Project 5: Cyclist Data Project Report

Data Management Final Project

Prepared by: Group 9

Eloise Chen

Yanjing Li

Michael Suzuki

Yigong Yuan

Yudong Zhang

TABLE OF CONTENTS

Executive Summary 2

Project Statement 2

Data Dictionary 3

Dimensional Modeling + ERD 3

1. Business and event processes 3
2. Grain 3
3. Facts and dimensions 3
4. ERD Diagram (Star Schema) 7

Data Transformation 7

Loading Data 7

Data Cleaning 8

Rides above 1500 Minutes 8

Populating Dimension and Fact Table 8

KPIs, Visualizations, and Insights 9

KPI 1: Ride Duration 9

KPI 2: The Most Common Rideable Type 11

KPI 3: Most Frequent Route 12

KPI 4: Max Bike Usage 13

KPI 5: Casual-Member Rate 14

Data Visualization: 16

Tableau Dashboards 16

Recommendations 20

Project Challenges 21

Technical Challenges 21

KPI Calculations Challenges 21

Further Research 21

Market Trend Research 22

Appendix 23

EXECUTIVE SUMMARY

The Cyclist Data Project by Group 9 focused on analyzing historical Divvy/Lyft bike trip data from 2021 in the Chicago area. The project aimed to provide strategic insights by examining ride duration, ride length, and user segmentation. A significant portion of the work involved data cleaning due to issues like missing station names, IDs, and geographical coordinates, resulting in the exclusion of rides with null values. This process reduced the original dataset size significantly. The team also decided to omit rides above 1500 minutes, considering them outliers. This comprehensive approach allowed the team to analyze the data through various lenses, including temporal, segmentation, and geographical perspectives, providing a nuanced understanding of the cycling trends in Chicago.

The project's methodology revolved around dimensional modeling, using a star schema ERD to organize data into six dimensions: User Type, Rideable Type, Trip Time, Time of Day, Location, and Date. This structuring was essential for examining five key performance indicators (KPIs): Ride Duration, Most Common Rideable Type, Most Frequented Route, Maximum Bike Usage, and Casual-Member Rate. These KPIs provided insights into aspects such as customer engagement, satisfaction, and demographic preferences. For instance, the analysis of Ride Duration revealed patterns like peak average ride durations in May, longer rides on weekends, and different usage patterns between casual and member users. These findings were crucial for strategic decisions regarding pricing, marketing, and operational strategies.

The project encountered several challenges, particularly in data integrity and processing. The raw dataset presented issues with missing values and precision in geographic coordinates. Additionally, the complexity of defining the "docked bike" category and its impact on average trip duration presented challenges in KPI calculations. Despite these hurdles, the team managed to produce insightful analyses, with recommendations for future strategic decisions. Their work underscored the importance of adapting strategies to evolving market trends and highlighted the potential for Divvy to optimize resource management, marketing plans, and pricing strategies in response to changing demand patterns.

PROJECT STATEMENT

The project aims to analyze the historical Divvy/Lyft bike trip data from 2021 in the Chicago area. By analyzing ride duration, and ride length and segmenting users, the team provides findings for aiding strategic decisions.

DATA DICTIONARY

OLTP							OLAP		
DB.schema	TABLE_NAME	COLUMN_NAME	data_type	max length	IS_NULLABLE	is needed?	Analytical Value	Fact/Dim Tables	Description
mydb.project5	DIVVY_TRIPDATA_CLEANED	ride_id	VARCHAR(255)	255	No		Yes	trip_time_dim	Unique Ride id
mydb.project5	DIVVY_TRIPDATA_CLEANED	rideable_type	VARCHAR(50)	50			Yes	Rideable_type_dim	Type of Bike: Classic, Electric, Docked
mydb.project5	DIVVY_TRIPDATA_CLEANED	started_at	TIMESTAMP_NTZ				Yes	date_dim	Start Timestamp. Date and Time
mydb.project5	DIVVY_TRIPDATA_CLEANED	ended_at	TIMESTAMP_NTZ				Yes	date_dim	End Timestamp. Date and Time
mydb.project5	DIVVY_TRIPDATA_CLEANED	start_station_name	VARCHAR(255)	255			Yes	Location_dim	Name of Starting Station
mydb.project5	DIVVY_TRIPDATA_CLEANED	start_station_id	VARCHAR(50)	50			Yes	Location_dim	ID of Starting Station
mydb.project5	DIVVY_TRIPDATA_CLEANED	end_station_name	VARCHAR(255)	255			Yes	Location_dim	Name of Ending Station
mydb.project5	DIVVY_TRIPDATA_CLEANED	end_station_id	VARCHAR(50)	50			Yes	Location_dim	ID of Ending Station
mydb.project5	DIVVY_TRIPDATA_CLEANED	start_lat	FLOAT				Yes	Location_dim	Starting Latitude
mydb.project5	DIVVY_TRIPDATA_CLEANED	start_lng	FLOAT				Yes	Location_dim	Starting Longitiude
mydb.project5	DIVVY_TRIPDATA_CLEANED	end_lat	FLOAT				Yes	Location_dim	Ending Latitude
mydb.project5	DIVVY_TRIPDATA_CLEANED	end_lng	FLOAT				Yes	Location_dim	Ending Longitude
mydb.project5	DIVVY_TRIPDATA_CLEANED	member_casual	VARCHAR	50			Yes	user_type_dim	Rider type: Member or Casual

DIMENSIONAL MODELING + ERD

1. Business and event processes

The business process is a bike rental service. This involves customers/riders of different membership types (member or casual) renting a bike (electronic, classic, docked). Each customer picks up a bike at a start station, uses the bike, and returns it to an end station. The event process involves the time and location of each bike ride trip, the types of customers, and the types of rideable bikes.

2. Grain

It is crucial to determine the grain of the fact table for dimensional modeling. The grain refers to the granularity level of information each row in the fact table contains. In this analysis, the grain is defined as an individual bike ride record. Each row in the fact table represents a unique ride from start to end, with information about the start and end time, the start and end stations, the type of bike used, and the type of user.

3. Facts and dimensions

Using the 7W's framework, we defined 6 dimensions as below:

Dimensions			
7Ws	**Dimensions**	**Table Name**	**Attributes**
Who	User Type Dimension	user_type_dim	1. user_type_key 2. user_type
What	Rideable Type Dimension	rideable_type_dim	1. rideable_type_key 2. rideable_type
When	Trip Time Dimension	trip_time_dim	1. ride_key 2. ride_id 3. started_at 4. Ended_at
	Time Dimension	date_dim	1. date_key 2. month_name 5. month_num 6. year 7. quarter 8. week_name (weekday or weekend) 9. week_num 10. day_of_week
	Time of Day Dimension	time_of_day_dim	1. time_of_day_key 2. hour 3. am_pm 4. hour_24
Where	Location Dimension	location_dim	1. station_key 2. station_id 3. station_name 4. station_max_lat 5. station_min_lat

			6. station_max_lng 7. station_min_lng
Why	Not relevant to this dataset		
How	Not relevant to this dataset		

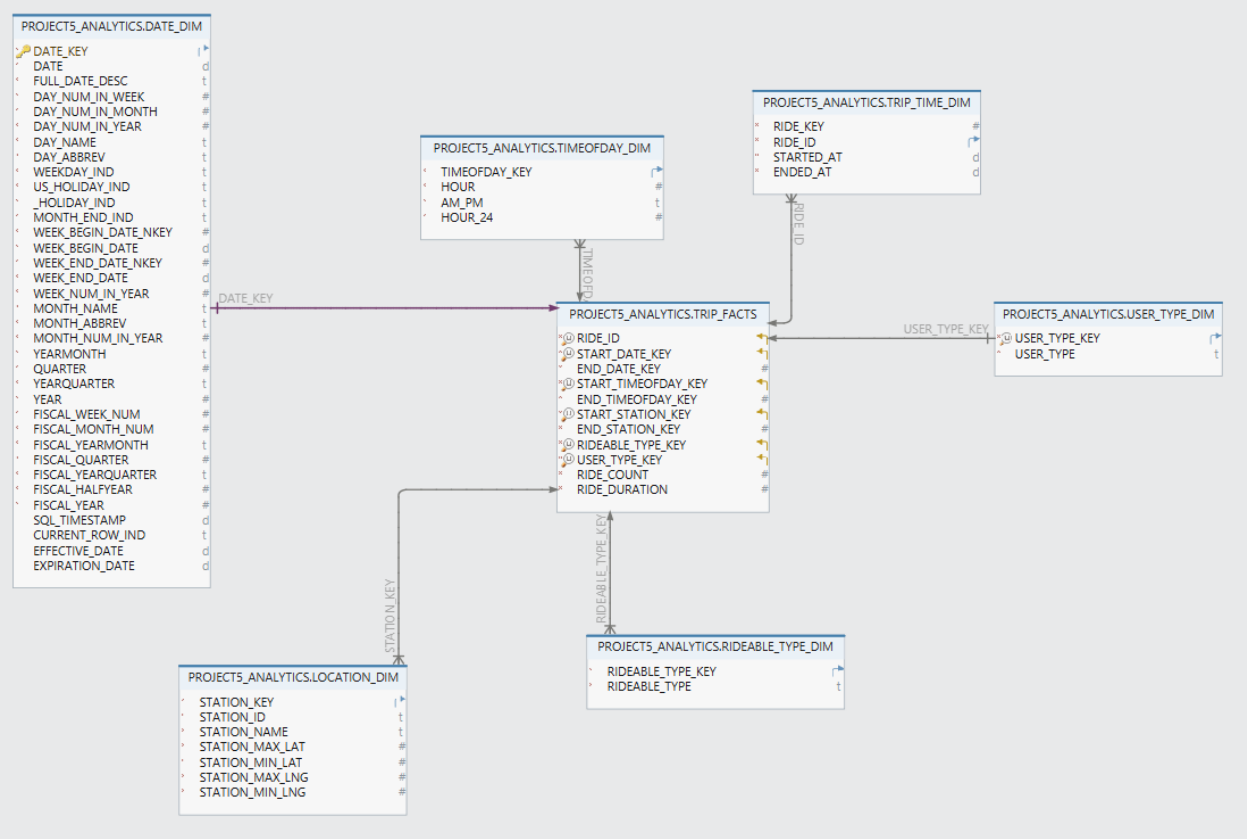
Based on the 5 major KPIs and the defined Dimensions, we defined the fact table as below:

Fact		
Table Name	**Attributes**	**Definitions**
trip_fact_table	1. ride_id 2. start_date_key 3. end_date_key 4. start_timeofday_key 5. end_timeofday_key 6. start_station_key 7. end_station_key 8. rideable_type_key 9. user_type_key 10. ride_count 11. ride_duration	● ride_id: the unique identifier of each bike ride record for each row ● keys attributes: foreign keys used to link to different dimension tables ○ start/end date keys: date_dim ○ start/end time of day keys: time_of_day_dim ○ start/end station keys: location_dim ○ rideable type key: rideable_type_dim ○ user_type_key: user_type_dim ● ride_count: count of the ride, counting 1 ride for each row ● ride_duration: timespan of each ride trip in minutes

Dimensions			
7Ws	**Table Name**	**Attributes**	**Why the Measures?**
Who	user_type_dim	user_type_key user_type	● KPI: Casual-Member Rate ● Discover the difference in patterns of member vs. casual users
What	rideable_type_dim	rideable_type_key rideable_type	● KPI: Most Common Rideable Type ● Discover the most common type of bikes for transportation
When	trip_time_dim	ride_key ride_id started_at ended_at	● Relates to all 5 KPIs ● Three dimensions to find out the time attributes of the trip because time can be looked at in many ways ● Discover how trips differ within different timespans (e.g. weekdays vs. weekends; morning vs. afternoon vs. night) ● Discover the lengths of trips in different situations, specifically by using "trip_time_dim", which includes starting and ending timestamps of each ride
	date_dim	date_key month_name month_num year quarter week_name week_num day_of_week	
	time_of_day_dim	time_of_day_key hour am_pm hour_24	
Where	location_dim	station_key station_id station_name station_max_lat	● KPI: Most Frequented Route ● Discover optimal bike deployment and collection points ● Each unique combination of ID and name accounts for one row due to

		station_min_lat station_max_lng station_min_lng	the many-to-many relationship between the two variables • Each station has a range of coordinates split into four attributes: max/min latitudes and longitudes

4. ERD Diagram (Star Schema)



DATA TRANSFORMATION

Loading Data

Our team created the raw data table, divvy_tripdata, in Snowflake and uploaded the 12 files into the table. From there, the team cleaned the data to remove rides with null values and moved the data to the DIVVY_TRIPDATA_CLEANED table.

Our team managed to load the data through two different approaches: loading them directly to Snowflake and copying them into Snowflake through AWS.

After creating the raw data table and setting the data type correctly, we went over to the Database tab in Snowflake and uploaded the data using the “Load Data” button. We simply dragged and

dropped the data files downloaded from Divvy and all data are uploaded with no glitch. However, we did notice that this operation utilized a large amount of processing power, which could be costly for large corporate clients loading big data on a regular basis.

Data Cleaning

Our team analyzed the data and found significant missing information. Many station names, IDs, and geographical coordinates (starting and ending longitudes and latitudes) were absent. Additionally, some latitude and longitude data were only accurate to two decimal places, inadequate for reliably linking station names and IDs to specific locations. To ensure high-quality data, we removed all rows containing null values. This process resulted in a reduction of the dataset from the original 5,595,063 rows to 4,588,302 rows, amounting to a 17.99% deletion of the original data.

```
CREATE OR REPLACE TABLE "MYDB"."PROJECT5".DIVVY_TRIPDATA_CLEANED AS
SELECT * FROM "MYDB"."PROJECT5".DIVVY_TRIPDATA
WHERE (start_station_name is not null AND start_station_id is not null)
    AND (end_station_name is not null AND end_station_id is not null)
    AND end_lat is not null
    AND end_lng is not null;
```

Rides above 1500 Minutes

When looking at average and max ride duration, the team noticed outliers over 50,000 minutes. Bike rentals are supposed to be at the daily level and the max standard bike ride duration was 1,500 minutes so the team decided to exclude rides above 1,500 minutes in duration as outliers due to poor data. This was 1,167 rides from the 4,588,302.

Populating Dimension and Fact Table

After removing nulls from the raw data, we created a new schema, “project5_analytics”, to store dimension and fact tables.

Constructing dimension tables, user_type_dim, rideable_type_dim, and trip_time_dim, was straightforward. The process involved creating the tables and specifying each variable and its data type, setting up sequences for key generations, and populating the tables with corresponding variables from the cleaned dataset.

One specification is around location_dim. The raw data contains overlapping start and end stations because start stations can also be end stations, and vice versa. We also recognized that station IDs and names are not unique identifiers of stations, with their many-to-many relationship, and are associated with multiple geographic coordinates. Therefore, we merged all station data using the UNION ALL command to capture the full extent of station names, IDs, latitudes, and longitudes. Then, by wrapping the combined station information into a subquery, we aggregated the maximum and minimum latitude and longitude values for each unique station

ID and name combination. This approach allowed us to form a holistic list of stations that accurately incorporates the range of coordinates associated with each station, grouping them by their IDs and names. The code is shown below:

```
INSERT INTO "MYDB"."PROJECT5_ANALYTICS".LOCATION_DIM (
    STATION_ID,
    STATION_NAME,
    STATION_MAX_LAT,
    STATION_MIN_LAT,
    STATION_MAX_LNG,
    STATION_MIN_LNG
)
SELECT
    STATION_ID,
    STATION_NAME,
    MAX(STATION_LAT) AS STATION_MAX_LAT,
    MIN(STATION_LAT) AS STATION_MIN_LAT,
    MAX(STATION_LNG) AS STATION_MAX_LNG,
    MIN(STATION_LNG) AS STATION_MIN_LNG
FROM (
    SELECT
        START_STATION_ID AS STATION_ID,
        START_STATION_NAME AS STATION_NAME,
        START_LAT AS STATION_LAT,
        START_LNG AS STATION_LNG
    FROM "MYDB"."PROJECT5".DIVVY_TRIPDATA_CLEANED
    UNION ALL
    SELECT
        END_STATION_ID AS STATION_ID,
        END_STATION_NAME AS STATION_NAME,
        END_LAT AS STATION_LAT,
        END_LNG AS STATION_LNG
    FROM "MYDB"."PROJECT5".DIVVY_TRIPDATA_CLEANED
) AS combined_stations
GROUP BY STATION_ID, STATION_NAME;
```

Constructing the fact table followed a similar process to the dimension tables, with the exception that data selection was based on joining different dimension tables on ride_id. For calculated fields, each row was assigned a ride_count of 1, allowing us to easily sum up the number of rides for future KPI establishments. In addition, ride_duration was calculated by joining the trip_time_dim table and applying the TIMEDIFF command to compute the time difference between the ride's start and end times in minutes, simplifying future analysis and dashboarding.

KPIS, VISUALIZATIONS, AND INSIGHTS

In this project, our team has developed five key performance indicators – ride duration, most common rideable type, most frequent route, max bike usage, and casual member rate, to analyze the trend in the cyclist data. Our analysis comes from three perspectives: temporal analysis, segmentation, and geographical analysis, which measures performance from a comprehensive view.

KPI 1: Ride Duration

The ride duration metric quantifies the duration of each ride, deriving from the difference between each trip's start and end times. This indicator is instrumental in assessing levels of

customer engagement and satisfaction. It provides critical insights necessary for refining pricing strategies, enhancing customer experience, and contributing positively to the company's financial outcomes.

Our analysis encompasses the following key aspects:

1. Temporal Analysis: Examines the average and maximum ride durations (in minutes) overall, alongside average durations analyzed for each month in 2021. It also differentiates between average ride durations on different days in a week, and hourly analysis based on the ride's start time.
2. Segmentation: Breaks down average ride durations by user type and rideable type, providing a nuanced view of usage patterns.
3. Geographical Analysis: Focuses on the average ride duration associated with each start station, offering a spatial perspective on ride durations.

The analysis provides the management team with insights into seasonality trends, and demographic preferences.

1. Seasonality and Trends: The data highlights a peak in average ride durations in May, with a noticeable decline from October to January, indicating significant seasonality effects.
2. Weekday Versus Weekend Usage: Rides are significantly longer on weekends than on weekdays, suggesting variations in ride purposes and user availability.
3. Hourly Trends: Early morning rides (5 am to 8 am) are shorter in duration, potentially reflecting commuter patterns.
4. Behavioral Differences by User Types: Casual users tend to have ridden more than double the length of those by members, indicating different usage patterns between the two groups.
5. Rideable Type Preferences: Classic bikes show a slightly longer ride duration, on average, over electric bikes in terms of ride duration.

Those insights are crucial in driving strategic management decisions on pricing, marketing, and operational strategies, which further optimizes resources and enhances long-term customer value for better financial outcomes.

1. Pricing strategies: The seasonal, daily, and hourly demand fluctuations suggest Divvy implement a dynamic pricing system. Divvy could increase the unit price during peak time/season and offer discounts for off-peak times. Further studies on price-demand elasticities are needed to determine the specific price settings.

2. Marketing plans: Considering a lower average trip duration for members, the company needs to focus on improving customers' perceived values by adding services and benefits to the member user group and encouraging longer rides.
3. Operational enhancements: Divvy needs to design its operational systems based on the demand level. They need to create an inventory management system, ensuring sufficient demand for bikes during peak season/days/hours, as well as stations with high demands. In addition, Divvy should schedule maintenance activities during periods of low activity to minimize disruptions and ensure seamless operations.

KPI 2: The Most Common Rideable Type

This key performance indicator offers valuable insights into pinpointing demand hotspots, which guides strategic directions for potential partnerships with local businesses or government entities. Such insights are essential in determining marketing priorities and customizing services to be more user-centric, enhancing overall customer engagement and satisfaction.

Our analysis is structured into three major perspectives:

1. Temporal Analysis: Identifies the most popular rideable types across time frames, analyzing trends across months in a year, days in a week, and hours in a day.
2. Segmentation: Explores preferences by user type (casual vs. members) and by rideable types. Informing decision-makers with targeted data.
3. Geographical Analysis: Assesses preferred rideable types at each station, guiding an overview of preference distribution patterns.

The analysis provides insights into the seasonality trends and geographical differences in the most popular rideable type.

1. Seasonality and Trends: Despite the lower number of used frequencies during winter months, classic bikes are always the most popular rideable. Its popularity consistently outperforms electric bikes, regardless of the time frame and user type (casual vs. members).
2. Geographical Differences: Rideable-type popularities vary across stations. Despite a majority of preference for classic bikes, some stations do have a higher popularity for electric bikes.

Those insights provide critical strategic information on resource allocation, product development, and marketing planning.

1. Inventory Management and Resource Allocation: Given the consistent popularity of classic bikes over time, it is critical to implement an inventory management system that

ensures sufficient supply matching demand. It is also critical to customize the inventory supply based on geographical preferences. For example, Divvy needs to allocate more electric bikes to stations with a higher electric bike demand.

2. Product Development: the contrast in demand between classic and electric bikes offers Divvy a direction for its product development efforts. The enduring popularity of classic bikes underscores the potential for diversifying and tailoring products to meet the unique needs of different geographic markets. Simultaneously, there's a strategic imperative to upgrade the features of electric bikes, aiming to boost their appeal and usage. However, it is essential to assess the profitability of each rideable type, guiding the company towards a precise resource allocation strategy that balances investment with expected returns.
3. Strategic Partnerships: Divvy could seek potential partnerships with local governments and entities to enhance the infrastructure for classic bike storage in stations where classic bikes are popular.

KPI 3: Most Frequent Route

The most frequent route indicates the most commonly traveled path by users throughout 2021, showcasing the start and end stations of each trip. This data offers valuable insights for strategic decision-making, including identifying key areas for infrastructure investments, informing marketing promotion strategies, and pinpointing opportunities for service expansion.

1. Temporal Analysis: Identifies the most frequented route across all data, and further dissects route popularity by different time frames—monthly, daily, and hourly.
2. Segmentation: Examines the preferred routes among different user types (casual riders vs. members) and across various rideable types.
3. Geographical Analysis: Maps the distribution of route popularity in the city of Chicago.

The analysis delves deep into understanding user preferences and identifying seasonal trends.

1. Consistent Route Preference: The analysis reveals that the route between Street Dr & Grand Ave is the most frequently chosen overall. This route maintains its dominance across each day of the week and among various user types.
2. Hourly Route Patterns: The most frequent route tends to maintain the same start and end stations throughout the day, except the period from 3 am to 7 am.
3. User Type Preferences: There are discernible differences in route preferences between member and casual users. Members exhibit distinct preferences compared to casual users, suggesting varying commute patterns.

To convert those insights into financial outcomes, Divvy could implement strategic planning on infrastructure investments, route diversity promotion, and potential service expansions.

1. Infrastructure Investments: Because of the high popularity of the Street Dr & Grand Ave route, Divvy could focus its investment on enhancing the infrastructure in the station. Potential actions include adding more bike docks, scheduling regular maintenance activities, and improving road conditions nearby.
2. Route Diversity Promotion: To ensure that the popular routes are not overcrowded during peak hours, Divvy could implement promotional activities to promote alternative routes for the destinations. This could also ensure a balance in resource utilization.
3. Potential Service Expansions: Divvy could use information on hourly and daily popular route trends to provide additional customer service, informing customers on road conditions to avoid congestion areas. They could also offer customizable services to allow users to customize their route based on the collected information Divvy provides on the app.

KPI 4: Max Bike Usage

This KPI measures the total number of bikes being used in the entire network (all stations), during a period of time. The aim is to understand what moments of the day have the peak usage of bikes.

Our analysis includes the following:

1. Temporal analysis: Identifies the usage situation of bikes hourly as well as in different periods —- Morning Rush Hour (5-8), Noon (9-15), Evening Rush Hour (16-19), and Night (21-4).
2. Segmentation: Examining the bike usage situation among different user types and rideable types to find the type with the highest usage frequency.
3. Geographical analysis: the top 10 start stations with the highest bike usage number.

The analysis provides insights into the peak bike usage condition:

1. Hourly Bike Usage Pattern: The maximum bike usage hour is 17:00 whereas the minimum hour is 4:00, with a local max value at 8:00. The evolving tendency of bike usage is quite smooth, indicating a higher tendency of bike usage in the afternoon and evening rather than morning, which is according to the time period analysis results.
2. Bike Usage in Time Periods: The results indicate that bike users of the company have a far more strong tendency to use the bike during the evening rush hour rather than in the

morning. This is probably due to a broad usage intention during the evening rather than morning where the only usage for bikes is to cover the short distance to the workplace.

3. User Type with Usage Count: The general number of casual and member usage is almost the same, with the number of member usage slightly higher. This indicates a higher membership frequency since a smaller user pool of members created a bigger usage number.
4. Bike Type with Usage Count: The classic bike is the most frequently used type of bike compared with docked bikes and electric ones. However, there could be an omitted variable bias due to the total amount of different types of bike. We can only conclude that the classic type is the most popular if taking the total number of different types into consideration.
5. Start Station with Usage Count: Among the most popular routes, Streeter Dr & Grand Ave has an irreplaceable leading position in the average hourly rides.

To convert those insights into financial outcomes, Divvy could implement strategic planning on bike distribution, user engagement, and bike type analysis.

1. Bike Distribution: The company should schedule more bike distribution during noon time and night time, which are the low ebb of bike usage, to ship more bikes to the start station with a higher demand for bike usage such as Streeter Dr.
2. User Engagement: Member users constitute a crucial part of the total usage (over 50%). We need to further study the average usage of member users for each time period (for example month), to test the member user engagement. To keep or improve user engagement, put forward member favorable policies.
3. Bike Type Analysis: We need further study on the total number of different types of bikes to determine if electric bikes are more popular. Profitability analysis can be conducted for the benefit of putting in more electric bikes to the financial and market competitive situation.

KPI 5: Casual-Member Rate

This KPI measures the percentage rate between casual rides and member rides. It is calculated as (# casual / # member) * 100 for a given location/route/destination and/or time period.

Our analysis includes the following:

1. Temporal analysis: Identifies the casual member rate of bikes hourly as well as on different weekdays and on different months.
2. Segmentation: Examines the casual member rate of different rideable types of bikes.

3. Geographical analysis: the top 10 start stations as well as the end stations with the highest causal member rates.

The analysis provides insights into the peak bike usage condition:

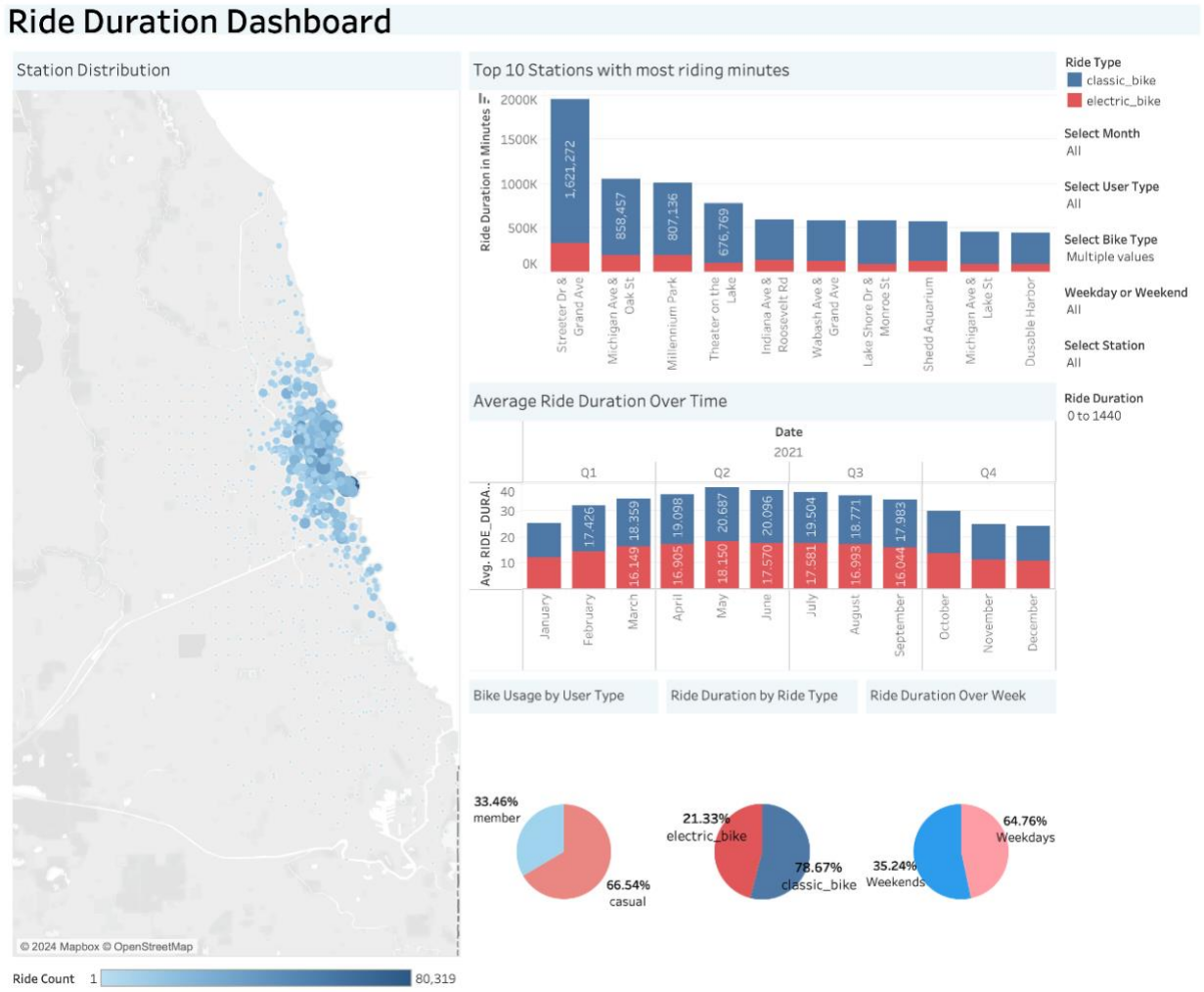
1. Hourly Difference of Casual Member Rate: The results show that the minimum point of casual member rate in a day is 6:00, with a local minimum at 17:00. Both of them are rush hours, and bike usage peak time. So we can conclude that the ratio of casual users reduces at rush hours whereas in normal time periods, casual users are more likely to use the bikes. This may be related to the accessibility of different types of users.
2. Week Day Differences: We can see that the casual member rate on weekends is higher than on work days, this is probably because during work days there is a higher demand for bike usage and casual members' accessibility is weaker than members.
3. Monthly Differences: The casual member rate is highest in July and lowest in February with a quite smooth evolution. Apart from accessibility, there should be further reasons which require further research.
4. Rideable Type Difference of Casual Member Rate: The casual member rate of electric bikes is higher than classic bikes, which needs further research to find the reasons.
5. Casual Member Rate of Start Station: When doing start station and end station casual member rate analysis, we can not see a significant difference between the list of top 10 stations with the highest casual member rate. Some of these stations are the busiest stations in the city, with more bike supply attracting more casual users, whereas some of them are stations where most users live around them and are not members.

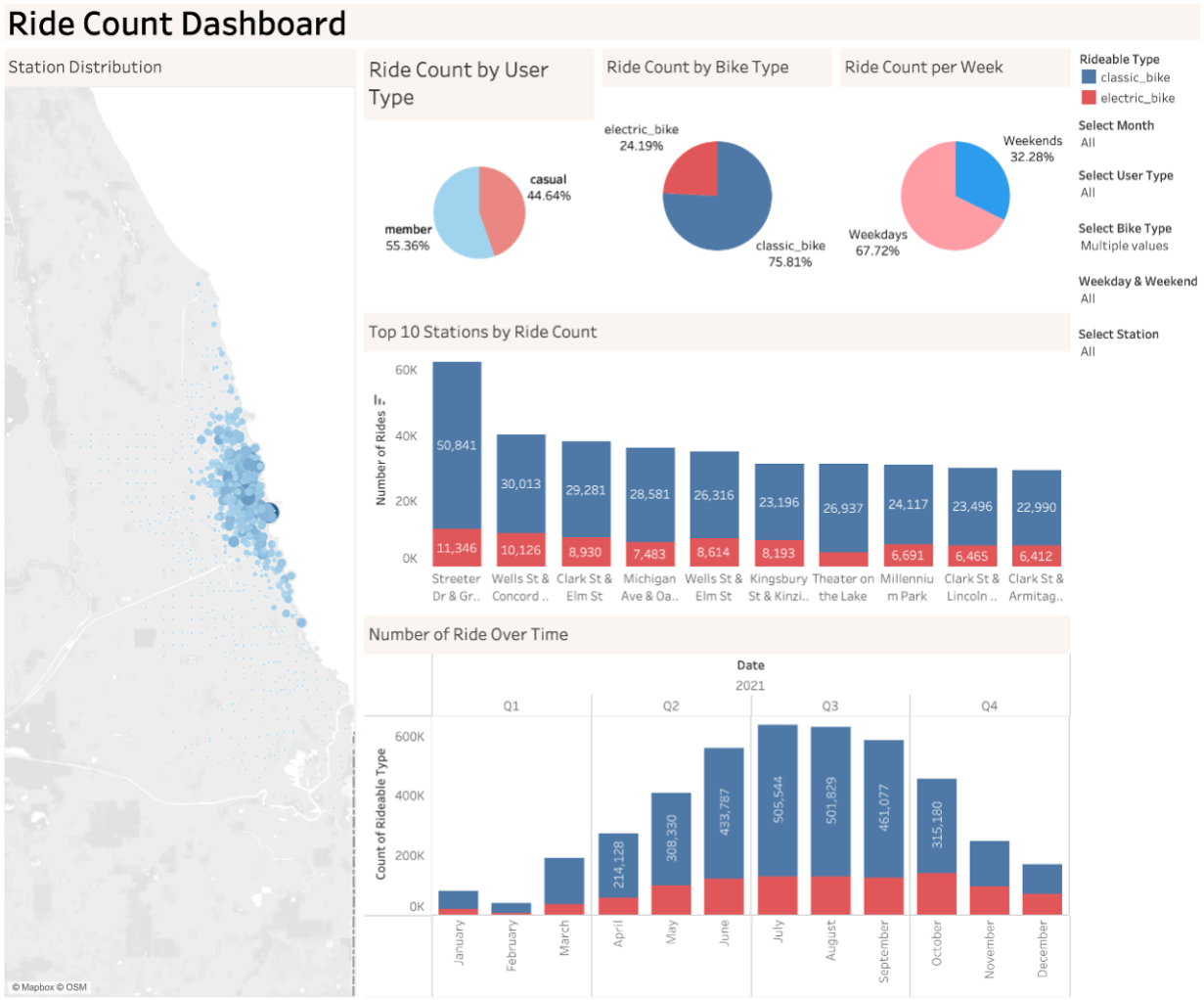
To convert those insights into financial outcomes, Divvy could implement strategic planning on bike distribution, user engagement, and bike type analysis.

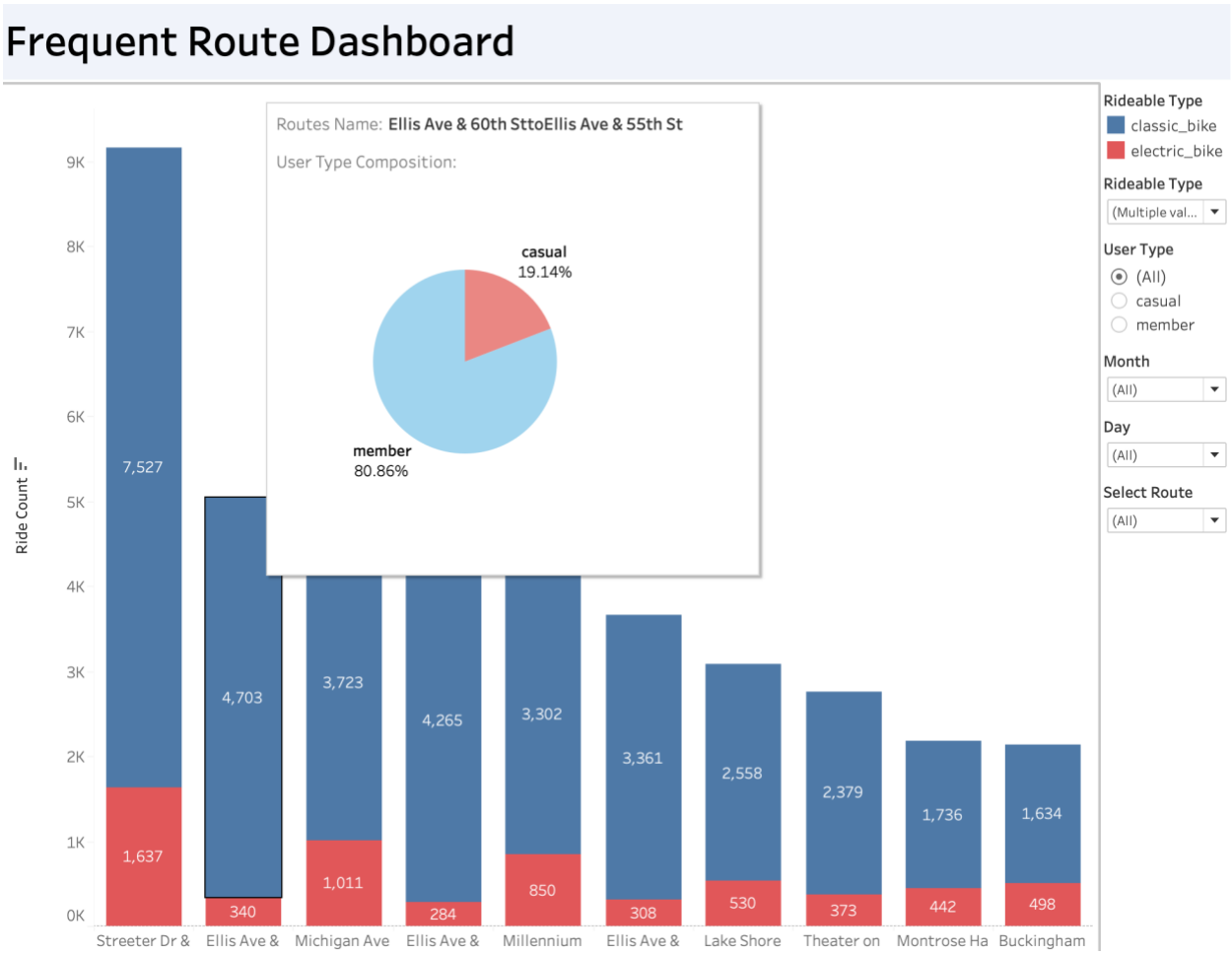
1. Bike Accessibility Analysis: We can see that during any period of highest demand for bike usage, the casual member rate is at a low ebb. This could be due to a lower accessibility of casual users to bike at rush hours and more bike implementation may help the company absorb their demand thereby boosting market shares.

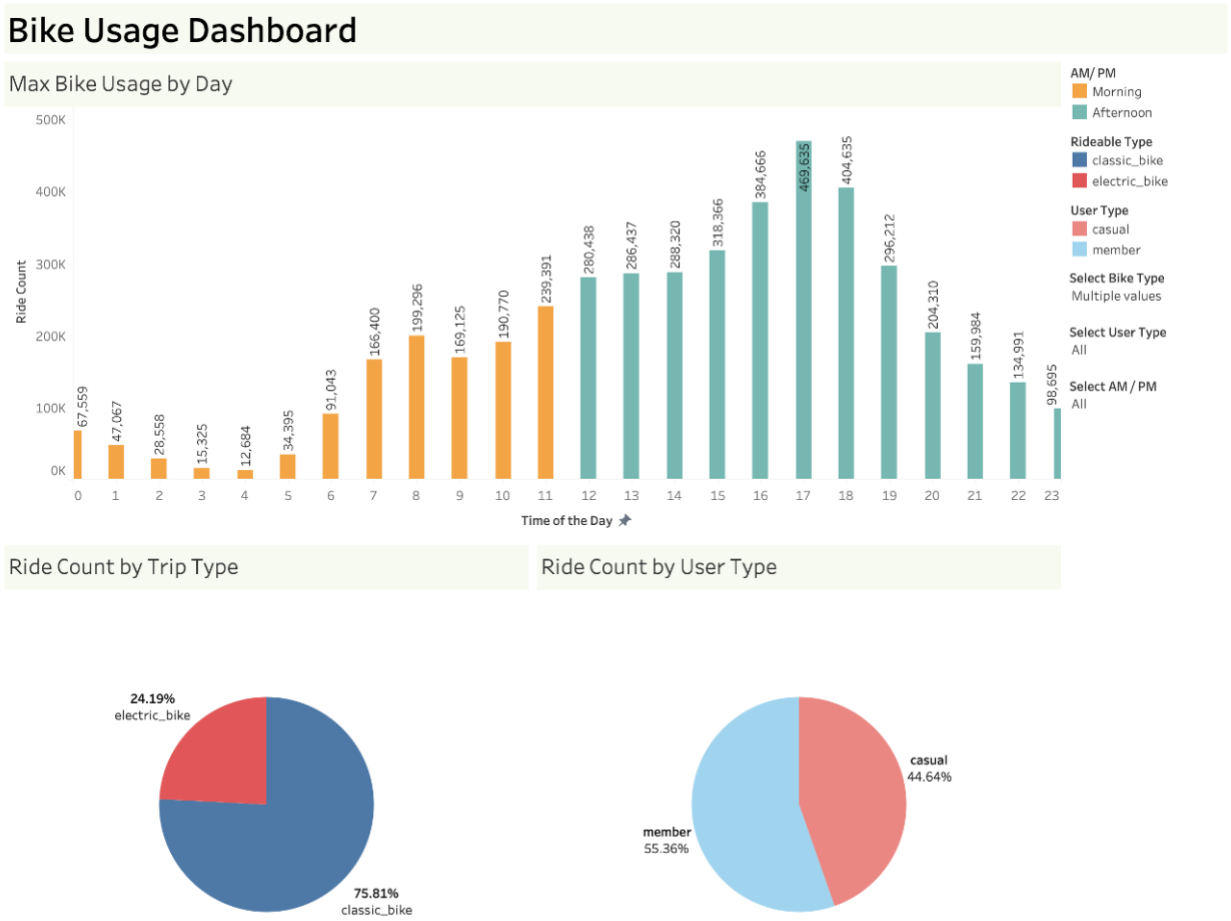
DATA VISUALIZATION:

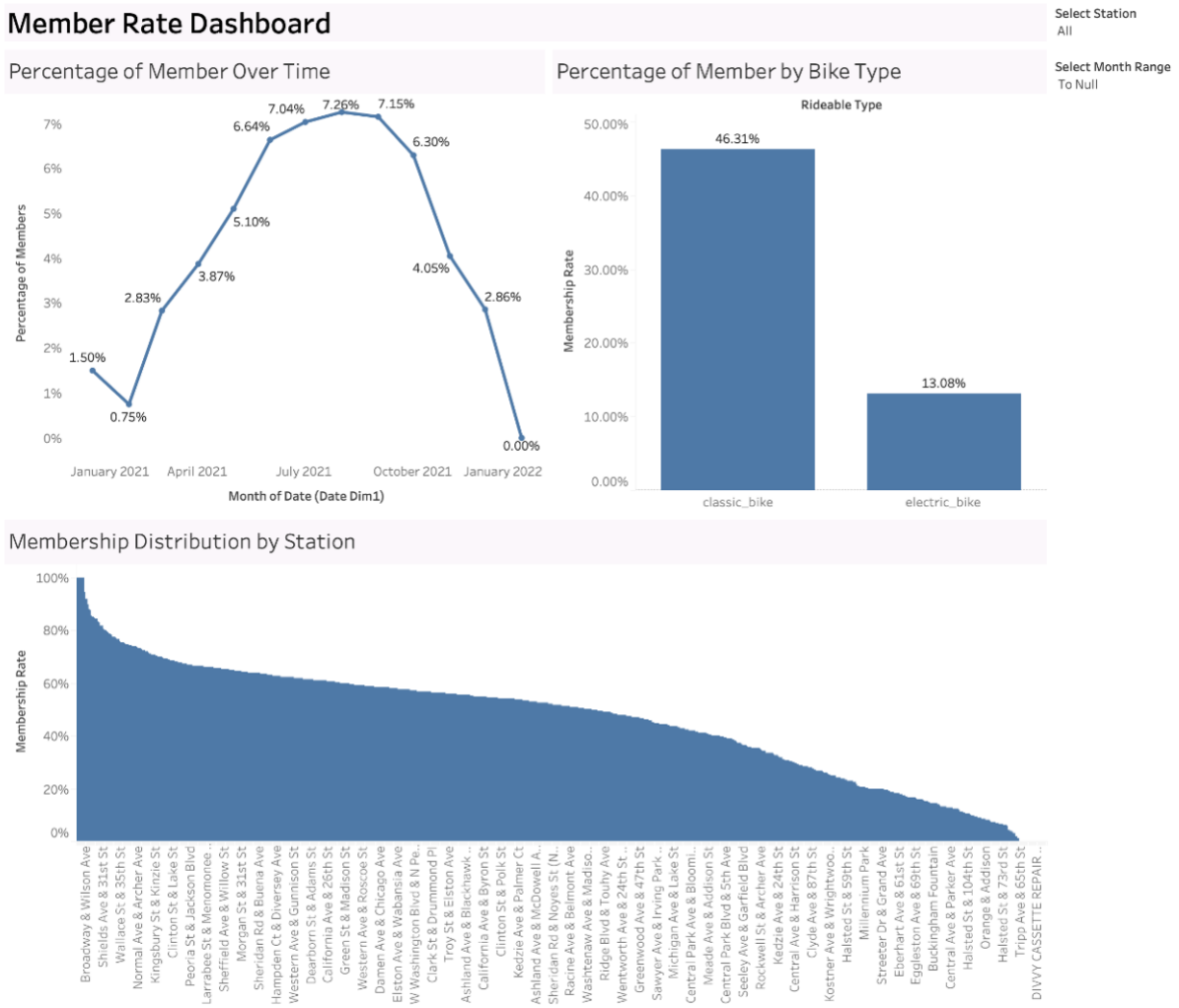
Tableau Dashboards




Ride Count Dashboard
Station Distribution
© Mapbox © OSM
Ride Count by User Type
casual
44.64%
member
55.36%
Ride Count by Bike Type
electric_bike
24.19%
classic_bike
75.81%
Ride Count per Week
Weekends
32.28%
Weekdays
67.72%
Rideable Type
classic_bike
electric_bike
Select Month
All
Select User Type
All
Select Bike Type
Multiple values
Weekday & Weekend
All
Select Station
All
Top 10 Stations by Ride Count
Number of Rides
60K
40K
20K
0K
50,841
11,346
30,013
10,126
29,281
8,930
28,581
7,483
26,316
8,614
23,196
8,193
26,937
24,117
6,691
23,496
6,465
22,990
6,412
Streeter Dr & Gr..
Wells St & Concord ..
Clark St & Elm St
Michigan Ave & Oa..
Wells St & Elm St
Kingsbury St & Kinzi..
Theater on the Lake
Millennium Park
Clark St & Lincoln ..
Clark St & Armitag..
Number of Ride Over Time
Date
2021
Q1
Q2
Q3
Q4
Count of Rideable Type
600K
400K
200K
0K
214,128
308,330
433,787
505,544
501,829
461,077
315,180
January
February
March
April
May
June
July
August
September
October
November
December


Frequent Route Dashboard
Routes Name: Ellis Ave & 60th SttoEllis Ave & 55th St
User Type Composition:
casual
19.14%
member
80.86%
Ride Count
9K
8K
7K
6K
5K
4K
3K
2K
1K
0K
7,527
1,637
4,703
340
3,723
1,011
4,265
284
3,302
850
3,361
308
2,558
530
2,379
373
1,736
442
1,634
498
Streeter Dr &
Ellis Ave &
Michigan Ave
Ellis Ave &
Millennium
Ellis Ave &
Lake Shore
Theater on
Montrose Ha
Buckingham
Rideable Type
classic_bike
electric_bike
Rideable Type
(Multiple val...
User Type
(All)
casual
member
Month
(All)
Day
(All)
Select Route
(All)


Bike Usage Dashboard
Max Bike Usage by Day
Ride Count
500K
400K
300K
200K
100K
0K
67,559
47,067
28,558
15,325
12,684
34,395
91,043
166,400
199,296
169,125
190,770
239,391
280,438
286,437
288,320
318,366
384,666
469,635
404,635
296,212
204,310
159,984
134,991
98,695
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Time of the Day
AM/ PM
Morning
Afternoon
Rideable Type
classic_bike
electric_bike
User Type
casual
member
Select Bike Type
Multiple values
Select User Type
All
Select AM / PM
All
Ride Count by Trip Type
Ride Count by User Type
24.19%
electric_bike
75.81%
classic_bike
casual
44.64%
member
55.36%



Recommendations

During peak hours, there's a notable decrease in casual bike member rates, indicating that casual users face accessibility challenges, and addressing this by increasing bike availability could potentially boost market shares. This trend is particularly evident during rush hours at 6:00 AM and 5:00 PM, suggesting a significant drop in casual usage at these times. In contrast, weekends witness a higher casual member rate than weekdays, likely due to reduced accessibility for casual users during the busier workdays. Interestingly, there's a seasonal pattern in casual member rates, with a peak in July and a low in February, hinting at underlying factors affecting usage that merit further investigation. Evening rush hours see more bike usage, implying a variety of uses beyond morning commutes. For efficient operation, it's crucial to manage inventory effectively, ensuring an adequate supply of classic bikes and adapting electric bike distribution based on local demand. Finally, Divvy's focus should be on refining its products, particularly enhancing classic and electric bikes, and establishing strategic partnerships to improve bike storage facilities, especially where classic bikes are in high demand.

PROJECT CHALLENGES

Technical Challenges

As outlined in previous sections, we faced challenges in data sanity and integrity. The raw dataset contains a large number of missing values across station names, IDs, and geographic coordinates. While in theory, it might be possible to associate some missing station names or IDs with corresponding latitudes and longitudes, the dataset had certain inherent limitations that complicated this approach: 1. Geographic coordinates lacked precision, being accurate to only two decimal places and having a many-to-one relationship with station names and IDs; 2. The many-to-many relationship between station IDs and names resulted in non-unique identifiers. These constraints made the process of replacing missing values not only time-consuming and error-prone but also too demanding for our limited Snowflake capacity.

KPI Calculations Challenges

The key challenges in KPI calculation stem from the vague definition of "docked bike". Our team discovered that docked bikes have a significantly higher average ride duration, with multiple duration values above 24 hours. Since many docked bike trips have different start and end stations, we were unclear about the business process of docked bikes. Therefore, our team was unsure whether they should be counted as part of the analysis. Since the long trip duration from the docked bikes distorts the average trip duration, we decided to exclude the trips with an average duration above 1,500 minutes (25 hours).

FURTHER RESEARCH

The main drawback of the present research is that we lack a user information dataset and a bike information dataset so we can not conduct the user engagement and bike type profitability analysis necessary for profitability assessment. To make further decisions, the company should improve its records to include such information.

Suppose that such a requirement could be fulfilled, some interesting facts may be worth further notice and inquiries:

1. Monthly Differences in Bike Casual Member Rate: the evolution of casual member rates in different months is so smooth that there must be causal factors that lead to such results. A further study on this may help us better understand the periodical member preference for bike usage and make catering decisions.
2. Docked Bike Usage: Dicked bike is the most unpopular type of bike in the three different bike types, especially no member users would like to use this kind of bike. The company may consider reducing the implementation of such types and switching to some more popular types. However, this decision should take the data on profitability as well as casual user preference into consideration.

Market Trend Research

Since all the insights from this report are based on data from 2021, it is critical to adapt future strategic decisions to the evolving market trends. For example, there is a potential rise in demand from people who need to do daily commutes in future years, as many companies start to require return-to-office. This may drive a different trend in demand, thereby influencing the strategic decisions on resource management, marketing plans, and pricing strategies.

APPENDIX

Checklist: [Data Management Final Project: Project 5 - Cyclist Data - Google Sheets]

KPI Calculations

1. Ride Durations

```
-- Average & max durations of each ride (in minutes)
SELECT
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)),2) AS Average_Duration_Minutes,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)),2) AS Max_Duration_Minutes
FROM TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
WHERE TIMESTAMPDIFF(MINUTE, Started_At, Ended_At) <= 1500;
```

Results | Chart

	AVERAGE_DURATION_MINUTES	MAX_DURATION_MINUTES
1	19.92	1500

FINAL.PROJECT5 | Settings

```
-- Average monthly/yearly/daily ride duration
SELECT
    MONTH(b.Started_At) AS Month,
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Average_Duration_Minutes,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Max_Duration_Minutes
FROM TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
WHERE TIMESTAMPDIFF(MINUTE, Started_At, Ended_At) <= 1500
GROUP BY MONTH(Started_At)
ORDER BY MONTH(Started_At);
```

Results | Chart

	MONTH	AVERAGE_DURATION_MINUTES	MAX_DURATION_MINUTES
1	1	13.87	1490
2	2	18.10	1479
3	3	20.52	1499
4	4	21.49	1494
5	5	23.43	1498
6	6	22.45	1495
7	7	21.64	1498
8	8	20.43	1494
9	9	19.32	1498
10	10	16.92	1500
11	11	13.49	1499
12	12	13.05	1500

```sql
-- Average ride duration for each day of week
SELECT
    CASE
        WHEN DAYOFWEEK(Started_At) = 0 THEN 'Sunday'
        WHEN DAYOFWEEK(Started_At) = 1 THEN 'Monday'
        WHEN DAYOFWEEK(Started_At) = 2 THEN 'Tuesday'
        WHEN DAYOFWEEK(Started_At) = 3 THEN 'Wednesday'
        WHEN DAYOFWEEK(Started_At) = 4 THEN 'Thursday'
        WHEN DAYOFWEEK(Started_At) = 5 THEN 'Friday'
        WHEN DAYOFWEEK(Started_At) = 6 THEN 'Saturday'
    END AS DayOfWeek,
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Average_Duration_Minutes,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Max_Duration_Minutes
FROM
    TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
WHERE TIMESTAMPDIFF(MINUTE, Started_At, Ended_At) <= 1500
GROUP BY DAYOFWEEK(Started_At)
ORDER BY DAYOFWEEK(Started_At);
```

	DAYOFWEEK	···	AVERAGE_DURATION_MINUTES	MAX_DURATION_MINUTES
1	Sunday		24.94	1499
2	Monday		19.11	1474
3	Tuesday		17.22	1498
4	Wednesday		16.70	1500
5	Thursday		16.78	1484
6	Friday		18.74	1496
7	Saturday		23.72	1499

```sql
-- The average duration of each ride per hour (use start time)
SELECT
    HOUR(b.Started_At) AS HourOfDay,
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Average_Duration_Minutes,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, Started_At, Ended_At)), 2) AS Max_Duration_Minutes
FROM TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
WHERE TIMESTAMPDIFF(MINUTE, Started_At, Ended_At) <= 1500
GROUP BY HOUR(Started_At)
ORDER BY HOUR(Started_At);
```

	HOUROFDAY	AVERAGE_DURATION_MINUTES	MAX_DURATION_MINUTES
1	0	22.09	1425
2	1	23.20	1450
3	2	23.82	1461
4	3	24.28	1427
5	4	19.22	1392
6	5	13.42	1486
7	6	13.05	1087
8	7	13.40	1467
9	8	14.33	1421
10	9	18.05	1476
11	10	21.41	1490
12	11	22.21	1500
13	12	21.79	1466
14	13	22.88	1496
15	14	23.02	1499
16	15	21.75	1499
17	16	20.04	1497
18	17	19.04	1498
19	18	18.82	1481
20	19	19.26	1476
21	20	19.85	1498
22	21	20.04	1500
23	22	20.59	1469
24	23	21.15	1500

```sql
-- Average ride duration by user type
SELECT
    user_type_dim.User_Type,
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At)), 2) AS Average_Trip_Duration,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At)), 2) AS Max_Trip_Duration
FROM
    TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
JOIN USER_TYPE_DIM user_type_dim ON a.USER_TYPE_KEY = user_type_dim.USER_TYPE_KEY
WHERE TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At) <= 1500
GROUP BY user_type_dim.User_Type
ORDER BY user_type_dim.User_Type;
```

Results Chart

	USER_TYPE	AVERAGE_TRIP_DURATION	MAX_TRIP_DURATION
1	casual	28.28	1500
2	member	13.18	1495

```sql
-- Average ride duration by rideable type
SELECT
    rideable_type_dim.Rideable_Type,
    ROUND(AVG(TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At)), 2) AS Average_Trip_Duration,
    ROUND(MAX(TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At)), 2) AS Max_Trip_Duration
FROM
    TRIP_FACTS a
JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
JOIN RIDEABLE_TYPE_DIM rideable_type_dim ON a.RIDEABLE_TYPE_KEY = rideable_type_dim.RIDEABLE_TYPE_KEY
WHERE TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At) <= 1500
GROUP BY rideable_type_dim.Rideable_Type
ORDER BY rideable_type_dim.Rideable_Type;
```

Results | Chart

	RIDEABLE_TYPE	AVERAGE_TRIP_DURATION	...	MAX_TRIP_DURATION
1	classic_bike	18.37		1500
2	docked_bike	50.40		1500
3	electric_bike	15.61		480

Query Details

Query duration

Rows

```sql
-- Average ride duration by Start_Station
WITH Station_Ride_Durations AS (
    SELECT
        c.Station_ID,
        c.Station_Name,
        ROUND(AVG(TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At)), 2) AS Average_Ride_Duration
    FROM TRIP_FACTS a
    JOIN TRIP_TIME_DIM b ON a.Ride_ID = b.Ride_ID
    JOIN LOCATION_DIM c ON a.START_STATION_KEY = c.STATION_KEY
    WHERE TIMESTAMPDIFF(MINUTE, b.Started_At, b.Ended_At) <= 1500
    GROUP BY c.Station_ID, c.Station_Name
)
SELECT
    Station_ID,
    Station_Name,
    Average_Ride_Duration
FROM Station_Ride_Durations
ORDER BY Station_ID;
```

	STATION_ID	STATION_NAME	...	AVERAGE_RIDE_DURATION
1	13001	Michigan Ave & Washington St		28.97
2	13006	LaSalle St & Washington St		16.55
3	13008	Millennium Park		38.80
4	13011	Canal St & Adams St		16.69
5	13016	St. Clair St & Erie St		21.34
6	13017	Franklin St & Chicago Ave		13.10
7	13021	Clinton St & Lake St		12.91
8	13022	Streeter Dr & Grand Ave		35.79
9	13028	900 W Harrison St		14.45
10	13029	Field Museum		32.54

2. Most Common Rideable Type

```sql
--==================================================
-- Most common rideable type overall
SELECT
    Rideable_Type AS Most_Common_Rideable_Type,
    COUNT(*) AS Counts
FROM
    TRIP_FACTS a
JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
GROUP BY Rideable_Type
ORDER BY Counts DESC
LIMIT 1;
```

Results | Chart

	MOST_COMMON_RIDEABLE_TYPE	COUNTS
1	classic_bike	3241988

```sql
-- Most common rideable type by month
WITH Monthly_Rideable_Counts AS (
    SELECT
        MONTH(c.Started_At) AS Month,
        b.Rideable_Type,
        COUNT(*) AS Counts,
        RANK() OVER (PARTITION BY MONTH(c.Started_At) ORDER BY COUNT(*) DESC) AS ranking
    FROM TRIP_FACTS a
    JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
    JOIN TRIP_TIME_DIM c ON a.Ride_ID = c.Ride_ID
    GROUP BY MONTH(c.Started_At), b.Rideable_Type
),
Ranked_Monthly_Rideable AS (
    SELECT
        Month,
        Rideable_Type,
        Counts
    FROM Monthly_Rideable_Counts
    WHERE ranking = 1
)
SELECT
    Month,
    Rideable_Type AS Most_Common_Rideable_Type,
    Counts
FROM Ranked_Monthly_Rideable
ORDER BY Month;
```

	MONTH	MOST_COMMON_RIDEABLE_TYPE ...	COUNTS
1	1	classic_bike	61414
2	2	classic_bike	34634
3	3	classic_bike	152163
4	4	classic_bike	214128
5	5	classic_bike	308330
6	6	classic_bike	433787
7	7	classic_bike	505544
8	8	classic_bike	501829
9	9	classic_bike	461077
10	10	classic_bike	315180
11	11	classic_bike	153630
12	12	classic_bike	100272

```
-- Most common rideable type by day of the week
WITH Daily_Rideable_Counts AS (
    SELECT
        DAYOFWEEK(c.Started_At) AS Day_Of_Week,
        b.Rideable_Type,
        COUNT(*) AS Counts,
        RANK() OVER (PARTITION BY DAYOFWEEK(c.Started_At) ORDER BY COUNT(*) DESC) AS ranking
    FROM TRIP_FACTS a
    JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
    JOIN TRIP_TIME_DIM c ON a.Ride_ID = c.Ride_ID
    GROUP BY DAYOFWEEK(c.Started_At), b.Rideable_Type
),
Ranked_Daily_Rideable AS (
    SELECT
        Day_Of_Week,
        Rideable_Type,
        Counts
    FROM Daily_Rideable_Counts
    WHERE ranking = 1
)
SELECT
    CASE
        WHEN Day_Of_Week = 0 THEN 'Sunday'
        WHEN Day_Of_Week = 1 THEN 'Monday'
        WHEN Day_Of_Week = 2 THEN 'Tuesday'
        WHEN Day_Of_Week = 3 THEN 'Wednesday'
        WHEN Day_Of_Week = 4 THEN 'Thursday'
        WHEN Day_Of_Week = 5 THEN 'Friday'
        WHEN Day_Of_Week = 6 THEN 'Saturday'
    END AS Day_Of_Week_Name,
    Rideable_Type AS Most_Common_Rideable_Type,
    Counts
FROM Ranked_Daily_Rideable
ORDER BY Day_Of_Week;
```

	DAY_OF_WEEK_NAME	MOST_COMMON_RIDEABLE_TYPE	COUNTS
1	Sunday	classic_bike	508537
2	Monday	classic_bike	405364
3	Tuesday	classic_bike	423495
4	Wednesday	classic_bike	437586
5	Thursday	classic_bike	424233
6	Friday	classic_bike	457386
7	Saturday	classic_bike	585387

```sql
-- Most common rideable type by hour of day
WITH Hourly_Rideable_Counts AS (
    SELECT
        HOUR(c.Started_At) AS Hour_Of_Day,
        b.Rideable_Type,
        COUNT(*) AS Counts,
        RANK() OVER (PARTITION BY HOUR(c.Started_At) ORDER BY COUNT(*) DESC) AS ranking
    FROM TRIP_FACTS a
    JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
    JOIN TRIP_TIME_DIM c ON a.Ride_ID = c.Ride_ID
    GROUP BY HOUR(c.Started_At), b.Rideable_Type
),
Ranked_Hourly_Rideable AS (
    SELECT
        Hour_Of_Day,
        Rideable_Type,
        Counts
    FROM Hourly_Rideable_Counts
    WHERE ranking = 1
)
SELECT
    Hour_Of_Day,
    Rideable_Type AS Most_Common_Rideable_Type,
    Counts
FROM Ranked_Hourly_Rideable
ORDER BY Hour_Of_Day;
```

	HOUR_OF_DAY	MOST_COMMON_RIDEABLE_TYPE ···	COUNTS
1	0	classic_bike	44175
2	1	classic_bike	31046
3	2	classic_bike	18522
4	3	classic_bike	9630
5	4	classic_bike	7948
6	5	classic_bike	23967
7	6	classic_bike	65531
8	7	classic_bike	121258
9	8	classic_bike	143782
10	9	classic_bike	120349
11	10	classic_bike	134436
12	11	classic_bike	168923
13	12	classic_bike	197650
14	13	classic_bike	199482
15	14	classic_bike	199653
16	15	classic_bike	220404
17	16	classic_bike	269812
18	17	classic_bike	341055
19	18	classic_bike	298021
20	19	classic_bike	215302
21	20	classic_bike	144355
22	21	classic_bike	109946
23	22	classic_bike	91051
24	23	classic_bike	65690

```sql
-- Most common rideable per user type
WITH UserType_Rideable_Counts AS (
    SELECT
        c.User_Type,
        b.Rideable_Type,
        COUNT(*) AS Counts,
        RANK() OVER (PARTITION BY c.User_Type ORDER BY COUNT(*) DESC) AS ranking
    FROM
        TRIP_FACTS a
    JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
    JOIN USER_TYPE_DIM c ON a.USER_TYPE_KEY = c.USER_TYPE_KEY
    GROUP BY c.User_Type, b.Rideable_Type
),
Ranked_UserType_Rideable AS (
    SELECT
        User_Type,
        Rideable_Type,
        Counts
    FROM UserType_Rideable_Counts
    WHERE ranking = 1
)
SELECT
    User_Type,
    Rideable_Type AS Most_Common_Rideable_Type,
    Counts
FROM Ranked_UserType_Rideable
ORDER BY User_Type;
```

Results | Chart

	USER_TYPE	MOST_COMMON_RIDEABLE_TYPE	COUNTS
1	casual	classic_bike	1261558
2	member	classic_bike	1980430

```sql
-- The most common rideable type for each station
WITH Station_Rideable_Counts AS (
    SELECT
        Station_ID,
        Station_Name,
        Rideable_Type,
        COUNT(*) AS Counts,
        RANK() OVER (PARTITION BY Station_Name ORDER BY COUNT(*) DESC) AS ranking
    FROM
        TRIP_FACTS a
    JOIN RIDEABLE_TYPE_DIM b ON a.RIDEABLE_TYPE_KEY = b.RIDEABLE_TYPE_KEY
    JOIN LOCATION_DIM c ON a.START_STATION_KEY = c.STATION_KEY
    GROUP BY Station_ID, Station_Name, Rideable_Type
),
Ranked_Station_Rideable AS (
    SELECT
        Station_ID,
        Station_Name,
        Rideable_Type,
        Counts
    FROM Station_Rideable_Counts
    WHERE ranking = 1
)
SELECT
    Station_ID,
    Station_Name,
    Rideable_Type AS Most_Common_Rideable_Type,
    Counts
FROM
    Ranked_Station_Rideable
ORDER BY Station_ID;
```

	STATION_ID	STATION_NAME	MOST_COMMON_RIDEABLE_TYPE
230	201022	Loomis St & 89th St	electric_bike
231	20103	Prospect Sq & 91st St	classic_bike
232	20104	State St & 95th St	classic_bike
233	20105	Halsted St & 96th St	classic_bike
234	20106	Chicago State University	electric_bike
235	20107	Walden Pkwy & 100th St	classic_bike
236	20108	Hale Ave & 107th St	electric_bike
237	20109	Vernon Ave & 107th St	classic_bike
238	20110	Eberhart Ave & 91st St	classic_bike
239	20111	Olive Harvey College	electric_bike
240	20112	Indiana Ave & 103rd St	classic_bike

```sql
-- Most frequented route overall
SELECT
    start_station.STATION_NAME AS Start_Station_Name,
    end_station.STATION_NAME AS End_Station_Name,
    COUNT(*) AS Trip_Count
FROM TRIP_FACTS
JOIN LOCATION_DIM AS start_station ON TRIP_FACTS.start_station_key = start_station.STATION_KEY
JOIN LOCATION_DIM AS end_station ON TRIP_FACTS.end_station_key = end_station.STATION_KEY
GROUP BY start_station.STATION_NAME, end_station.STATION_NAME
ORDER BY Trip_Count DESC
LIMIT 3;
```

Results Chart

	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	13035
2	Michigan Ave & Oak St	Michigan Ave & Oak St	6675
3	Millennium Park	Millennium Park	6470

```sql
-- Most frequented route by month
WITH MonthlyRouteCounts AS (
    SELECT
        MONTH(Started_At) AS Month,
        start_station.STATION_NAME AS Start_Station_Name,
        end_station.STATION_NAME AS End_Station_Name,
        COUNT(*) AS Trip_Count,
        RANK() OVER (PARTITION BY MONTH(Started_At) ORDER BY COUNT(*) DESC) AS Rank
    FROM TRIP_FACTS
    JOIN LOCATION_DIM AS start_station ON TRIP_FACTS.start_station_key = start_station.STATION_KEY
    JOIN LOCATION_DIM AS end_station ON TRIP_FACTS.end_station_key = end_station.STATION_KEY
    JOIN TRIP_TIME_DIM ON TRIP_FACTS.Ride_ID = TRIP_TIME_DIM.Ride_ID
    GROUP BY Month, Start_Station_Name, End_Station_Name
)
SELECT
    Month,
    Start_Station_Name,
    End_Station_Name,
    Trip_Count
FROM MonthlyRouteCounts
WHERE Rank = 1
ORDER BY Month;
```

	MONTH	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	1	Ellis Ave & 60th St	Ellis Ave & 55th St	182
2	2	Ellis Ave & 60th St	Ellis Ave & 55th St	86
3	3	Lake Shore Dr & Monroe St	Lake Shore Dr & Monroe St	637
4	4	Lake Shore Dr & Monroe St	Lake Shore Dr & Monroe St	963
5	5	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1786
6	6	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	2159
7	7	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	2623
8	8	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	2233
9	9	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1689
10	10	Ellis Ave & 60th St	University Ave & 57th St	1265
11	11	Ellis Ave & 60th St	University Ave & 57th St	966
12	12	Ellis Ave & 60th St	University Ave & 57th St	368

```sql
-- Most frequented route for day of week
WITH Daily_Route_Counts AS (
    SELECT
        DAYOFWEEK(TRIP_TIME_DIM.Started_At) AS Day_Of_Week,
        start_station.STATION_NAME AS Start_Station_Name,
        end_station.STATION_NAME AS End_Station_Name,
        COUNT(*) AS Trip_Count,
        RANK() OVER (PARTITION BY DAYOFWEEK(TRIP_TIME_DIM.Started_At) ORDER BY COUNT(*) DESC) AS Rank
    FROM TRIP_FACTS
    JOIN LOCATION_DIM AS start_station ON TRIP_FACTS.start_station_key = start_station.STATION_KEY
    JOIN LOCATION_DIM AS end_station ON TRIP_FACTS.end_station_key = end_station.STATION_KEY
    JOIN TRIP_TIME_DIM ON TRIP_FACTS.Ride_ID = TRIP_TIME_DIM.Ride_ID
    GROUP BY Day_Of_Week, Start_Station_Name, End_Station_Name
)
SELECT
    CASE
        WHEN Day_Of_Week = 0 THEN 'Sunday'
        WHEN Day_Of_Week = 1 THEN 'Monday'
        WHEN Day_Of_Week = 2 THEN 'Tuesday'
        WHEN Day_Of_Week = 3 THEN 'Wednesday'
        WHEN Day_Of_Week = 4 THEN 'Thursday'
        WHEN Day_Of_Week = 5 THEN 'Friday'
        WHEN Day_Of_Week = 6 THEN 'Saturday'
    END AS Day_Of_Week_Name,
    Start_Station_Name,
    End_Station_Name,
    Trip_Count
FROM Daily_Route_Counts
WHERE Rank = 1
ORDER BY Day_Of_Week;
```

	DAY_OF_WEEK_NAME	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	Sunday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	3172
2	Monday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1661
3	Tuesday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1165
4	Wednesday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1208
5	Thursday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1013
6	Friday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1571
7	Saturday	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	3245

```sql
-- Most frequented route hours in a day
WITH Hourly_Route_Counts AS (
    SELECT
        HOUR(TRIP_TIME_DIM.Started_At) AS Hour_Of_Day,
        start_station.STATION_NAME AS Start_Station_Name,
        end_station.STATION_NAME AS End_Station_Name,
        COUNT(*) AS Trip_Count,
        RANK() OVER (PARTITION BY HOUR(TRIP_TIME_DIM.Started_At) ORDER BY COUNT(*) DESC) AS Rank
    FROM TRIP_FACTS
    JOIN LOCATION_DIM AS start_station ON TRIP_FACTS.start_station_key = start_station.STATION_KEY
    JOIN LOCATION_DIM AS end_station ON TRIP_FACTS.end_station_key = end_station.STATION_KEY
    JOIN TRIP_TIME_DIM ON TRIP_FACTS.Ride_ID = TRIP_TIME_DIM.Ride_ID
    GROUP BY Hour_Of_Day, Start_Station_Name, End_Station_Name
)
SELECT
    Hour_Of_Day,
    Start_Station_Name,
    End_Station_Name,
    Trip_Count
FROM Hourly_Route_Counts
WHERE Rank = 1
ORDER BY Hour_Of_Day;
```

	HOUR_OF_DAY	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	0	Millennium Park	Millennium Park	206
2	1	Michigan Ave & 8th St	Michigan Ave & 8th St	126
3	2	Millennium Park	Millennium Park	84
4	3	Emerald Ave & 31st St	Clinton St & Roosevelt Rd	68
5	4	Desplaines St & Jackson Blvd	Peoria St & Jackson Blvd	89
6	5	Sheridan Rd & Irving Park Rd	Pine Grove Ave & Waveland Ave	197
7	6	Clinton St & Washington Blvd	LaSalle St & Jackson Blvd	248
8	7	Artesian Ave & Hubbard St	Wolcott Ave & Polk St	222
9	8	Ellis Ave & 60th St	Ellis Ave & 55th St	493
10	9	Ellis Ave & 60th St	Ellis Ave & 55th St	382
11	10	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	560
12	11	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	829
13	12	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	996
14	13	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1247
15	14	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1299
16	15	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1398
17	16	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1255
18	17	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1259
19	18	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1179
20	19	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	866
21	20	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	515
22	21	Millennium Park	Millennium Park	432
23	22	Millennium Park	Millennium Park	367
24	23	Millennium Park	Millennium Park	322

```sql
-- Most frequented route per user type
WITH User_Type_Route_Counts AS (
    SELECT
        user_type_dim.User_Type,
        start_station.STATION_NAME AS Start_Station_Name,
        end_station.STATION_NAME AS End_Station_Name,
        COUNT(*) AS Trip_Count,
        RANK() OVER (PARTITION BY user_type_dim.User_Type ORDER BY COUNT(*) DESC) AS Rank
    FROM TRIP_FACTS a
    JOIN LOCATION_DIM AS start_station ON a.start_station_key = start_station.STATION_KEY
    JOIN LOCATION_DIM AS end_station ON a.end_station_key = end_station.STATION_KEY
    JOIN USER_TYPE_DIM AS user_type_dim ON a.USER_TYPE_KEY = user_type_dim.USER_TYPE_KEY
    GROUP BY user_type_dim.User_Type, Start_Station_Name, End_Station_Name
)
SELECT
    User_Type,
    Start_Station_Name,
    End_Station_Name,
    Trip_Count
FROM User_Type_Route_Counts
WHERE Rank = 1
ORDER BY User_Type;
```

Results | Chart

	USER_TYPE	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	casual	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	11683
2	member	Ellis Ave & 60th St	Ellis Ave & 55th St	4082

```sql
-- Most frequented route per rideable type
WITH Rideable_Type_Route_Counts AS (
    SELECT
        rideable_type_dim.Rideable_Type,
        start_station.STATION_NAME AS Start_Station_Name,
        end_station.STATION_NAME AS End_Station_Name,
        COUNT(*) AS Trip_Count,
        RANK() OVER (PARTITION BY rideable_type_dim.Rideable_Type ORDER BY COUNT(*) DESC) AS Rank
    FROM TRIP_FACTS a
    JOIN LOCATION_DIM AS start_station ON a.start_station_key = start_station.STATION_KEY
    JOIN LOCATION_DIM AS end_station ON a.end_station_key = end_station.STATION_KEY
    JOIN RIDEABLE_TYPE_DIM AS rideable_type_dim ON a.RIDEABLE_TYPE_KEY = rideable_type_dim.RIDEABLE_TYPE_KEY
    GROUP BY rideable_type_dim.Rideable_Type, Start_Station_Name, End_Station_Name
)
SELECT
    Rideable_Type,
    Start_Station_Name,
    End_Station_Name,
    Trip_Count
FROM Rideable_Type_Route_Counts
WHERE Rank = 1
ORDER BY Rideable_Type;
```

Results | Chart

	RIDEABLE_TYPE	START_STATION_NAME	END_STATION_NAME	TRIP_COUNT
1	classic_bike	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	7528
2	docked_bike	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	3869
3	electric_bike	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	1638

Query Details

Query duration

Rows

Query ID 01b32715-0002-