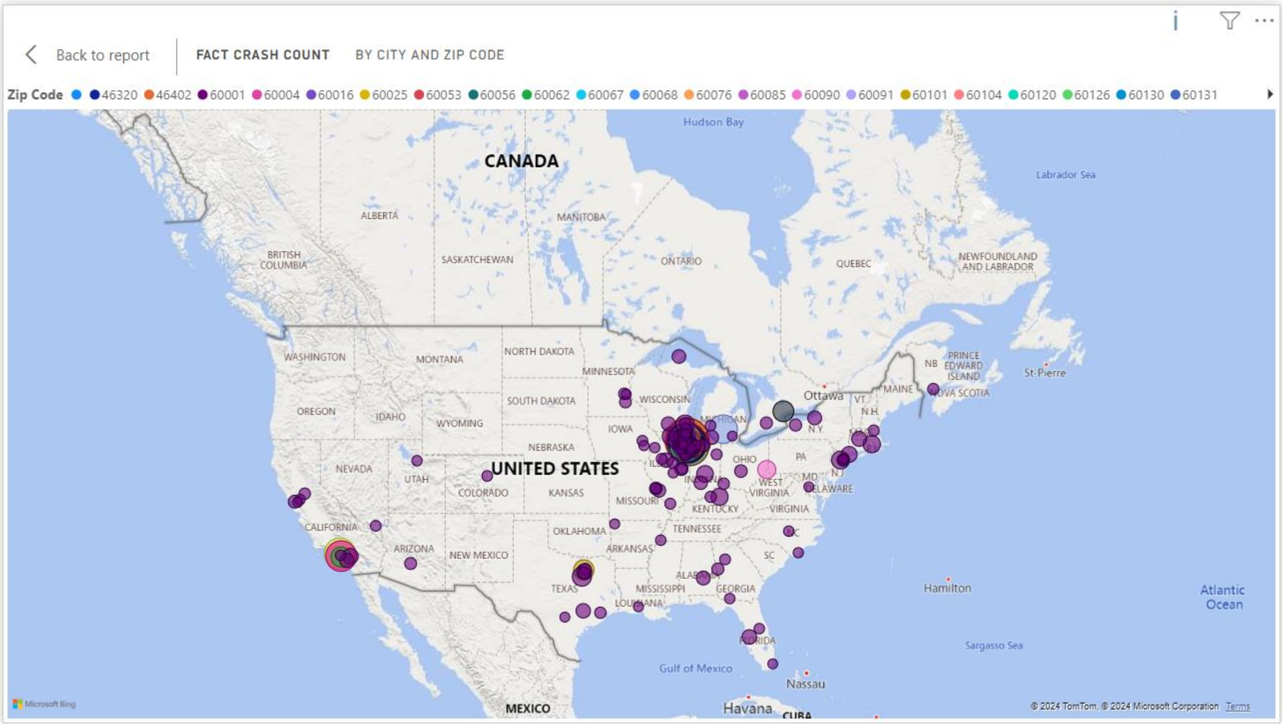


B. Crash count by lightening conditions and Incident severity with Time of the day filter to know which time crashes happed more.

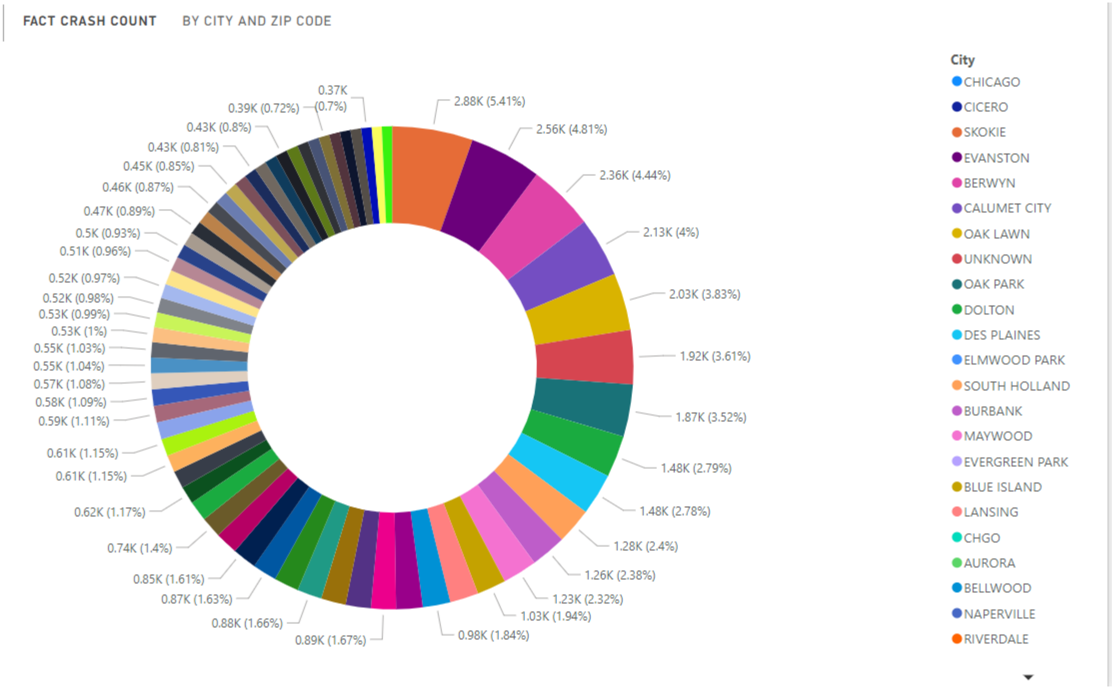


C. Fact Crash Count by City and Zip Code

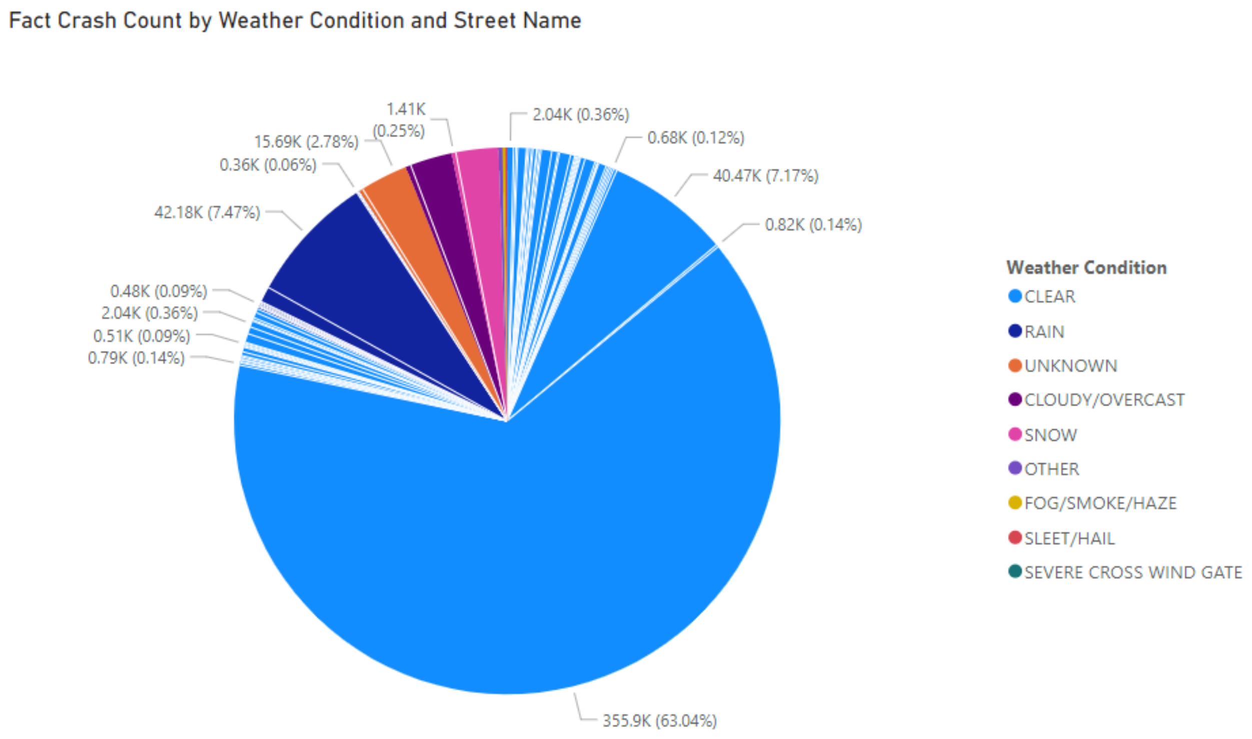
Map View

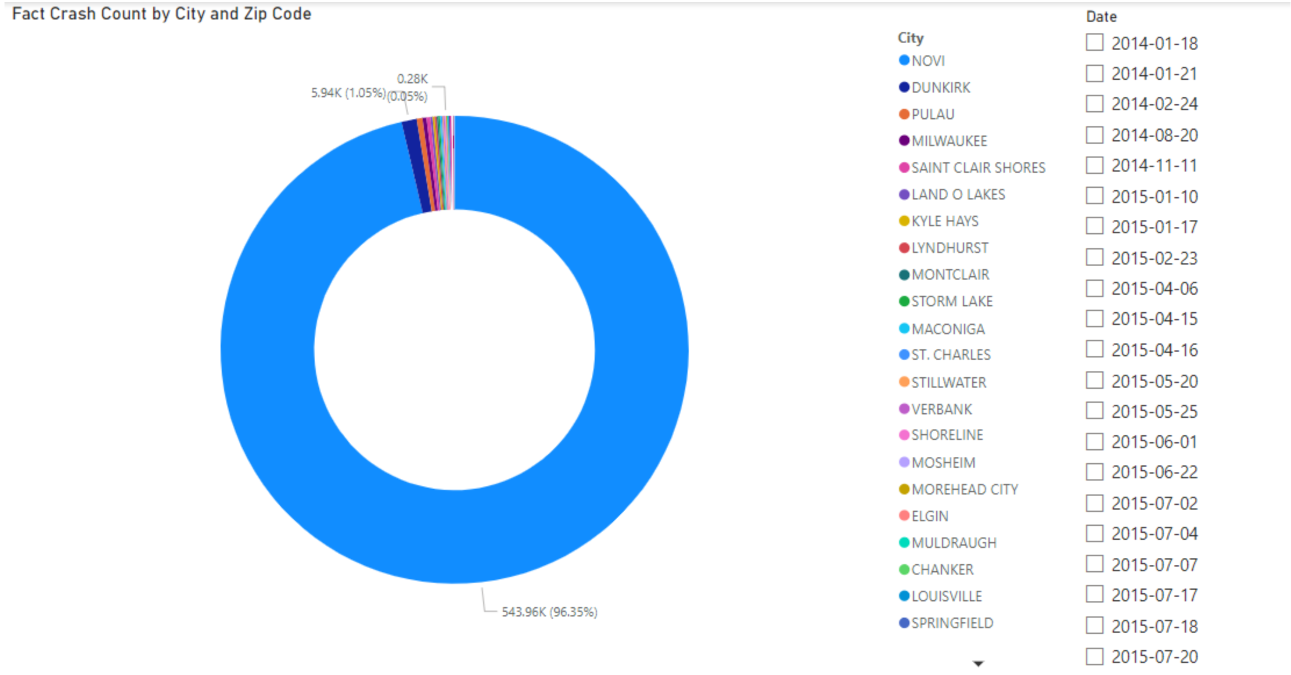

Back to report
FACT CRASH COUNT
BY CITY AND ZIP CODE
Zip Code ● ● 46320 ● 46402 ● 60001 ● 60004 ● 60016 ● 60025 ● 60053 ● 60056 ● 60062 ● 60067 ● 60068 ● 60076 ● 60085 ● 60090 ● 60091 ● 60101 ● 60104 ● 60120 ● 60126 ● 60130 ● 60131
CANADA
UNITED STATES
MEXICO
Hudson Bay
Labrador Sea
Gulf of Mexico
Atlantic Ocean
Sargasso Sea
© 2024 TomTom, © 2024 Microsoft Corporation Terms
Microsoft Bing

Donut View



D. Crash Count by Weather Condition and Street name

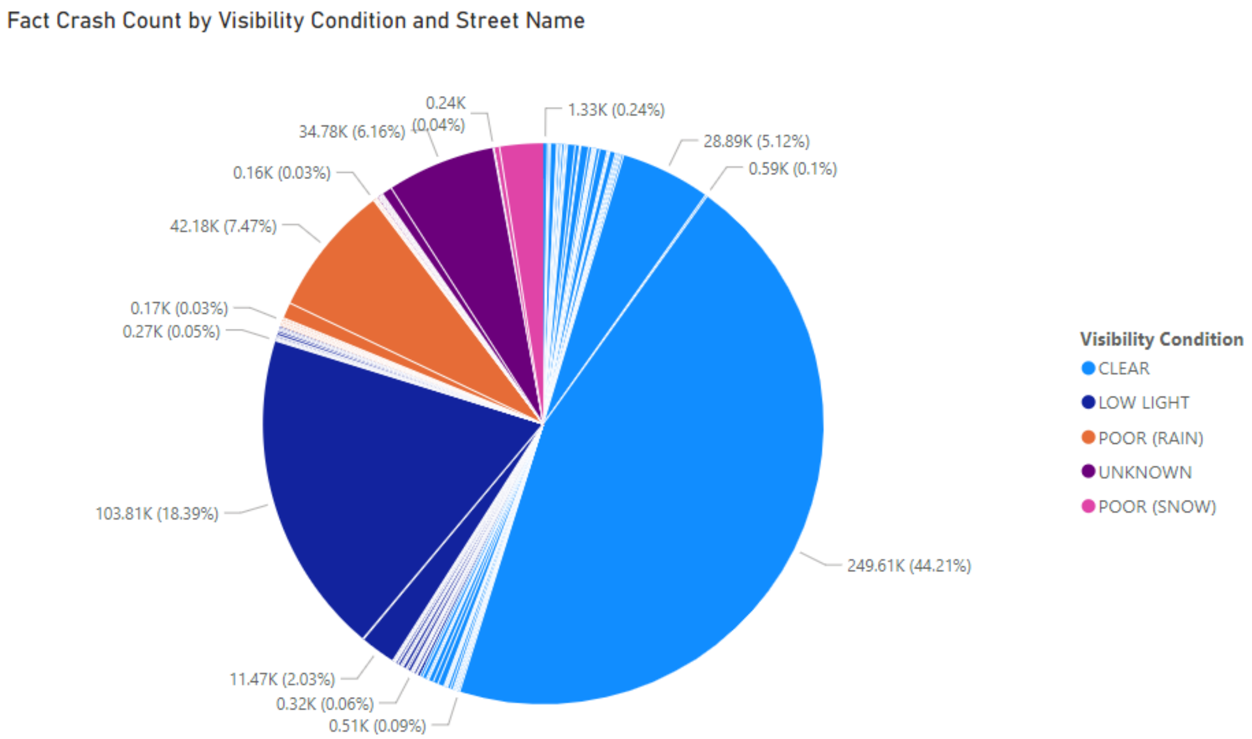




Date

- ☐ 2014-01-18
- ☐ 2014-01-21
- ☐ 2014-02-24
- ☐ 2014-08-20
- ☐ 2014-11-11
- ☐ 2015-01-10
- ☐ 2015-01-17
- ☐ 2015-02-23
- ☐ 2015-04-06
- ☐ 2015-04-15
- ☐ 2015-04-16
- ☐ 2015-05-20
- ☐ 2015-05-25
- ☐ 2015-06-01
- ☐ 2015-06-22
- ☐ 2015-07-02
- ☐ 2015-07-04
- ☐ 2015-07-07
- ☐ 2015-07-17
- ☐ 2015-07-18
- ☐ 2015-07-20

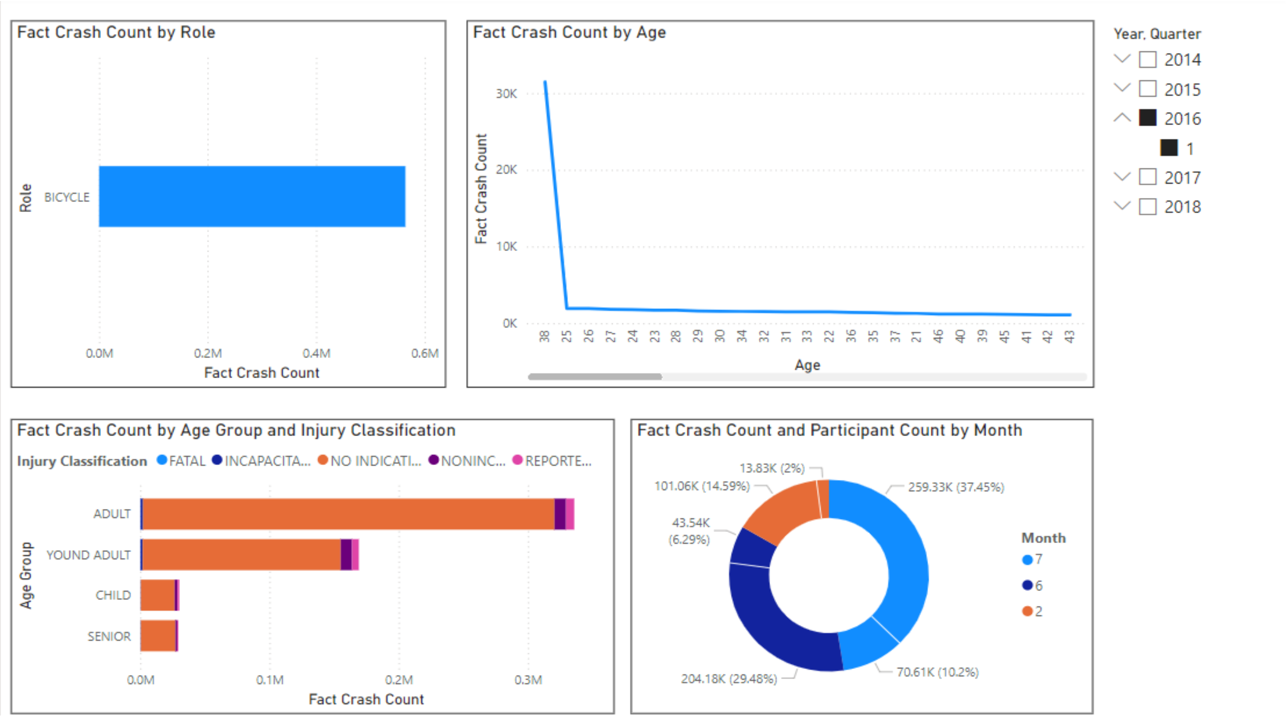
E. Crash Count by Visibility Condition and Street name



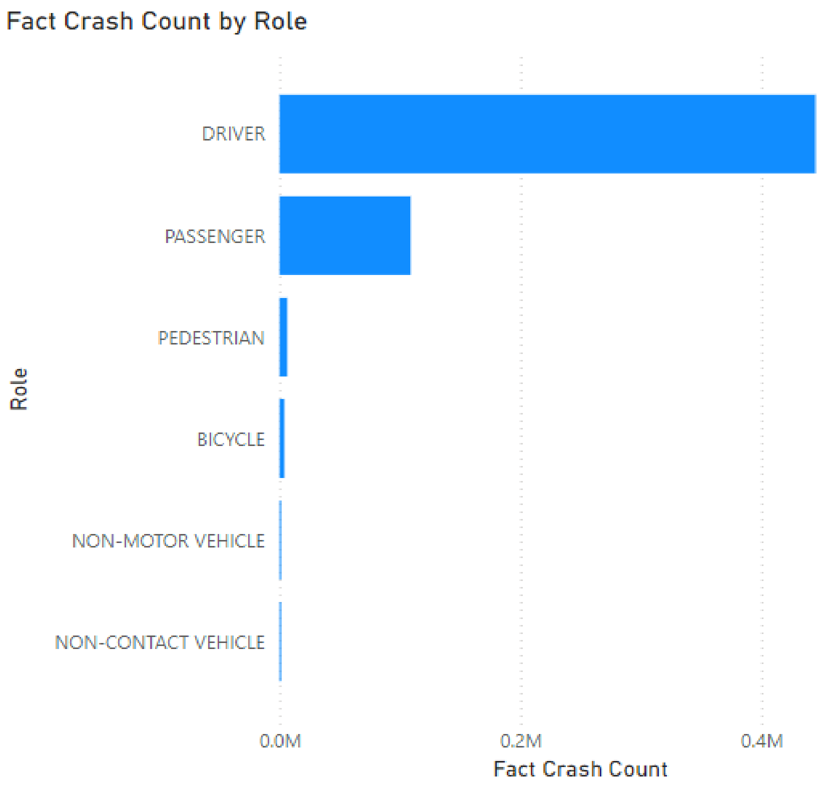
Assignment 11:

Create a plot/dashboard that you deem interesting w.r.t. the data available in your cube, focusing on data about the people involved in a crash.

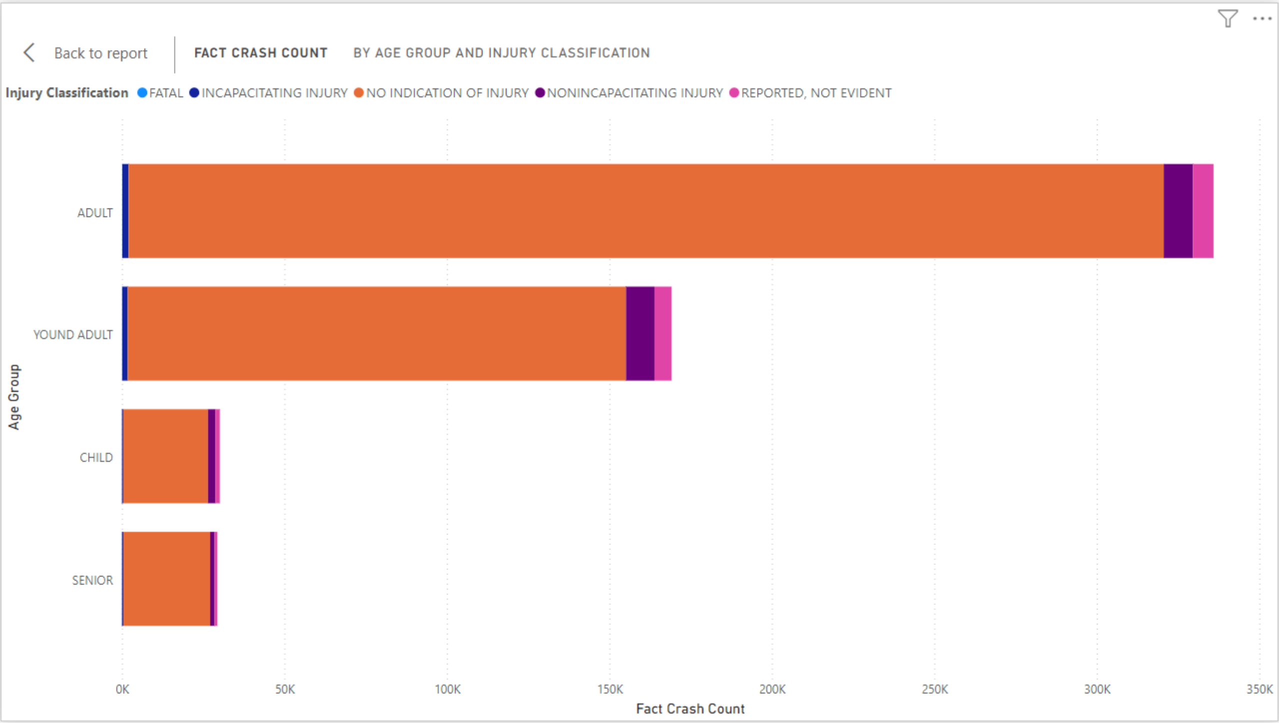
We used slicer by observing data by year and quarter by enabling edit formatting to apply filters on specific graphs as well.



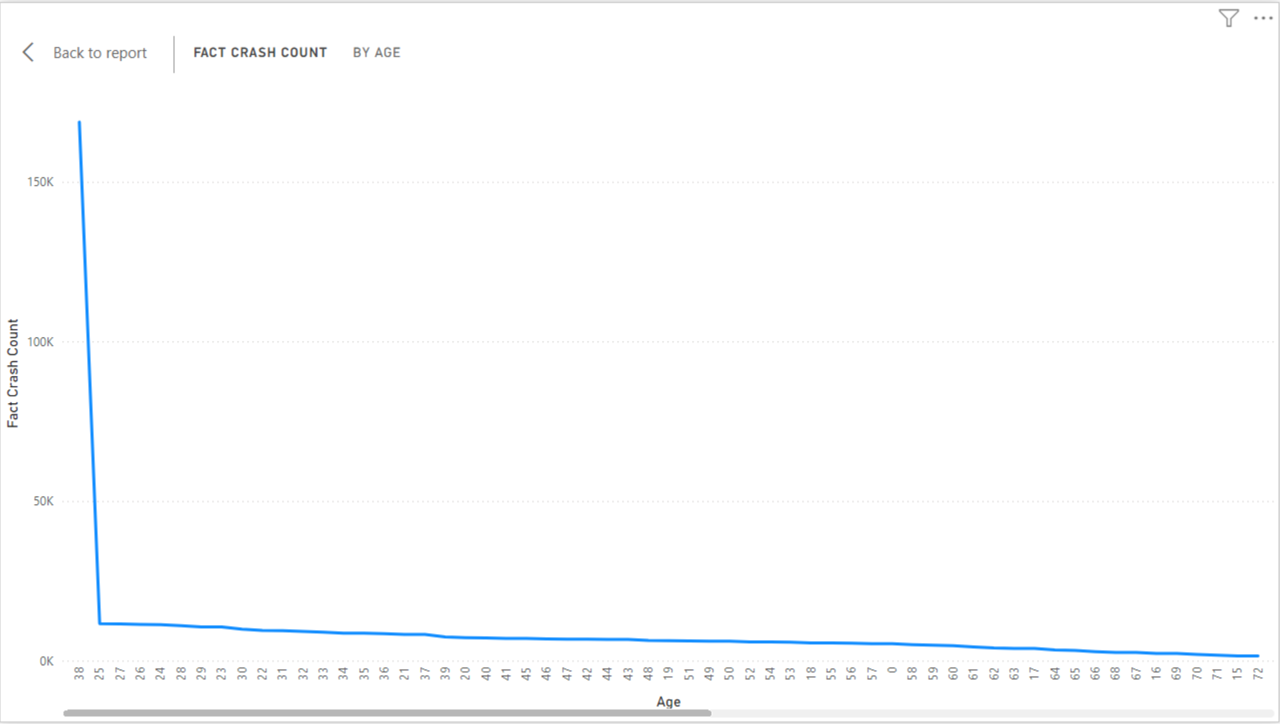
A. Role vs Crash Count



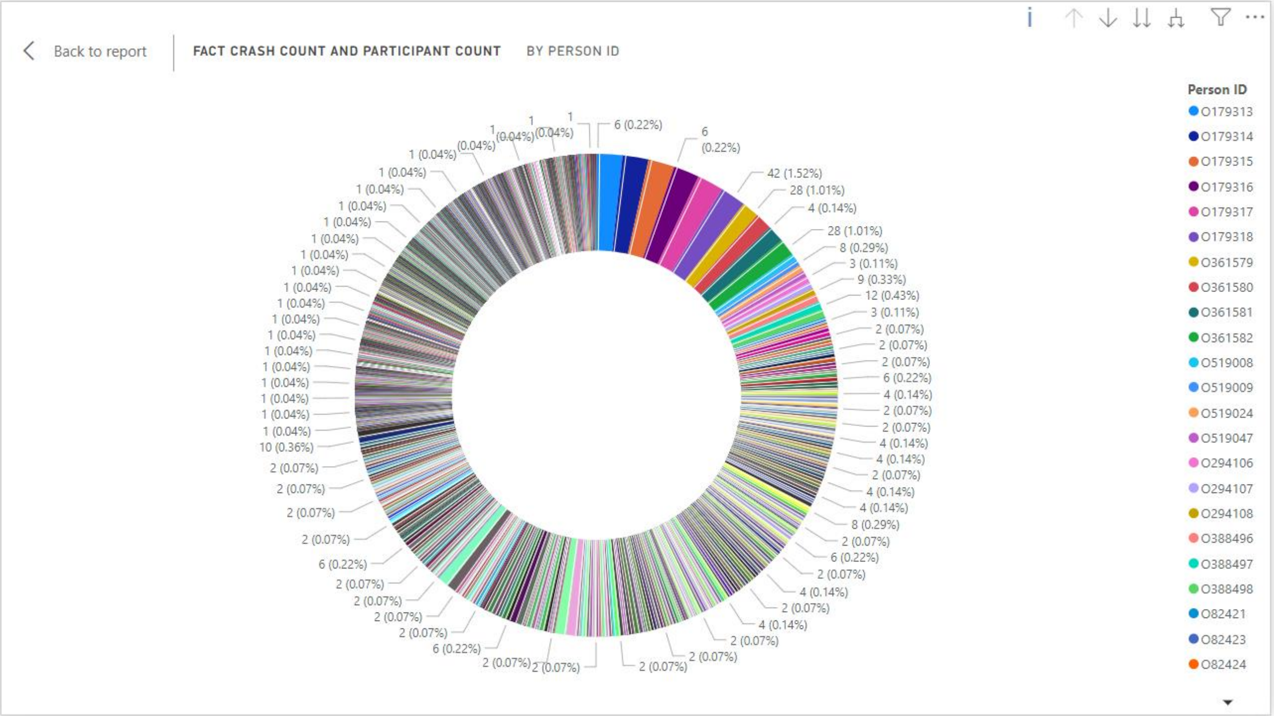
B. Injury Classification by Age Group

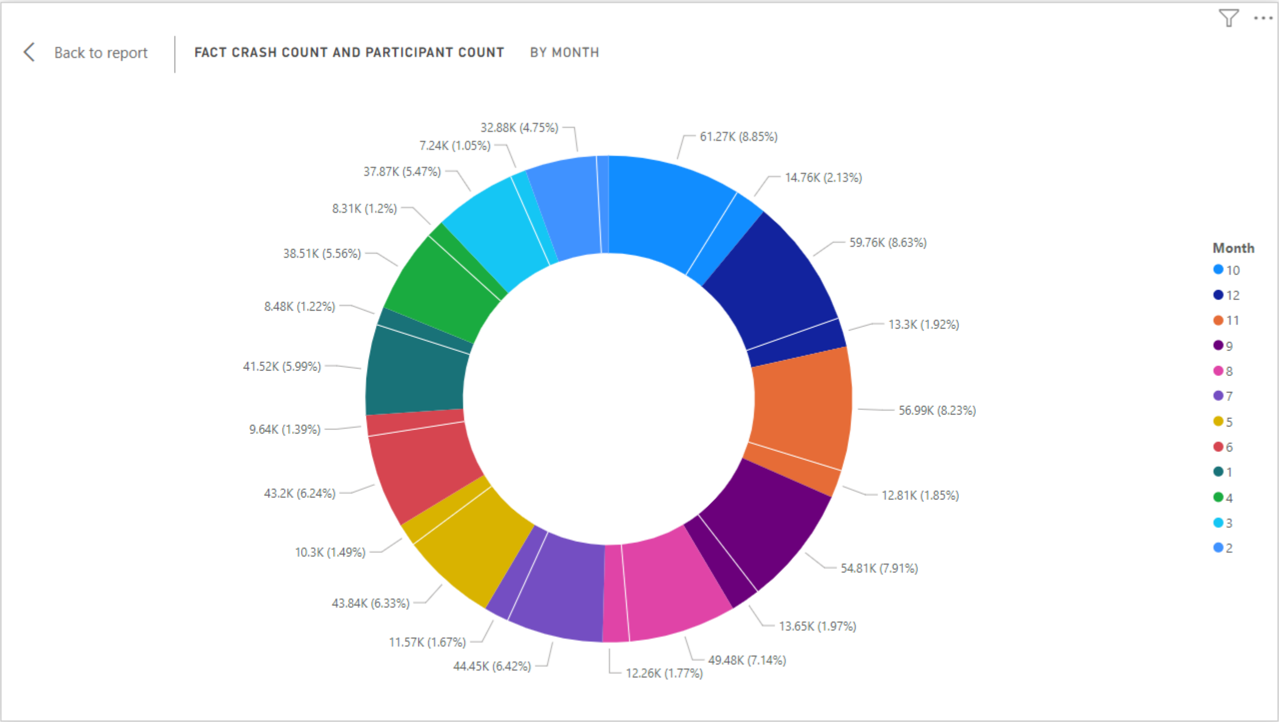


C. Crash Severity by Age



D. Crash Count and Participant Count by Person ID and Month.





Back to report

FACT CRASH COUNT AND PARTICIPANT COUNT BY MONTH