
# UNIVERSITÀ DI PISA

Department of Computer Science
Data Science and Business Informatics

Laboratory Of Data Science Project

## Data traffic incidents in Chicago

Thomas Gonzo
Davide Rizzello
Minh Duc Pham

---

Academic Year 2024/2025

# Contents

# 1 Introduction

In this report, we are going to describe the various pre-processing and ETL steps we performed on a given dataset regarding vehicle crashes in Chicago. We will then answer some business questions with a set of multiple tools. The aim of the first step is to create detailed tables to synthesize a Data Warehouse into SQL Server Management Studio so that it could be queried for business purposes.

The data was provided through three different CSV files: *crashes.csv*, *vehicles.csv*, and *people.csv*.

A preliminary Data Exploration task has been performed to check for incongruities. Two dimensions have been considered:

1. Missing Values
2. Data Integrity

## 1.1 Creating the Data Warehouse Star schema

Designing a data warehouse schema is a complex task that requires structuring data to facilitate decision-making regarding key business processes. To guide the design, specific business questions were formulated, helping to identify the proper naming and placement of attributes within each dimension.

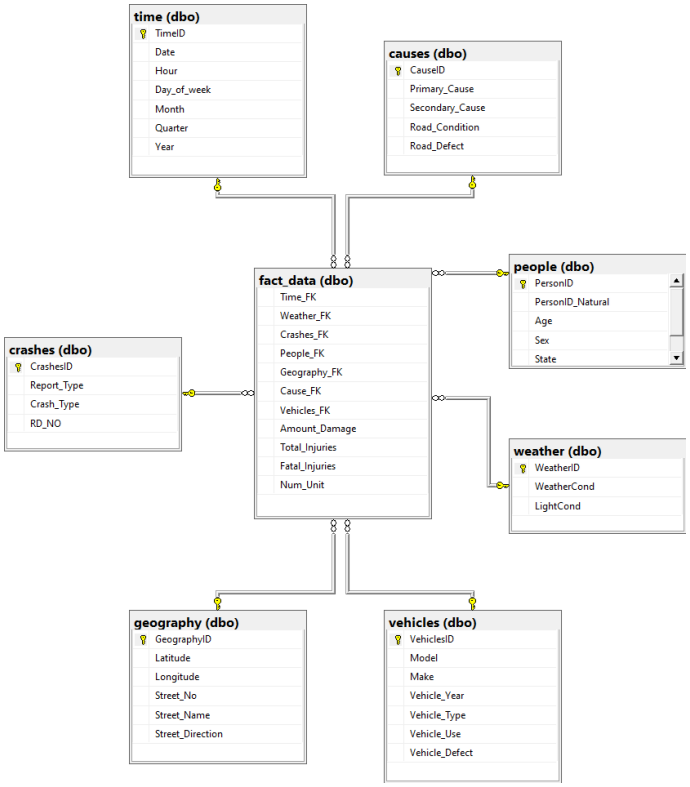


Figure 1: The datawarehouse schema

Only attributes deemed relevant for the analysis were included in the schema. Attributes considered irrelevant or exhibiting low variability across all records were excluded,

as they were not informative. Particular attention was paid to selecting appropriate measures for the fact table. The grain was meticulously defined to represent a single row for each person, per crash, per vehicle.

## 1.2 Data Preprocessing

Significant effort was dedicated to the cleaning phase, carefully addressing each instance of missing values based on the type of attribute involved.

The methods used for handling missing values in this phase were as follows:

- **Filling Based on the Distribution of Non-Missing Values:** The distributions of non-missing values were calculated, and missing values were generated to reflect these distributions. This approach was applied to the attributes *AGE* and *SEX*.
- **Filling Geographic Attributes Using Coordinates:** For records with missing values in *LATITUDE*, *LONGITUDE*, or *LOCATION*, the coordinates were calculated using the *GoogleV3* and *geopy* libraries, with *STREET_NAME* as the input. Conversely, records with missing *STREET_NAME* values were filled using these libraries with *LATITUDE*, *LONGITUDE*, and *LOCATION* as inputs.
- **Filling with the Mode or *UNKNOWN*:** Attributes were categorized based on whether *UNKNOWN* was a valid value. For each record with missing values, either *UNKNOWN* or the mode of the non-missing values was used to fill the gaps, depending on the specific attribute.
- **Refining the *DAMAGE* Category:** For the missing values in the *DAMAGE* attribute, a distinct approach was employed. First, the median value was calculated for each of the three damage categories. Then, for each record, the corresponding damage category was identified, and the median value of that category was imputed into the *DAMAGE* attribute.

The outcome of these operations is the partial filling of each row in each file. This filling is incomplete because no solution was found to address the missing values in the *VEHICLE_ID* attribute. Consequently, a drastic decision was made to remove all rows from the *People* dataset that had no value in this attribute.

According to our design, each row of the fact table corresponds to a person involved in a specific crash. Consequently, creating the fact table for the data warehouse required joining the three available datasets. This operation was performed sequentially: first, a join was conducted between the *People* and *Vehicles* datasets using the condition *People.VEHICLE_ID = Vehicles.VEHICLE_ID*. Subsequently, the resulting merged dataset (*MergedPV*) was joined with the *Crashes* dataset using the condition *MergedPV.RD_NO = Crashes.RD_NO*.

## 1.3 Uploading tables to the server

In the final stage of the project, the merged dataset was divided into eight separate CSV files: one for each dimension table and one for the fact table. The dimension tables were designed to contain only distinct rows, ensuring no duplicates. A surrogate key was generated for each table as a sequential number starting from 0 up to the total number of rows in the respective table. These surrogate keys were then mapped to the fact table using a lookup function, enabling efficient foreign key references. The final fact table contains only foreign keys linking to the dimension tables and the associated measures.

# 2 SSIS Solutions

In this section, we use SQL Server Integration Services (SSIS) to solve some business questions on the data warehouse. All computations are executed on the client side without the aid of any SQL query.

## 2.1 Assignment 6b

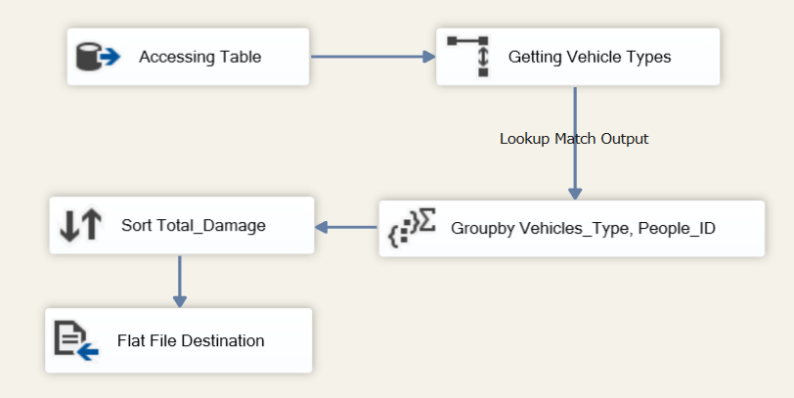


Figure 2: Assignment 6b - Data Flow

The initial Data Flow component is the OLE DB Source, used to import the fact table once again. We selected the columns *People_FK*, *Vehicles_FK*, and *Amount_Damage*. Then, we used the Lookup Transformation to match the *Vehicles_FK* from the fact table with the *Vehicles_ID* in the Vehicles table. In the Vehicles table, we selected the *Vehicle_Type* column to retrieve the vehicle type. After that, we used the Aggregate Transformation to group by *Vehicle_Type* and *People_FK*, and we selected *Amount_Damage* (renamed as *Total_Damage*) with the Sum function to calculate the total damage cost for each vehicle type. Next, we used the Sort Transformation to sort the values of *Total_Damage*. Finally, we used a Flat File Destination to write the output to a CSV file.

## 2.2 Assignment 7b

In the first step, we used the OLE DB Source to extract relevant data from the fact table. The columns used in this step include *Time_FK* and *Amount_Damage*. Then, we used the Lookup Transformation to match *Time_FK* in the fact table with *Time_ID* in the "time" table. We also extracted the columns *Hour*, *Month*, and *Year* in this step. Next, we derived the *Hour* column by categorizing the time ranges. Times from 8 A.M. to 9 P.M. were categorized as "8AM-9PM," and the remaining times were categorized as "9PM-8AM." The new column was renamed *Time_Range*. After this, we used the Aggregate Transformation to group data by *Time_Range*, *Month*, and *Year*, and calculated the total damage costs, renaming the output column as *Total_Damage_By_Range*. At this stage, we split the flow into two branches:

- In the first branch, we grouped the data by *Year* and calculated the average damage cost per year, renaming the new column *AvgDamage_ByYear*. We then sorted this data by *Year* in ascending order.
- In the second branch, we simply sorted the data by *Year* and *Month* in ascending order.

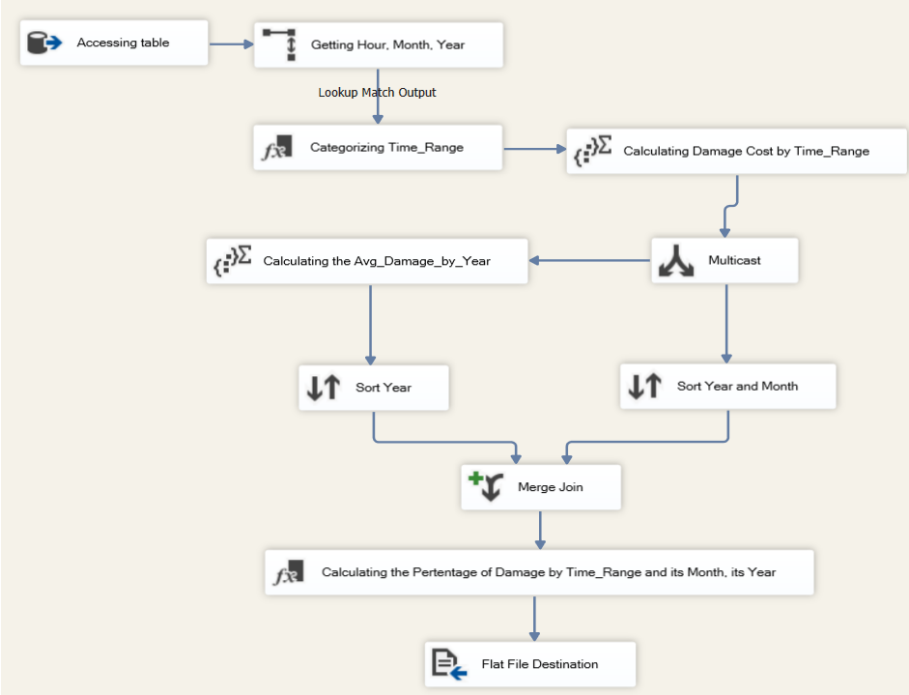


Figure 3: Assignment 7b - Data Flow

The two branches were merged using the Merge Join Transformation, joining on the *Year* column. We selected the columns *Year*, *Month*, *Time_Range*, *Total_Damage_By_Range*, and *AvgDamage_ByYear*. Then, we used the Derived Column Transformation to calculate the percentage of damage by *Time_Range* by dividing *Total_Damage_By_Range* by *AvgDamage_ByYear*. Finally, we used the Flat File Destination to save the output as a CSV.

## 2.3 Assignment 8b

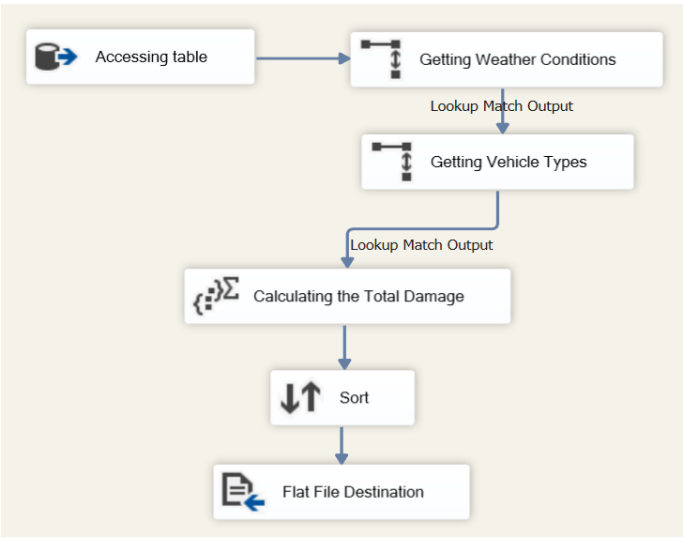


Figure 4: Assignment 8b - Data Flow

In this assignment, we also started by choosing the OLE DB Source. We used the fact table to select *Weather_FK*, *Vehicle_FK*, and *Amount_Damage*. Then, we used the Lookup Transformation to extract the value of *Weather Condition* from the Weather table. Next, we used the Lookup Transformation again to extract the data for *Vehicle Types* from the Vehicle table. After that, we used an Aggregate Transformation to group the data by *Vehicle_Type* and *WeatherCond*, and calculated the sum of *Amount_Damage*, renaming the result to *Total_Damage*. Then, we sorted the *Total_Damage* values in descending order. Finally, we used the Flat File Destination to save the result to our local system. The figure 4 shows the data flow represented by SSIS.

### 2.4 Assignment 9b

In this assignment, we generated the business question: "*Show the amount of damage for each weather condition by month and year*". In order to solve this problem, we created the solution as shown in the figure 5.

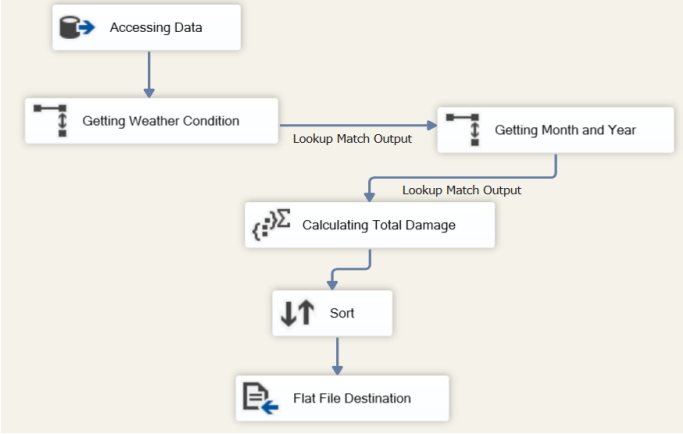


Figure 5: Assignment 9b - Data Flow

The Fact Table was accessed once again to retrieve the columns needed for the computation: *Weather_FK*, *Time_FK*, and *Amount_Damage*. Then, we used the Lookup Transformation twice. The first lookup was used to extract the *Weather Condition* from the Weather table, and the second was used to retrieve the *Month* and *Year* values from the Time table. Next, we used an Aggregate Transformation to group the data by *Year*, *Month*, and *WeatherCond*, and calculate the sum of *Amount_Damage*, renaming it to *Total_Damage*. After that, we sorted the *Total_Damage* column in descending order. Finally, we saved the results to a CSV file using the Flat File Destination.

## 3 MDX Queries

In this section, we solve the business questions using MultiDimensional eXpressions (MDX) within SQL Management Studio.

### 3.1 Assignment 2

The question is: "*For each month, show the total damage costs for each location and the grand total for each location.*"

The query returned a table with the following columns: location (represented as longitude and latitude), month (expressed as a number), the amount of damage for that location in that month, and the grand total for the location (i.e., the total sum of damage at that location).
It is evident that some locations have crashes distributed across multiple months, while others show a concentration of crashes within fewer months.

| | | Amount Damage | Grand total location |
|---|---|---|---|
| (41.645667484,-87.542509813) | 5 | 1415 | 8816 |
| (41.645667484,-87.542509813) | 7 | 1000 | 8816 |
| (41.645667484,-87.542509813) | 8 | 2228 | 8816 |
| (41.645667484,-87.542509813) | 9 | 2386 | 8816 |
| (41.645667484,-87.542509813) | 10 | 1787 | 8816 |
| (41.645823276,-87.542713717) | 10 | 12299 | 12299 |
| (41.645860133,-87.542351765) | 1 | 7512 | 7512 |
| (41.645941825,-87.542867575) | 2 | 1000 | 1000 |
| (41.646016079,-87.542555917) | 11 | 13099 | 13099 |
| (41.646020858,-87.542968192) | 11 | 10805 | 10805 |

Figure 6: The result of the assignment 2

### 3.2 Assignment 3

The question is: "*Compute the average yearly damage costs as follows: for each crash, calculate the total damage cost divided by the number of distinct people involved in the crash. Then, compute the average of these values across all crashes in a year.*"
The query results indicate that there is no specific trend associated with this metric. Instead, various factors can significantly influence the outcome each year.

| | Weighted Yearly Average Damage |
|---|---|
| 2014 | 3184.5 |
| 2015 | 2366.12159152879 |
| 2016 | 2444.98053277617 |
| 2017 | 2543.99879508532 |
| 2018 | 2592.14173103157 |
| 2019 | 2699.5248019291 |

Figure 7: The result of the assignment 3

### 3.3 Assignment 4

The question is: "*For each location, show the percentage increase or decrease in damage costs compared to the previous year.*"
The query returns a table where the first column is GeographyID, representing the unique identifier for each location. For each location, multiple years may appear, depending on the occurrence of crashes over time. The table allows for a year-by-year comparison of damage costs, making it possible to identify trends such as consistent increases, decreases, or fluctuating patterns in damages at each location.
This type of analysis can highlight locations with recurring safety issues or improvements and can help assess the effectiveness of mitigation measures implemented over time.

| | | Amount Damage | Difference | Difference(%) |
|---|---|---|---|---|
| 1 | 2019 | 6069 | 6069 | 0.00% |
| 2 | 2018 | 2700 | 2700 | 0.00% |
| 2 | 2019 | 10039 | 7339 | 271.81% |
| 3 | 2016 | 20823 | 20823 | 0.00% |
| 3 | 2018 | 1758 | 1758 | 0.00% |
| 3 | 2019 | 5352 | 3594 | 204.44% |
| 4 | 2019 | 500 | 500 | 0.00% |
| 5 | 2019 | 10040 | 10040 | 0.00% |
| 6 | 2019 | 11515 | 11515 | 0.00% |
| 7 | 2018 | 17876 | 17876 | 0.00% |
| 7 | 2019 | 8183 | -9693 | -54.22% |

Figure 8: The result of the assignment 4

### 3.4 Assignment 6

The question is: "*For each vehicle type and each year, show the information and the total damage costs of the person with the highest reported damage.*"
The query returns a table that includes the Person ID (which uniquely identifies the person), the vehicle type, the year, and the total damage costs associated with the person who reported the highest damage for that specific combination of vehicle type and year.
This analysis provides insights into the most severe individual incidents in terms of financial impact, helping to identify potential patterns related to specific vehicle types or time periods that may warrant further investigation. The figure 9 shows the results of the query.

| | | Person With Max Damage | Max Damage |
|---|---|---|---|
| 2015 | MOTORCYCLE (OVER 150CC) | O10548 | 6081 |
| 2015 | OTHER | O2434 | 9256 |
| 2015 | OTHER VEHICLE WITH TRAILER | P15 | 4477 |
| 2015 | PASSENGER | P376 | 13402 |
| 2015 | PICKUP | O3939 | 10793 |
| 2015 | SPORT UTILITY VEHICLE (SUV) | O11760 | 12010 |
| 2015 | TRACTOR W/ SEMI-TRAILER | O16720 | 7399 |
| 2015 | TRACTOR W/O SEMI-TRAILER | O19426 | 6952 |
| 2015 | TRUCK - SINGLE UNIT | O8547 | 10512 |

Figure 9: The result of the assignment 6

### 3.5 Assignment 8.1

The question is "*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*".
To answer this business question, we proposed different solutions, considering various circumstances. First, we defined the weighted measures for each cause by multiplying the Fact Data Count by 2 for the Primary Cause and by 1 for the Secondary Cause. Then, we calculated their sum. Next, we used the TopCount function to identify the cause with the highest weighted total for each year. After that, we calculated the total damage costs

for crashes caused by the most frequent cause in each year and filtered the data to include only crashes matching the most frequent cause. Finally, we determined the overall most frequent crash cause across all years based on weighted contributions. This approach focuses on the causes of crashes and provides an overview of the damage costs linked to the most common causes. The result of the query is showed at the figure 10.

| | Most Frequent Cause | Total Damage Costs | Overall Most Frequent Cause |
|---|---|---|---|
| 2014 | UNABLE TO DETERMINE | 20318 | UNABLE TO DETERMINE |
| 2015 | UNABLE TO DETERMINE | 15688930 | UNABLE TO DETERMINE |
| 2016 | UNABLE TO DETERMINE | 75525585 | UNABLE TO DETERMINE |
| 2017 | UNABLE TO DETERMINE | 139592828 | UNABLE TO DETERMINE |
| 2018 | UNABLE TO DETERMINE | 192618209 | UNABLE TO DETERMINE |
| 2019 | UNABLE TO DETERMINE | 4304453 | UNABLE TO DETERMINE |

Figure 10: The result of the assignment 8.1

# 4 Data Visualization

This section describes interactive dashboards created using the previously developed multidimensional model. These dashboards are essential tools as their visual nature enables management to uncover patterns and trends not immediately apparent in static data, providing in-depth insights that can be critical for supporting business decisions. The following assignments were developed using Power BI software.

## 4.1 Assignment 9

The chart 11 displays the geographical distribution of user injuries in relation to the type of vehicle involved in the crash. For clarity, this dashboard example focuses on injuries sustained by users who were passengers at the time of the crash. Blue dots represent the data points, with their size increasing according to the severity of the injury. Notably, the visualization reveals that crashes tend to cluster around intersections, where the risk of accidents seems higher due to braking (before reaching the intersections) and the convergence of vehicles from different directions. Interestingly, a significant concentration of crashes is observed near a river branch in Chicago's New Eastside neighborhood, possibly due to the area's higher population density. This insight could encourage the insurance company to adjust policy premiums based on clients' residential zones.

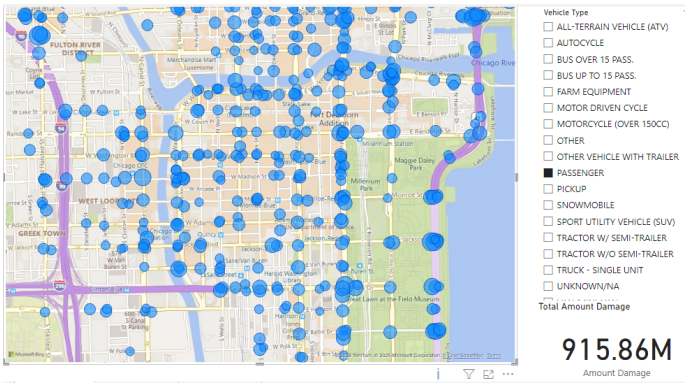


Figure 11: The Geographical Distribution of the Total Damage costs for each Vehicle Category

## 4.2 Assignment 10

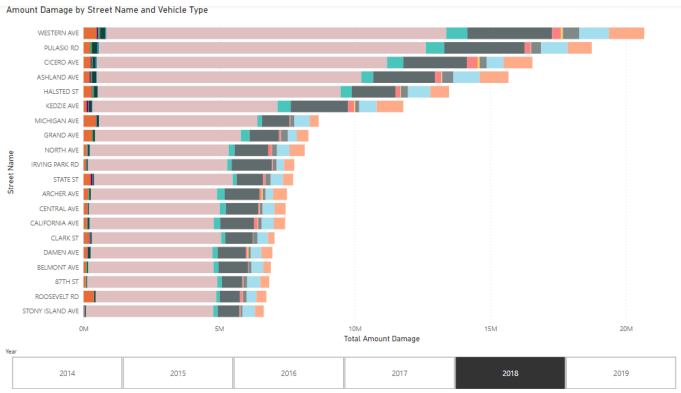


Figure 12: Top 20 Streets with the Highest Total Damage Amount (showed by each year)

The dashboard 12 ranks the top 20 streets in Chicago annually by the severity of user injuries. The chart shows the contribution of each vehicle category to the total injuries on eachstreet for the years analyzed. The example provided highlights data for 2018, where passengers contributed the most to total injuries, followed by SUVs and vans/minivans. The temporal dimension of the dashboard allows users to observe changes in street safety over time and identify vehicle categories that contribute the most to injuries. This information can be leveraged to update insurance premiums and offer tailored products to different user segments.

## 4.3 Assignment 11

The final assignment presents a dashboard 13 linking user injuries and types of crashes to gender and age groups. The dashboard consists of two distinct charts. First, it highlights that male users significantly contribute to the total injury count compared to females. Second, it shows that most crashes, regardless of age or gender, result in no injuries or are classified as "*drive-away*". On the left side of the dashboard, 20 age groups are ranked by injury severity, with half of the groups concentrated in the 20–30 age range. This dashboard can assist management in designing targeted solutions based on customer categories.

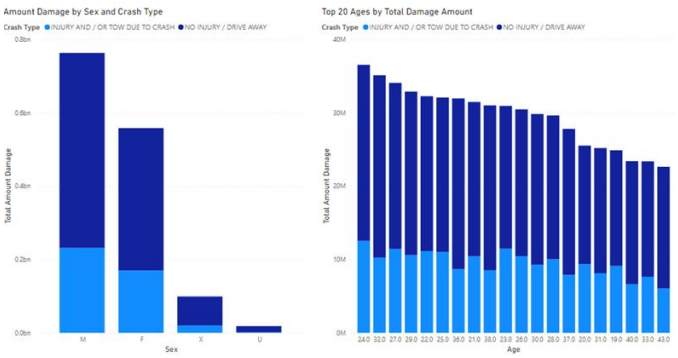


Figure 13: The Total Amount Damage per Gender (left) and Top 20 Ages by Total Damage Amount (right)