DIVVY BIKE SHARING ANALYSIS

**PowerBI Report**

by

William Gan

Table of Contents

[1. BI Data Source and Requirements 4](#_Toc195805721)

[1.1. Introduction to the Dataset 4](#_Toc195805722)

[1.2. Business Objective 4](#_Toc195805723)

[1.3. Key Performance Indicators (KPI) 5](#_Toc195805724)

[1.4. Key User Groups 5](#_Toc195805725)

[1.5. Importing Data Source 6](#_Toc195805726)

[1.6. Overview of the Dataset 8](#_Toc195805727)

[2. Data Cleansing and Pre-Processing 10](#_Toc195805728)

[2.1. Trimming the whitespaces of text fields 11](#_Toc195805729)

[2.2. Removing Blank Rows and Duplicates from the table 12](#_Toc195805730)

[2.3. Filtering the <ride\_id> column (M Language) 13](#_Toc195805731)

[2.4. Filtering the <rideable\_type> and <member\_casual> column 14](#_Toc195805732)

[2.5. Changing the data type of <started\_at> and <ended\_at> columns 15](#_Toc195805733)

[2.6. Changing the data type of columns containing coordinates 16](#_Toc195805734)

[2.7. Removing errors from the table 17](#_Toc195805735)

[3. Data Modelling and Relationship 18](#_Toc195805736)

[3.1. Creation of two ID columns in the <trips> table (M Language) 18](#_Toc195805737)

[3.2. Creating the <datetime> table 20](#_Toc195805738)

[3.3. Creating the <coordinates> Table 27](#_Toc195805739)

[3.4. Creating the <stations> table 28](#_Toc195805740)

[3.5. Transforming the <trips> table into a fact table 35](#_Toc195805741)

[3.6. Data Modelling Relations and Summary 36](#_Toc195805742)

[3.7. Filtering out invalid durations 39](#_Toc195805743)

[4. Executive Summary 41](#_Toc195805744)

[5. Data Analysis 42](#_Toc195805745)

[5.1. Introduction 42](#_Toc195805746)

[5.2. Analysis of Total Trips 44](#_Toc195805747)

[5.3. Analysis of Total Ride Duration (New column using DAX) 46](#_Toc195805748)

[5.4. Analysis of Median Ride Duration 48](#_Toc195805749)

[5.5. Distribution of Bike Usage by User Type 50](#_Toc195805750)

[5.6. Weekly Distribution of Ride Duration (New measure using DAX) 52](#_Toc195805751)

[5.7. Data Visualization Dashboard (New measure with DAX) 55](#_Toc195805752)

[6. Conclusions and Recommendations 56](#_Toc195805753)

[6.1. Pricing Information of Bike Sharing Service 56](#_Toc195805754)

[6.2. Correlation Between Usage Patterns and Pricing 57](#_Toc195805755)

[6.3. Recommendations and Justification 58](#_Toc195805756)

[7. References 61](#_Toc195805757)

**Gan Hansong**

**BUSINESS INTELLIGENCE DESIGN SCOPE**

***Divvy Bike Sharing Analysis***

# BI Data Source and Requirements

## Introduction to the Dataset

The dataset used in this analysis is sourced from <https://www.kaggle.com/datasets/hswg94/chicago-bike-rental/data> that is originated from Divvy Bikes at <https://divvybikes-marketing-staging.lyft.net/system-data>. Divvy bikes is a popular bike sharing service in Chicago, Illinois. It includes trip related data, such as user type (member or casual), start and end datetimes and the type of bike used (classic or electric). Analyzing this dataset brings an understanding to usage behavior of the bike sharing service and allows to identify key areas that require improvement. Additionally, it serves as a foundation dataset that can be used for performing and conducting data analysis for identifying patterns and trends to create strategic decisions, enhance service quality and understand customer demands.

## Business Objective

The objective of this project is to analyze the data and identify strategies that can aid the increase of non-member to membership conversions and attract new customers to utilize the bike sharing service. In the analysis, the specific questions will be answered and followed by providing recommendations.

* How do annual members and casual riders use the bike service differently?
* Why would casual riders switch to purchase memberships?
* How to influence new customers to utilize the bike sharing service?

## Key Performance Indicators (KPI)

To effectively measure the success of the business objectives, the following KPIs were identified:

* Total Number of Trips - Measures the overall usage of the bike-sharing service.
* Member vs. Casual Trips - Tracks the proportion of trips made by members compared to casual users.
* Ride Duration - Analyzes the typical duration of rides, segmented by user type and bike type.
* Peak Usage Times - Identifies the times of day and days of the week when the service is most utilized.

## Key User Groups

This report may serve useful for various stakeholders within the organization to aid in decision making.

For example, business analysts may use the insights to understand usage patterns and support strategic decision-making, while marketing teams will identify target segments and develop campaigns to increase memberships and attract new users. High-level executives could make high-level strategic decisions based on the insights generated from the report.

## Importing Data Source

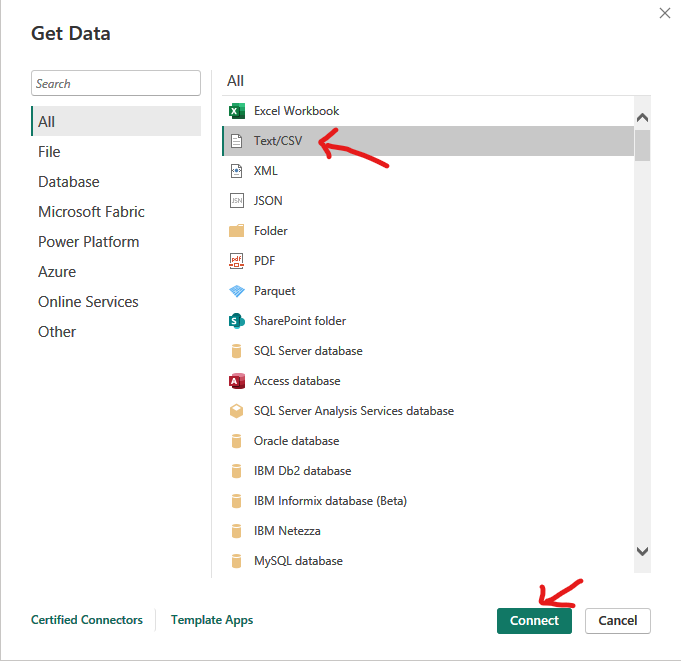


Figure - Importing Data as Text/CSV

The initial steps begin by connecting to the dataset file **chicago\_bike\_sharing\_service.csv** with the steps shown in Figure 1.

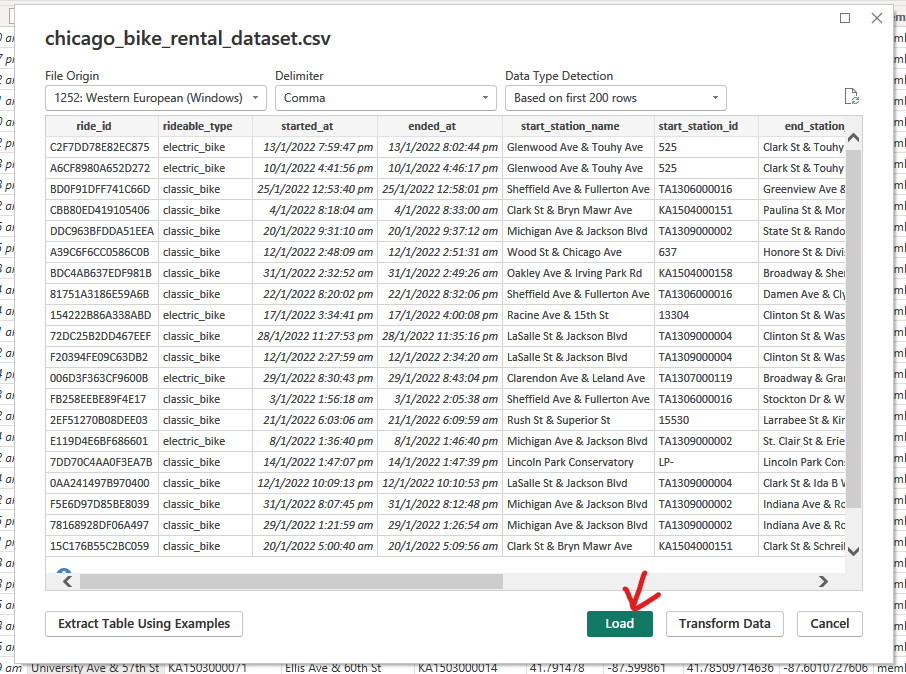


Figure - Loading Dataset

After the dataset is connected, load the data. As shown in Figure 2, the **chicago\_bike\_rental\_dataset.csv** file will be previewed, allowing to verify the data structure and content.

## Overview of the Dataset

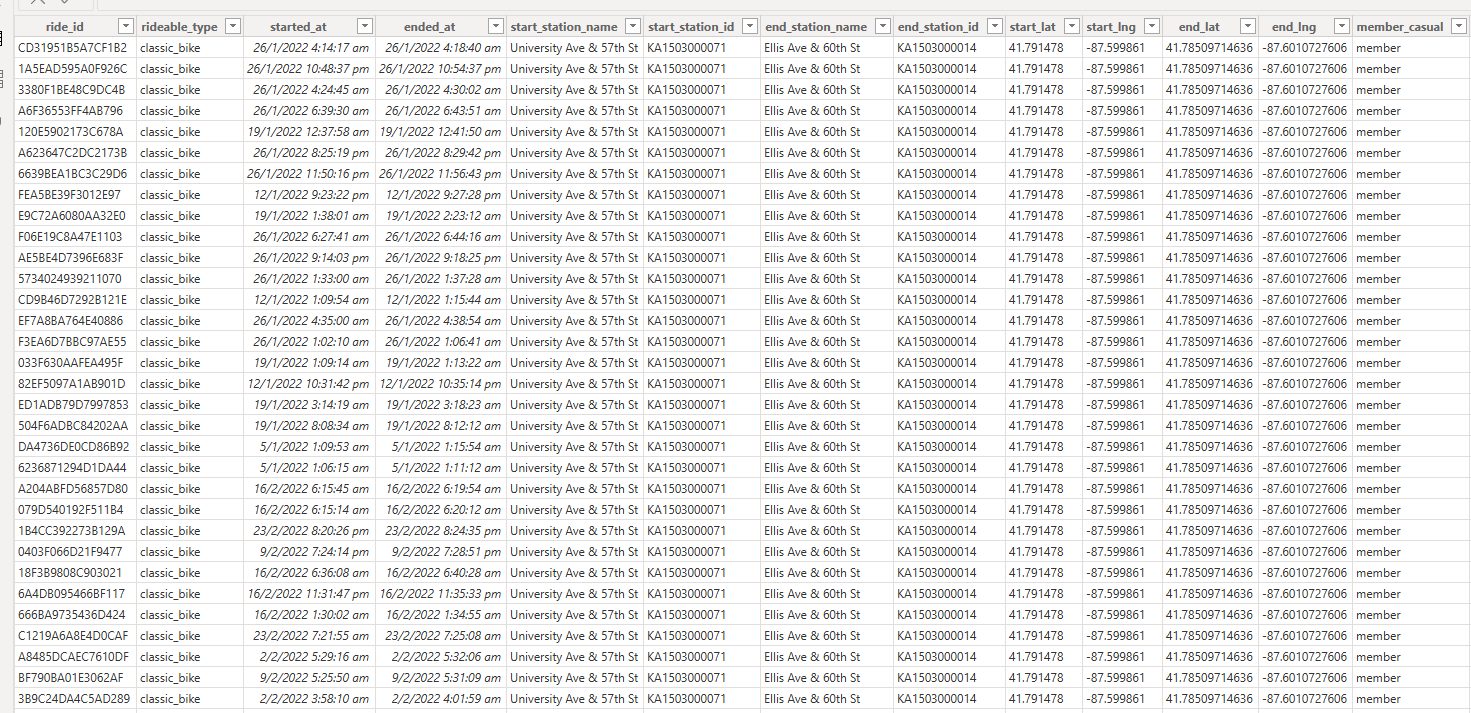


Figure - Overview of the Dataset

Figure 3 is a preview of the dataset that was loaded, details of the columns can be found in Table 1, including the column name and its description.

|  |  |
| --- | --- |
| **Data Description** | |
| **Column Name** | **Description** |
| ride\_id | Unique identifier for each ride |
| rideable\_type | Type of bike used for the ride |
| started\_at | Timestamp when the ride started |
| ended\_at | Timestamp when the ride ended |
| start\_station\_name | Name of the station where the ride started |
| start\_station\_id | Unique identifier for the start station |
| end\_station\_name | Name of the station where the ride ended |
| end\_station\_id | Unique identifier for the end station |
| start\_lat | Latitude of the start location |
| start\_lng | Longitude of the start location |
| end\_lat | Latitude of the end location |
| end\_lng | Longitude of the end location |
| member\_casual | Type of user (member or casual) |

Table - Data Description

# Data Cleansing and Pre-Processing

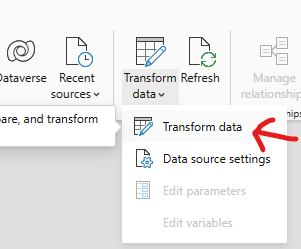


Figure - Transform Data Menu

To start off the data preparation process, the feature ‘transform data’ in PowerBI as shown in Figure 4 is used, it opens a window that allows allow performing actions related to data preparation, such as data cleansing, and it will be primarily used to prepare the data before performing the steps for data modelling.

The process of data cleansing and pre-processing is to acquire a clean and reliable dataset that is ready for producing accurate analysis and report, the goal is to free it from inconsistencies and errors.

## Trimming the whitespaces of text fields

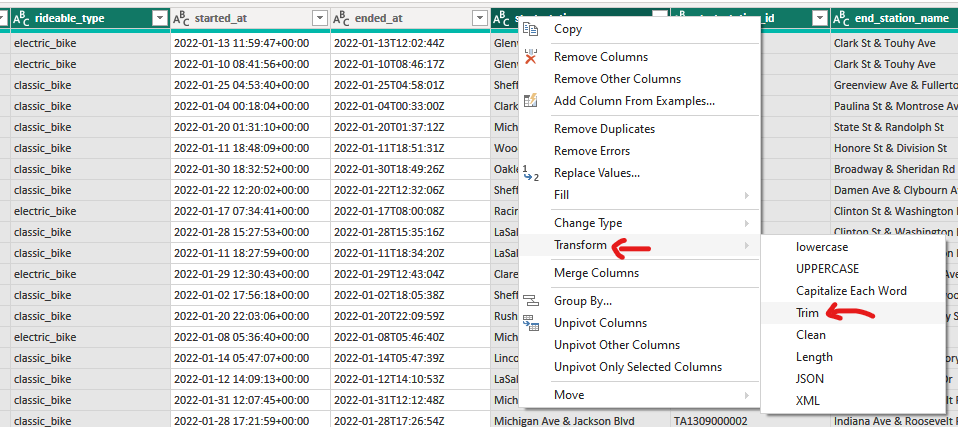


Figure - Trimming Whitespaces

The first step in the data cleaning process involves trimming whitespaces from text fields to remove any leading, trailing, or double spaces within the fields (Figure 5). This helps maintain data consistency by preventing unintended issues with entries being treated as distinct values.

## Removing Blank Rows and Duplicates from the table

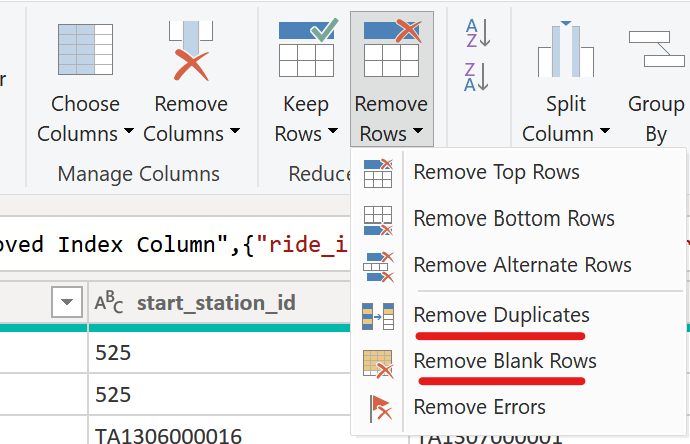
****

Figure – Removing Blank Rows and Duplicates

To ensure that all entries are complete and to prevent any errors during analysis, blank rows and duplicates were removed from the table, illustrated in Figure 6. This is essential for maintaining the accuracy of any subsequent calculations and analysis.

## Filtering the <ride\_id> column (M Language)

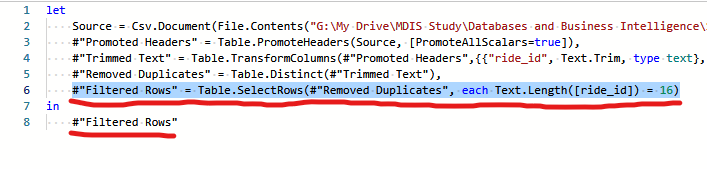


Figure – Filtering Column to 16 Characters in Length

The following lines shown in Figure 7 was added in the advanced editor to display only rows where the <ride\_id> column contains 16 characters. This ensures that all entries are of the correct length, and to eliminate any invalid entries while maintaining integrity.

## Filtering the <rideable\_type> and <member\_casual> column

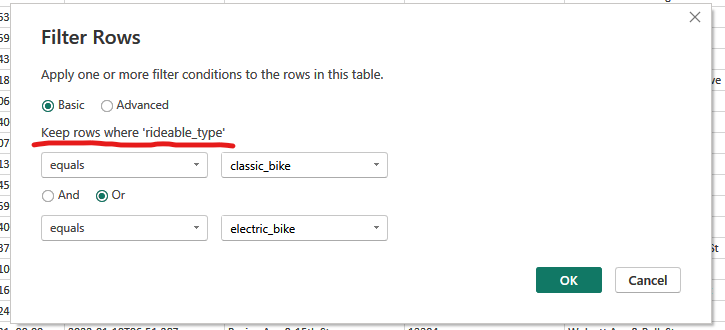


Figure - Filtering the columns to include only relevant data

Filtering the <rideable\_type> column shown in Figure 8 ensures that only the relevant bike types are included in the analysis. Keeping only rows that are either "classic\_bike" or "electric\_bike" will exclude any irrelevant or dirty data, which could skew the analysis results. Likewise, the <member\_casual> column was filtered to include only rows where the field were either "casual" or "member"

## Changing the data type of <started\_at> and <ended\_at> columns

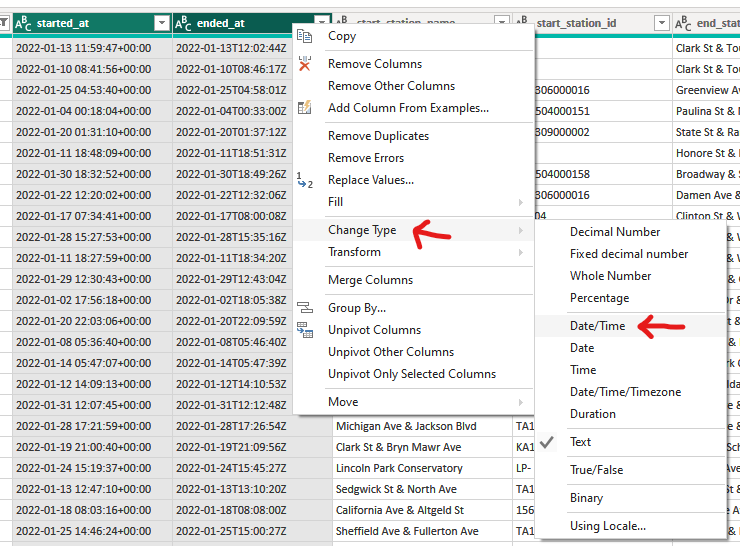


Figure - Changing the data type of <started\_at> and <ended\_at> columns

To ensure proper representation and functionality of date and time related calculations, the data types of the <started\_at> and <ended\_at> columns were changed to Date/Time. Figure 9 illustrates this process, which is crucial for extracting day information and calculating ride durations.

## Changing the data type of columns containing coordinates

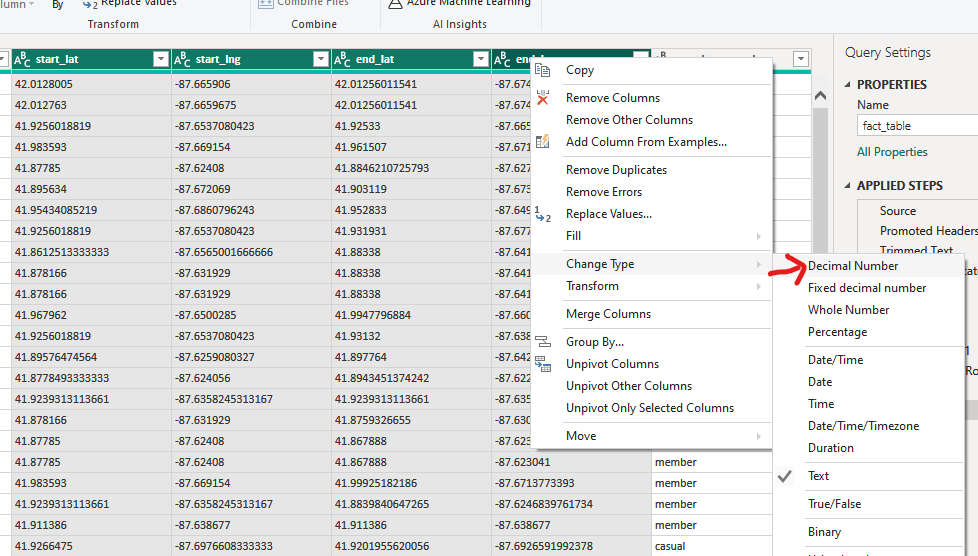


Figure - Adding the day\_of\_week\_name column using a custom column formula

The data type of the coordinate related columns was changed to better represent the type of data as geographical and mapping related tools may require a decimal format to function as intended, converting the data type is essential to ensure compatibility. Figure 10 illustrates the process of changing the data type to Decimal Number.

## Removing errors from the table

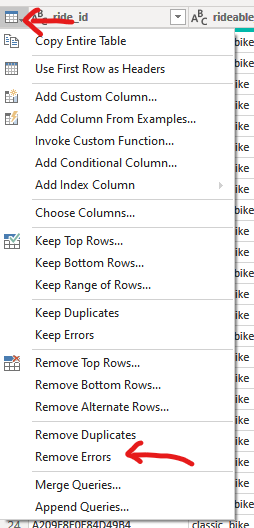


Figure - Removing Errors from the Table

The final step in the data cleansing process is to remove any errors from the table as shown in Figure 11. This step is critical to ensure the integrity and accuracy of the dataset. Errors can occur during data cleaning, such as converting the data types on fields with dirty data. Removing these errors ensures that any issues introduced during the cleansing process are resolved before finalizing the dataset.

# Data Modelling and Relationship

## Creation of two ID columns in the <trips> table (M Language)

In this step, columns for unique identifiers of the coordinates and datetime information is created to serve its purpose in establishing a clear relationship to the <trips> tables. This step helps in organizing the data effectively and to facilitates future querying and analysis.

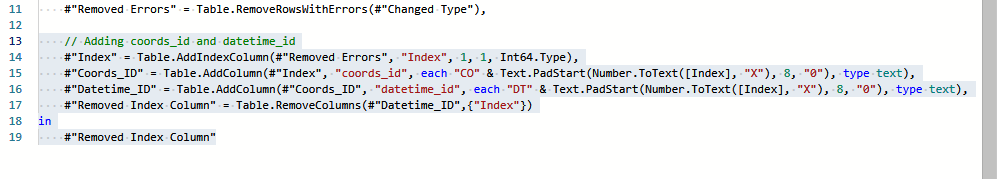


Figure – Creation of Columns with M Language

In the advanced editor, the highlighted text shown in Figure 12 are lines of code to generates unique IDs, this is to provide a basis for the new IDs.

The <coords\_id> field is created by concatenating a custom prefix "CO" with values from the created index column in line 14, the indexes are then converted to hexadecimal and padded onto 8 digits before the concatenation, this is done to create unique identifiers for the coordinate fields.

A similar process was replicated to create the <datetime\_id> column using the prefix “DT”, shown in line 16

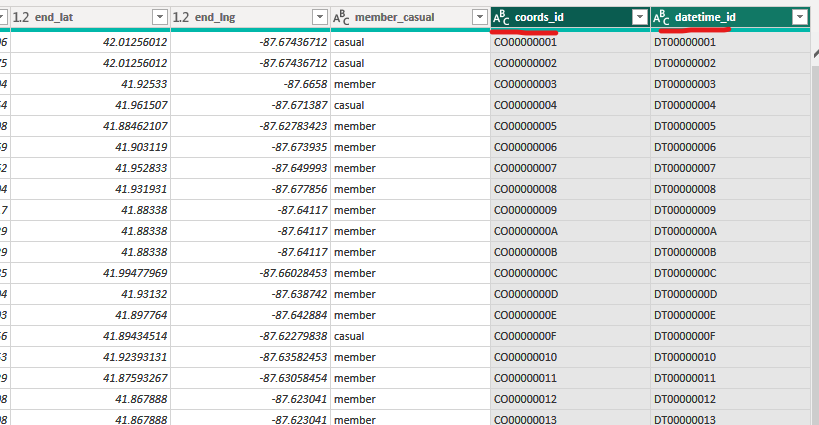


Figure - Updated Table with Coordinate and Datetime Identifiers

The highlighted columns in Figure 13, <coords\_id> and <datetime\_id> were results of the columns created into the <trips> table using M language shown in Figure 12. This concludes the creation of the two IDs.

## Creating the <datetime> table

### Duplicating Table and Removing Columns

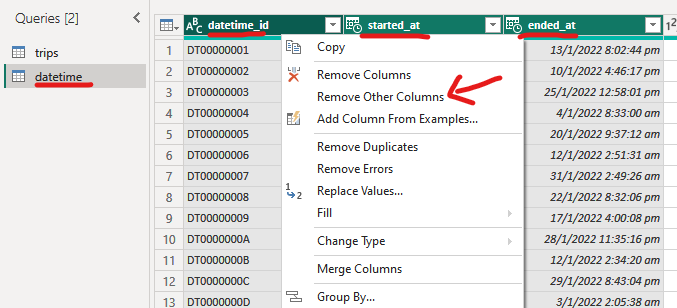


Figure - Duplicating the <trips> Table and Removing Columns

The datetime table is created to separate the date and time related information of trips into a dedicated table to help reduce redundancy and improve efficiency of queries. The trips table was duplicated, with all other columns removed, except the columns, this process is shown in Figure 14.

### Adding <duration\_sec> column (custom column formula).

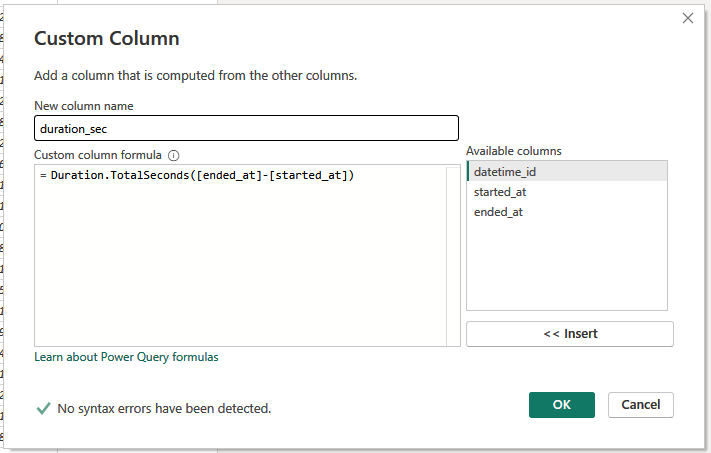


Figure - Adding the <duration\_sec> Column

Figure 15 demonstrates the addition of a custom column named <duration\_sec> using a custom column. The formula calculates the duration of each ride in seconds by subtracting the started\_at timestamp from the ended\_at timestamp and converting the result to seconds using the “Duration.TotalSeconds()” function. The purpose of this <duration\_sec> column is to provide a measurement of the duration of each trip in seconds.

### Adding <day\_of\_week> column (custom column formula).

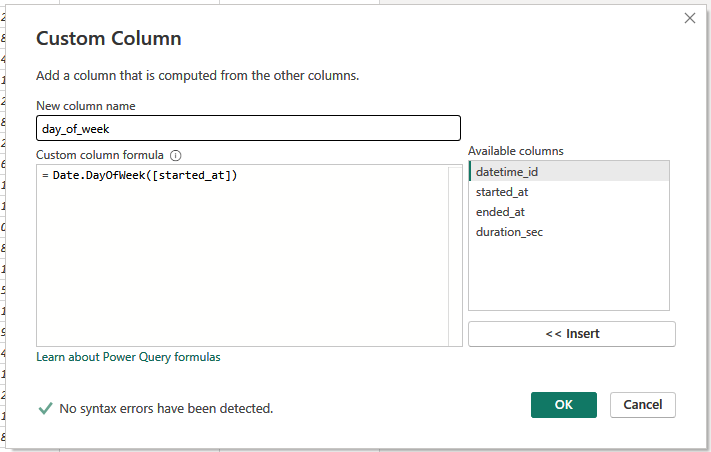


Figure - Adding the <day\_of\_week> Column

The image shown in Figure 16 is the step taken to produce a custom column named <day\_of\_week> using a custom column formula. The formula “Date.DayOfWeek([started\_at])” is used to extract the day of the week from the “started\_at” timestamp. The purpose of this <day\_of\_week> column is to identify the day of the week for the start of each trip.

### Changing data type of <duration\_sec> and <day\_of\_week> columns.

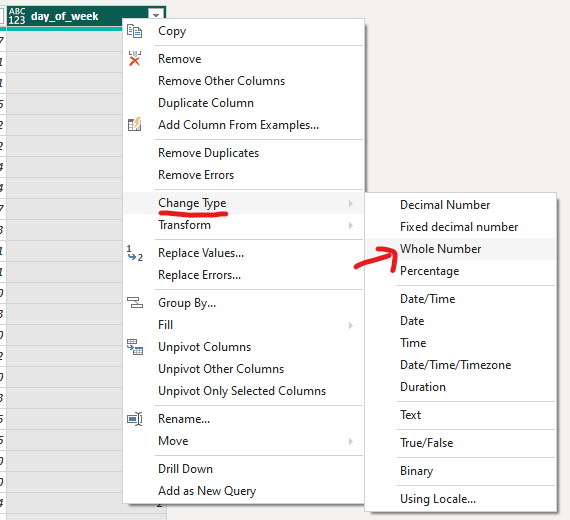
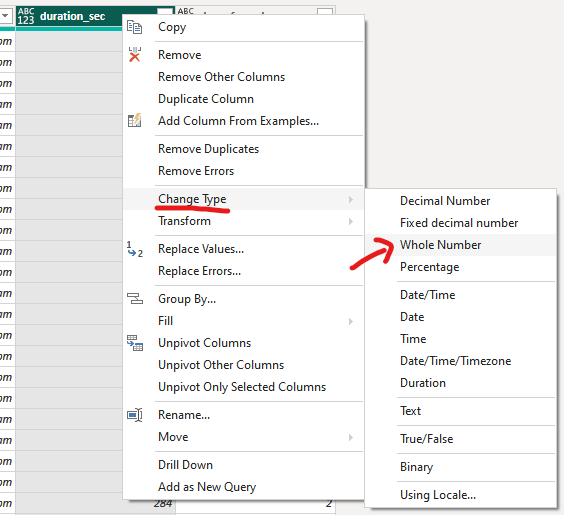


Figure - Changing Data Type of <duration\_sec> and <day\_of\_week>

The image in Figure 17 illustrates the process of changing the data types of the <duration\_sec> and <day\_of\_week> columns. The <duration\_sec> is changed to store the duration which are calculated in seconds as integers. The <day\_of\_week> column is changed to ensure that the day of the week is stored as an integer, with values ranging from 0 (Sunday) to 6 (Saturday). This is to facilitate compatibility with datetime related data processing and analysis functions, such as data sorting that expect these values to be in integer format.

### Adding <day\_of\_week\_name> column (custom column formula).

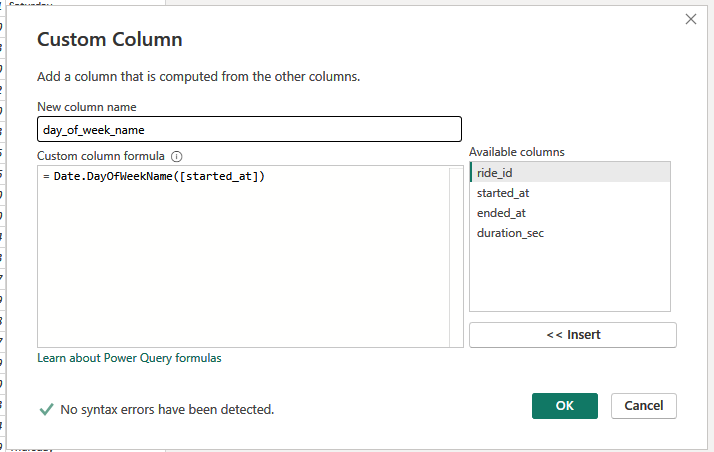


Figure - Adding the <day\_of\_week\_name> Column

The image in Figure 18 shows the addition of a custom column named <day\_of\_week\_name> using a custom column formula. The formula “Date.DayOfWeekName([started\_at])” is used to extract the name of the day of the week from the <started\_at> column.

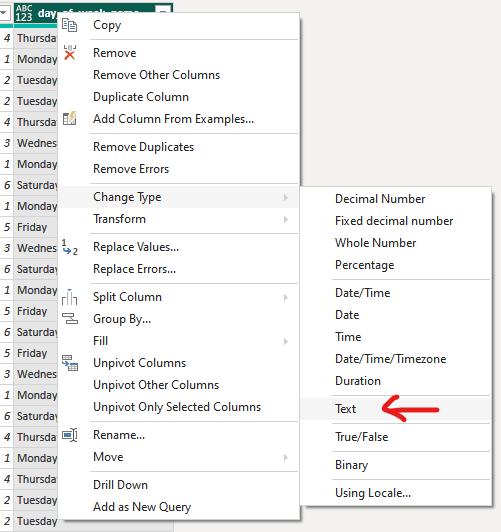


Figure - Changing Data Type of <day\_of\_week\_name> Column

Figure 19 demonstrates the process of changing the data type of the <day\_of\_week\_name> column to "Text" to better represent the data as day names are treated as text strings rather than any other data type, which aligns with their nature as names.

### Overview of the <datetime> table

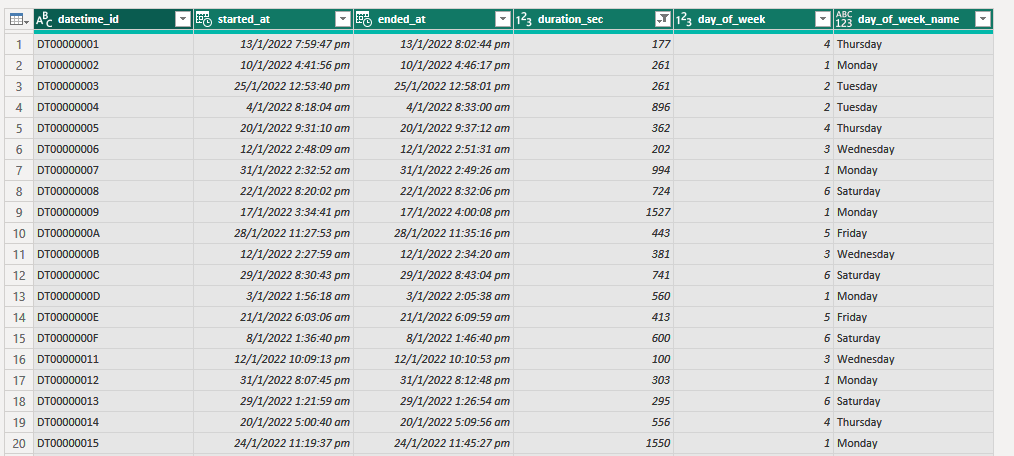


Figure - Overview of the <datetime> Table

Figure 20 provides an overview of the <datetime> table from the outcome of the data transformation process. The table is now structured in a way to facilitate data analysis, such as identifying trends based on the day of the week, calculating average ride durations etc.

## Creating the <coordinates> Table

### Duplicating <trips> table and Removing Columns

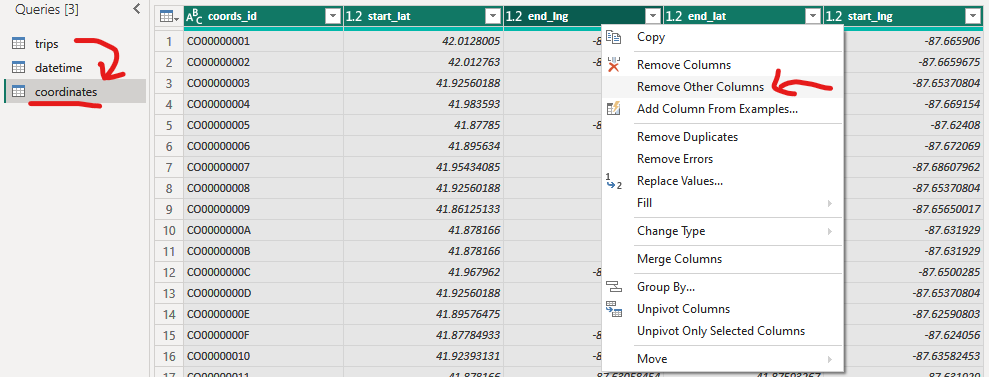


Figure - Duplicating <trips> table and Removing Columns

Figure 21 shows the process of creating a <coordinates> table by duplicating the <trips> table and removing unnecessary columns.

### Overview of the <coordinates> table

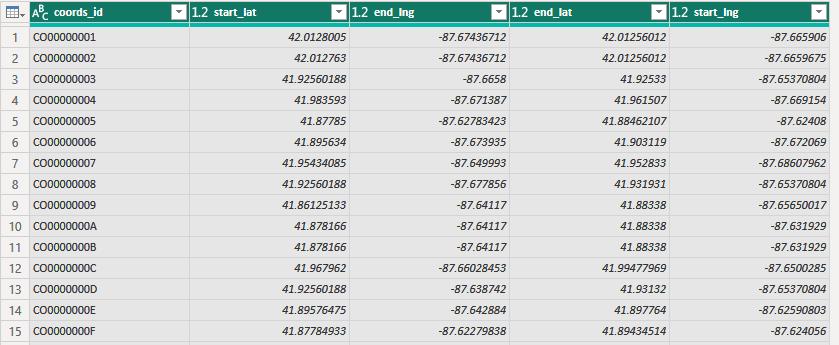


Figure - Overview of the <coordinates> Table

This outcome of the <coordinates> table is shown in Figure 22, it contains the geographical coordinates associated with the start and end points of each ride.

## Creating the <stations> table

### Duplicating the <fact\_table> and rename to <stations>

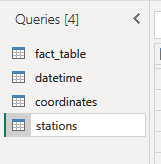


Figure - Duplicating and Renaming the <fact\_table>

Figure 23 illustrates the initial step in creating the <stations> table by duplicating and renaming the <fact\_table>. This goal of this table is to provide unique records for each station, this table provides unique station IDs and its corresponding name. The purpose of this table is to reduce data redundancy and improve data integrity

### Constructing the <stations> table (M Language)

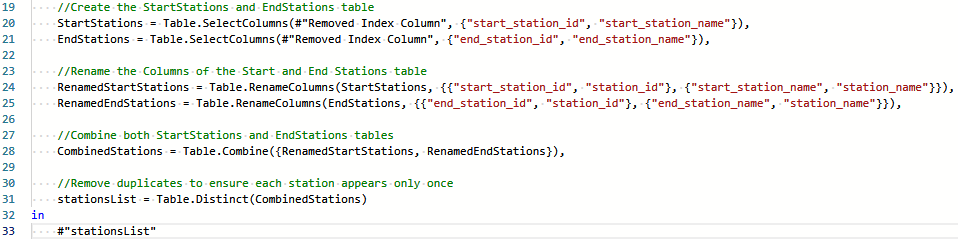


Figure – Using M Language to construct the <stations> table

Figure 24 shows the M Language code used to construct the <stations> table. First, columns for start and end stations are selected from the original table and renamed to have a consistent naming convention for station\_id and station\_name. Next, the renamed start and end stations columns are appended together. Finally, duplicates are removed.

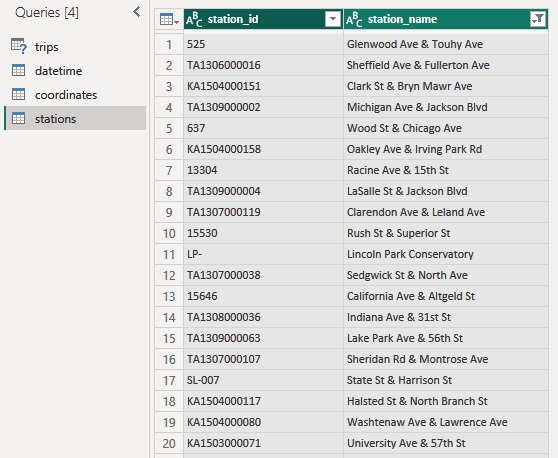


Figure - Result of <stations> Table with M Language Script

Figure 25 shows the outcome after running the M Language script to construct the <stations> table. The resulting table includes columns for station\_id and station\_name, that should display unique records for each station.

### Ensuring the values in “station\_id” column is unique (DAX Queries)

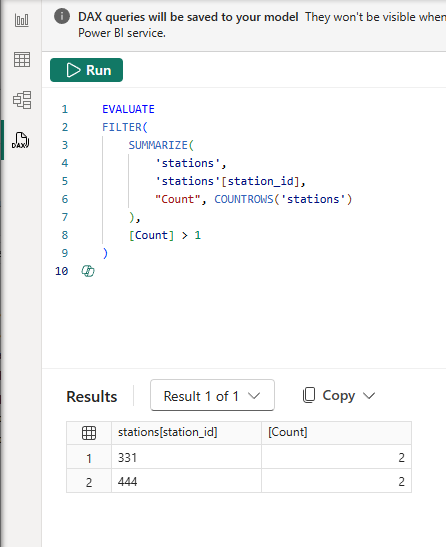


Figure – Dax Query for identifying non unique station id values

Before finalizing the creation of the <stations> table, it is crucial to identify any non-unique IDs to prevent potential errors. The following DAX query shown in Figure 26 was used to discover any non-unique values in the station\_id column. The query results reveal that station\_id values "331" and "444" each have two entries, the duplicates will need to be investigated and resolved. Having unique station IDs is crucial to maintain data integrity and to allow forming a relationship with the fact table <trips>.

### Investigating duplicate station\_id <331>

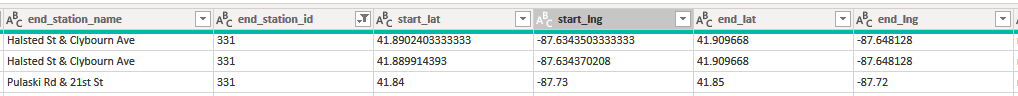


Figure – Investing the names and coordinates of station ID 331

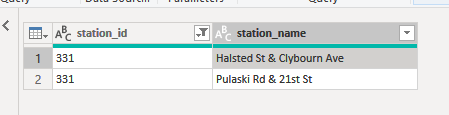


Figure – Station Names that are found under Station ID 331

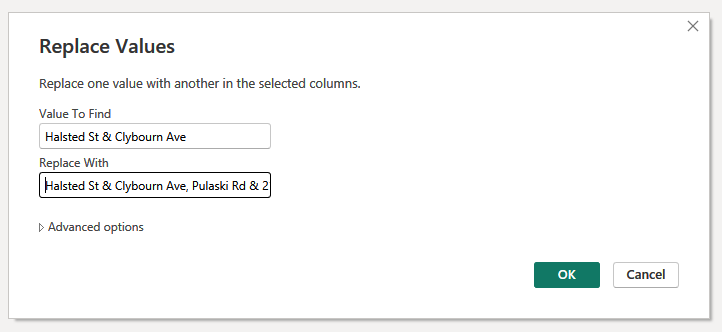


Figure – Merging the Station Names

During the investigation of the duplicate station\_id <331>, it was observed that the coordinates for "Halsted St & Clybourn Ave" and "Pulaski Rd & 21st St" are in proximity (shown in Figure 27), this likely have led to a glitch that may have caused them to be assigned the same station\_id. To correct this, the two locations are combined under a single ID, with the name updated to "Halsted St & Clybourn Ave, Pulaski Rd & 21st St", this step is shown in Figure 29 to accurately reflect both locations. The redundant record was then removed in section 3.4.6.

### Investigating duplicate station\_id <444> (DAX Queries)

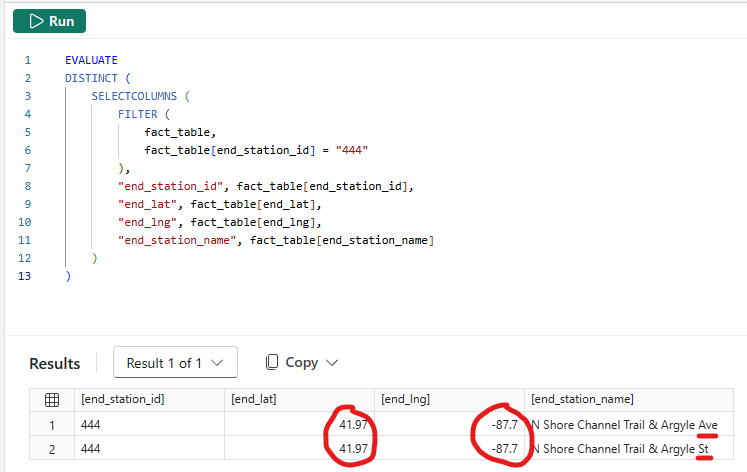


Figure – Dax Query for identifying non unique station id values

The following DAX Query shown in Figure 30 is executed to identify the station names and coordinates associated with station\_id <444>. The findings indicate that there is a high likelihood for the error in the name as the coordinates are nearly identical.

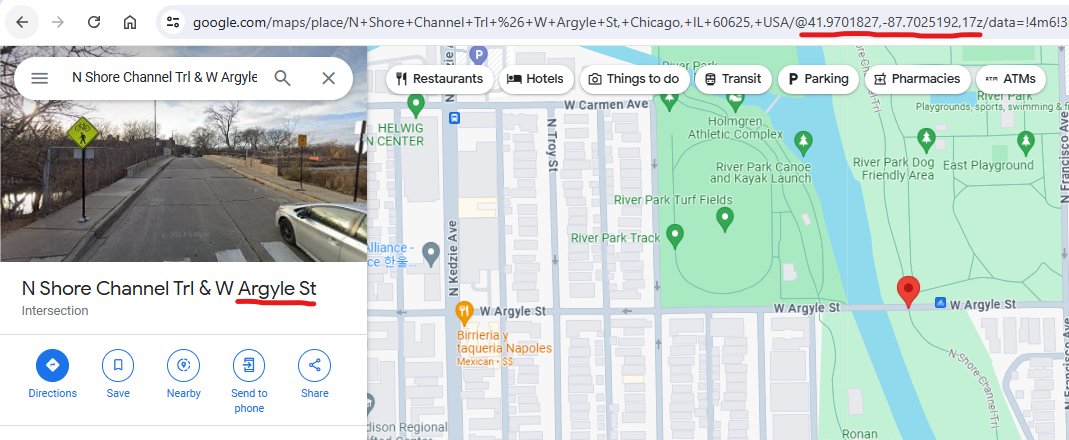


Figure – Identifying the correct station name

The coordinates placed in google maps shows the location pointing to Argyle St, which should be the correct name of the station as shown in Figure 31.

### Removing the two invalid records from <stations> table

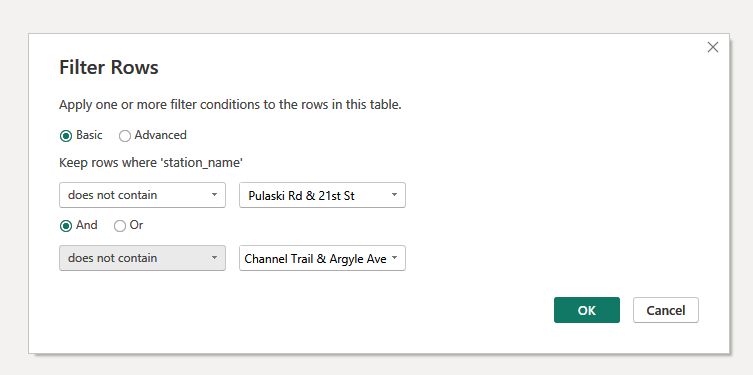


Figure – Removing any invalid records by filtering rows

To remove the two invalid records from the <stations> table, the "Filter Rows" dialog box shown in Figure 32 is utilized. The filter conditions are then set to exclude rows where the station\_name contains "Pulaski Rd & 21st St" or "Channel Trail & Argyle Ave". These filter conditions will effectively removing the duplicate entries highlighted at section 3.4.4 and section 3.4.5 from the table.

### Conforming the data in <station\_id> column is unique (DAX Queries)



Figure - Using DAX queries to ensure the station\_id column contains unique values

DAX Query as shown on Figure 33 is used to verify the uniqueness of the station\_id column. The resulting empty table confirmed that all station\_id values are unique, conforming the integrity of the data and finalizing the creation of the <stations> table.

## Transforming the <trips> table into a fact table

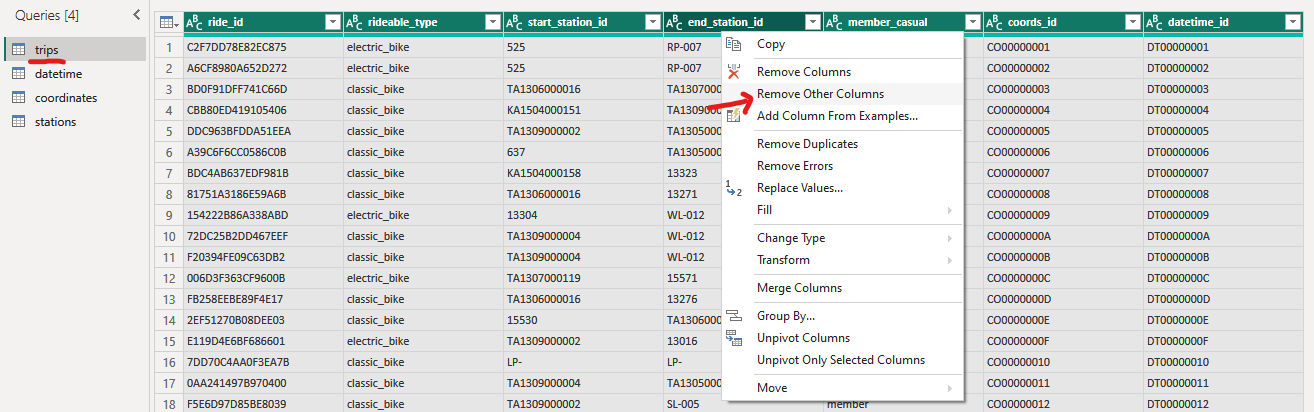


Figure – Transforming the trips table to a fact table

The image in Figure 34 shows the process of removing unnecessary columns from the <trips> table. This step is to prepare the <trips> table for use as a fact\_table.

## Data Modelling Relations and Summary

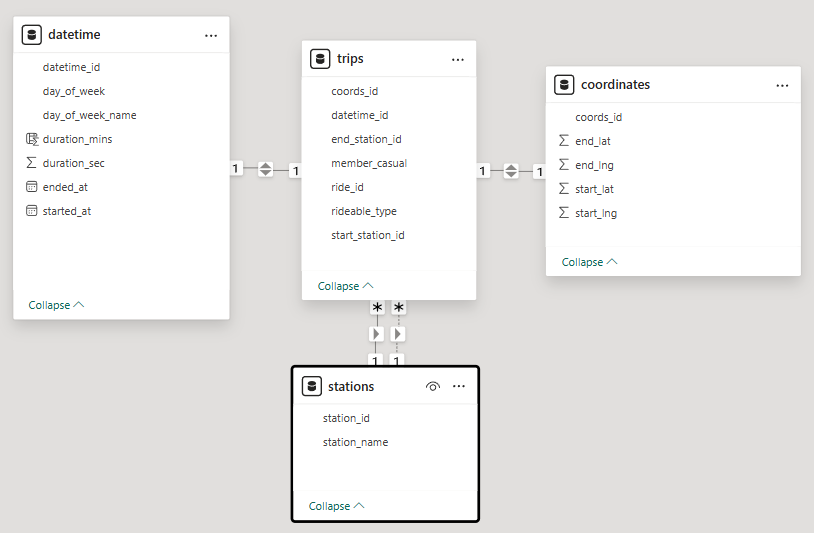


Figure - Star schema data model showing relationships between tables

The diagram in Figure 35 represents a star schema model consisting of four interconnected tables: <datetime>, <trips>, <coordinates>, and <stations>. In this star schema, the <trips> table serves as the fact table.

Relationships between the tables are: <datetime> to <trips> via datetime\_id, <coordinates> to <trips> via coords\_id, and <stations> to <trips> via start\_station\_id and end\_station\_id.

This star schema model organizes data, ensures integrity, and optimizes analytical queries. In Table 2, Table 3, Table 4, and Table 5 are data dictionaries that outlines the columns, data types, and descriptions of each table.

|  |  |  |
| --- | --- | --- |
| **trips table** | | |
| **Column Name** | **Data Type** | **Description** |
| coords\_id | String | A unique identifier for each coordinate record. |
| datetime\_id | String | A unique identifier for each datetime record. |
| end\_station\_id | String | The ID of the station where the ride ended. |
| member\_casual | String | Indicates if the rider is a member or a casual user. |
| ride\_id | String | A unique identifier for each ride. |
| rideable\_type | String | The type of bike used for the ride (e.g., electric, classic). |
| start\_station\_id | String | The ID of the station where the ride started. |

Table – Data Dictionary of trips table

|  |  |  |
| --- | --- | --- |
| **datetime table** | | |
| **Column Name** | **Data Type** | **Description** |
| datetime\_id | String | A unique identifier for each datetime record. |
| day\_of\_week | Integer | The day of the week when the ride started, represented as a whole number (0 for Sunday, 1 for Monday, etc.). |
| day\_of\_week\_name | String | The full name of the day of the week when the ride started (e.g., "Monday", "Tuesday"). |
| duration\_mins | Integer | The duration of the ride in minutes. |
| duration\_sec | Integer | The duration of the ride in seconds. |
| ended\_at | DateTime | The timestamp indicating when the ride ended. |
| started\_at | DateTime | The timestamp indicating when the ride started. |

Table - Data Dictionary of datetime table

|  |  |  |
| --- | --- | --- |
| **coordinates table** | | |
| **Column Name** | **Column Name** | **Column Name** |
| coords\_id | String | A unique identifier for each coordinate record. |
| end\_lat | Decimal | The latitude coordinate where the ride ended. |
| end\_lng | Decimal | The longitude coordinate where the ride ended. |
| start\_lat | Decimal | The latitude coordinate where the ride started. |
| start\_lng | Decimal | The longitude coordinate where the ride started. |

Table - Data Dictionary of coordinates table

|  |  |  |
| --- | --- | --- |
| **stations table** | | |
| **Column Name** | **Data Type** | **Description** |
| station\_id | String | A unique identifier for each station. |
| station\_name | String | The name of the station. |

Table - Data Dictionary of stations table

## Filtering out invalid durations

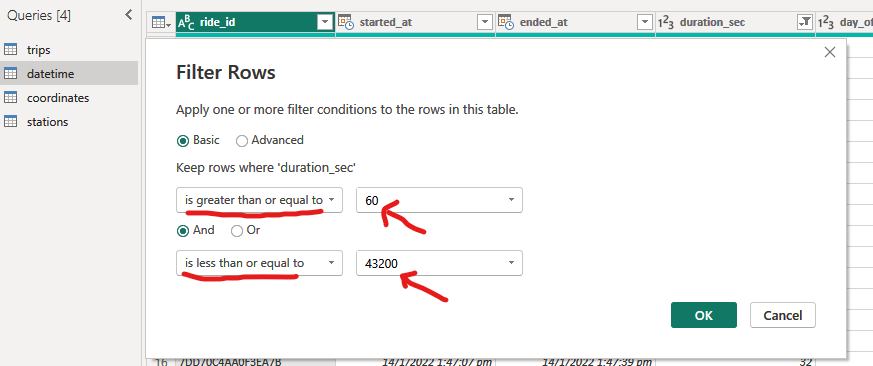


Figure - Filtering the datetime table to include only durations between 60 seconds and 12 hours.

After establishing the relationships between the tables, an important step to take prior to generating any data visualizations is to filter the <datetime> table to retain only data with durations between 60 seconds and 12 hours as shown in Figure 36.

This step enhances the data integrity by removing data that does not reflect typical user riding behaviour. Ensuring that the data falls within this range helps to eliminate any outliers that could skew the analysis and lead to inaccurate insights.

The purpose of structuring and cleansing of the data is to ensure data integrity during data analysis.

**William Gan**

**BUSINESS INTELLIGENCE SOLUTION**

***Divvy Bike Sharing Analysis***

# **Executive Summary**



Figure - Bike Sharing Utilisation Dashboard

This Business Intelligence report analyzed the Divvy Bikes dataset of 3 months from January 2022 to March 2022 to identify strategies for increasing membership conversions and attracting new customers. The objective is to fill the service gap and convert casual riders or attract new customers. Figure 37 is a dashboard that was created.

The key findings include that members account for 77.48% of total trips, while casual riders who make up the remaining 22.52% tend to have longer ride durations. Casual riders prefer electric bikes and use the service more on weekends. Recommendations include introducing a new weekly pass introduced to provide significant cost benefits for the identified targeted audience, tourists and short-term visitors, and strategies for advertisements to promote the new plan discussed to bring awareness to the new plans to increase user engagement.

# **Data Analysis**

## Introduction

This section showcases the steps to discover solutions by performing data analysis and addresses the questions highlighted in and provides information about the data collected.

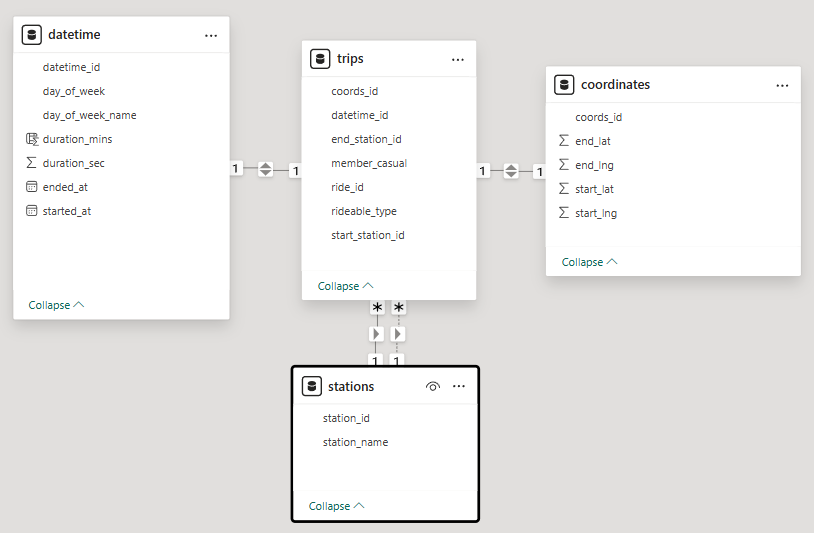


Figure - Overview of the star schema model

The dataset used in this analysis is sourced from Divvy Bikes, a popular bike-sharing service in Chicago, Illinois.

Figure 38 shows the data model that was produced from the steps taken in section 3, providing a visual representation of how the tables are interconnected. It includes trip-related data such as user type (member or casual), start and end datetimes, and the type of bike used (classic or electric).

The data will be examined to answer the following key questions:

* How do annual members and casual riders use the bike service differently?
* Why would casual riders switch to purchase memberships?
* How to influence new customers to utilize the bike sharing service

## **Analysis of Total Trips**

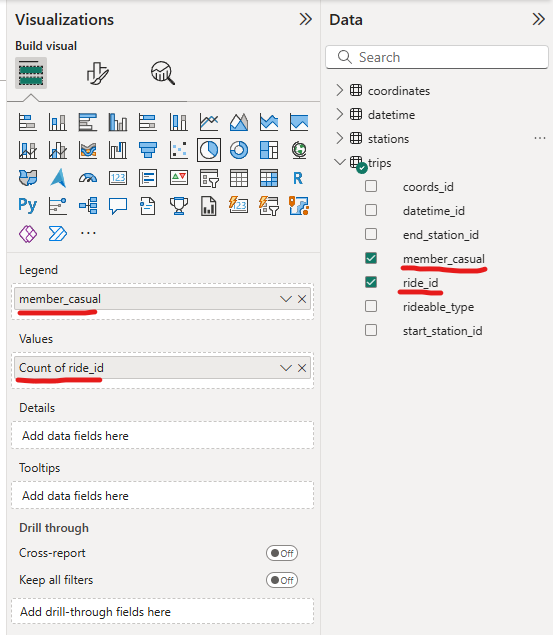


Figure - Visualization Settings for Total Trips Pie Chart

The <Total Trips> pie chart is created from the steps taken as shown in Figure 39. In the legend, member\_casual field was selected, and in the values field, the count of ride\_id.

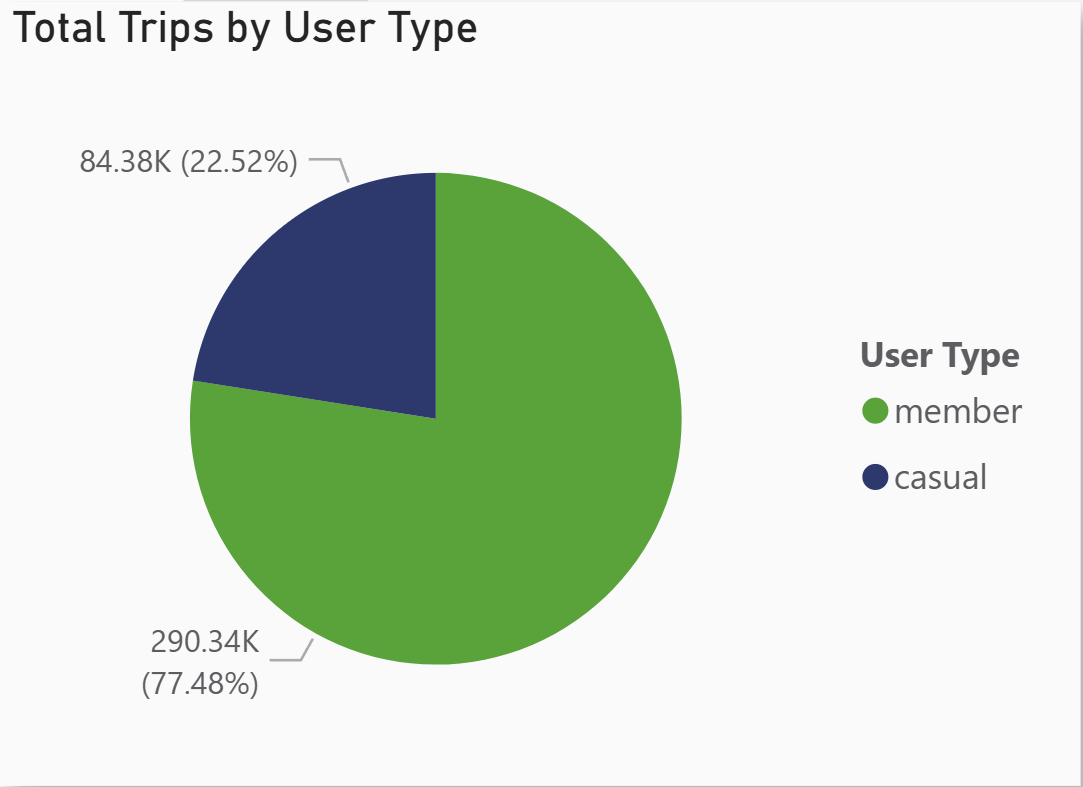


Figure – Pie Chart of Total Trips by User Type

The resulting pie chart, shown in Figure 40 illustrates the proportion of trips taken by casual riders versus annual members. This visualization is crucial for identifying which group utilizes the bike-sharing service more frequently.

* Members account for 77.48% (290.34K) of the total trips.
* Casual users account for 22.52% (84.38K) of the total trips.

## **Analysis of Total Ride Duration (New column using DAX)**



Figure - Formula to calculate duration in minutes

To conduct this analysis, a new column using DAX expression for total ride duration in minutes named ‘duration\_mins’ was created, this calculates the ride duration in minutes. The formula shown in Figure 41 was used to convert the duration from seconds to minutes to provide a more relatable understanding of the ride times.

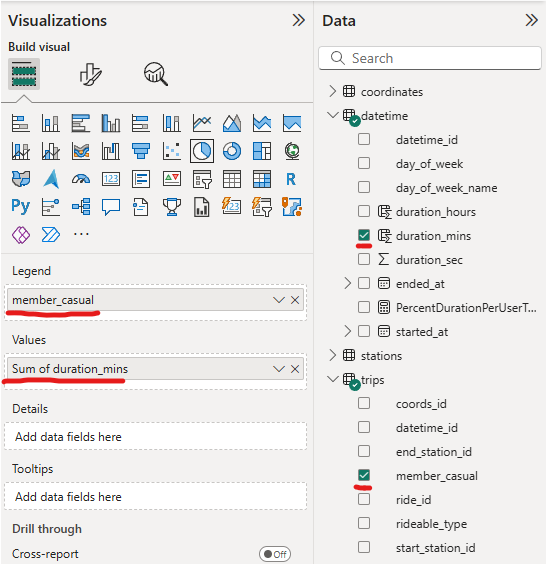


Figure - Visualization Settings for Total Duration Pie Chart

A pie chart was generated with the setting shown in Figure 42 to display the total ride duration segmented by user type. The member\_casual field is used to distinguish between annual members and casual riders, while the “duration\_mins” measure is used for aggregating the total ride time in minutes for each user type.

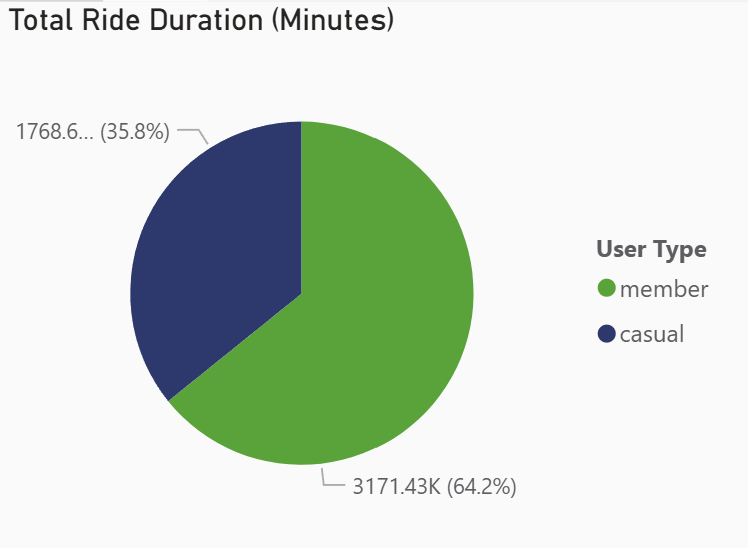


Figure – Total Ride Duration Pie Chart

The resulting pie chart shown in Figure 43 visualizes the total ride duration for each user group, providing a view of how much time annual members and casual riders spend on rides.

* Members have a total ride duration of ~3171 thousand minutes, making up 64.2% of the total duration.
* Casual users have a total ride duration of ~1768 thousand minutes, making up 35.8% of the total duration

## **Analysis of Median Ride Duration**

The purpose of this analysis is to visualize the median ride duration for different user types (annual members versus casual riders. Understanding the median ride duration helps in identifying a typical user usage behavior.

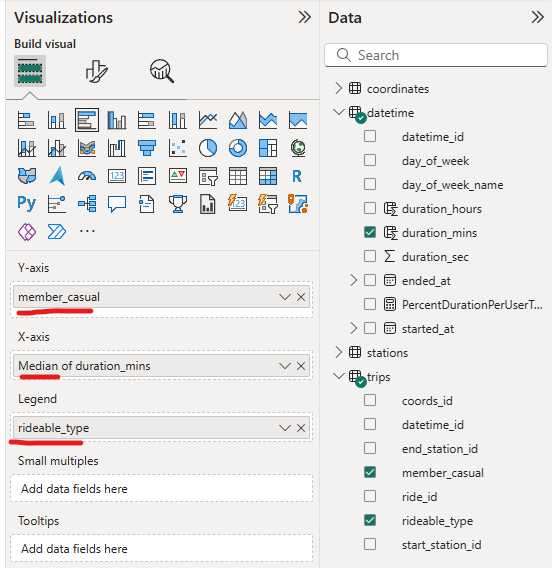
****

Figure - Visualization Settings for Median Ride Duration Bar Chart

To perform this analysis, a measure for the median ride duration in minutes was calculated with the setting shown in Figure 44. A median form measure provides a central value that represents the typical ride duration for each user type, excluding the influence of outliers.

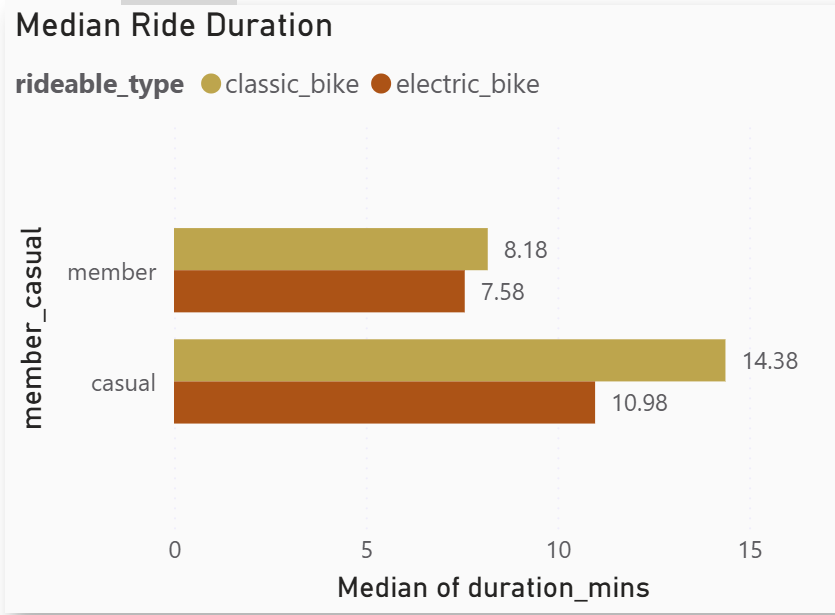


Figure – Bar Chart for Median Ride Duration

|  |  |  |
| --- | --- | --- |
|  | **Classic bikes** | **Electric bikes** |
| **Members** | 8.18 minutes | 7.58 minutes |
| **Casual** | 14.38 minutes | 10.98 minutes |

Table - Median Ride Duration of Users by Bike Type

The resulting bar chart, as shown in Figure 45, illustrates the median ride duration for annual members and casual riders, segmented by the type of bike used. This visualization helps in understanding the typical ride duration for each user group and bike type, offering valuable insights into user preferences and behaviour. Table 6 summarizes the findings of median ride duration with user type and bike type.

## **Distribution of Bike Usage by User Type**

The aim of this analysis is to visualize the distribution of bike-sharing trips by user type and bike type. Understanding the distribution helps in identifying user and bike preferences.

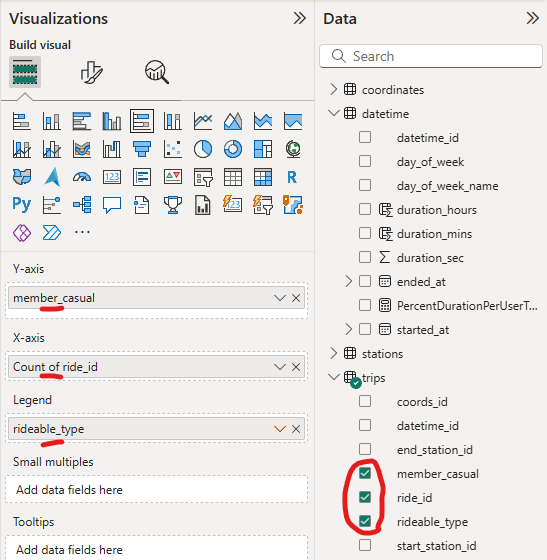


Figure - Visualization Settings for Distribution of Bike Usage

A stacked bar chart was created using the settings shown on Figure 46 to display the distribution of trips by user type and bike type shown on Figure 47.

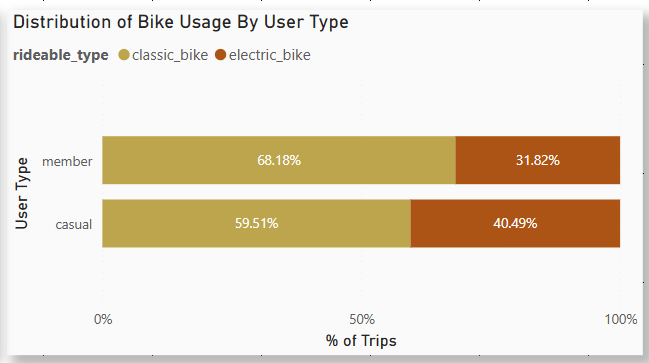


Figure – Bar Chart for Distribution of Bike Usage By User

|  |  |  |
| --- | --- | --- |
|  | **Classic bikes** | **Electric bikes** |
| **Members** | 68.18% | 31.82% |
| **Casual** | 59.51% | 40.49% |

Table – Readings of Distribution of Bike Usage

The resulting bar chart shown in Figure 47 illustrates the distribution of trips for annual members and casual riders, segmented by the type of bike used. This visualization provides a clear comparison of preferences between user groups. Table 7 summarizes the distribution percentages for user type and bike type.

## **Weekly Distribution of Ride Duration (New measure using DAX)**

The purpose of this analysis is to visualize the weekly distribution of ride duration for annual members and casual riders. Understanding how ride durations vary throughout the days in a week helps in identifying peak usage times.



Figure – DAX Measure formula for Percent Duration Per User Type

**In** Figure 48**, a DAX measure, “PercentDurationPerUserType” was created to calculate the percentage of the total ride duration for each user type. This involves summing up the duration in seconds for casual and member riders separately, and then calculating the percentage of the group duration relative to the total duration for each day of the week. It is used to show a comparative view of how ride durations are distributed across different days and user types.**

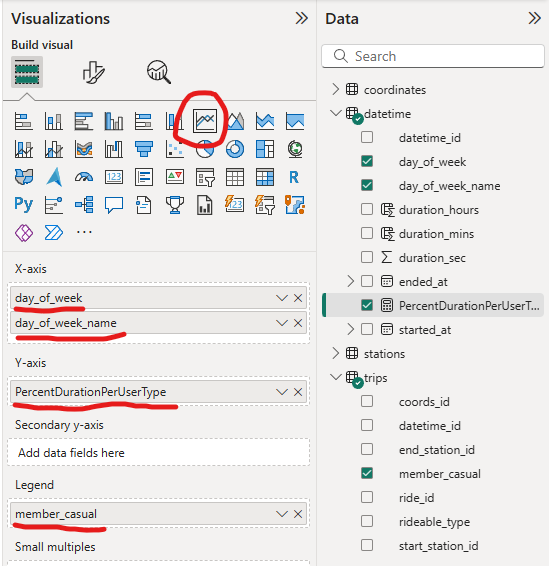


Figure – Visualization Setting for Weekly Distribution of Ride Duration

A line chart was created to display the weekly distribution of ride duration using the settings shown in Figure 49.

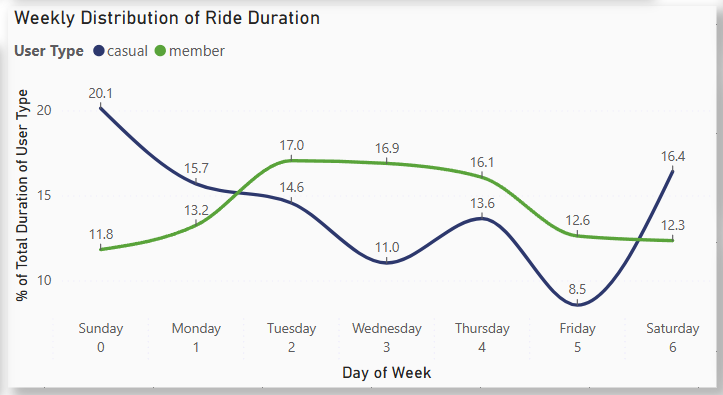


Figure – Line Chart for Weekly Distribution of Ride Duration

The resulting line chart shown in Figure 50 illustrates the weekly distribution of ride duration for both members and casual riders. The chart provides a visual comparison of how ride durations are distributed throughout the week for each user group.

## Data Visualization Dashboard (New measure with DAX)

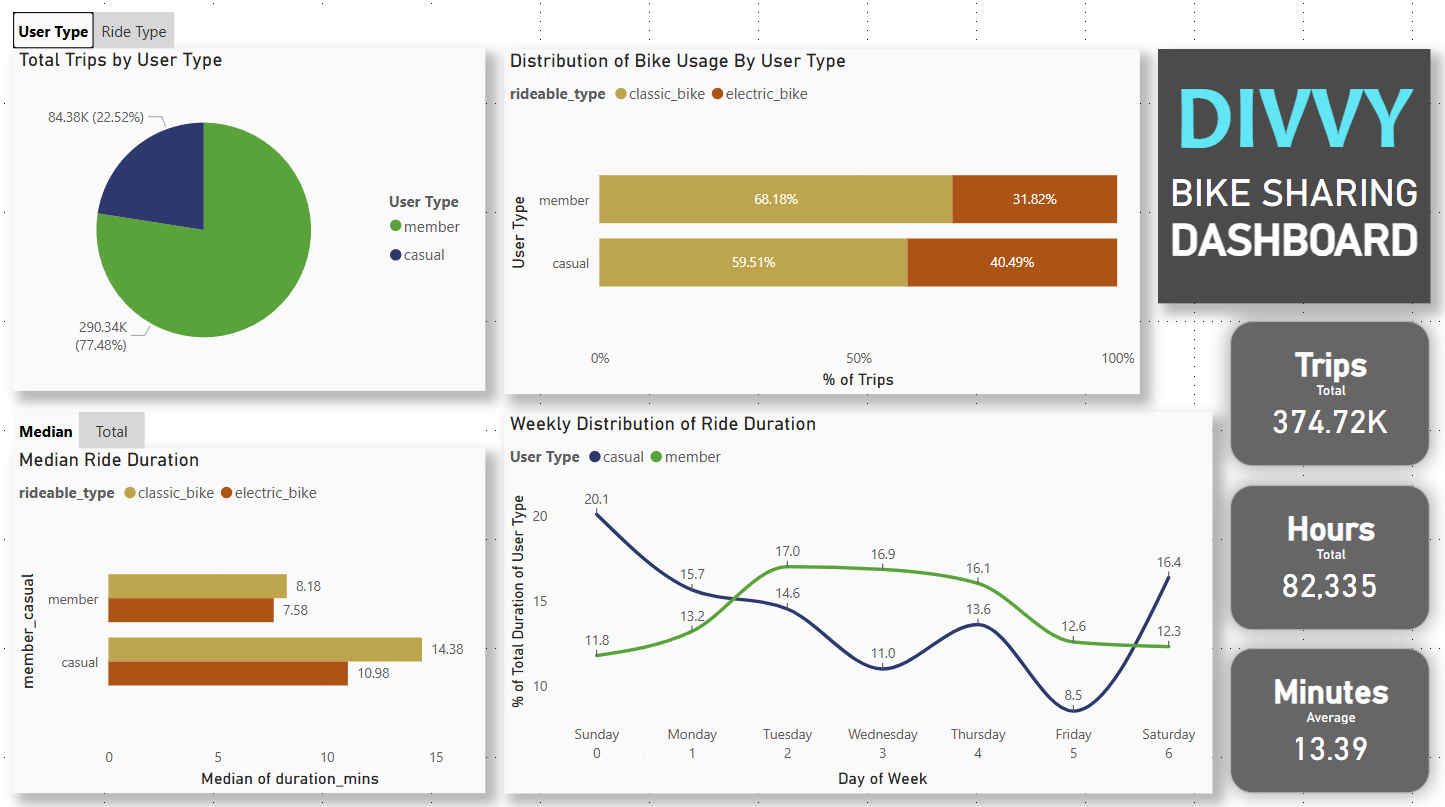


Figure – Bike Sharing Dashboard



Figure - DAX Measure Formula used for Hours Total

The data visualization dashboard produced in Figure 51 consolidates the charts created for data analysis, it offers an interactive view of the usage of the bike sharing service.

Figure 52 is a DAX measure for the total hours card that was created for the dashboard shown in Figure 51. This dashboard is created to allow relevant stakeholders to grasp key trends and patterns, and to aid in decision making and strategic planning.

# Conclusions and Recommendations

Before continuing, its crucial to understand the current pricing information on the bike sharing service.

## Pricing Information of Bike Sharing Service

### **Pricing Model of Classic Bike Rental**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classic Bike Rental** | | | | |
|  | | Unlock Price | Free Duration | Cost/Min |
| **Casual** | Single Ride | $1 | - | $0.18 |
| Day Pass  ($18.10/Day) | Free | **180 Mins/Ride** | $0.18 |
| **Member** | Divvy  ($143.90/Year) ($11.99/Month) | Free | **45 Mins/Ride** | $0.18 |

Table – Pricing Information of Classic Bike

### **Pricing Model of Electric Bike Rental**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Electric Bike Rental** | | | | |
|  | | Unlock Price | Free Duration | Cost/Min |
| **Casual** | Single Ride | $1 | - | $0.44 |
| Day Pass  ($18.10/Day) | Free | - | $0.44 |
| **Member** | Divvy  ($143.90/Year)  ($11.99/Month) | Free | - | **$0.18** |

Table – Pricing Information of Electric Bike

## Correlation Between Usage Patterns and Pricing

### **Casual Riders have a higher median duration**

The higher median duration on both type of bikes for casual riders can be attributed to the **180 free minutes provided on the day pass compared to 45 minutes on the member plan**.

### **Preference for Electric Bikes for Casual Riders**

The pricing structure indicates that electric bikes are significantly more expensive per minute compared to classic bikes. However, **casual riders preference for electric bikes suggests they are willing to pay more for the convenience and speed of electric bikes, or possibly for leisure purposes.**

### **Weekday vs Weekend Usage**

**Casual users have higher usage on weekends compared to weekdays, indicating a preference for leisure rides.** It may make less sense to casual riders to commit to a full month membership

## Recommendations and Justification

### **A New Pricing Plan and Explanation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Weekly Pass (New)** ($24.99 / Week) | | | |
|  | Unlock Price | Free Duration | Cost/Min |
| **Classic Bike** | Free | 60 Mins / Ride | **$**0.18 |
| **Electric Bike** | Free | 90 Mins / Week | $0.18 |

Table – Illustration of a recommended new pricing plan

The new pricing plan shown in Table 10 is an illustration to target new customers and convert casual riders on Single Ride or Day Pass to a newly proposed weekly pass plan.

To justify, casual Riders make fewer but longer trips on average and have a stronger preference for electric bikes than Members. Casual Riders also have a higher duration usage on weekends than weekdays. In addition, the cost efficiency of the new plan is shown in Table 11.

The target audience for this plan are tourists and short term visitors as they are likely visiting the city for leisure or business and may not be familiar with the infrastructure of the city. Tourists and visitors are usually in for a short stay and would most likely prefer a convenient and flexible transportation which will allow for easier navigation and better comfort.

### Cost Efficiency of Electric Bike Usage per Plan

|  |
| --- |
| **Single Ride**   1. Unlock Fee: $1 2. Cost per Minute: $0.44 3. Cost for 90 Minutes: 90\* $0.44 = $39.60 4. Total Cost for 90 Minutes: $1 + $39.60 = $40.60   **Day Pass**   1. Day Pass Price: $18.10 2. Cost per Minute: $0.44 3. Cost for 90 Minutes: 90 \* $0.44 = $39.60 4. Total Cost for 90 Minutes: $18.10 + $39.60 = $57.70   **Monthly Pass**   1. Monthly Pass Cost: $11.99 2. Cost per Minute: $0.18 3. Cost for 90 Minutes: 90\* $0.18 = $16.20 4. Total Cost for 90 Minutes: $11.99 + $16.20 = $28.19   **Weekly Pass**   1. Pass Cost: $24.99 2. Free Duration: 90 minutes 3. Cost/Minute Exceeding Free Duration: $0.18 4. Total Cost for 90 Minutes: $24.99 |

Table – Cost Efficiency Calculations and Summary

Table 11 compares the cost of riding an e-bike for the first 90 minutes using different pricing plans. The weekly pass is the most cost-effective option, followed by the monthly pass. The single ride option is more expensive, with the day pass being the most expensive overall. Hence, the illustrated weekly pass is a compelling choice for users who are committing short term, with the habit of longer ride durations and preference for electric bikes.

### **Targeted advertisements to promote illustrated weekly plan**

Targeted advertisements can be used to promote the new weekly membership plan to casual riders who often use the service. This can potentially increase user retention and result in a steady revenue stream. The new membership plan could also be promoted at non-customers for those who disinterested with the previous existing options.

# References

DougKlopfenstein (2024) Expressions, values, and let expression - powerquery m, PowerQuery M | Microsoft Learn. Available at: https://learn.microsoft.com/en-us/powerquery-m/expressions-values-and-let-expression (Accessed: 01 July 2024).

Kfollis (2024) Switch function (DAX) - dax, function (DAX) - DAX | Microsoft Learn. Available at: https://learn.microsoft.com/en-us/dax/switch-function-dax (Accessed: 07 July 2024).

Kfollis and jeroenterheerdt (2023) DAX Queries Microsoft Learn. Available at: https://learn.microsoft.com/en-us/dax/dax-queries (Accessed: 04 July 2024).

Peter-Myers (2022) Use variables to improve your dax formulas - dax, DAX | Microsoft Learn. Available at: https://learn.microsoft.com/en-us/dax/best-practices/dax-variables (Accessed: 11 August 2024).

Power BI Resources (2024) Power BI Guide. Available at: https://pbi.guide/resources/ (Accessed: 01 July 2024).