# Chicago Traffic Crashes Data Analysis

## Laboratory of Data Science

Sara Hoxha
Alessandro Carella
Rafael Ignacio Urbina Hincapie

01

# Data Understanding & Cleaning

# Data Understanding

## Datasets

Three datasets regarding data about incidents of crashes, the people and vehicles involved in the crashes.

## Observations

Some key issues we identified across all three datasets were:

❏ Missing data
❏ Inconsistent formatting
❏ Erroneous values

# Data Cleaning: People

## Data Inconsistency

- **AGE**: Set *'NaN'* for drivers < 10, assuming errors or anomalies.
- **VEHICLE ID & AGE**: Convert from float to integer for consistency.
- **CITY**:Set "*Unknown*" when numeric, too short (<2), or starts with "*UNK*."
- **STATE:** Set "*Unknown*" when CITY is "*Unknown*" or STATE = "*XX*."

## Missing values

- **Columns with "Unknown" values**: Fill missing observations with existing "*Unknown*" equivalents.
- **Driver-related columns (for passengers):** Use "*N/A*".
- **SEX**: Map NaN values to "*U*" (Unknown) for consistency.

# Data Cleaning: Vehicles

## Data Inconsistency

**YEAR**: Years recorded beyond 2024 were marked as *’UNKNOWN’*. *’999’* was identified as a typo for *’1999’*.

**LICENSE PLATE STATE**: Set "*Unknown*" for *NaN* or “*XX*".

**MAKE & MODEL:** Duplicates removed. *’UNKNOWN’* and *’UKNOW’* values were standardized to *’UNKNOWN’*.

## Missing values

**Columns with "Unknown" values**: Fill missing observations with existing "*Unknown*" equivalents.
Example columns:

- *VEHICLE_TYPE*
- *VEHICLE_USE*
- *VEHICLE_DIRECTION*
- etc…

# Data Cleaning: Crashes

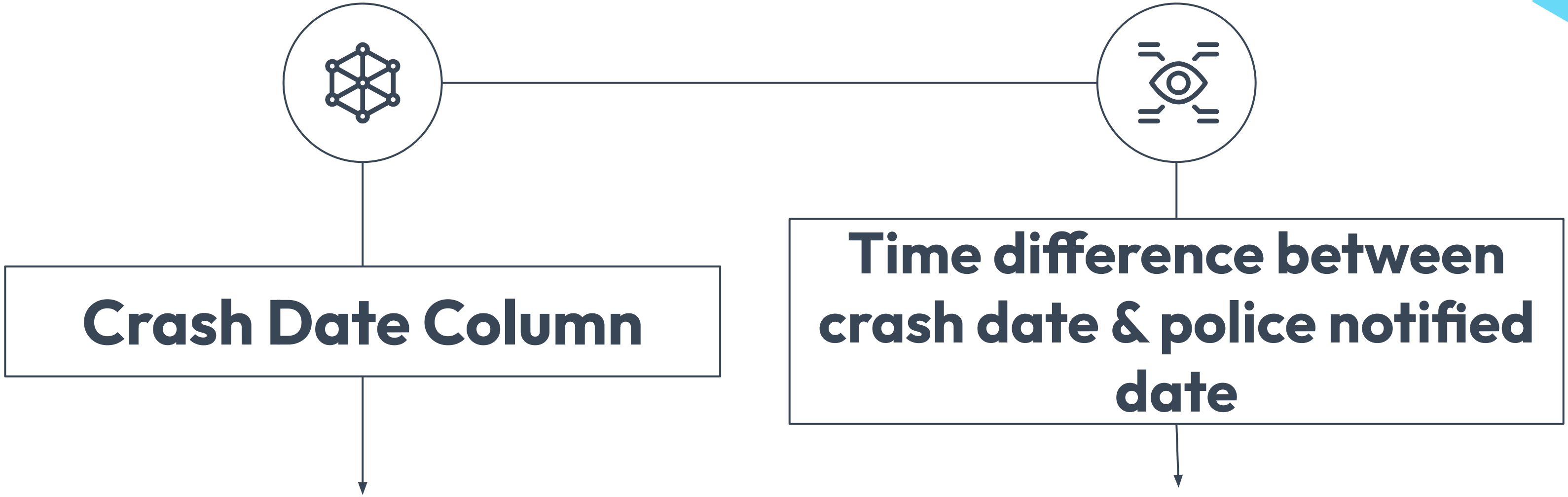
## Data Inconsistency

- **RD_NO:** Two entries began with lower-case letters, differently from the usual format (*hz273623* and *hz125235*).
- **NUM_UNITS & INJURIES TOTAL**: Convert from float to integer for consistency.

## Missing values

- **Columns with "Unknown" values**: Fill missing observations with existing "*Unknown*" equivalents.
- **LATITUDE, LONGITUDE, POINT:** Use API and STREET_NAME & STREET_NO to fill values.
- **BEAT_OF_OCCURRENCE:** Filled using discovered values from the previous 3 columns.

# Data Transformation



## Crash Date Column

Split into four separate columns: DAY, MONTH, YEAR, and TIME to improve granularity and analysis flexibility.

## Time difference between crash date & police notified date

New derived column called "*DELTA_TIME_CRASH_DATE_POLICE_REPORT_DATE*". This feature is expressed in the format "days minutes seconds."

# 02

# Data Warehouse Schema

# Fact Table

## Granularity

Each entry corresponds to a reimbursement  transaction for damages caused by vehicle crashes.

## Composite Primary Key

Comprised of three foreign keys Crash ID, Person ID, and Vehicle ID linking the fact table to the respective dimensions.

## Measures

‘Cost’ allows for aggregation and analysis of total reimbursement amounts. It is an additive measure.

## Attributes

‘Cost_Category’ classifies the reimbursement based on its monetary value.

# Dimensions

## Crash

Contains information about crashes. Normalized into three subdimensions: *CrashLocation*, *CrashCondition*, *Injury*

## DateTime

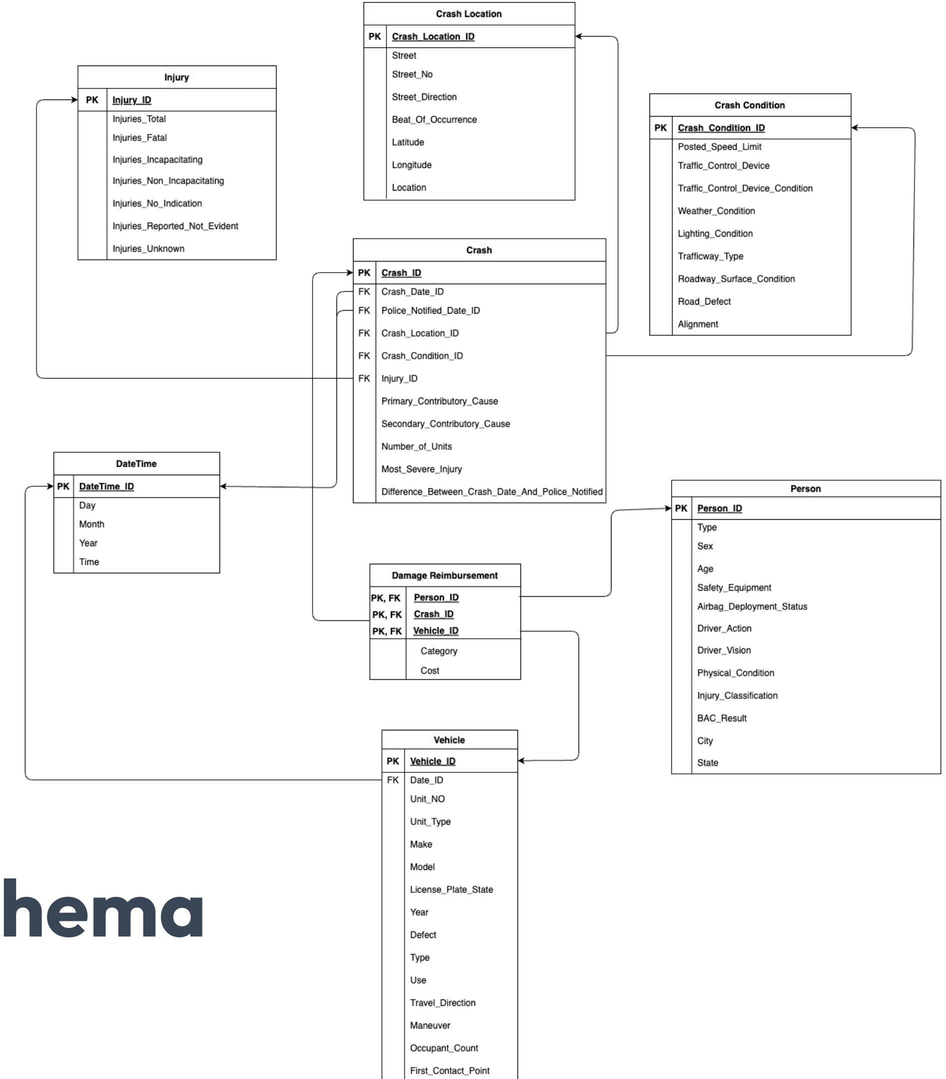
Contains temporal data. Shared dimension between Crash and Vehicle

## Person

Contains information about people in crashes.

## Vehicle

Contains information about vehicles in crashes.



| Crash Location | |
|---|---|
| PK | Crash_Location_ID |
| | Street |
| | Street_No |
| | Street_Direction |
| | Beat_Of_Occurrence |
| | Latitude |
| | Longitude |
| | Location |

| Injury | |
|---|---|
| PK | Injury_ID |
| | Injuries_Total |
| | Injuries_Fatal |
| | Injuries_Incapacitating |
| | Injuries_Non_Incapacitating |
| | Injuries_No_Indication |
| | Injuries_Reported_Not_Evident |
| | Injuries_Unknown |

| Crash Condition | |
|---|---|
| PK | Crash_Condition_ID |
| | Posted_Speed_Limit |
| | Traffic_Control_Device |
| | Traffic_Control_Device_Condition |
| | Weather_Condition |
| | Lighting_Condition |
| | Trafficway_Type |
| | Roadway_Surface_Condition |
| | Road_Defect |
| | Alignment |

| Crash | |
|---|---|
| PK | Crash_ID |
| FK | Crash_Date_ID |
| FK | Police_Notified_Date_ID |
| FK | Crash_Location_ID |
| FK | Crash_Condition_ID |
| FK | Injury_ID |
| | Primary_Contributory_Cause |
| | Secondary_Contributory_Cause |
| | Number_of_Units |
| | Most_Severe_Injury |
| | Difference_Between_Crash_Date_And_Police_Notified |

| DateTime | |
|---|---|
| PK | DateTime_ID |
| | Day |
| | Month |
| | Year |
| | Time |

| Person | |
|---|---|
| PK | Person_ID |
| | Type |
| | Sex |
| | Age |
| | Safety_Equipment |
| | Airbag_Deployment_Status |
| | Driver_Action |
| | Driver_Vision |
| | Physical_Condition |
| | Injury_Classification |
| | BAC_Result |
| | City |
| | State |

| Damage Reimbursement | |
|---|---|
| PK, FK | Person_ID |
| PK, FK | Crash_ID |
| PK, FK | Vehicle_ID |
| | Category |
| | Cost |

| Vehicle | |
|---|---|
| PK | Vehicle_ID |
| FK | Date_ID |
| | Unit_NO |
| | Unit_Type |
| | Make |
| | Model |
| | License_Plate_State |
| | Year |
| | Defect |
| | Type |
| | Use |
| | Travel_Direction |
| | Maneuver |
| | Occupant_Count |
| | First_Contact_Point |

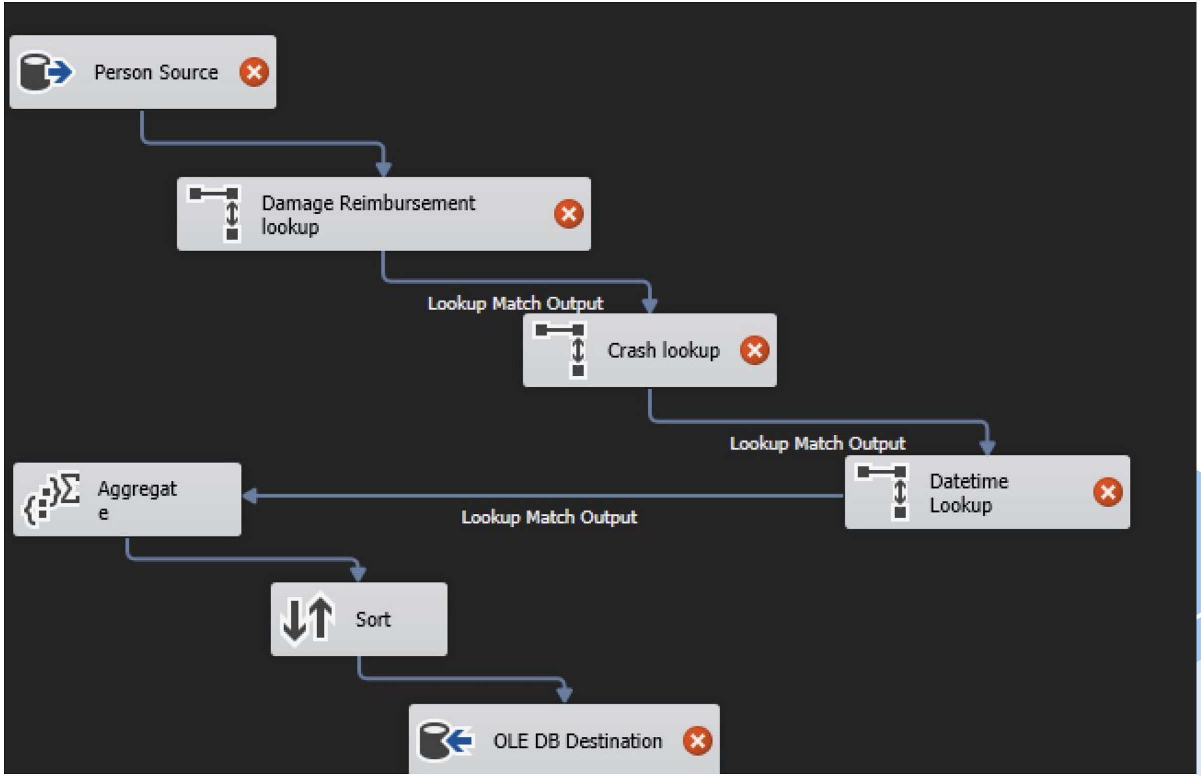
# Snowflake Schema

# 03

# SSIS Queries

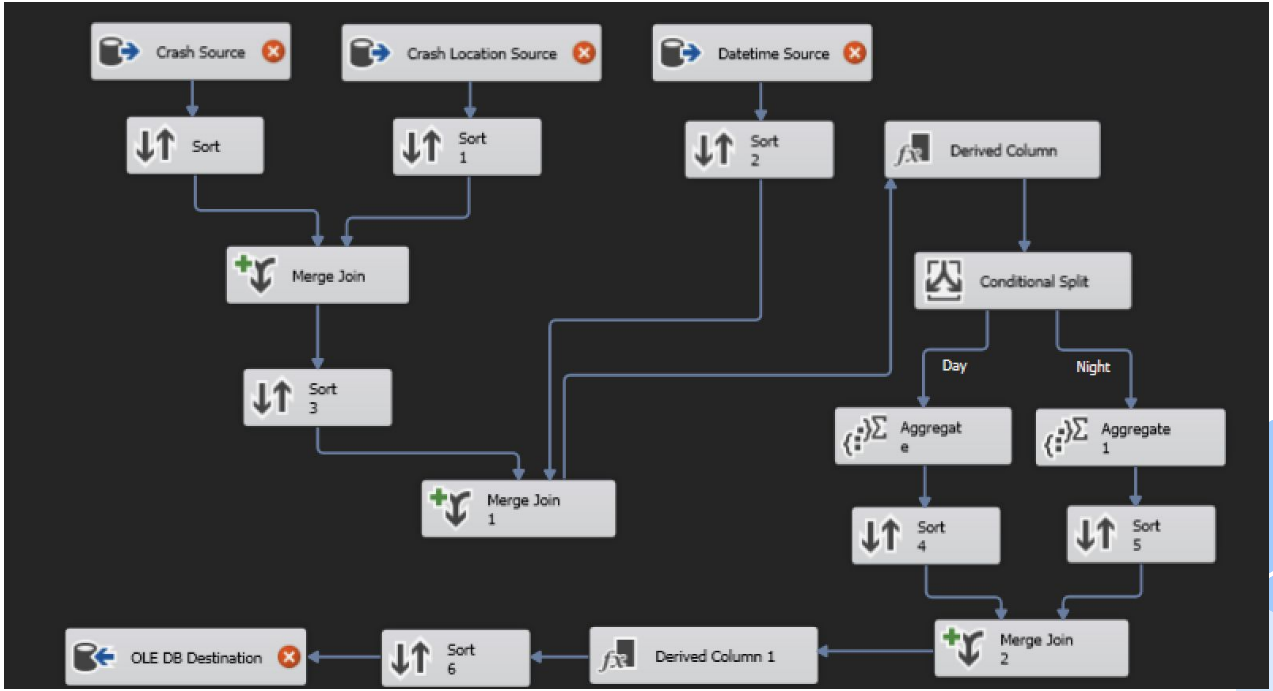
# Query 6a

1. Start with  Person table, extracting key demographic details such as ID, Type, Sex, and Age.
2. Link data to the DamageReimbursement table to retrieve crash IDs.
3. Link data tothe Crash table to obtain crash dates.
4. Link data to the DateTime table to extract the year of the crash.
5. Aggregate data by counting the number of crashes for each participant per year, generating a “Total Crashes” metric.
6. Sort results by year and total crashes.
7. Store in the “Query 6 result” table.



# Query 7a

1. Extract relevant columns from the Crash(Number of Units), CrashLocation(Beat of occurrence), and DateTime tables, joining them sequentially on their IDs.
2. Create a Time Category field ("Day" or "Night") based on timestamps.
3. Split by Time Category, aggregated by Beat of occurrence, and summed for vehicle counts (Number of Units Day and Night)
4. Final ratio calculation is performed using a derived column transformation that handles potential division by zero cases, setting the index to 0 when no daytime crashes exist.
5. Store in the "Query 7 result" table.



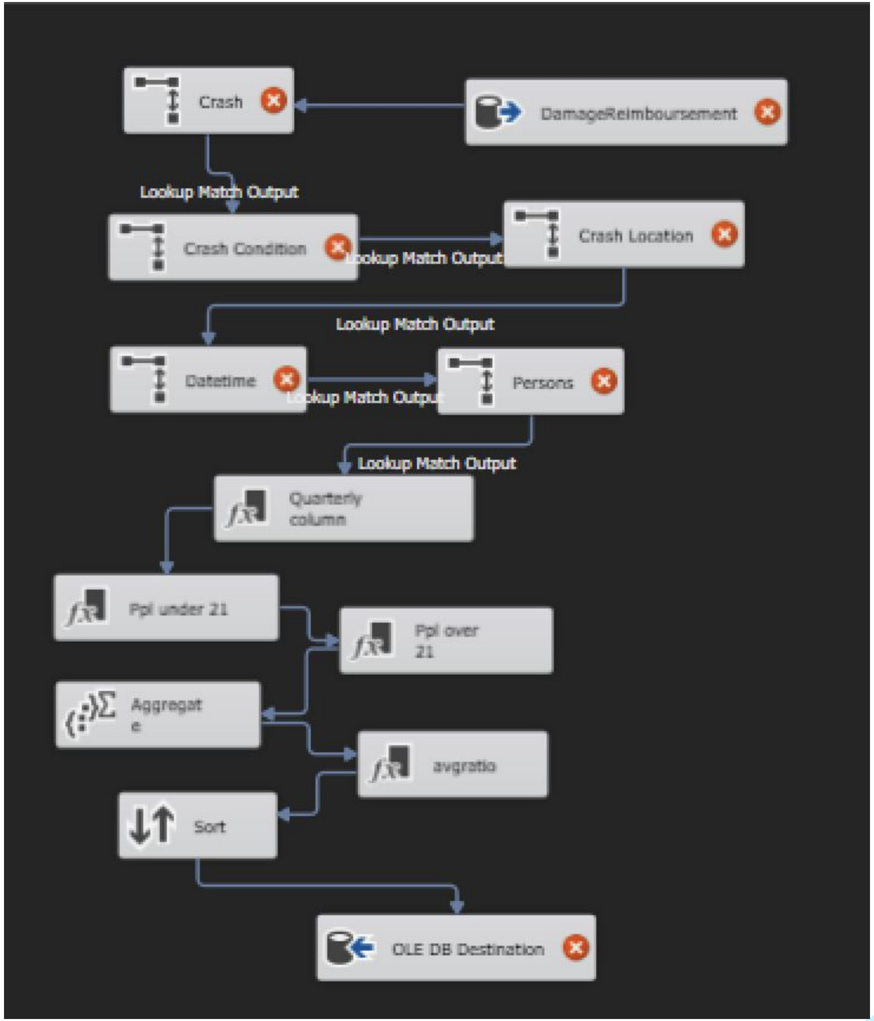
# Query 8a

1. Join the DamageReimbursement, Crash, CrashLocation, CrashCondition, DateTime, and Person tables using lookup.

2. Counting individuals over and under 21, we aggregate by the required columns

3. Create new column containing the ratio.

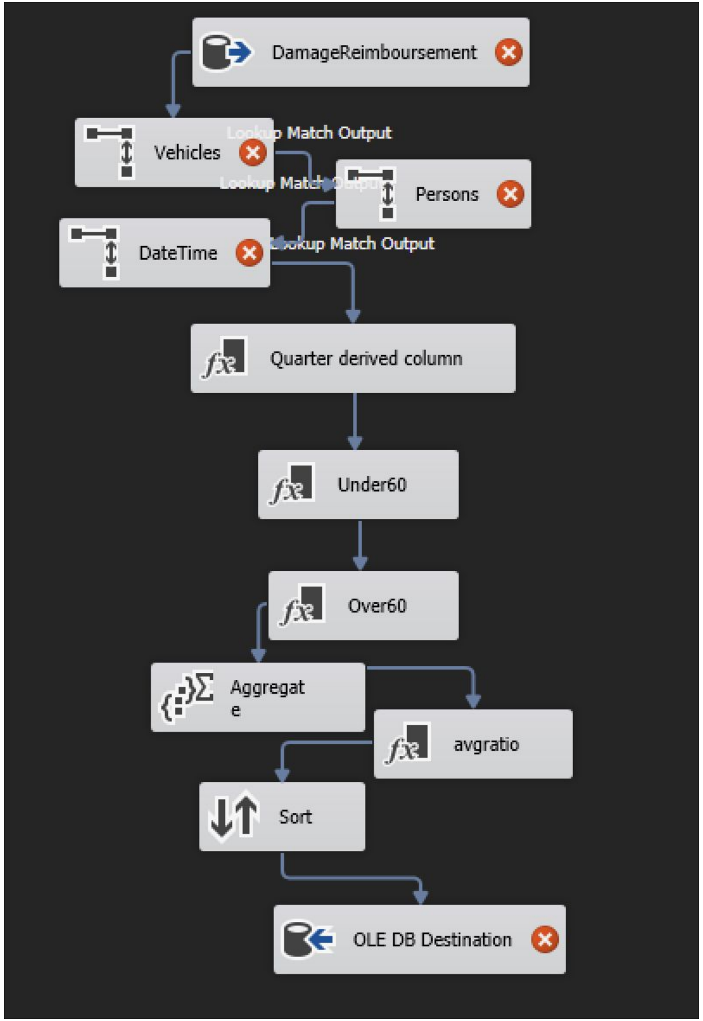
4. Store in the “Query 6 result” table.



# Query 9a

**Query to examine the ratio of older individuals (over 60) involved in interstate movements in Chicago.**

1. Use lookup to join relevant tables together.
2. Classifying individuals as over or under 60.
3. Aggregate the data by quarter and license plate registration.
4. Ratio is calculated, sorted using a sorting operator.
5. Store in the "Query 9 result" table.

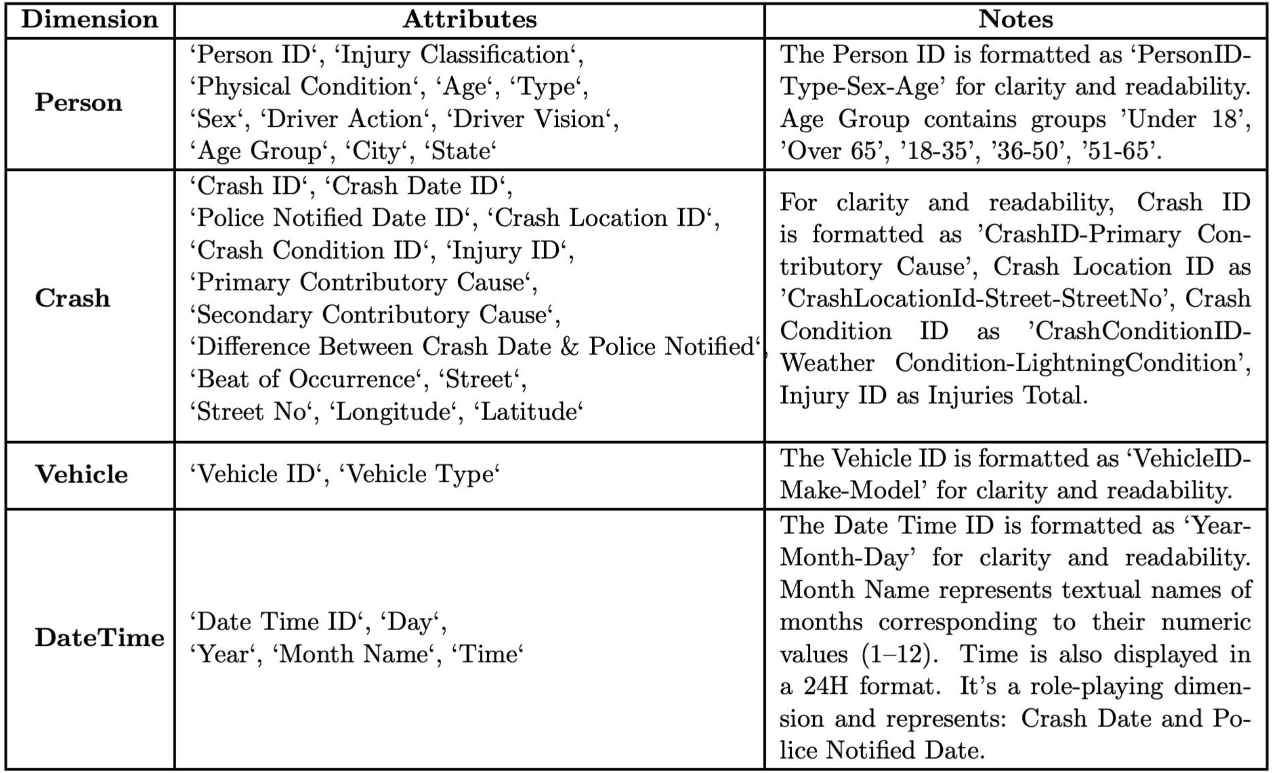


# 04

# OLAP Cube

# Structure

The data cube is organized into dimensions and measure groups.



| Dimension | Attributes | Notes |
|---|---|---|
| **Person** | ‘Person ID‘, ‘Injury Classification‘, ‘Physical Condition‘, ‘Age‘, ‘Type‘, ‘Sex‘, ‘Driver Action‘, ‘Driver Vision‘, ‘Age Group‘, ‘City‘, ‘State‘ | The Person ID is formatted as ‘PersonID-Type-Sex-Age’ for clarity and readability. Age Group contains groups ’Under 18’, ’Over 65’, ’18-35’, ’36-50’, ’51-65’. |
| **Crash** | ‘Crash ID‘, ‘Crash Date ID‘, ‘Police Notified Date ID‘, ‘Crash Location ID‘, ‘Crash Condition ID‘, ‘Injury ID‘, ‘Primary Contributory Cause‘, ‘Secondary Contributory Cause‘, ‘Difference Between Crash Date & Police Notified‘, ‘Beat of Occurrence‘, ‘Street‘, ‘Street No‘, ‘Longitude‘, ‘Latitude‘ | For clarity and readability, Crash ID is formatted as ’CrashID-Primary Contributory Cause’, Crash Location ID as ’CrashLocationId-Street-StreetNo’, Crash Condition ID as ’CrashConditionID-Weather Condition-LightningCondition’, Injury ID as Injuries Total. |
| **Vehicle** | ‘Vehicle ID‘, ‘Vehicle Type‘ | The Vehicle ID is formatted as ‘VehicleID-Make-Model’ for clarity and readability. |
| **DateTime** | ‘Date Time ID‘, ‘Day‘, ‘Year‘, ‘Month Name‘, ‘Time‘ | The Date Time ID is formatted as ‘Year-Month-Day’ for clarity and readability. Month Name represents textual names of months corresponding to their numeric values (1–12). Time is also displayed in a 24H format. It’s a role-playing dimension and represents: Crash Date and Police Notified Date. |

# Measure Groups

The measure groups include Damage Reimbursement and Person.

❑ Damage Reimbursement captures financial data, including Cost, Damage Reimbursement Count, and Average Cost, which are essential for assessing the financial impact of crashes.

❑ The Person measure group provides insights into the number and type of individuals involved in crashes, including Person Count, Count of Person Type, and Fatal Crashes.

# Hierarchies

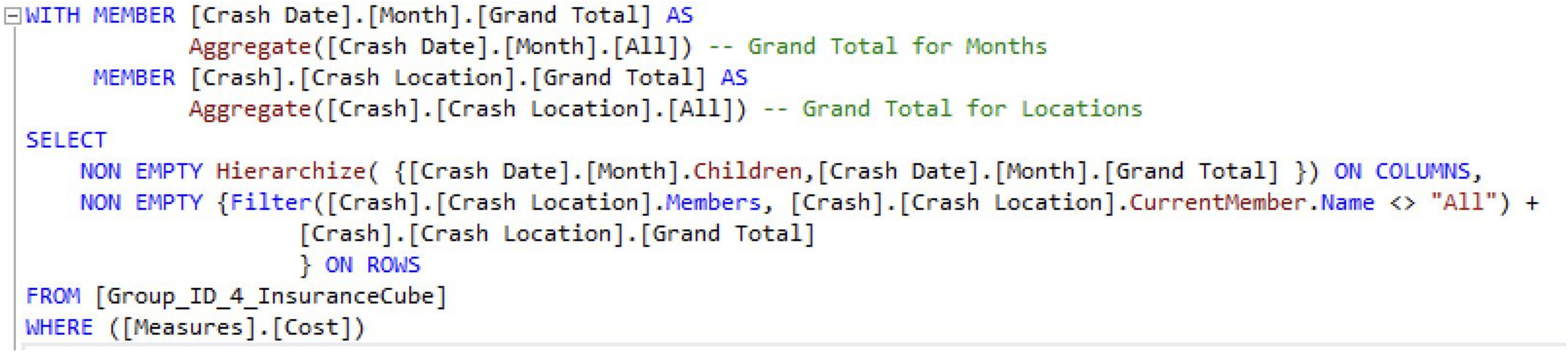
The DateTime dimension includes a hierarchy on the date which is specified as
Year → Month → Day → Time.

# 05

# MDX Queries

# Query 2

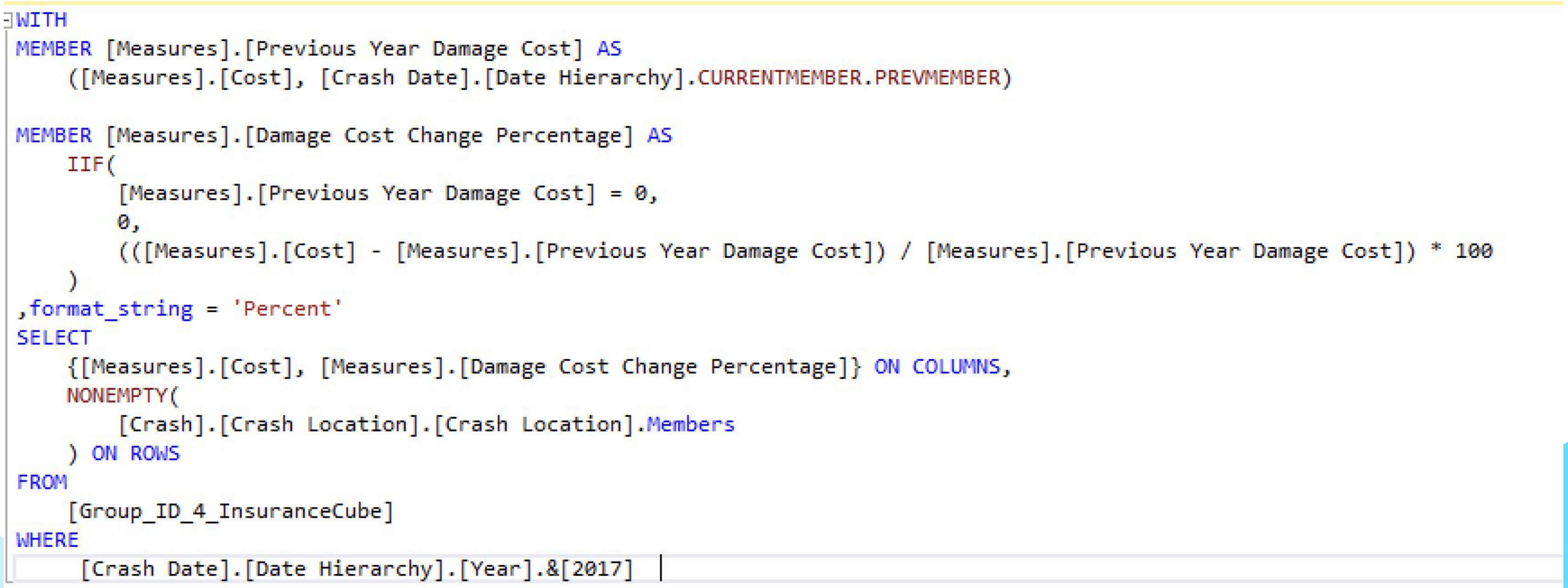
Show the total damage costs for each location and each month, as well as the total.



```
WITH MEMBER [Crash Date].[Month].[Grand Total] AS
            Aggregate([Crash Date].[Month].[All]) -- Grand Total for Months
     MEMBER [Crash].[Crash Location].[Grand Total] AS
            Aggregate([Crash].[Crash Location].[All]) -- Grand Total for Locations
SELECT
    NON EMPTY Hierarchize( {[Crash Date].[Month].Children,[Crash Date].[Month].[Grand Total] }) ON COLUMNS,
    NON EMPTY {Filter([Crash].[Crash Location].Members, [Crash].[Crash Location].CurrentMember.Name <> "All") +
                    [Crash].[Crash Location].[Grand Total]
                    } ON ROWS
FROM [Group_ID_4_InsuranceCube]
WHERE ([Measures].[Cost])
```

# Query 4

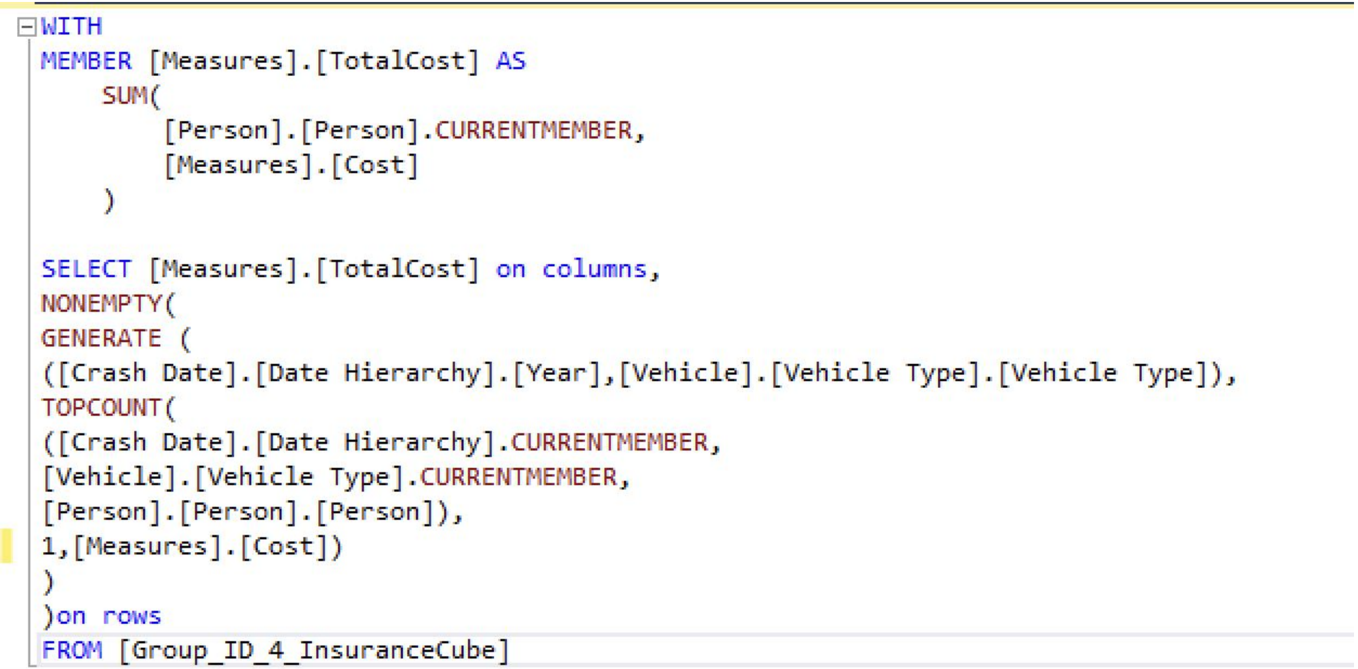
Show for every location, damage costs increase or decrease, percentage-wise, relative to the previous year.



```
WITH
MEMBER [Measures].[Previous Year Damage Cost] AS
    ([Measures].[Cost], [Crash Date].[Date Hierarchy].CURRENTMEMBER.PREVMEMBER)

MEMBER [Measures].[Damage Cost Change Percentage] AS
    IIF(
        [Measures].[Previous Year Damage Cost] = 0,
        0,
        (([Measures].[Cost] - [Measures].[Previous Year Damage Cost]) / [Measures].[Previous Year Damage Cost]) * 100
    )
,format_string = 'Percent'
SELECT
    {[Measures].[Cost], [Measures].[Damage Cost Change Percentage]} ON COLUMNS,
    NONEMPTY(
        [Crash].[Crash Location].[Crash Location].Members
    ) ON ROWS
FROM
    [Group_ID_4_InsuranceCube]
WHERE
    [Crash Date].[Date Hierarchy].[Year].&[2017]
```

# Query 6

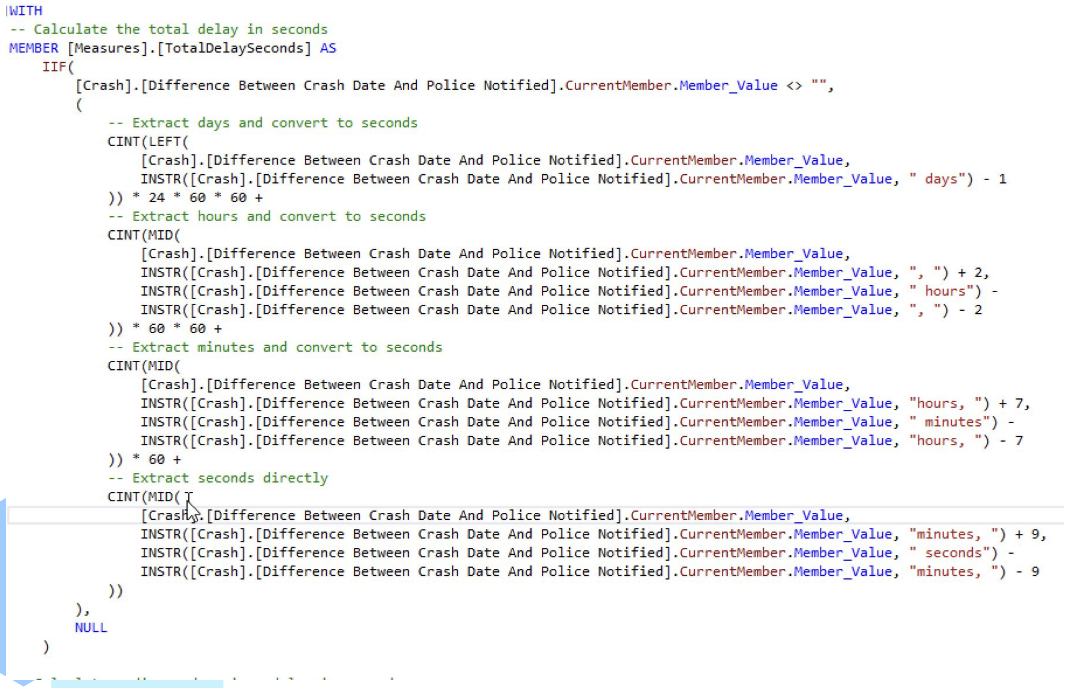
Show for each vehicle type and each year, show the information and the (total) damage costs of the person with the highest reported damage.



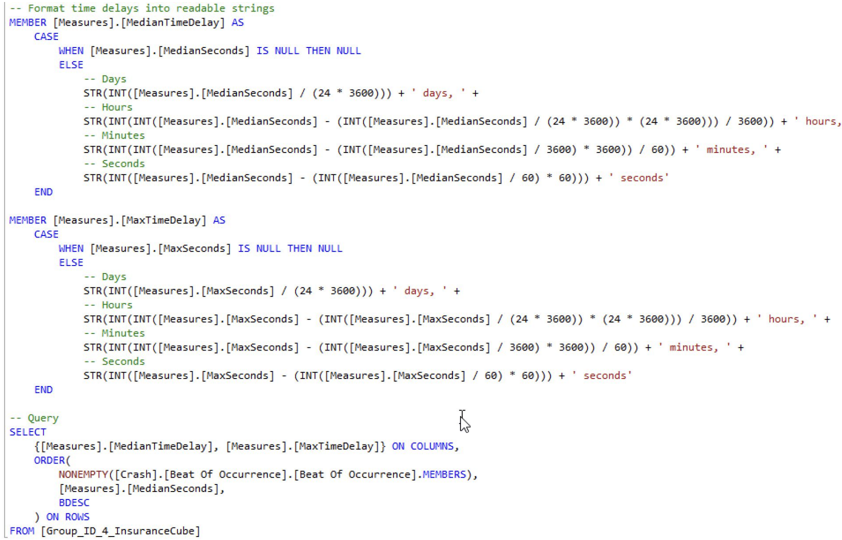
```
WITH
MEMBER [Measures].[TotalCost] AS
    SUM(
        [Person].[Person].CURRENTMEMBER,
        [Measures].[Cost]
    )

SELECT [Measures].[TotalCost] on columns,
NONEMPTY(
GENERATE (
([Crash Date].[Date Hierarchy].[Year],[Vehicle].[Vehicle Type].[Vehicle Type]),
TOPCOUNT(
([Crash Date].[Date Hierarchy].CURRENTMEMBER,
[Vehicle].[Vehicle Type].CURRENTMEMBER,
[Person].[Person].[Person]),
1,[Measures].[Cost])
)
)on rows
FROM [Group_ID_4_InsuranceCube]
```

# Query 7

Calculate and display the median and maximum time delays (delta time) between when crashes occurred and were reported for each beat of occurrence.



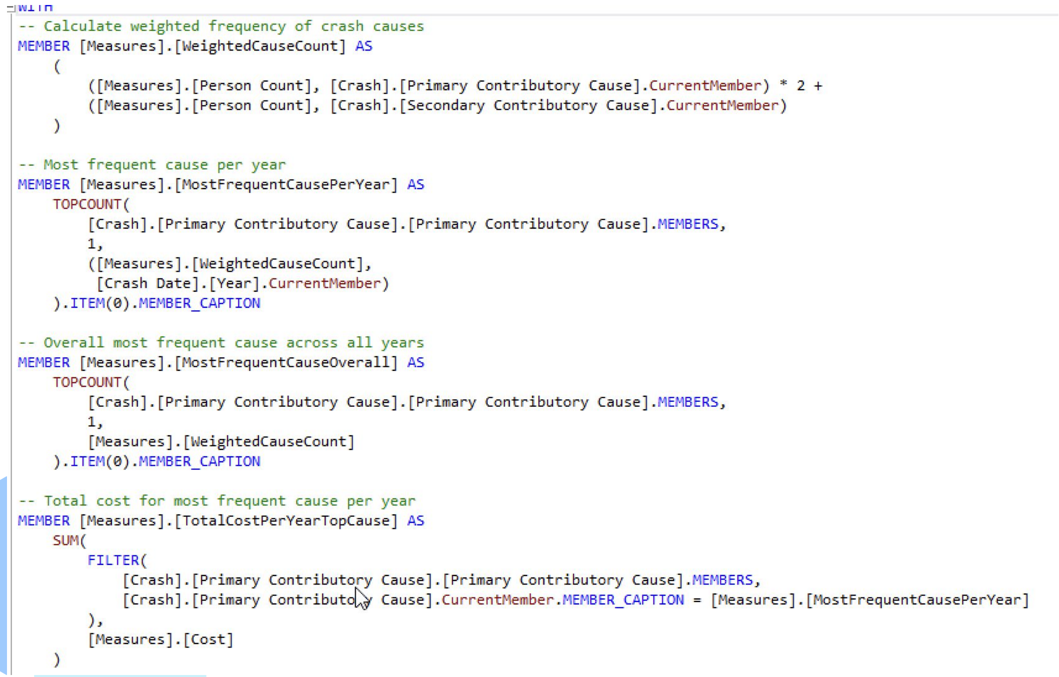
```
WITH
-- Calculate the total delay in seconds
MEMBER [Measures].[TotalDelaySeconds] AS
    IIF(
        [Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value <> "",
        (
            -- Extract days and convert to seconds
            CINT(LEFT(
                [Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, " days") - 1
            )) * 24 * 60 * 60 +
            -- Extract hours and convert to seconds
            CINT(MID(
                [Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, ", ") + 2,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, " hours") -
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, ", ") - 2
            )) * 60 * 60 +
            -- Extract minutes and convert to seconds
            CINT(MID(
                [Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, "hours, ") + 7,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, " minutes") -
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, "hours, ") - 7
            )) * 60 +
            -- Extract seconds directly
            CINT(MID(
                [Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, "minutes, ") + 9,
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, " seconds") -
                INSTR([Crash].[Difference Between Crash Date And Police Notified].CurrentMember.Member_Value, "minutes, ") - 9
            ))
        ),
        NULL
    )
```



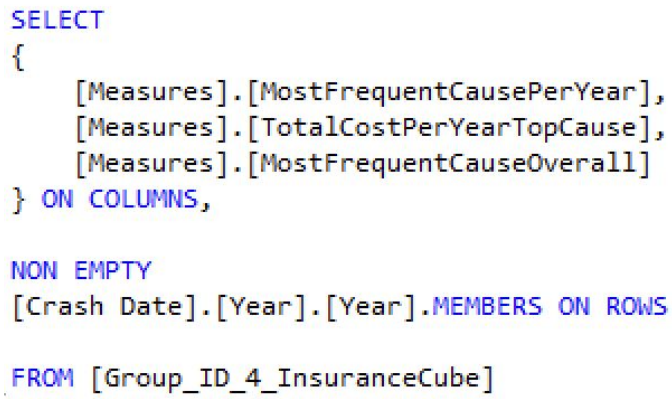
```
-- Format time delays into readable strings
MEMBER [Measures].[MedianTimeDelay] AS
    CASE
        WHEN [Measures].[MedianSeconds] IS NULL THEN NULL
        ELSE
            -- Days
            STR(INT([Measures].[MedianSeconds] / (24 * 3600))) + ' days, ' +
            -- Hours
            STR(INT(INT([Measures].[MedianSeconds] - (INT([Measures].[MedianSeconds] / (24 * 3600)) * (24 * 3600))) / 3600)) + ' hours,
            -- Minutes
            STR(INT(INT([Measures].[MedianSeconds] - (INT([Measures].[MedianSeconds] / 3600) * 3600)) / 60)) + ' minutes, ' +
            -- Seconds
            STR(INT([Measures].[MedianSeconds] - (INT([Measures].[MedianSeconds] / 60) * 60))) + ' seconds'
    END

MEMBER [Measures].[MaxTimeDelay] AS
    CASE
        WHEN [Measures].[MaxSeconds] IS NULL THEN NULL
        ELSE
            -- Days
            STR(INT([Measures].[MaxSeconds] / (24 * 3600))) + ' days, ' +
            -- Hours
            STR(INT(INT([Measures].[MaxSeconds] - (INT([Measures].[MaxSeconds] / (24 * 3600)) * (24 * 3600))) / 3600)) + ' hours, ' +
            -- Minutes
            STR(INT(INT([Measures].[MaxSeconds] - (INT([Measures].[MaxSeconds] / 3600) * 3600)) / 60)) + ' minutes, ' +
            -- Seconds
            STR(INT([Measures].[MaxSeconds] - (INT([Measures].[MaxSeconds] / 60) * 60))) + ' seconds'
    END

-- Query
SELECT
    {[Measures].[MedianTimeDelay], [Measures].[MaxTimeDelay]} ON COLUMNS,
    ORDER(
        NONEMPTY([Crash].[Beat Of Occurrence].[Beat Of Occurrence].MEMBERS),
        [Measures].[MedianSeconds],
        BDESC
    ) ON ROWS
FROM [Group_ID_4_InsuranceCube]
```

# Query 8a

Identify the most frequent cause of crashes for each year, calculate the associated total damage costs, and determine the overall most frequent crash cause across all years.



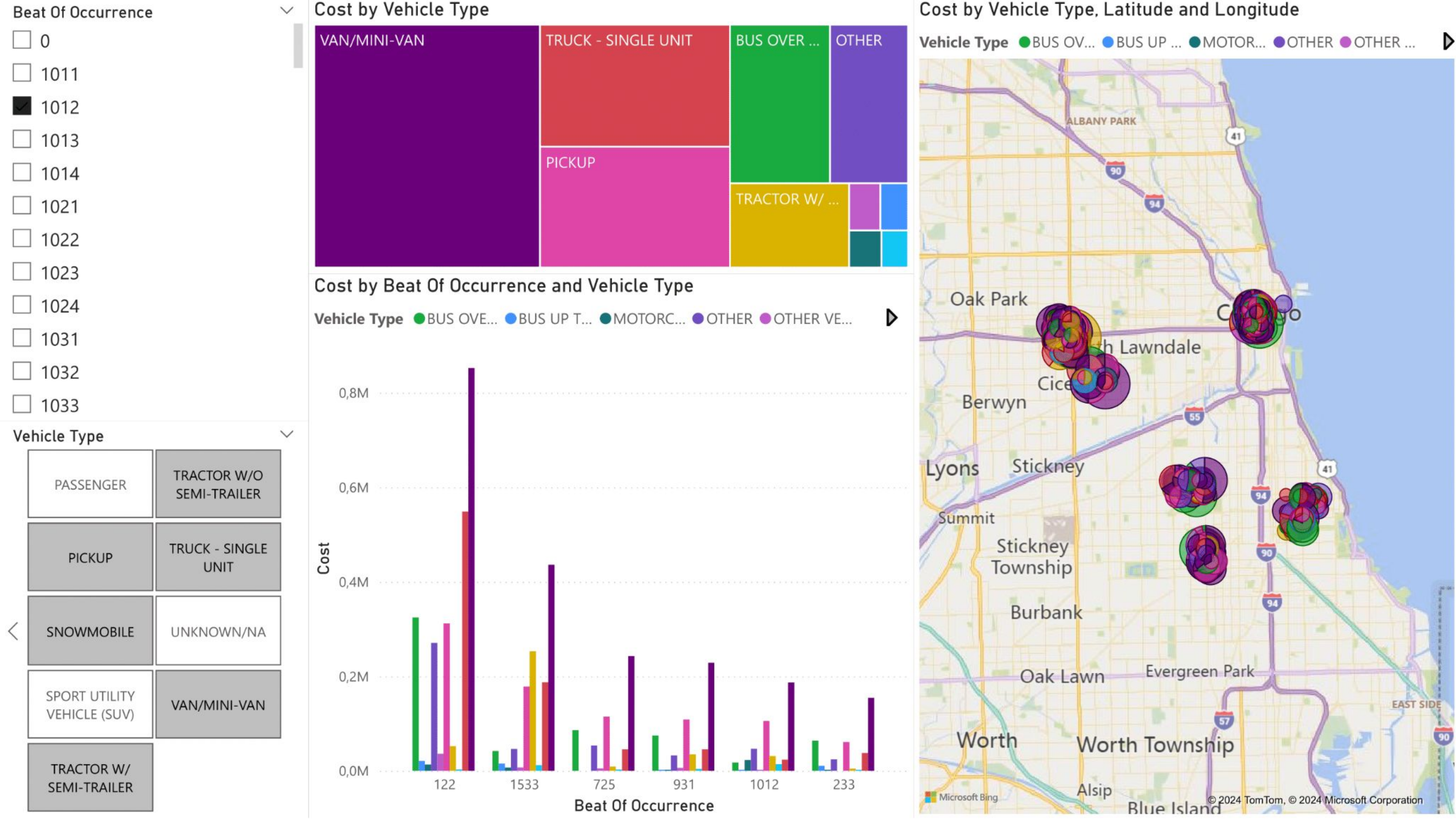
```
WITH
-- Calculate weighted frequency of crash causes
MEMBER [Measures].[WeightedCauseCount] AS
    (
        ([Measures].[Person Count], [Crash].[Primary Contributory Cause].CurrentMember) * 2 +
        ([Measures].[Person Count], [Crash].[Secondary Contributory Cause].CurrentMember)
    )

-- Most frequent cause per year
MEMBER [Measures].[MostFrequentCausePerYear] AS
    TOPCOUNT(
        [Crash].[Primary Contributory Cause].[Primary Contributory Cause].MEMBERS,
        1,
        ([Measures].[WeightedCauseCount],
         [Crash Date].[Year].CurrentMember)
    ).ITEM(0).MEMBER_CAPTION

-- Overall most frequent cause across all years
MEMBER [Measures].[MostFrequentCauseOverall] AS
    TOPCOUNT(
        [Crash].[Primary Contributory Cause].[Primary Contributory Cause].MEMBERS,
        1,
        [Measures].[WeightedCauseCount]
    ).ITEM(0).MEMBER_CAPTION

-- Total cost for most frequent cause per year
MEMBER [Measures].[TotalCostPerYearTopCause] AS
    SUM(
        FILTER(
            [Crash].[Primary Contributory Cause].[Primary Contributory Cause].MEMBERS,
            [Crash].[Primary Contributory Cause].CurrentMember.MEMBER_CAPTION = [Measures].[MostFrequentCausePerYear]
        ),
        [Measures].[Cost]
    )
```



```
SELECT
{
    [Measures].[MostFrequentCausePerYear],
    [Measures].[TotalCostPerYearTopCause],
    [Measures].[MostFrequentCauseOverall]
} ON COLUMNS,

NON EMPTY
[Crash Date].[Year].[Year].MEMBERS ON ROWS

FROM [Group_ID_4_InsuranceCube]
```

# 06

# Dashboards

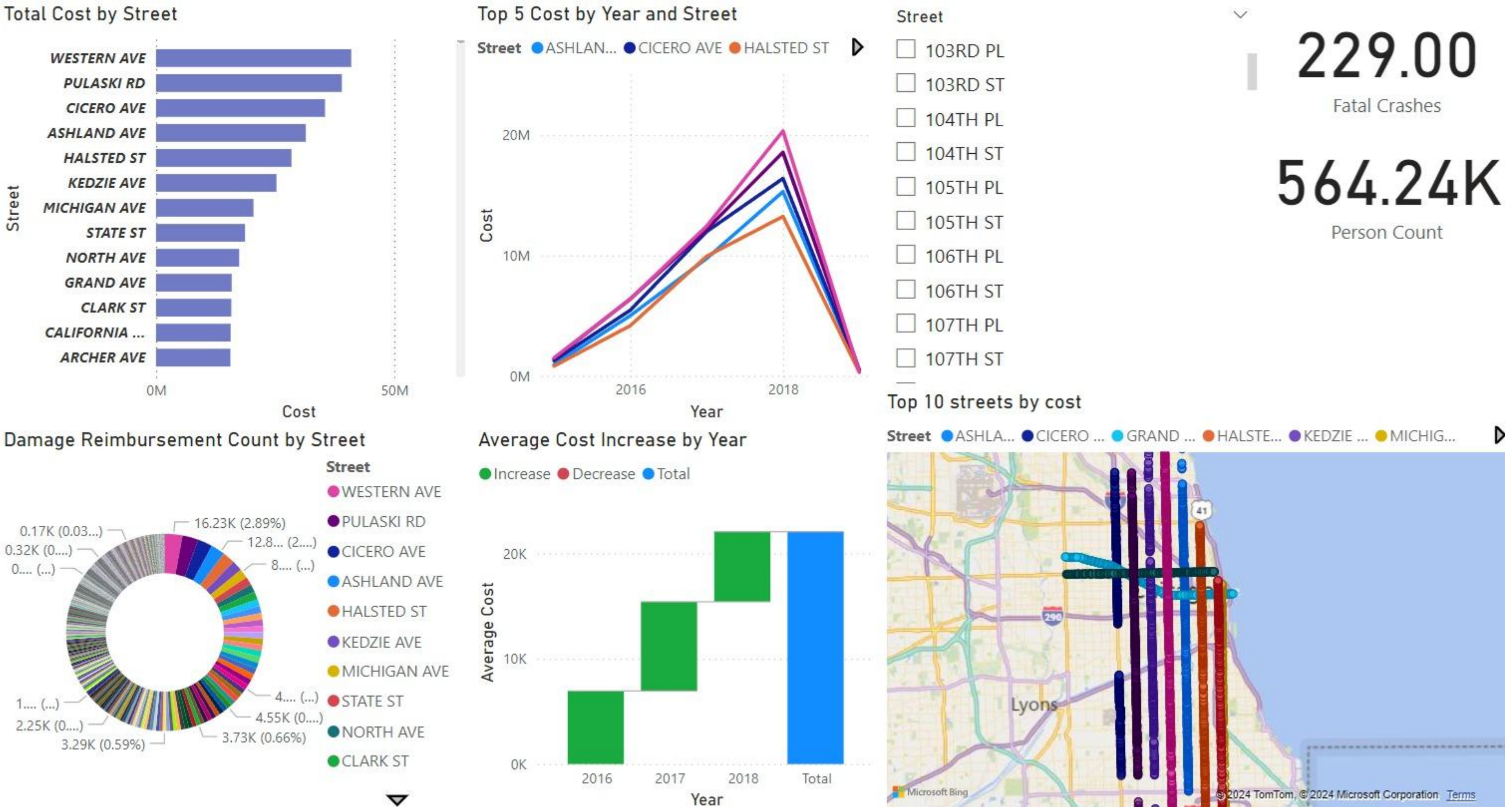


Beat Of Occurrence

- ☐ 0
- ☐ 1011
- ☑ 1012
- ☐ 1013
- ☐ 1014
- ☐ 1021
- ☐ 1022
- ☐ 1023
- ☐ 1024
- ☐ 1031
- ☐ 1032
- ☐ 1033

Vehicle Type

- PASSENGER
- TRACTOR W/O SEMI-TRAILER
- PICKUP
- TRUCK - SINGLE UNIT
- SNOWMOBILE
- UNKNOWN/NA
- SPORT UTILITY VEHICLE (SUV)
- VAN/MINI-VAN
- TRACTOR W/ SEMI-TRAILER



Total Cost by Street

Top 5 Cost by Year and Street

Street

□ 103RD PL
□ 103RD ST
□ 104TH PL
□ 104TH ST
□ 105TH PL
□ 105TH ST
□ 106TH PL
□ 106TH ST
□ 107TH PL
□ 107TH ST

229.00

Fatal Crashes

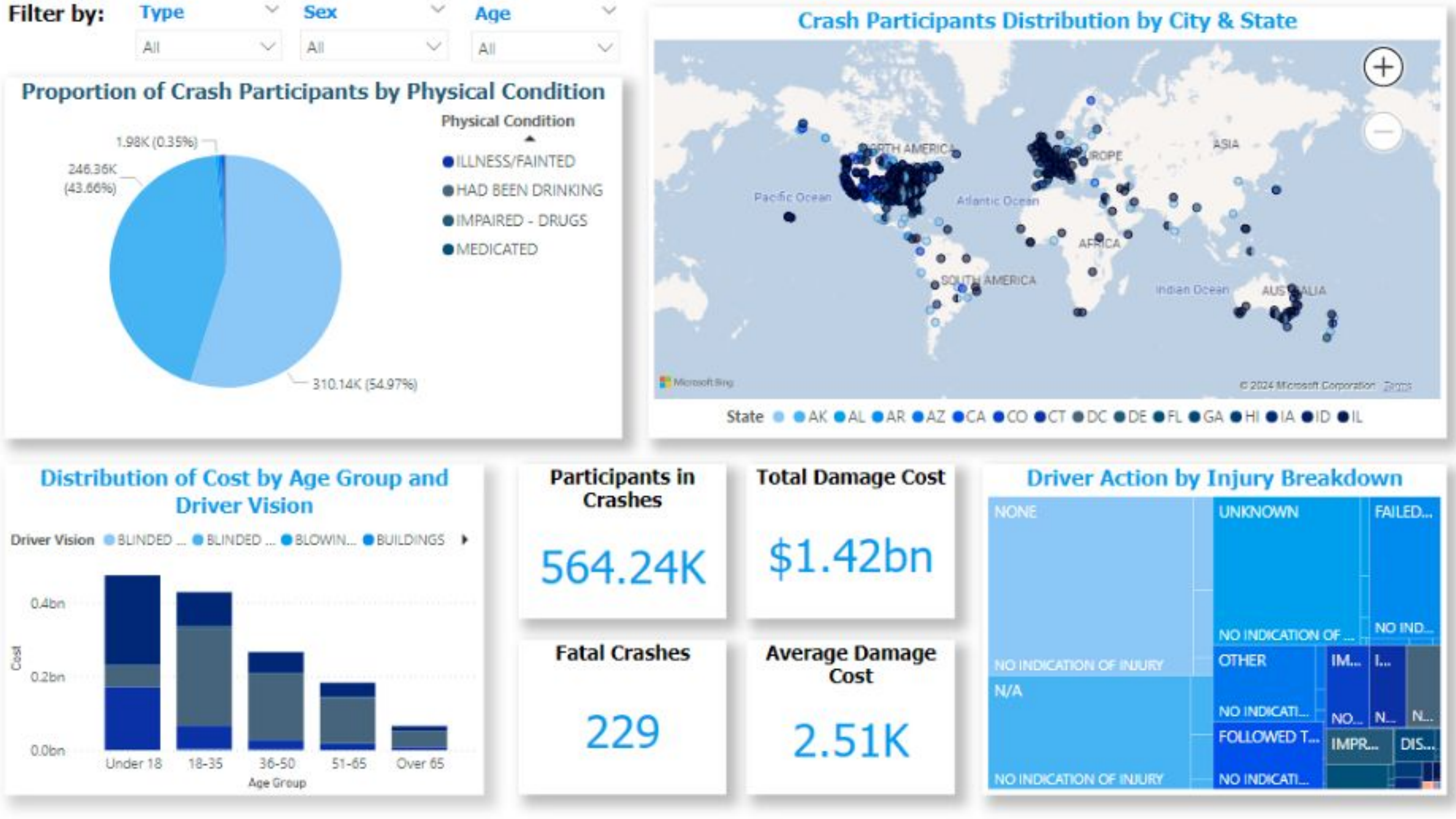
564.24K

Person Count

Damage Reimbursement Count by Street

Average Cost Increase by Year

Top 10 streets by cost



Filter by: Type All | Sex All | Age All

## Proportion of Crash Participants by Physical Condition

## Crash Participants Distribution by City & State

## Distribution of Cost by Age Group and Driver Vision

Participants in Crashes

564.24K

Total Damage Cost

$1.42bn

Fatal Crashes

229

Average Damage Cost

2.51K

## Driver Action by Injury Breakdown

# Thank you for the attention!