DADABI Capstone Project: Motor Collision Analysis – An end-to-end project

*Agash Uthayasuriyan*

*Ram Prashanth Rao G*

Contents:

1. Overview of the project
2. About the Datasets
3. Profiling - Inferences, and Staging
4. Data Cleaning, Transformations, and Intermediate Staging.
5. ER Modeling, Fact and Dimensional Loading
6. Visualizations

Overview of the Project:

This final project on Motor Collisions - focuses on data modeling, data engineering, analysis with the help of visualizations pipeline using the datasets provided (Austin, Chicago, Montgomery, NYC).

Project Objectives

1. Analyze the given data sets thoroughly
2. Design and implement an ETL (Extract, Transform, Load) pipeline
3. Create a dimensional model for efficient data storage and retrieval
4. Develop visualizations to showcase findings
5. Present a narrative that tells the story hidden within the data

This project serves as a capstone, allowing for the application of skills acquired throughout the course in a real-world scenario. It provides an opportunity to demonstrate proficiency in data handling, analysis, and presentation, showcasing of abilities in the field of data engineering and analytics.

About the Datasets

Each of the datasets has been obtained from the government websites and profiling has been done individually for each one of them, and the details of the same are presented in the profiling document. The following section covers only the key details about the datasets.

1. Austin Dataset:

No. of Columns: 43

No. of Rows: 212,834

Source: <https://data.austintexas.gov/Transportation-and-Mobility/Austin-Crash-Report-Data-Crash-Level-Records/y2wy-tgr5/about_data>

1. Chicago Dataset:

No. of Columns: 48

No. of Rows: 896,756

Source: <https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if/about_data>

1. New York City Dataset:

No. of Columns: 29

No. of Rows: 2139048

Source: <https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/about_data>

1. Montgomery Dataset:

No. of Columns: 37

No. of Rows: 107003

Source: <https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Incidents-Data/bhju-22kf/about_data>

Profiling - Inferences, and Staging

Using Alteryx Designer and Y-Data Profiling, we have performed the profiling on all the given datasets. The following section showcases only the important inferences obtained from the datasets.

NYC Profiling:

A diagram of a computer

Description automatically generated  
Key Findings

1. Data Integrity: A critical data integrity issue was identified and resolved in record 1269803, where a missing close quote was fixed.
2. Location Data Incompleteness: Significant null values (>10% on average) exist in location-related columns such as latitude, longitude, zipcode, and street names. Approximately 30,000 records lack all location data.
3. Time Format Inconsistency: The Crash\_Time column uses a 24-hour format, differing from other datasets.
4. Primary Key Issues: The Collision\_ID column, intended as a unique identifier, contains null values and inconsistent ID lengths.
5. Derivable Data: Potential to derive zipcode information from the Borough column.
6. Vehicle Classification: The Vehicle type codes column requires categorization into broader groups.
7. Recommendations
8. Address Completion: Develop a strategy to complete the accident\_reported\_address using a combination of available location data.
9. Time Standardization: Convert Crash\_Time to include AM/PM designations for consistency with other datasets.
10. Unique Identifier Standardization: Implement a uniform Primary Key format to ensure each record is uniquely and consistently identified.
11. Data Enrichment:
12. Derive missing zip codes from Borough information where possible.
13. Categorize vehicle types into broader, more manageable groups.
14. Data Cleaning Protocol: Establish a robust data cleaning protocol to address missing values and ensure data consistency across all fields.

Austin Dataset:  
A diagram of a company

Description automatically generated with medium confidence

Key Observations:

1. The Austin dataset has three unique columns, namely ID, Crash\_ID, and Case\_ID. However, after profiling, it is found that the Case\_ID column has 1.4% missing values, making it unsuitable for assigning a PK to move forward.
2. Both the Crash Timestamp and Crash Timestamp (US/Central) columns have multi-valued data separated by spaces. These columns can either be divided further for finer granularity or cleaned to standardize the values into a proper format.
3. The Crash Speed Limit column contains the value -1 for accidents that are either under investigation or uncertain. The count for these values is 38,244.
4. Significant null values are found in rpt\_street\_sfx and rpt\_block\_num, which can be derived from the rpt\_street\_name column.
5. Latitude and Longitude have an equal number of null values, which need to be addressed by assigning a non-existent latitude and longitude coordinate instead of using "NA" or 0.
6. There are no records where both the Latitude and Longitude columns are empty, but the Point column is not empty.
7. The reported\_street\_prefix column is entirely null and can be removed down the line.
8. Fields like micromobility\_death\_count, bicycle\_death\_count, and motorcycle\_death\_count have limited unique values, indicating categorical data.

Chicago Dataset:  
A diagram of a company

Description automatically generated with medium confidence

Key Observations and Recommendations

1. Columns like DOORING\_I (99.7% missing), WORKERS\_PRESENT\_I (99.9%), WORK\_ZONE\_I (99.4%), and WORK\_ZONE\_TYPE (99.6%) have an overwhelmingly high percentage of missing values and could potentially be dropped.
2. Although columns like LANE\_CNT and INTERSECTION\_RELATED\_I have 77.8% and 77.0% null values respectively, they might still be relevant for analysis. These columns can also be dropped if proven to have no significant analytical value.
3. Columns such as STATEMENTS\_TAKEN\_I, PHOTOS\_TAKEN\_I, and HIT\_AND\_RUN\_I have Boolean-like values (Y/N) but include missing entries. These should be converted into a binary format (1/0) for easier processing, with missing values explicitly handled, for example, assigning a 0 for missing entries.
4. For geographical analysis, apart from latitude and longitude, as well as street\_no and street\_name, other columns such as street\_direction and location can be dropped as they do not add much value.
5. The CRASH\_DATE and DATE\_POLICE\_NOTIFIED columns can be split into separate DATE and TIME components to enable temporal analysis.
6. There are no records where latitude and longitude are null while the location column is not.
7. The INJURIES\_TOTAL column could be recalculated by summing the values of other columns denoting injuries.

It was also observed that there are no records where the INJURIES\_TOTAL is greater than the aggregated value of related columns, indicating that no category has been left out.

A screen shot of a computer

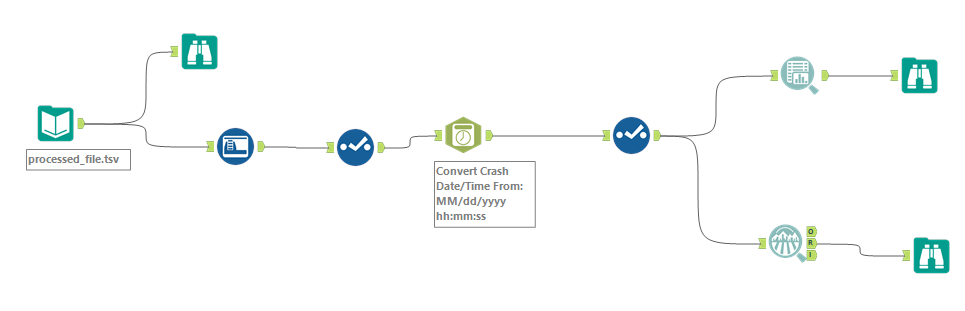
Description automatically generated

Additionally, around 876k records were found where the INJURIES\_TOTAL does not match the sum of other injury columns. Therefore, the INJURIES\_TOTAL column could be recalculated for consistency.

A screenshot of a computer

Description automatically generated

4. Montgomery Profile:

  
Key Observations:

1. There are few records that have \n in between the values, causing a record to breakdown into multiple records. This must be addressed.

Staging-As-Is Tables Using ADF:  
  
 ADF was used to create pipelines for the initial staging-as-is tables. The "Medallion Architecture" was implemented to separate operations. The "bronze" layer was utilized to stage all source files, and the "silver" layer was used to stage all Parquet files in the first step of the pipeline. Source files were converted to Parquet format and then loaded into Snowflake tables. Screenshots of the job and the counts in Snowflake are provided.

For Austin:  
A screenshot of a computer

Description automatically generated

Since the Austin columns are not as per ADF, we had to create a dataflow to address this issue and make the column names compliant.

A screenshot of a computer

Description automatically generated

Counts of the as-is Staging Tables and the dataset:

A screenshot of a computer

Description automatically generated

A white background with black lines

Description automatically generated with medium confidence

For Chicago:  
 A screen shot of a computer

Description automatically generated  
Chicago’s dataset contains valid column names hence we didn’t use dataflow to correct the column names.

A screenshot of a computer

Description automatically generated

A black and grey rectangular object

Description automatically generated

3. For NYC:

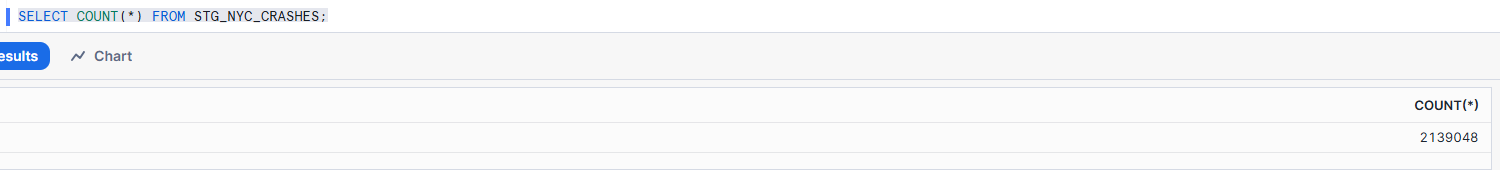
Like Austin we used Dataflows to modify the column names.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



4. For Montgomery:

Similarly for Montgomery.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black and blue background

Description automatically generated with medium confidence

Mapping Document:

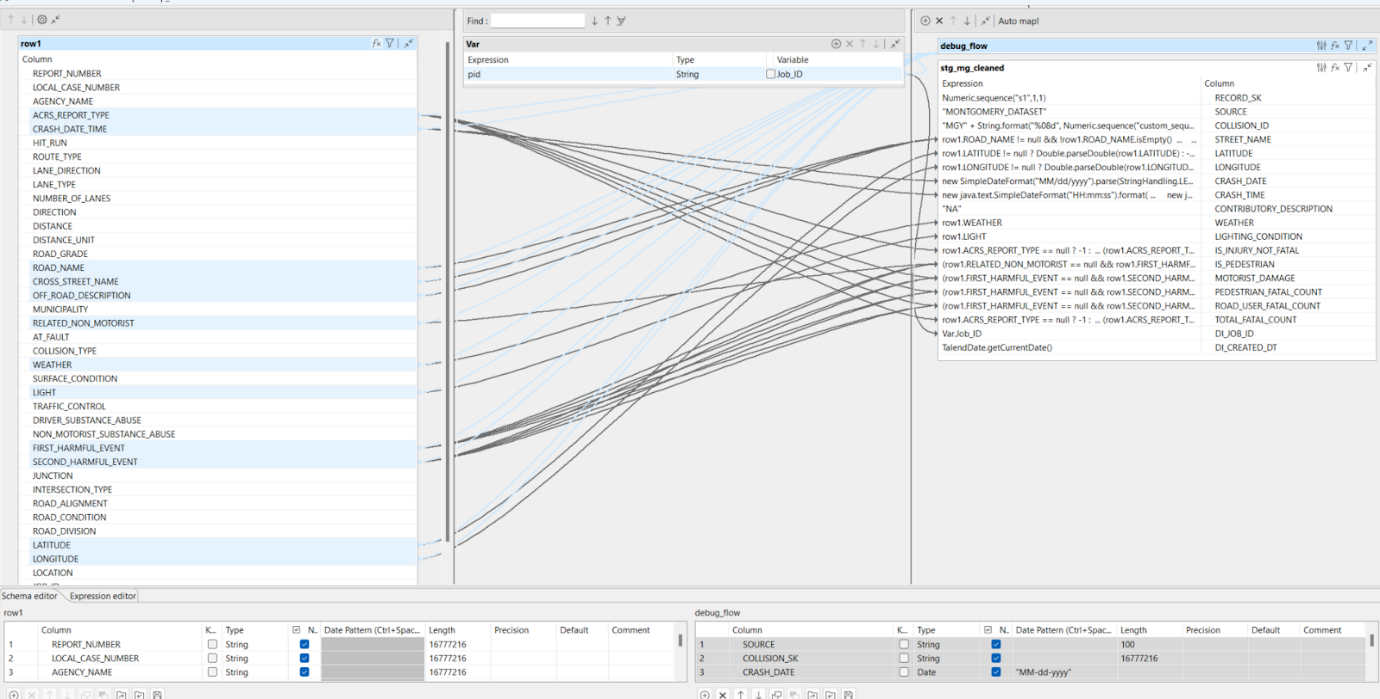
A Mapping Document was created, where all the columns in all the datasets were examined at an overall level. Based on the business requirements, the necessary data present in columns of each of the datasets were observed and the final mapping document was created, that is in a separate file.

ETL Pipelines Using Talend for Cleaning and Transforming the Intermediate Tables

After staging the tables as-is in the dataset, Talend was used to load the cleaned data, incorporating both cloud and local pipelines. The columns in each dataset were finalized for cleaning and transformation based on the specified business requirements to be achieved.

For Montgomery:  
A green line with black text

Description automatically generated



Transformation logic was added in the TMap component of Talend to address null values, clean improper data, and implement logic to meet business requirements by finalizing the columns from the actual dataset.

For Chicago:

A diagram of a computer

Description automatically generated

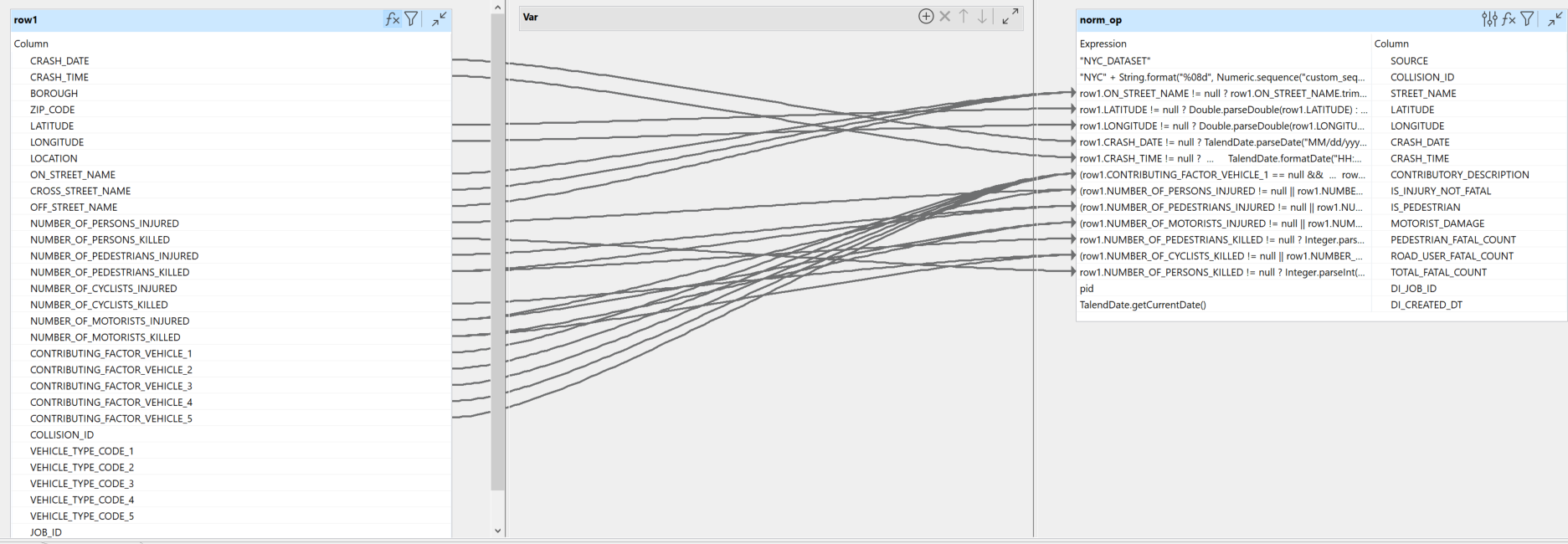
The Chicago dataset required additional transformations, including the application of the tNormalize component in Talend to address the contribution factors of accidents. As per the CR in phase-2, a transformation was implemented to assign proper codes to the contribution factors. The tUniq component was used to filter unique records and write them to the target table in Snowflake.



For NYC:

A screenshot of a computer

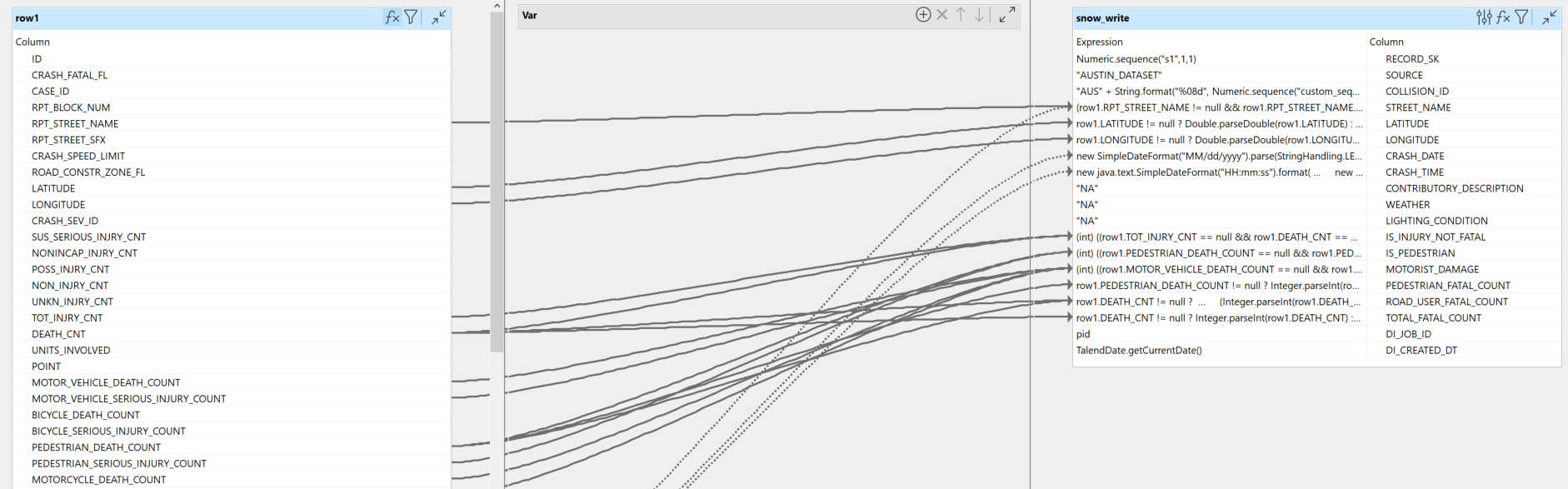
Description automatically generated



For Austin:

A green and black line with arrows

Description automatically generated



For the final Merged Staging table:

Since all the column names, datatypes are same across all datasets, they are be directly union-ed.

A screenshot of a computer

Description automatically generated

Number of rows:

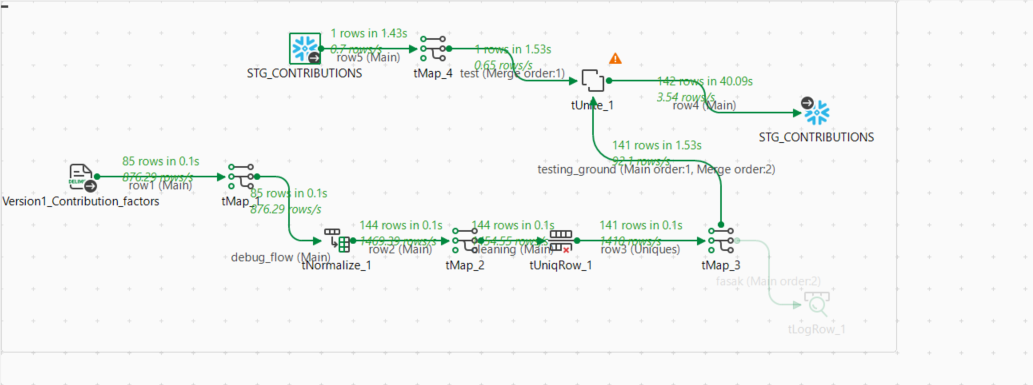
A screenshot of a computer

Description automatically generated

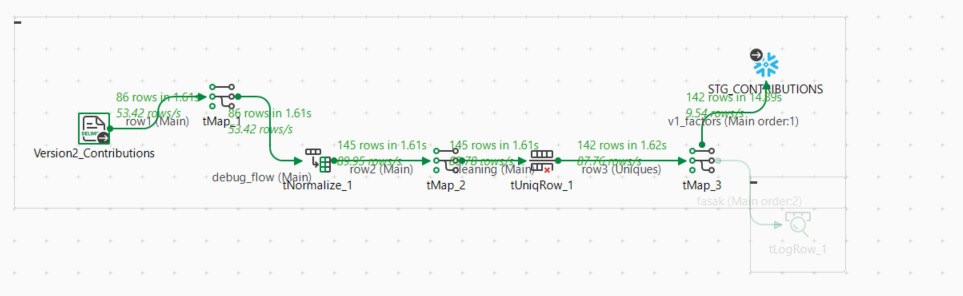
SCD Type-2 Implementation

As per the Change Request in Phase-2 of the project, logic was implemented to capture data changes and maintain a history. Talend was used to create pipelines for this purpose. Two files, version-1 and version-2, were provided to track changes in the code and description of four datasets.

A pipeline was created to load version-1 changes into the STG\_CONTRIBUTIONS table and assign proper codes based on the description.



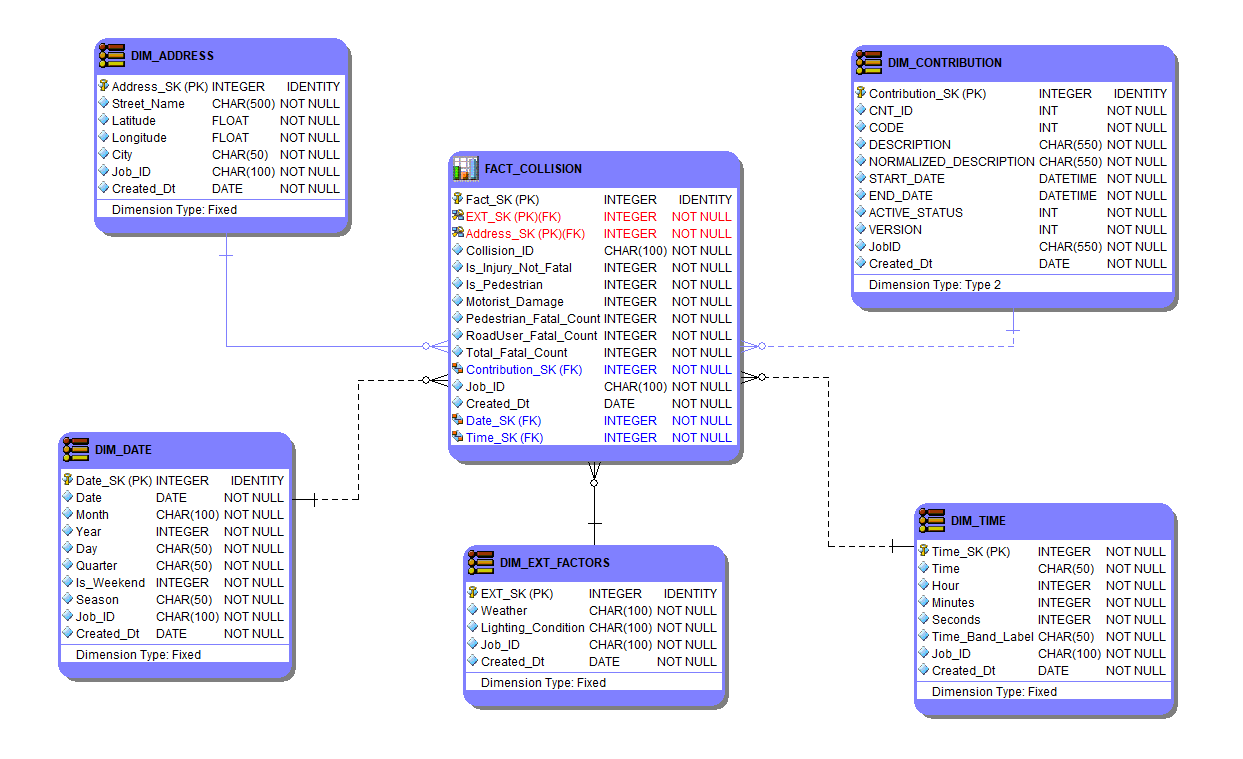
The tJDBCELT component in Talend was used to implement Slowly Changing Dimensions (SCD) for the datasets. Another pipeline was created to upsert any changes based on the version-2 file, targeting the STG\_CONTRIBUTIONS table.



This is another pipeline where it will upsert any changes as per the version-2 file. The target is STG\_CONTRIBUTIONS table.

DIMENSIONAL MODELING & LOADING THE FACT AND DIMENSION TABLES

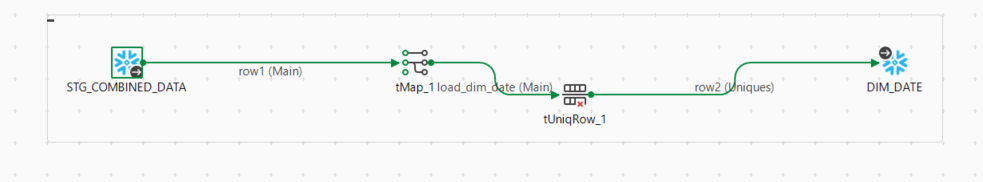
Using ER Studio, the dimensional model was designed, and DDL scripts for dimension tables and the fact table were set up.



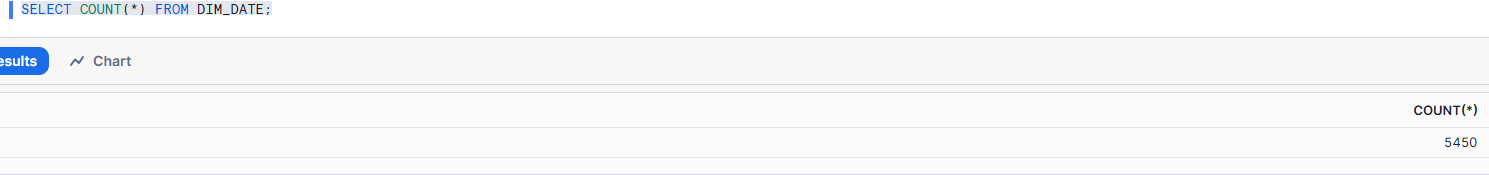
The relationships between entities have been assigned appropriately.

With the dimensional model design finalized, the DDL scripts provided by ER Studio were used to create the respective tables in Snowflake. Pipelines were then created to load these tables and apply any necessary transformations.

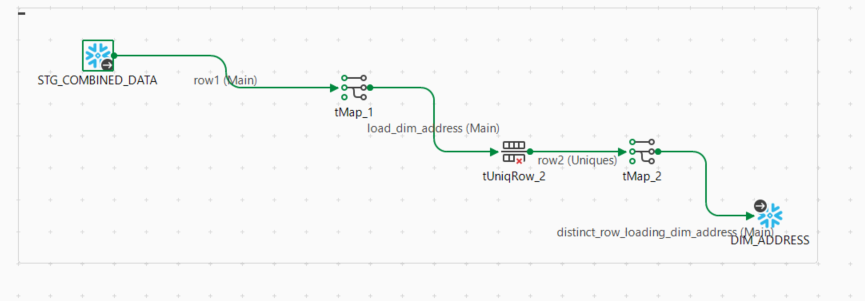
Pipeline to load DIM\_DATE:

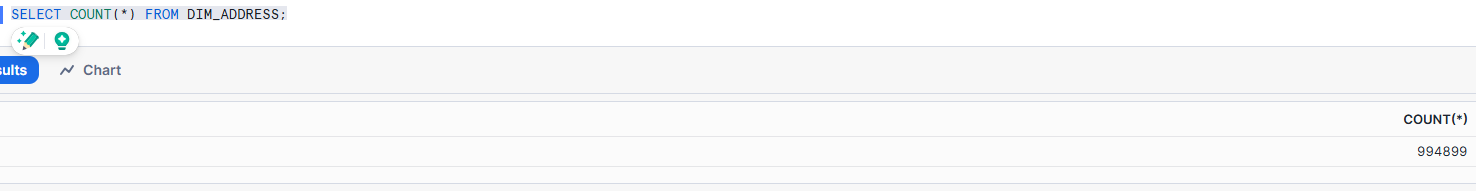


The stg\_combined\_data will be acting as a source to load all the dimension and fact tables.



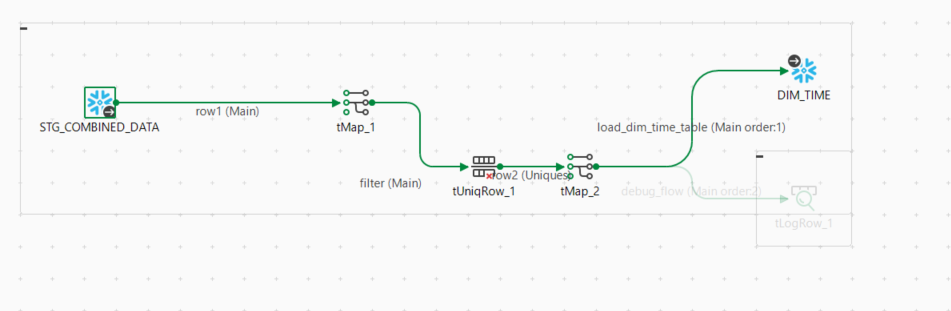
Pipeline to load DIM\_ADDRESS:





Pipeline to load DIM\_TIME:

This DIM has only 1440 records – satisfying the logic of 24 hours \* 60 minutes in a day = 1440 unique time.



A white background with black lines

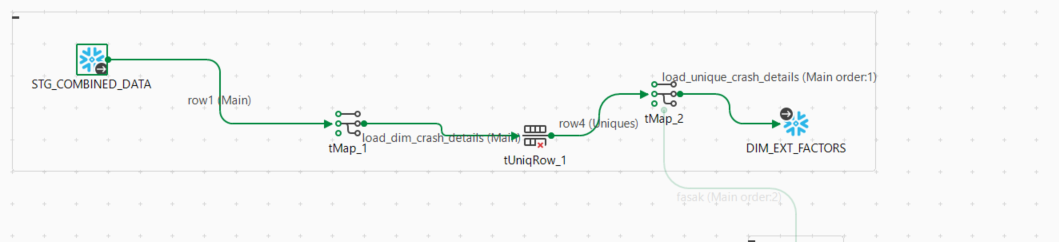
Description automatically generated

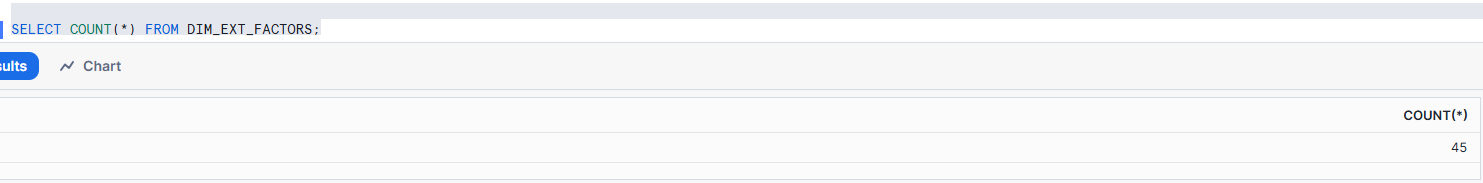
Pipeline to load DIM\_CONTRIBUTIONS:

A screenshot of a computer

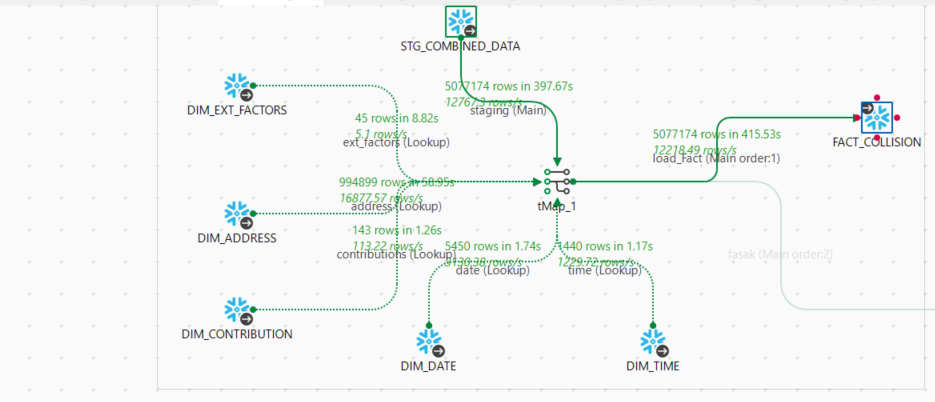
Description automatically generated

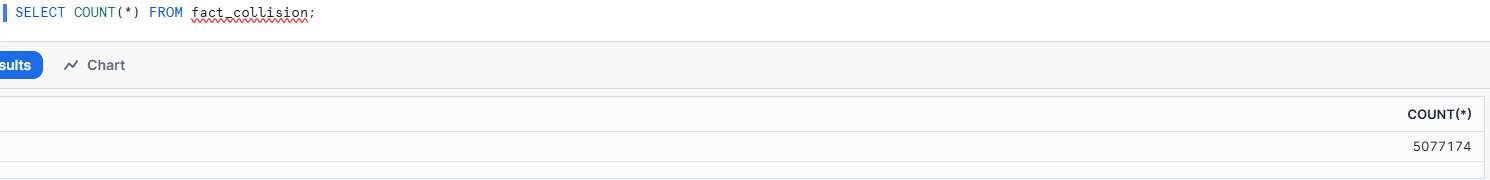
Pipeline to load DIM\_EXTERNAL\_FACTORS:





Pipeline to load Fact\_Table:





The counts match the number of records in stg\_combined\_table.

Visualizations:

**Design Rationale:**

The design strategy was to position the title and filters at the top, with the most important KPIs placed directly below them. This layout optimizes space for visualizing charts while adhering to best practices.

The most important KPI, Total Accidents, was positioned in the top-left corner, aligning with the natural eye movement of the viewer.

Both Power BI and Tableau dashboards were designed to maintain a similar structure for consistency.

**Power BI:**

A screenshot of a computer

Description automatically generated

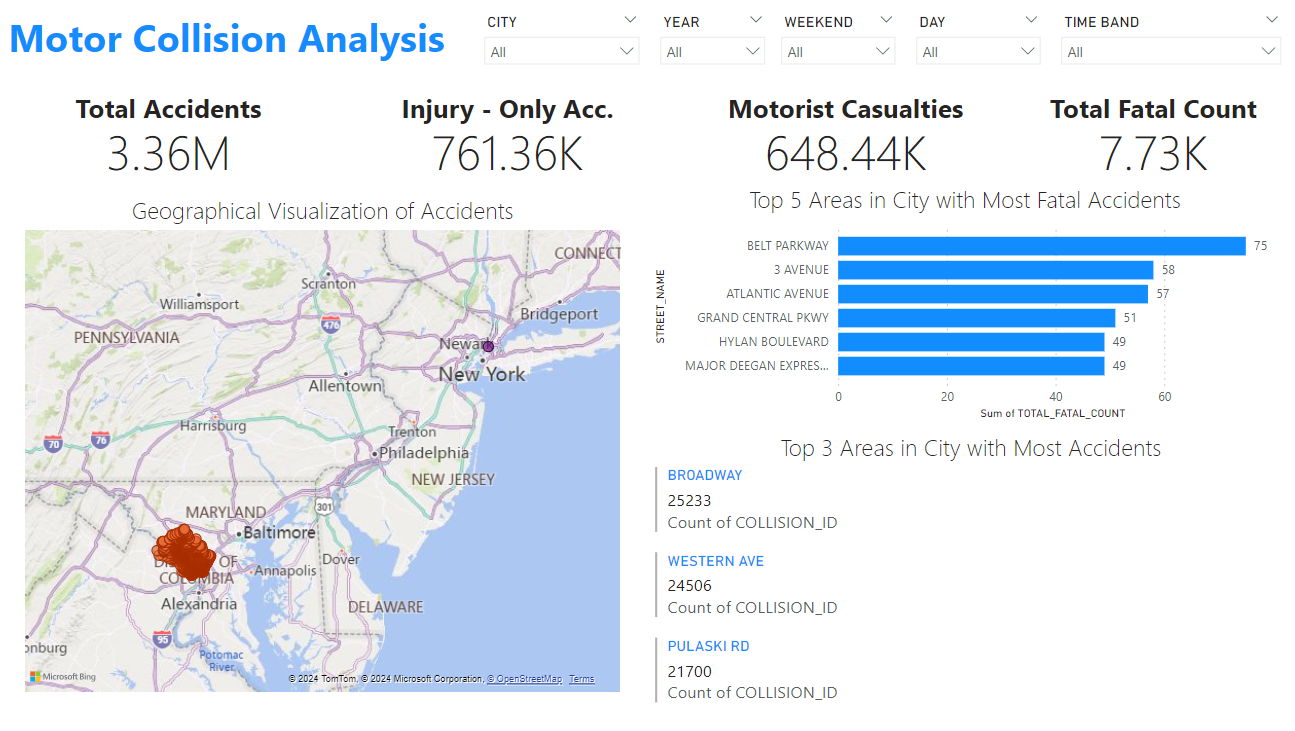


Tableau:

A screenshot of a data analysis

Description automatically generated

A screenshot of a computer

Description automatically generated