IMPACT OF CITI BIKE RIDERSHIP ON ROAD SAFETY

CIS 9440 [PMWA] – Group 4

Patrick Peralta, patrick.peralta@baruchmail.cuny.edu
James Weiner, james_weiner@hotmail.com
Alex Dorsinville, alex11896@gmail.com
John Ortiz, playcube14@gmail.com

Contents

Introduction

The government of New York City, led by Mayor Bill de Blasio, launched the Vision Zero program in 2014 in an effort to significantly reduce traffic injuries and fatalities on city streets. Bicycle ridership in New York City has soared in recent years, and its impact on safety must be examined as part of this initiative. The Citi Bike bicycle sharing system is a factor contributing to the increased bicycle ridership, and the government is interested in reviewing its effects on safety in greater detail. To accomplish this, a data warehouse needs to be created that integrates Citi Bike trip data with other data concerning road safety.

Using analysis of integrated data in the data warehouse, the government would like to be able to answer questions such as:

1. Are accidents more likely to occur on days of high Citi Bike ridership?
2. Are accidents more likely to occur near heavily used Citi Bike stations?
3. Are there any aspects of a Citi Bike rider's profile that have an effect on accident rates?

If trends can be identified, the government can devise and implement the appropriate measures to improve safety on the streets of New York City.

Identification of Data Sources/Information Needs

To be able to answer the questions posed above, two data sources were utilized. These are:

1. **Citi Bike Trip History Data**. This data can be found at:

   https://www.citibikenyc.com/system-data

   This public data is published by Citi Bike on their website. It contains details of Citi Bike trips such as date and time of the start and end of a trip, time duration, start and end stations, bike, user type, gender, and rider birth year.

   Below is the description of the data provided on the website:

   We publish downloadable files of Citi Bike trip data. The data includes:

   - *Trip Duration (seconds)*
   - *Start Time and Date*
   - *Stop Time and Date*
   - *Start Station Name*
   - *End Station Name*
   - *Station ID*
   - *Station Lat/Long*
   - *Bike ID*
   - *User Type (Customer = 24-hour pass or 3-day pass user; Subscriber = Annual Member)*

- *Gender (Zero=unknown; 1=male; 2=female)*
- *Year of Birth*

This data has been processed to remove trips that are taken by staff as they service and inspect the system, trips that are taken to/from any of our "test" stations (which we were using more in June and July 2013), and any trips that were below 60 seconds in length (potentially false starts or users trying to re-dock a bike to ensure it's secure).[1]

2. **NYPD Motor Vehicle Collisions Data**. This data can be found at:

   https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

   This is public safety data provided by the NYPD available on NYC OpenData. It contains details of motor vehicle collisions in New York City such as date, time, location, injuries, contributing factors, and vehicle types. Bicycles are included among the vehicle types.

   Below is the description of the data provided on the website:

   This is a breakdown of every collision in NYC by location and injury. This data is collected because the NYC Council passed Local Law #11 in 2011. This data is manually run every month and reviewed by the TrafficStat Unit before being posted on the NYPD website. Each record represents a collision in NYC by city, borough, precinct and cross street. This data can be used by the public to see how dangerous/safe intersections are in NYC. The information is presented in pdf and excel format to allow the casual user to just view the information in the easy to read pdf format or use the excel files to do a more in-depth analysis.[2]

List of Dimensions and Facts

After review of the data sources, the following facts and dimensions were determined:

1. **Citi Bike Trip History Data**
    - Fact
        - Citi Bike Trip
    - Dimensions
        - Date
        - Time
        - Bike Station
        - Rider Profile
2. **NYPD Motor Vehicle Collisions Data**
    - Fact
        - Accident
    - Dimensions
        - Date
        - Time
        - Location
        - Vehicle

Key Performance Indicators (KPIs)

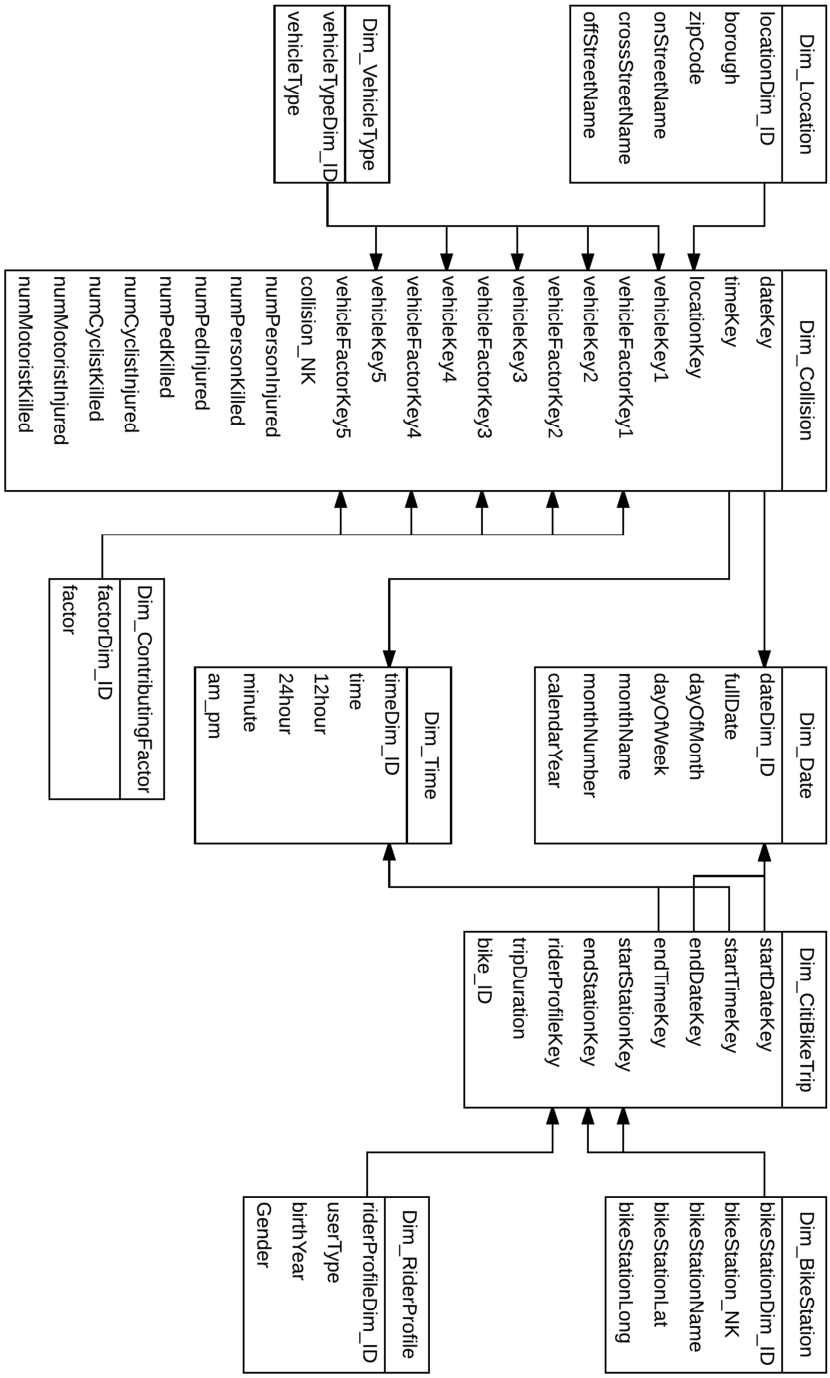
Below is a brief list of KPIs that the date warehouse could be used to provide.

- Dates/times of most/fewest Citi Bike trips
- Busiest and least busy Citi Bike stations
- Dates/times of most/fewest motor vehicle collisions
- New York City areas (zip codes) with most/fewest vehicle collisions
- Correlation of Citi Bike ridership and vehicle collision frequency
- Vehicle collision rates near Citi Bike stations
- Citi Bike rider profile effect (if any) on vehicle collision rates

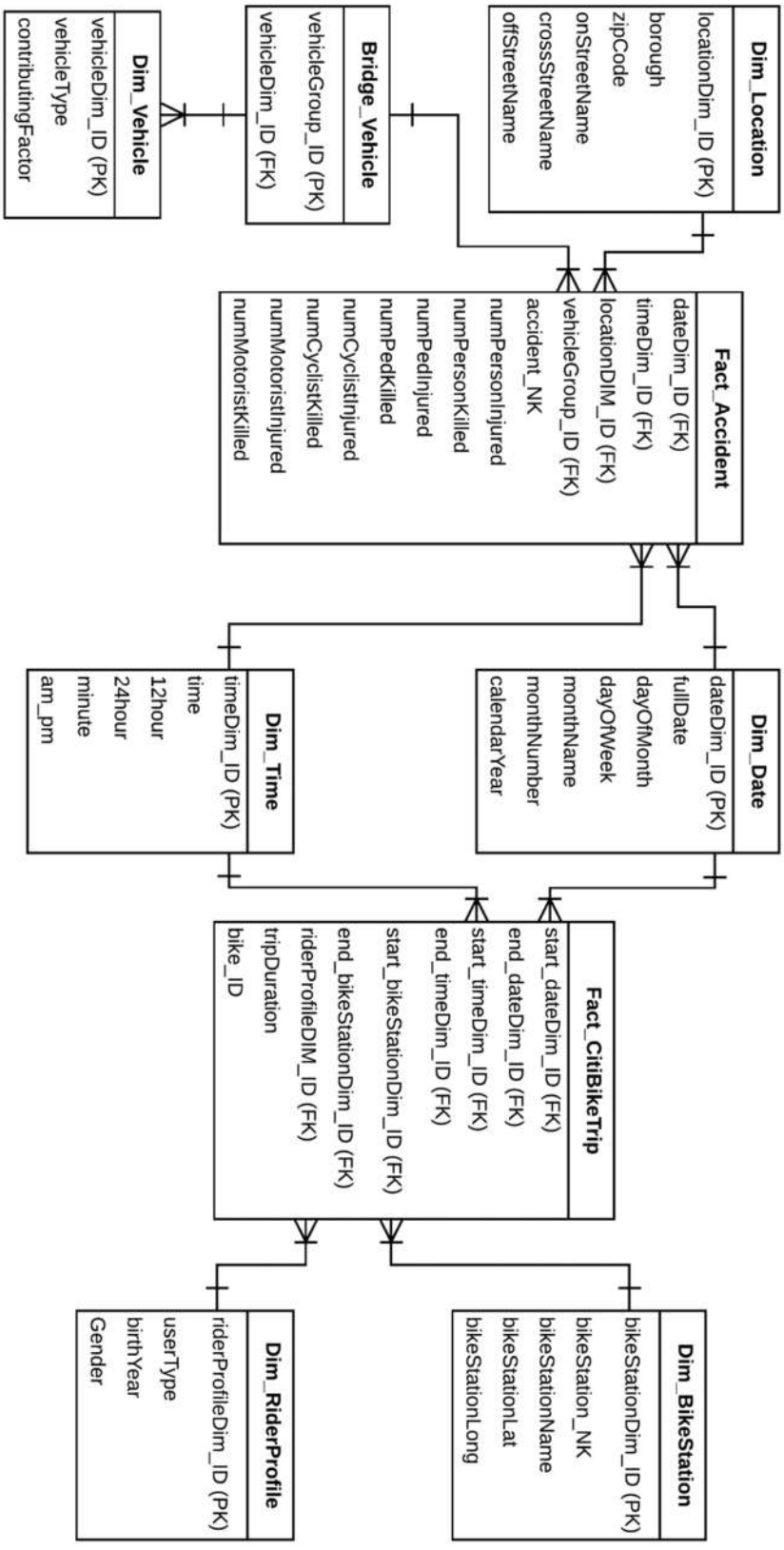
Dimensional Model

The dimension model went through several revisions. All versions of the model have been included.

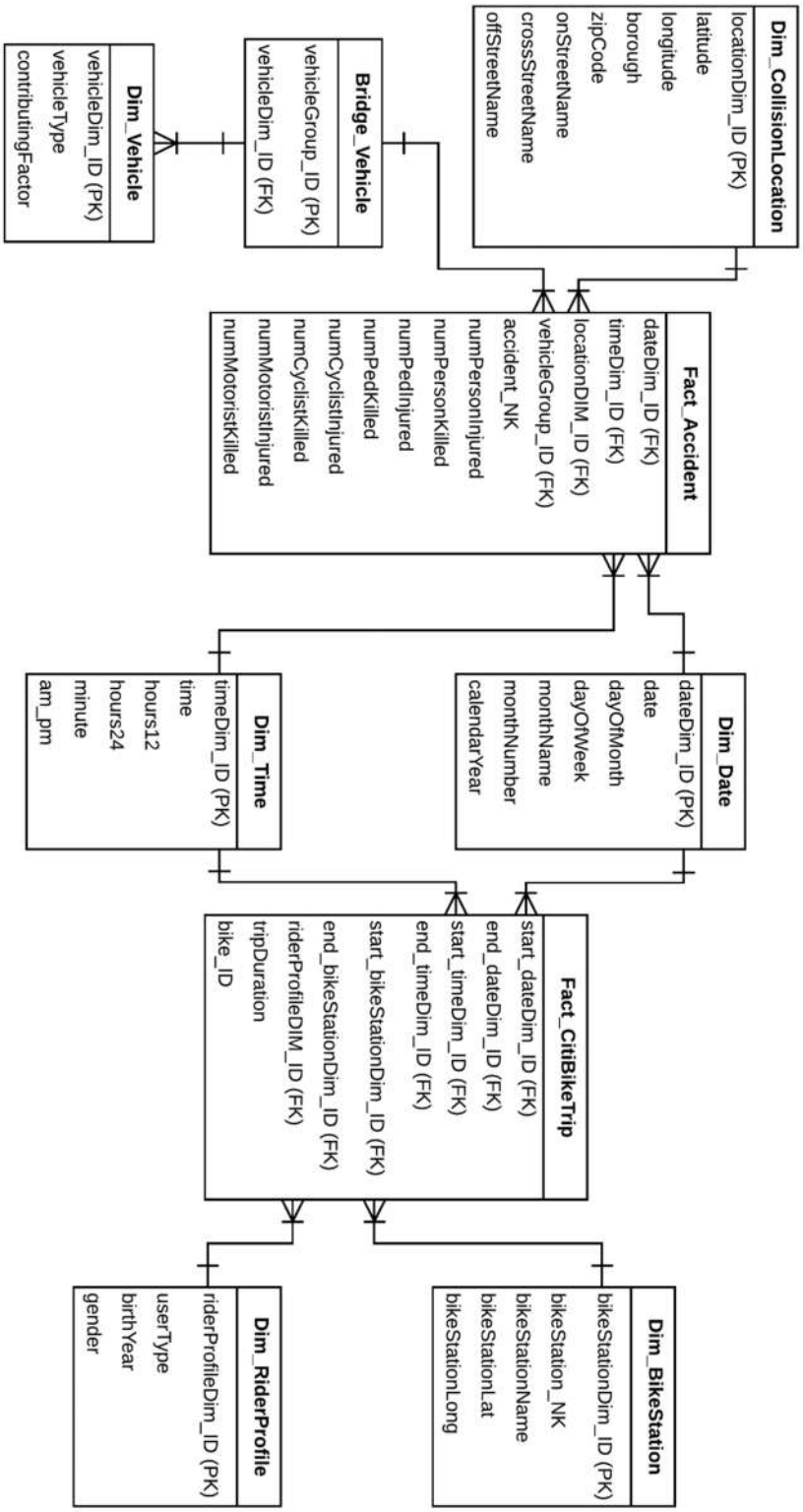
Version 1 (November 6th, 2017)



Version 2 (November 13th, 2017)

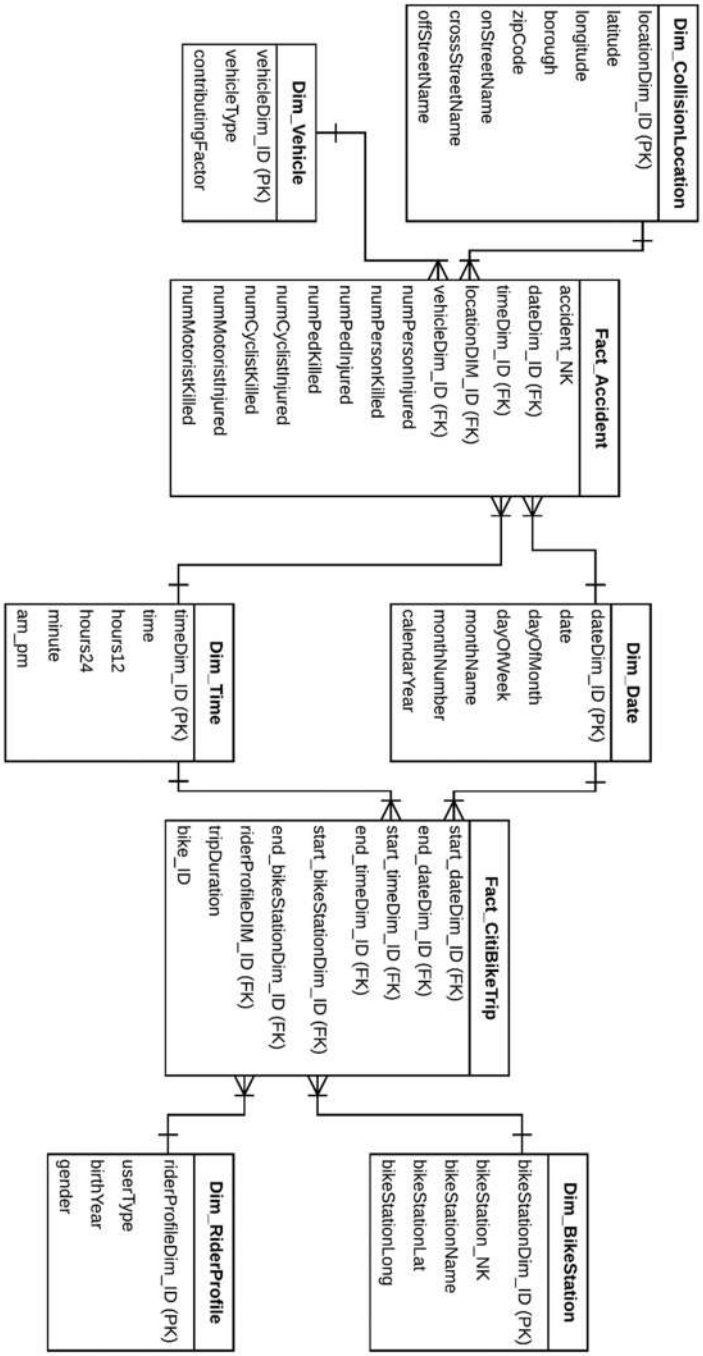


Version 3 (November 30th, 2017)



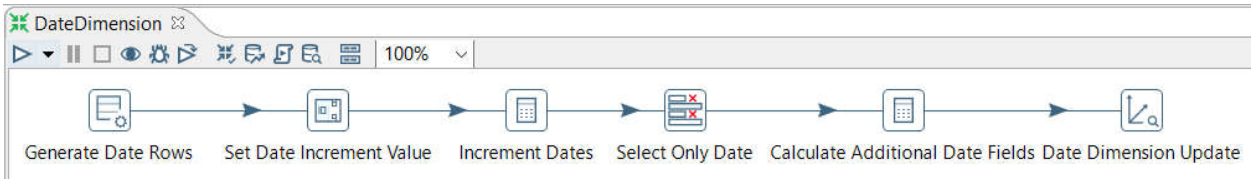
Version 4 – Final (December 16th, 2017)

The bridge table between **Fact_Accident** and **Dim_Vehicle** was removed by using additional records in the **Fact_Accident** fact table. Every accident has multiple rows in the fact table; one for each vehicle involved.



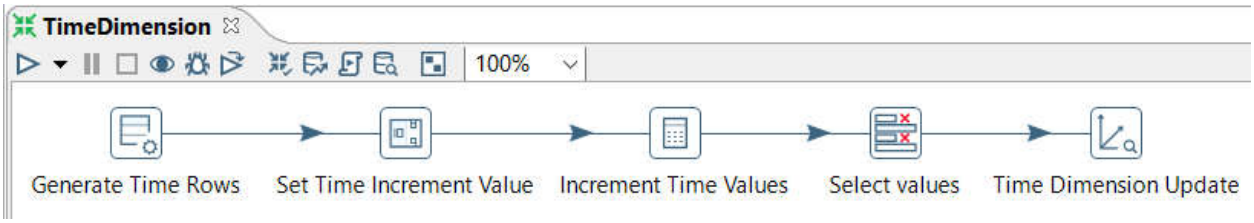
Pentaho Data Integration – ETL Processes

Date Dimension



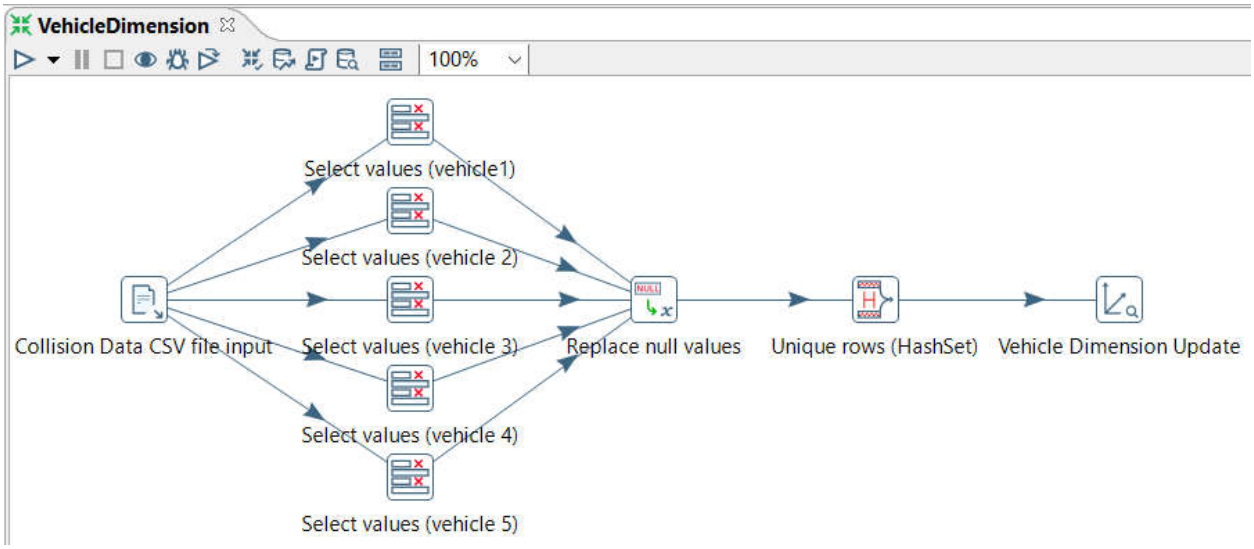
This transformation was used to create and automate the population of the **Date** dimension. The generated dates span from *January 1st, 2012* to *December 31st, 2017*. A total of *2192* rows were generated to cover that date range, and the date values were incremented by one day. The dimension was enhanced with the addition of *day of month, day of week, month name, month number,* and *calendar year* fields. This dimension was implemented as a Type 0 slowly-changing dimension as it will not require updating.

Time Dimension



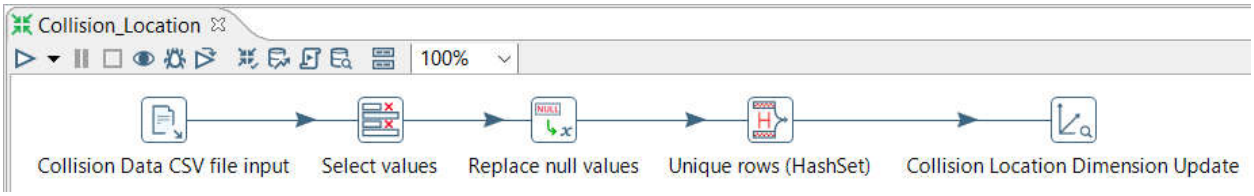
This transformation was used to create and automate the population of the **Time** dimension. A total of 1440 rows were generated to record every minute of a day, and the time values were incremented by one minute. The dimension was enhanced with the addition of *hours12* (AM/PM time), *hours24* (24-hour clock time), *minute*, and *am_pm* (AM/PM) fields. This dimension was implemented as a Type 0 slowly-changing dimension as it will not require updating.

Vehicle Dimension



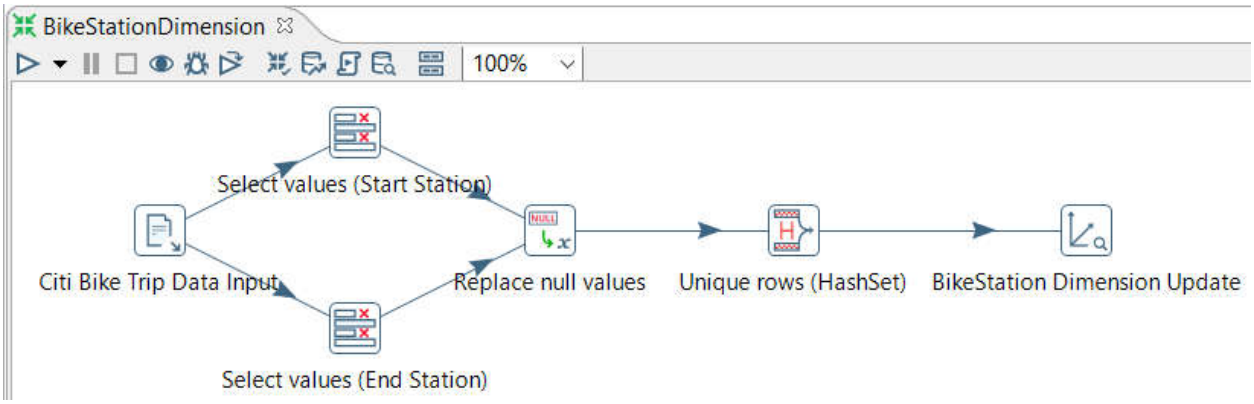
This transformation was used to create and populate the **Vehicle** dimension. This is a dimension from the **NYPD Motor Vehicle Collisions** data. It populates the dimension with all unique occurrences of the *vehicle type* and *contributing factor* values of a vehicle involved in a collision. The source data contains five columns for up to five vehicles involved in a collision, and five columns for up to five contributing factors for each of the corresponding vehicles. Five *Select / Rename values* steps were used to extract all five occurrences of vehicle/contributing factor combinations; the streams are then united when sent to the *Replace null value* step. This step sets null string values to "*N/A*". This dimension was implemented as a Type 1 slowly-changing dimension since it may be necessary to correct errors but we did not see the value in keeping a history of the changes.

Collision Location Dimension



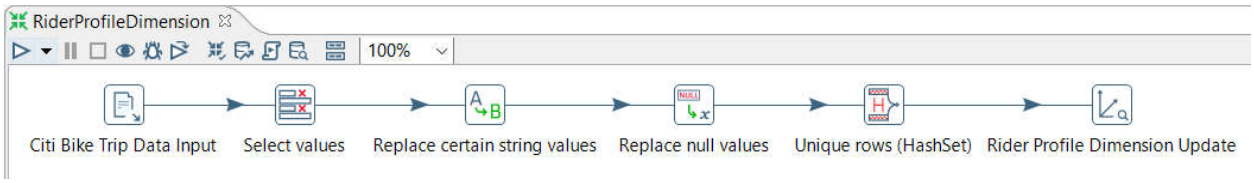
This transformation was used to create and populate the **Collision Location** dimension. This is a dimension from the **NYPD Motor Vehicle Collisions** data. The *Replace null value* step was used to replace null strings with "*NA*" and null numerical values with *0*. To populate the dimension with only unique location values, the *Unique rows (HashSet)* step was added. This dimension was implemented as a Type 1 slowly-changing dimension since it may be necessary to correct errors but we did not see the value in keeping a history of the changes.

Bike Station Dimension



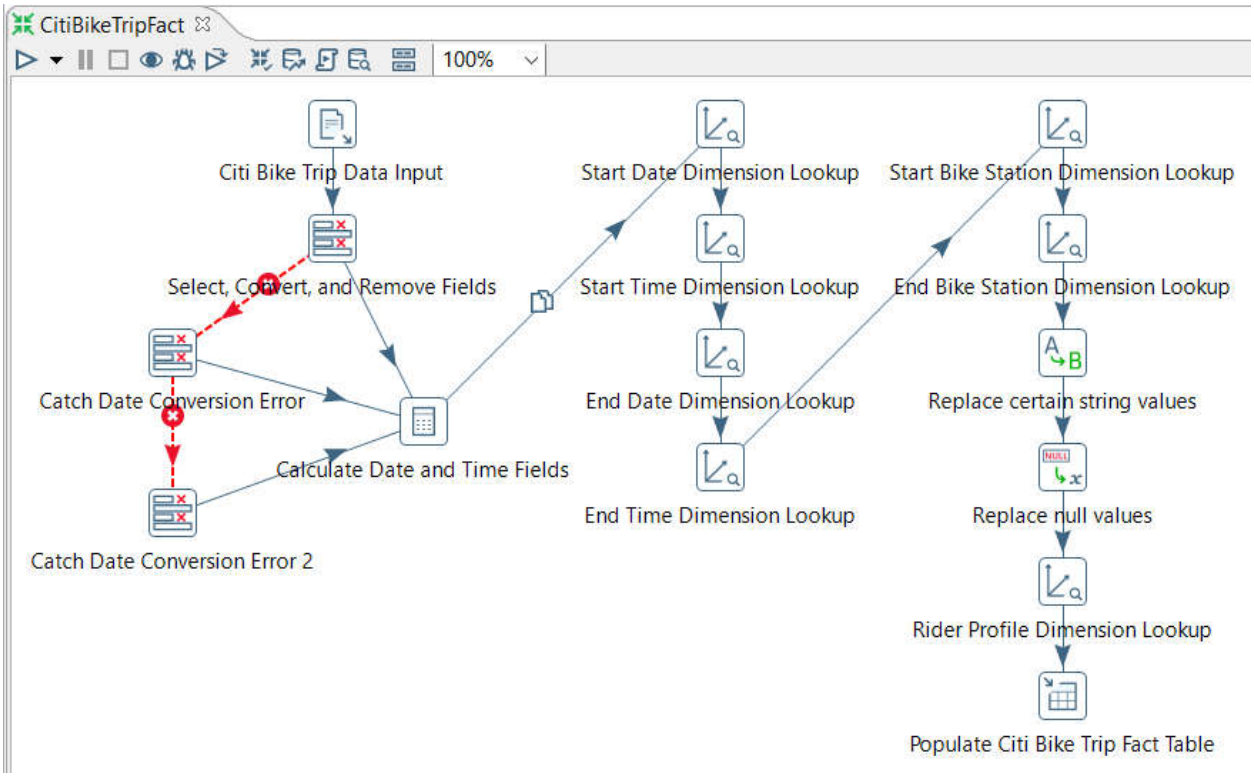
This transformation was used to create and populate the **Bike Station** dimension. This is a dimension from the **Citi Bike Trip History** data. Since there is a start and end bike station for every Citi Bike trip, it was necessary to use two *Select / Rename values* steps to separate the values of each bike station into two separate streams of data. The streams were then joined to be cleaned up using a *Replace null value* step, which replaced null strings with "*NA*" and null numerical values with *0*. The *Unique rows (HashSet)* step was used to ensure that only unique occurrences were sent to the dimension update step. This dimension was implemented as a Type 2 slowly-changing dimension with the start/end station ID from the source data used as the natural key.

Rider Profile Dimension



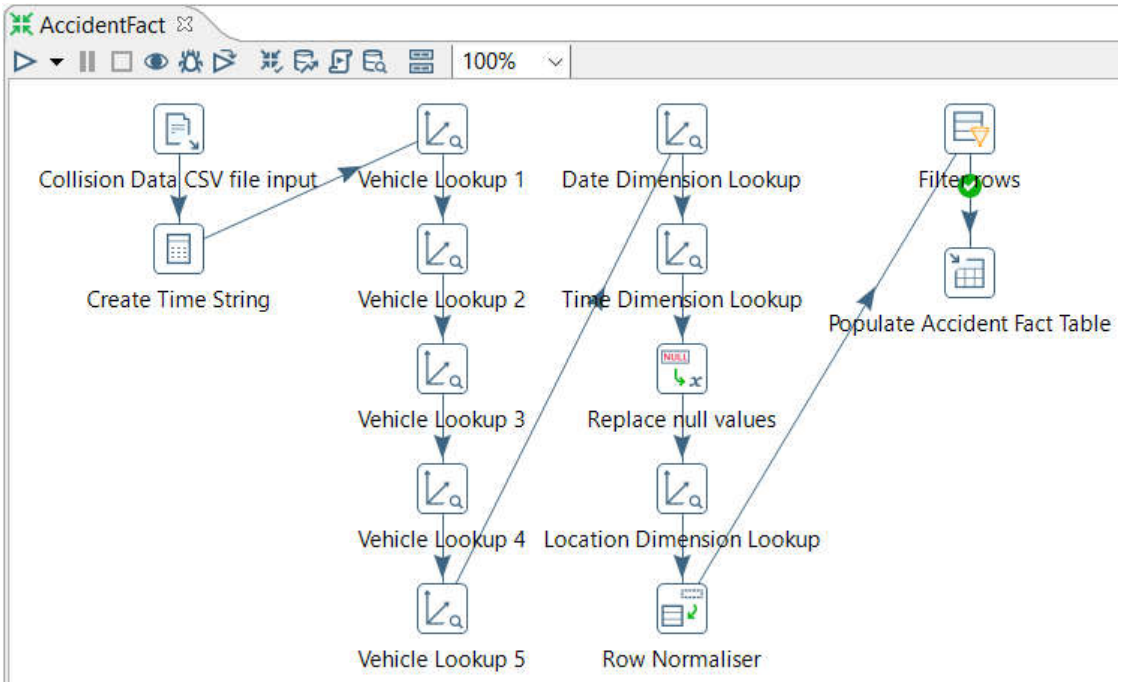
This transformation was used to create and populate the **Rider Profile** dimension. This is a dimension from the **Citi Bike Trip History** data. It populates the dimension with all unique occurrences of the *user type*, *birth year*, and *gender* values of a Citi Bike rider. To clean the data, it was necessary to use the *Replace in string* and *Replace null value* steps. The former performed actions such as changing *gender* value *1* to *male* and value *2* to *female*. The latter replaced null values with "*NA*" string values. The *Unique rows (HashSet)* step was used to ensure that only unique occurrences were sent to the dimension update step. This dimension was implemented as a Type 0 slowly-changing dimension as there was no natural key in the source data and there is no need to update the existing rows.

Citi Bike Trip Fact



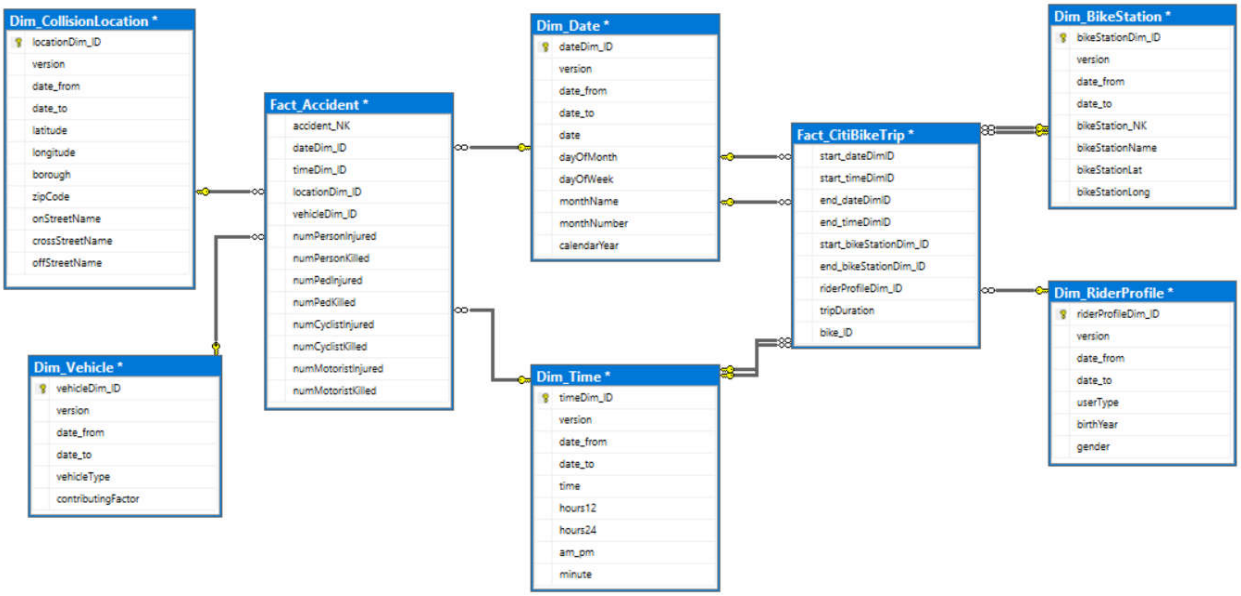
This transformation was used to create and populate the **Citi Bike Trip** fact table using the **Citi Bike Trip History** data. The *Text file input* step was used to ingest all of the .csv files that were contained in a folder. The *Select / Rename values* step was used to select only the columns necessary to perform the dimension lookups and to set the conversion mask for datetime fields. Since there were three different datetime formats across the files, it was necessary to use two error-catching *Select / Rename values* steps to be able to parse all of the values successfully. The *Calculator* step was used to break out the datetime fields into separated date and time columns. That was necessary in order to be able to perform the **Date** and **Time** dimension lookups that follow. The *Replace in string* step was used to replace gender values such as *0* to "*NA*", *1* to "*MALE*", and *2* to "*FEMALE*". The *Replace null value* step was used to set null numerical values to *0*, and null string values to *"NA"*. Finally, a *Table output* step was used to populate the fact table.

Accident Fact



This transformation was used to create and populate the **Accident** fact table using the **NYPD Motor Vehicle Collisions** data. The *Calculator* step was used to create a new column that contained the time value after applying a conversion mask and changing it to a string value type. This was necessary to facilitate the lookup to the **Time** dimension. A *Replace null value* step was used to set null numerical values to *0*, and null string values to *"NA"*. A *Row Normaliser* step was used to de-pivot the data by having each row repeat for every vehicle involved. It was then necessary to filter out the rows where the vehicle dimension lookup returned 0 (no vehicle) by using a *Filter rows* step. Finally, a *Table output* step was used to populate the fact table.

Final Schema

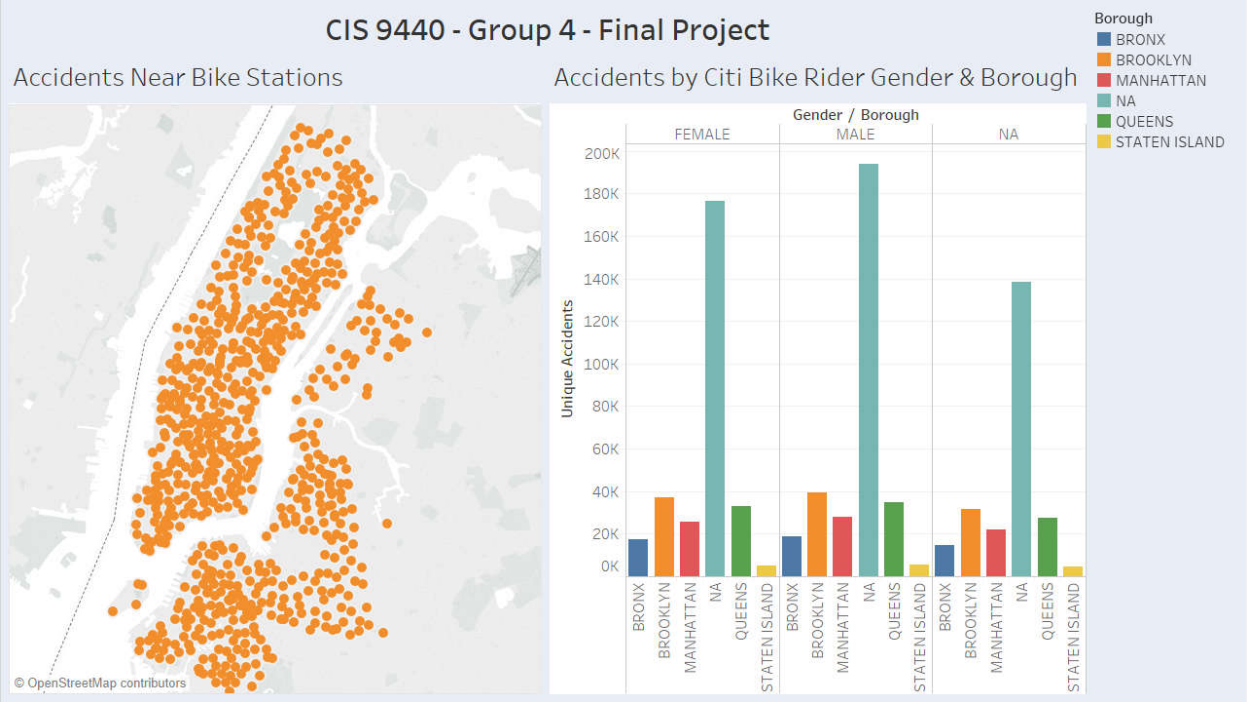


Above is the database diagram of the final schema of the data warehouse. It was generated in Microsoft SQL Server. When the tool was used, it only output the tables; the relationships between facts and dimensions had to be manually created. This is because the SQL that Pentaho Data Integration generated to create the tables did not specify foreign keys.

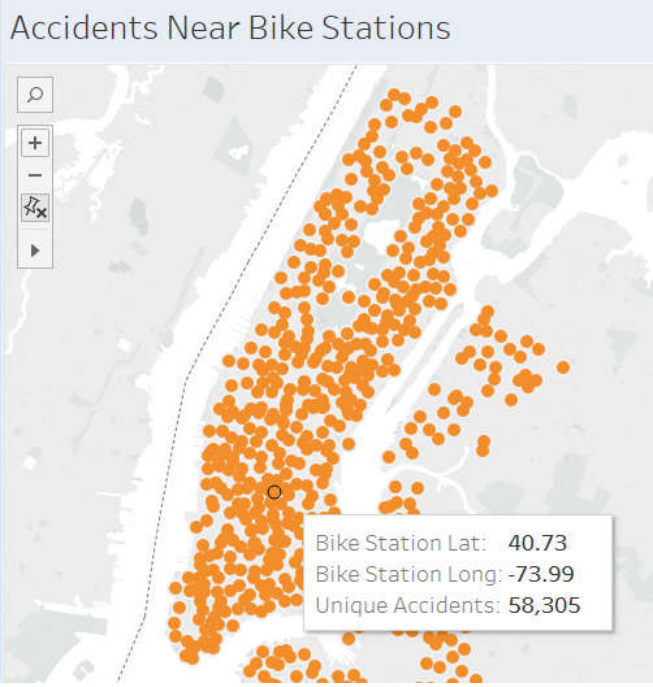
The final schema matches that of the final dimensional model with the exception of the addition of version and history tracking fields (*version*, *date_from*, *date_to*).

Dashboard Analytics

The dashboard and analytics were generated using Tableau.



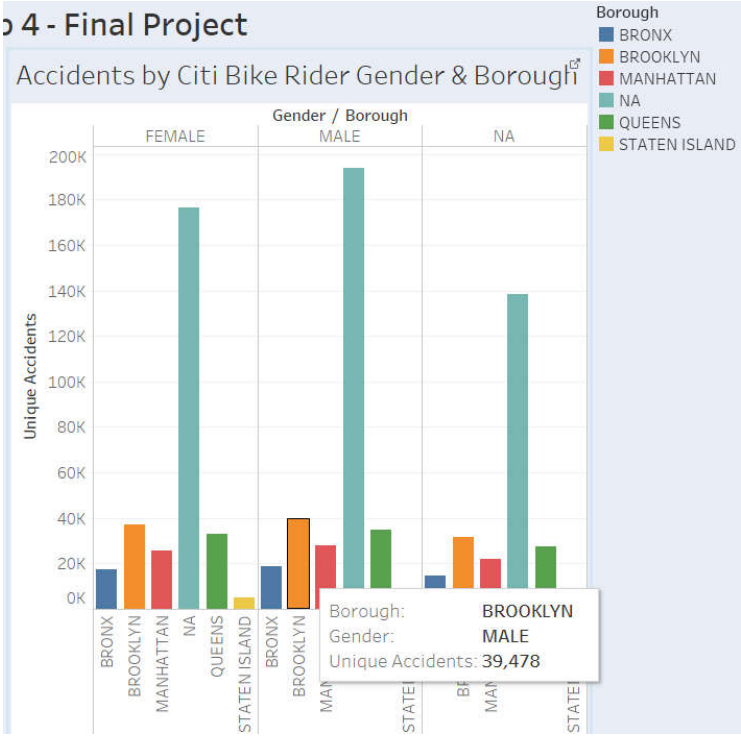
Accidents Near Bike Stations



This visualization plots all of the Citi Bike stations on a geographic map. When the cursor is hovered over the location of a Citi Bike station, the user can see how many accidents are associated with that particular station.

Since the accident facts repeat for every vehicle involved, the distinct count of accident natural keys (**accident_NK** from the **Fact_Accident** fact table) was used rather than the sum of rows of data.

Accidents by Citi Bike Rider Gender & Borough



This visualization shows the accident count by borough and Citi Bike rider gender.

Unfortunately, the majority of data did not have a borough recorded. However, it is possible to see that of the data with borough information, Brooklyn has the highest count of accidents. Secondly, accidents appear to have a higher correlation with male Citi Bike riders than female.

Description of Tools

Lucidchart

Lucidchart[3] is an online diagram application that was used to create the dimensional diagrams for our data warehouse. This tool was selected over other options, such as Microsoft Visio, because of the ease of collaboration.

Pentaho Data Integration

Pentaho Data Integration 7.1[4] is the application that was used to build the Extract, Load, and Transform (ETL) processes necessary to build the data warehouse. It is a component of the Pentaho Unified Data Integration and Analytics Platform.

Microsoft SQL Server 2017 Express

Microsoft SQL Server 2017 Express[5] was the database engine used to host the data warehouse on a local computer. It is provided for free by Microsoft and can host databases up to 10 GB in size.

Microsoft SQL Server Management Studio 17

Microsoft SQL Server Management Studio 17[6] was the application used to query and manage the data warehouse. It is provided for free by Microsoft.

Tableau

Tableau[7] was the application used to create data visualizations and the dashboard.

Conclusion

We encountered several challenges throughout the course of the project. First among them was the fact that the data for Citi Bike trips was provided in a series of files, divided by month. After downloading all of the available data, we had a total of 51 files. After some experimentation, we were able to figure out how to process all of them using the **Text File Input** step. However, this lead to the discovery of another issue; the date formats for all of the files were not consistent. This caused the ETL job that populated the **Citi Bike Trip** fact table to fail midway when parsing the dates. To resolve this, we needed to add two **Select / Rename values** steps to catch errors. Two error-catching steps were needed because there were three different date formats in the files. The ETL process was able to complete successfully after this change was made.

When populating the **Collision Location** dimension table, we queried the database to view the results and noticed that there appeared to be many duplicates. We could not immediately figure out why as we had included steps in the transformation to ensure that only unique locations were sent to the dimension table. We eventually realized that null values were the root cause of the problem. We added a *Replace null value* step and this resolved the issue. After this, we added the same step to other ETL processes to prevent null values from creating any additional problems.

Dimension lookups to the **Date** and **Time** dimensions were a challenge when populating the two fact tables. When previewing the data stream in Pentaho Data Integration, we could not see an error as the date and time formats had been properly matched between the fact and dimension tables. It was necessary to use conversion masks in some cases to match these fields. In addition, we were certain that all dates and times used in the source data were present in the dimensions. However, we noticed that in the preview of the data stream, the date and time dimension lookup values were 0. This indicated that the lookups were not working as expected, despite the key values matching. To troubleshoot further, we took a look at the data in the Microsoft SQL Server database. While there, we noticed that the date/time formats seen in Pentaho did not match that of the records in the database despite the use of conversion masks. To work around this and resolve the lookup issue, we decided to use date and time values in string format.

When populating the **Fact_Accident** table, the ETL process was very resource intensive and took several hours to complete. Despite running it for hours, the transformation ultimately failed on the first attempt. Upon review of the error log, we were able to see that the problem was related to inadequate memory. To resolve this, we edited the **Spoon.bat** file used to launch Pentaho Data Integration to double the amount of memory available to it; we raised the limit from 2 GBs to 4 GBs. In addition, we changed the value of the commit size of the *Table output* step from 1,000 to 200,000 to significantly reduce the trips to the database engine and improve performance. The instructions for increasing available memory were found in the Pentaho documentation available online.[8] These changes allowed us to fully execute the fact table transformation.

We did not experience much difficulty with several aspects of the project. Creating the dimensional model was fairly straightforward upon review of the available columns in the data. With the exception of the problems described above, the ETL processes were easy to implement. This was especially true for the dimension ETL processes after the first few were created. Some trial and error was required, but that was expected. Creating the visual analytics in Tableau required some trial and error with the table joins, moving of measure and dimension fields, and generation of calculated fields.

However, that was part of the learning process and we were able to complete it without too much effort.

If we were to recreate this data warehouse, we would make several changes. First, we would have used the *Combination lookup/update* step to maintain certain dimensions, such as **Date** and **Time**, as true Type 1 slowly-changing dimensions without the need for columns such as *version*, *date_from*, and *date_to*. We also would have sought ways to optimize the populating of the **Fact_Accident** table; it took several hours to complete and had to be done multiple times after issues were discovered, such as a lookup not functioning properly. For ease of collaboration, we also would have hosted the data warehouse on a cloud. However, we're uncertain if that would have resulted in performance issues due to the volume of data that we were working with. Lastly, we would have created a job to run through all of the transformations rather than run them individually.

The proposed benefits of the data warehouse can be realized, as made evident particularly by the data visualizations. Further review in Tableau or other visual analytics tools can yield information that can be used to drive the New York City government's decision making on the issue of road safety. While we only explored a small subset of the questions and KPIs listed in the introduction, the data warehouse is capable of much more. In order to ensure accuracy and to draw the appropriate conclusions, it may be necessary to modify the table joins between the available fact and dimension tables. It would also be a good idea to review that all of the data fields are correctly configured in Tableau, such as when we changed the latitude and longitude number values to be read as geographic latitude and longitude values. We have no doubt that there is a long list of interesting conclusions that have yet to be drawn from the data warehouse.

References

1. International, Inc. Motivate. “Citi Bike System Data.” Citi Bike NYC, www.citibikenyc.com/system-data.
2. Calgary, Open. *NYPD Motor Vehicle Collisions | NYC Open Data*, data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95.
3. “Flowchart Maker & Online Diagram Software.” *Lucidchart*, 18 Dec. 2017, www.lucidchart.com/.
4. “Data Integration.” *Pentaho*, www.pentaho.com/product/data-integration.
5. “SQL Server 2017 Express edition.” *Microsoft SQL Server - US (English)*, www.microsoft.com/en-us/sql-server/sql-server-editions-express.
6. Stevestein. “Download SQL Server Management Studio (SSMS).” *Microsoft Docs*, docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms.
7. “Tableau Software.” *Tableau Software*, www.tableau.com/.
8. “Increase the Spoon Memory Limit.” Pentaho Documentation, 3 Nov. 2016, help.pentaho.com/Documentation/6.1/0H0/070/020/010.