University of Pisa
Laboratory of Data Science
A.Y 2024-2025

LDS Project - Part 1

Submitted to:
Prof. Anna Monreale

Submitted by:
Alessandro Carella
Sara Hoxha
Rafael Ignacio Urbina Hincapie

December 2, 2024

Contents

1 Data Understanding & Cleaning

This section describes the steps we've taken to understand the Chicago traffic crash data, which was organized into three files `People.csv` , `Vehicles.csv`, and `Crashes.csv`. The data cleaning was performed over these files, and the results were saved into three respective files with the suffix '_Preprocessed'.

1.1 People Data

The *People* table provides detailed information about individuals involved in traffic crashes in Chicago. It includes information such as person's address of residence, their age, their actions during the crash, etc. To better understand the data, we reviewed the data types and unique values for each column which gave us an idea of what the dataset included. We explored potential duplicates but found none. However, we noticed that certain columns, such as `AGE`, were incorrectly represented as floating-point numbers instead of integers. Anther peculiarity was found in the `SEX` column, where a lot of rows had the value "X". Additionally, some columns included specific values to represent "Unknown," but also included missing values. We identified 14 columns with missing values. We utilized visualizations, including bar plots and histograms, which revealed some interesting trends, particularly regarding the `AGE` column. We discovered that a large number of entries, close to 20k, had an age recorded as 0 or had ages that seemed unusually young for drivers.

Based on the findings from data exploration on the `People` table, we decided to implement the following changes to clean the data:

1. **Use the `CITY` column to determine the `STATE` column when the latter is empty.** Initially, we considered using GeoCode API to fill the missing values, yet due to potential API usage limits and time constraints, we opted to use an external CSV file containing a comprehensive list of U.S. cities and their corresponding states. As such, we use `CITY` to infer the `STATE` by referencing the aforementioned external dataset.

2. **Map NaN `'SEX'` values → `"U"` (Unknown).** The `SEX` column sometimes contains `'U'` or missing values, which indicate unknown gender. We decided to map NaN values to `"U"` to maintain consistency and signify an unknown gender. As for the uncommon value "X", we decided to leave as-is, as it may refer to a person who doesn't wish to specify their sex intentionally.

3. **For columns that have already an "Unknown" value value, fill missing observations with this value.** The following mapping was used: For `CITY`, `DRIVER_ACTION`, `DRIVER_VISION`, `PHYSICAL_CONDITION`, and `EJECTION` → `UNKNOWN`. For `AIRBAG_DEPLOYED` → `DEPLOYMENT UNKNOWN` and for `SAFETY_EQUIPMENT` → `USAGE UNKNOWN`.

4. **Convert `VEHICLE_ID` & `AGE` to an integer.** As they were found to be stored as floats, we will convert both to integers to standardize the data and avoid any future processing issues.

5. **Add new value `'N/A'` for missing values in `DRIVER_VISION` & `DRIVER_ACTION` for passengers.** For rows that meet this condition, we will fill these missing values with `'N/A'` which stands for `Non Applicable`. This ensures that we don't have missing data in driver-related columns for passengers.

6. **Split `CRASH_DATE` into `DAY`, `MONTH`, `YEAR`, and `TIME` columns.** The `CRASH_DATE` column contains both the date and time information and to make the data more granular and easier to analyze, we will split this column into the four separate columns.

7. **Use NaN value for `AGE` when `AGE` is less than 10 and `PERSON_TYPE` is `DRIVER`.** This

is based on the assumption that anyone under the age of 10 cannot possibly be the driver, and these entries likely represent errors or anomalies.

8. **Set `CITY` to `Unknown` when `CITY` is numeric, has a length less than 2, or starts with `UNK`.** We will set the `CITY` value to `Unknown` in these cases as they are likely to represent errors, misspellings or placeholders.

9. **Set `STATE` as Unknown when `CITY` is UNKNOWN or `STATE` == XX**. This ensures that the `STATE` column remains consistent in representing missing or unknown data.

1.2 Vehicles Data

The vehicles table provides comprehensive information about the vehicles involved in crashes. Each record in this table represents a distinct unit involved in a crash, with multiple units potentially being involved in the same incident. This granular structure allows for detailed analysis of multi-vehicle accidents. The table contains the following interesting attributes, such as `UNIT_TYPE`: Categorizes the operational status of the unit at the time of the crash, including classifications such as 'DRIVER', 'PARKED', or 'NON-MOTOR VEHICLE', `LIC_PLATE_STATE`: A two-letter state code indicating where the vehicle is registered, helping track interstate involvement in crashes, `VEHICLE_DEFECT`: Indicates any mechanical or structural defects that may have contributed to the crash, etc... Prior to loading the data into our data warehouse, several standardization and cleaning operations were performed on the Vehicles dataset. The primary focus was on handling missing values (NaN) and standardizing inconsistent entries. The following modifications were implemented:

1. **Missing Values Standardization:** Multiple attributes containing NaN values were standardized to 'UNKNOWN' for consistency and better querying capabilities. This included: UNIT_TYPE, VEHICLE_DEFECT, VEHICLE_TYPE, VEHICLE_USE, VEHICLE_DIRECTION, MANEUVER, and FIRST_CONTACT_POINT.

2. **License Plate State:** Both NaN values and 'XX' entries in LIC_PLATE_STATE were consolidated to 'UNKNOWN' to maintain consistency in representing unidentified states.

3. **Vehicle Make and Model:** Duplicate entries caused by typographical errors were identified and consolidated. The MODEL field contained both 'UNKNOWN' and 'UKNOW' which were standardized to 'UNKNOWN'.

4. **Vehicle Year Cleaning:** Years recorded beyond 2024 were marked as 'UNKNOWN' as they represent impossible future dates. A special case was identified where '999' appeared in the data, potentially representing a typo for '1999'.

1.3 Crashes Data

The *Crashes* dataset provides information about traffic incidents in Chicago, including spatial, temporal, and severity-related aspects of crashes. We cleaned the data by performing the following: **Handling Incorrect Values** Two entries in the `RD_NO` column had lower-case letters, differently from the usual format. These anomalies (`hz273623` and `hz125235`) were corrected to align with the dataset's standard.

Imputation of Missing Values Several columns contained missing data that required careful consideration. For the `REPORT_TYPE` column, 4,996 out of 257,925 entries lacked values. Since reliable inference methods were unavailable and the page on the Chicago police department website indicated missing values as `AMENDED`, these entries were left as missing. The `STREET_DIRECTION` column had two missing values, and attempts to impute them using the `TRAVEL_DIRECTION` column from the Vehicles dataset were unsuccessful, so these values were retained. Initially there were many missing values for the columns LATITUDE, LONGITUDE, POINT and BEAT_OF_OCCURRENCE. Using the fields STREET_NAME and STREET_NO and the API available at this link we were able to obtain most of the missing values for LATITUDE, LONGITUDE and, obviously POINT. The STREET_NAME column had one missing value. This record also lacked LATITUDE and LONGITUDE, making it impossible to identify the corresponding street name. Using the found values for the 3 columns previously listed we were able to fill almost all missing values in the column BEAT_OF_OCCURRENCE by finding perfect match of the column POINT in other records. Just one record misses the field BEAT_OF_OCCURRENCE and is in an unique point. For the MOST_SEVERE_INJURY column, 7 missing values were identified. No discernible correlation with other injury-related columns or predictors was found, so these entries were left blank.

Derived Features and Enhancements A new column, DELTA_TIME_CRASH_DATE_POLICE_REPORT_DATE, was created to quantify the time difference between crash occurrence (CRASH_DATE) and police notification (DATE_POLICE_NOTIFIED). This feature was expressed in the format "days minutes seconds." Numeric columns such as NUM_UNITS, INJURIES_TOTAL, and various injury subcategories were converted to integer type for consistency.

Limitations and Unresolved Missing Data The REPORT_TYPE column retained 4,996 missing values, while STREET_DIRECTION and STREET_NAME had two and one missing entries, respectively. The BEAT_OF_OCCURRENCE column had one unresolved value, and the MOST_SEVERE_INJURY column retained seven missing entries due to a lack of predictive relationships with other variables. Additionally, one record missing LATITUDE, LONGITUDE, LOCATION, and STREET_NAME could not be resolved because no viable recovery method was identified. While these missing cases represent a very small proportion of the dataset, they were retained without further imputation to preserve the dataset's integrity.

2 Data Warehouse Schema

The data warehouse schema resembles a snowflake design, consisting of a central fact table and multiple dimension tables, which are further normalized. This structure was chosen to optimize query performance and reduce data redundancy, especially when dealing with changing or repeated attribute values across different records which we found to be the case. Below is a breakdown of the schema and the rationale behind the design choices.

2.0.1 Fact Table

The central fact table is DamageReimbursement. It contains the transactional data related to reimbursements for damages caused by crashes, as such the most useful information for the insurance agency.It has a composite primary key, comprised of three foreign keys Crash_ID, Person_ID, and Vehicle_ID linking the fact tables to the respective dimensions. The attribute Cost_Category classifies the reimbursement based on its value, and the measure Cost quantifies the total reimbursement. This structure was chosen because it enables efficient tracking of reimbursements related to specific crashes, vehicles, and people. The foreign keys ensure a comprehensive relationship with other tables and ability to delve into more details if needed. The Cost measure allows for aggregation and analysis of total reimbursement amounts, and the Cost_Category attribute allows segmentation by value categories.

2.0.2 Dimensions

As part of our design, several dimensions were structured to provide a clear and efficient representation of the data. In particular, the `Crash` dimension is normalized into three sub-dimensions: `CrashLocation`, `CrashCondition`, and `Injury`. This normalization approach was primarily chosen to avoid redundancy, such as the repeating location data for multiple crashes. Moreover, it improves data integrity, and enables more granular analysis of specific aspects of a crash if needed. The `DateTime` dimension is shared between two dimensions: `Crash` and `Vehicle`, which allows both to reference temporal attributes consistently, and thus ensuring consistency.

CrashLocation stores information about the geographic and physical location of each crash. Primary key is `Crash_Location_ID`. Key attributes include: `Street_No`, `Latitude`, `Longitude` etc...

CrashCondition captures information about the conditions surrounding each crash, such as weather, road conditions, and traffic control. Primary key is `Crash_Condition_ID`. This dimension allows us to analyze crashes based on various environmental and situational factors, which can provide insights into crash risk and/or eligibility for reimbursement.

Injury contains numerical data on the types and severity of injuries sustained during the crash. Primary key is `Injury_ID`. It is composed of measures, where the main ones include: `Injuries_Total`, `Injuries_Fatal`, `Injuries_No_Indication`, etc..

DateTime stores temporal information about crashes. Primary key is `DateTime_ID`. Attributes are`Day`, `Month`, `Year`, `Time`. It allows for time-based analysis, helping us understand patterns in crash frequency.

Person contains detailed information about the individuals involved in a crash. Primary key is `Person_ID`. Attributes include demographic information such as `Sex`, `Age`, whether safety measures were used `Safety_Equipment`, etc...

Vehicle stores details about the vehicles involved in the crash. Primary key is `Vehicle_ID`. It has attributes like `Make`, `Model`, `Defect`, etc.. The dimension helps us identify patterns related to specific vehicle types, defects and their potential influence to crash severity.

3 Data Uploading

In regards to assignment 6, given the constraints imposed by the relationships between tables in our data warehouse schema, it was not feasible to obtain an exact 10% sample of the data without compromising referential integrity. To address this, we employed a top-down sampling approach. Parent tables (without foreign keys) were sampled using SSIS's percentage sampling operation to achieve a 10% subset. Child tables (with foreign key dependencies) were filtered to include only the records associated with the sampled parent data. While this approach resulted in less than 10% of the data in the child tables, it ensured the database's referential integrity. Uniform sampling across all tables was not viable, as it would have disregarded hierarchical dependencies, leading to inconsistencies such as orphaned records or invalid references. This method, while imperfect in its yield, offered several advantages as referential integrity was preserved and the sampled dataset remained coherent and suitable for query testing.

4 Queries

4.1 Assignment 6a

In this query, we aim to analyze the frequency of crash involvement for all participants across different years to identify individuals with higher crash frequencies on a yearly basis. To achieve this, we begin with the Person table, extracting key demographic details such as ID, Type, Sex, and Age. We then link this data to the DamageReimbursement table to retrieve crash IDs, followed by the Crash table to obtain crash dates, and finally, the DateTime table to extract the year of the crash. The data is aggregated by counting the number of crashes for each participant per year, generating a "Total Crashes" metric. The results are sorted by year and total crashes to provide a clear chronological overview of crash frequency patterns. Finally, the results are stored in the "Query_6_result" table.

4.2 Assignment 7a

This query calculates a day-night crash index per police beat by comparing vehicle counts in nighttime (9 PM–8 AM) vs. daytime (8 AM–9 PM) incidents. To implement this, we extract relevant columns from the Crash(Number_of_Units), CrashLocation(Beat_of_occurrence), and DateTime tables, joining them sequentially on their IDs. A derived column transformation creates a Time_Category field ("Day" or "Night") based on timestamps. Data is split by Time_Category, aggregated by Beat_of_occurrence, and summed for vehicle counts (Number_of_Units_Day and Night). The 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. The results are sorted by the calculated index and stored in the "Query_7_result" table.

4.3 Assignment 8a

This query calculates the ratio of people under 21 to those over 21 for each quarter, weather condition, and beat of occurrence, analyzing how seasonality, weather, and location in Chicago affect crash proportions by age. This helps insurance companies understand young drivers' behavior and improve decision-making. To implement this, we join the DamageReimbursement, Crash, CrashLocation, CrashCondition, DateTime, and Person tables using SSIS lookup operations to extract relevant data. After counting individuals over and under 21, we aggregate by the required columns, thus creating a new column containing the ratio. The results are sorted and stored in the "Query_8_result" table.

4.4 Assignment 9a

Building on the previous analysis of day-night and age-based ratios, we propose a query to examine the ratio of older individuals (over 60) involved in interstate movements in Chicago. This includes analyzing whether they reside in Chicago and whether their vehicles are registered in other states. Additionally, we aim to identify significant quarterly variations, incorporating seasonality into the analysis. To implement this, we use lookup transformations to retrieve only the necessary attributes from dimensional tables, avoiding full joins. After classifying individuals as over or under 60, we aggregate the data by quarter and license plate registration. The ratio is calculated, sorted using a sorting operator, and saved in the "Query_9_result" table.