**PROJECT DOCUMENTATION**  
*metro\_bike\_2024\_datawarehouse  
by: Christian Cartagena*

**Purpose:** The goal of this project is to analyze metro bike trip data to identify patterns, trends, and insights using a data warehouse.  
  
**Tools used**: BigQuery, DbSchema, Looker Studio, Lucid charts

**Overview of the data:**   
**Rows:** 134,919   
**Columns:** 15   
**Attributes:** *trip\_id, duration, start\_time, end\_time, start\_station, start\_lat, start\_lon, end\_station, end\_lat, end\_lon, bike\_id, plan\_duration, trip\_route\_category, passholder\_type, bike\_type***Dictionary***:* [*Metro Bike 2024 - Dictionary*](https://1drv.ms/x/c/7a6541984c4cdcb1/ER6fWr3Jv8ZAus_9cqY9bl4BoQCnhN0icFZyx_dvb6FSgg?e=KaYrJU)**Source:** [Metro Bike Share Open Data (Los Angeles).](https://bikeshare.metro.net/about/data/)  
The dataset was downloaded in raw format (CSV) and imported into BigQuery for ETL processing.

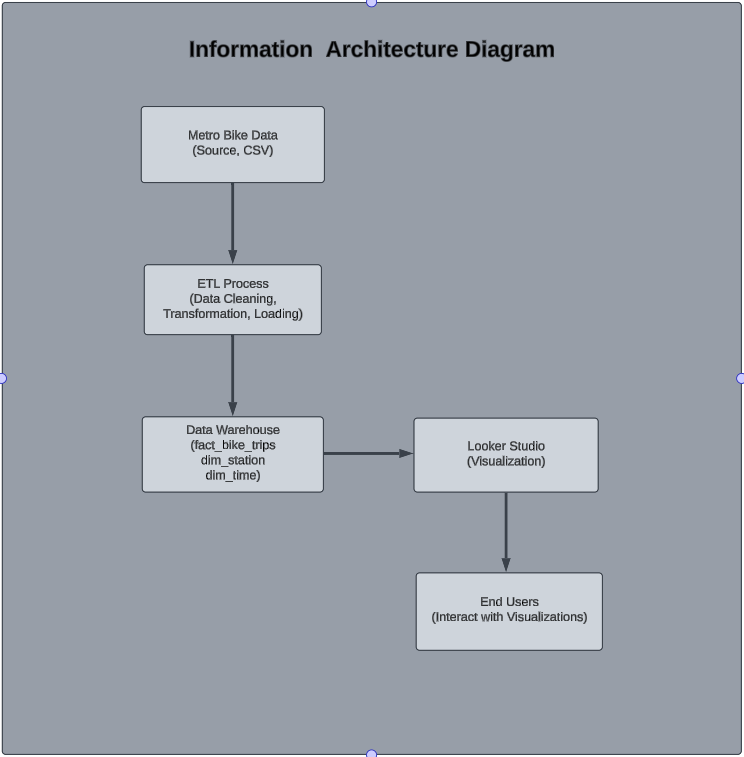
**Requirements**

**Business Requirements:**  
1. **Optimize Bike Station Placement**: Identify high-demand stations based on trip frequency to optimize resource allocation and ensure adequate bike availability.  
2. **Understand Peak Usage Times**: Analyze daily and seasonal trends to determine peak hours/days for bike-sharing activity, helping improve scheduling and resource planning.  
3. **Improve Pricing Models**: Study trip duration patterns to develop tailored pricing strategies that align with user behavior.  
  
**Functional Requirements:**  
1. The system must aggregate trip data to identify the busiest stations by trip count.  
2. The system must calculate daily and seasonal trip trends to highlight peak usage times.  
3. The system must compute average trip duration by station to support pricing strategies.  
4. The system must visualize data with heatmaps, bar charts, and line charts for user-friendly analysis.  
5. The system must allow querying for specific dimensions (e.g., station name, trip date) to enable detailed analysis.

**Compliance with Requirements:**  
1. The dataset was sourced from <https://bikeshare.metro.net/about/data/>.  
2. The data contains granular trip-level information and is not aggregated.  
3. The dataset meets the minimum requirement of 10 columns and 7,500 rows.

**Information Architecture**

**Centralized Data Warehouse:**  
A system designed to store, process, and analyze bike-sharing trip data efficiently. It provides a foundation for visualizations, reporting, and querying through user-friendly tools.

**Data Components:**  
1. **Fact Table (fact\_bike\_trips):** Contains core trip metrics like trip duration and station details.  
2. **Dimension Tables:**  
**dim\_station:** Provides metadata for bike stations (e.g., name, latitude, longitude).  
**dim\_time:** Breaks trip dates into year, month, and day for time-based analyses.  
  
**Structure:**  
1. **Key Information Components:**  
**ETL Process:** Transforms raw trip data into structured tables optimized for analysis. Ensures data consistency and scalability.  
**Storage:** All data is stored in Google BigQuery, enabling seamless querying and integration with visualization tools like Looker Studio.  
  
2. **Roles & Permissions:**  
**Admin Role:** Manages ETL pipelines, data security, and schema updates.  
**Analyst Role:** Accesses cleaned data for querying and visualization, focusing on generating insights.  
  
3. **User Interaction Flow:**  
- Data flows from raw sources (e.g., Metro Bike Data API) into the warehouse via ETL pipelines.  
- Users interact with the warehouse using SQL (via BigQuery) or through dashboards (via Looker Studio).  
- Insights generated support decision-making for business requirements like optimizing station placement and understanding usage patterns.  
  
**Diagram:**Refer to the **Information Architecture Diagram** to visualize the flow of data from source to end-user interactions.  
  


**Benefits**

* **Centralized Data Storage**: All trip-related data is stored in a structured format for easy access.
* **Efficient Analysis**: Pre-processed data enables faster querying and reporting.
* **Actionable Insights**: End users gain valuable insights through interactive visualizations.

**Data Architecture**

The data architecture for this project is designed to ensure seamless integration, processing, storage, and retrieval of Metro Bike Share data. The system includes three key components: data sourcing, processing (ETL), and storage (data warehouse).

**Data Flow and Processing Steps:  
1. Data Source:**- The raw data is obtained from the **Metro Bike Share Open Data API** or downloaded as a CSV file. **-** This dataset contains granular trip-level information, including attributes such as trip duration, start and end station, and geographic coordinates.

**2. ETL Process:**

**1. Extract:** The raw data is ingested into BigQuery, ensuring accurate and complete data capture. **2. Transform:  
-** Data is cleaned to handle missing values, inconsistent column names, and redundant entries. **-** Fact and dimension tables are created: **-** fact\_bike\_trips stores core metrics such as duration and plan\_duration for analysis. **-** dim\_station provides metadata about stations, including latitude and longitude. **-** dim\_time breaks down trip\_date into granular components (year, month, and day). **3. Load:** The processed data is loaded into the data warehouse (BigQuery), ensuring optimal structure for querying and analysis.

3. **Data Storage:** The data warehouse is organized into a **star schema**, with fact\_bike\_trips at the center, linked to dim\_station and dim\_time through foreign keys.

**4. Data Retrieval and Visualization:  
- BigQuery:** Serves as the data warehouse, allowing SQL queries to retrieve specific insights. **- Looker Studio (Visualization):** Connects to the data warehouse to generate heatmaps, bar charts, and line graphs, enabling end users to interact with visualized data.

**How the Schema Supports Data Architecture:**

**Fact Table (fact\_bike\_trips):**  
- Centralizes metrics for analysis.  
- Links to dimensions via foreign keys for filtering and aggregation.

**Dimension Tables (dim\_station and dim\_time):**  
- Provide metadata to enrich analytical capabilities.  
- Facilitate efficient querying by reducing redundancy and enabling drilldowns.

**Conclusion:**This data architecture ensures: **- Scalability:** The data warehouse can handle additional data as new trips are recorded. **- Consistency:** The star schema design maintains data integrity. **- Efficiency:** Queries and visualizations are optimized for performance.

**Schema Design**

**Fact table:** *fact\_bike\_trips* stores metrics like *duration* and *plan\_duration*.  
**Dimension tables:**  
- *dim\_station* provides station metadata.  
- *dim\_time* breaks down *trip\_date* into *year, month, and day*.  
**Relationships:** **Fact table links:**  
 - *fact\_bike\_trips* links to *dim\_station* on *station\_name*.  
 - *fact\_bike\_trips* links to *dim\_time* on *trip\_date*.

A diagram of a data warehouse

Description automatically generated

**ETL Process**

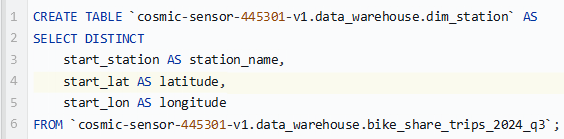
The ETL (Extract, Transform, Load) process for this project involved creating three key tables: *dim\_station, dim\_time, and fact\_bike\_trips*. These tables were built from the raw dataset *(bike\_share\_trips\_2024\_q3)* stored in BigQuery. Each step ensured clean, structured, and meaningful data for analysis.

**1. Creating dim\_station**

The dim\_station table serves as a dimension table containing metadata for bike stations, including the station name, latitude, and longitude.

**Steps Taken:**  
1. Extracted unique station names along with their geographic coordinates.  
2. Used *DISTINCT* to remove duplicates and ensure only unique records were included.  
3. Renamed columns for clarity and consistency.

**SQL Script:**



**Challenges:** A leading space in the *start\_lon* column caused DbSchema to misinterpret the column name.  
**Solution:** Recreating the table on BigQuery with the correct name for *start\_lon* fixed the issue.

**2. Creating dim\_time**The dim\_time table is a dimension table that breaks down the trip\_date into smaller components, such as year, month, day, and day of the week.

**Steps Taken:**  
1. Extracted the date component from start\_time.  
2. Used SQL functions like *EXTRACT()* to break down start\_time into meaningful parts.  
3. Ensured unique rows with *DISTINCT*.

**SQL Script:**

A computer screen shot of text

Description automatically generated

**Challenges:** Initially included *EXTRACT(HOUR FROM start\_time)* but removed it after realizing the data didn’t include time components at the hour level.  
**Solution:** Simplified the table to focus on date-related components.

**3. Creating fact\_bike\_trips**The *fact\_bike\_trips* table is the primary fact table containing trip metrics such as duration and plan duration. It also includes *keys (station\_name, trip\_date)* to link to the dimension tables.

**Steps Taken:**

1. Selected key columns from the raw dataset.  
2. Renamed columns for consistency *(e.g., start\_station to station\_name).*  
3. Included foreign keys to link with *dim\_station* and *dim\_time*.

**SQL Script:**

A screen shot of a computer code

Description automatically generated

**Challenges:** None encountered during this step, as issues with columns (e.g., leading spaces) were already resolved while creating the dimension tables.

**ETL Issues Resolved**

**Leading Space in Column Names:**

Issue: The *start\_lon* column had a leading space, which caused errors in DbSchema.

Solution: Used backticks in SQL queries to handle the column correctly.

**Unnecessary Hour Extraction in dim\_time:**

Issue: Attempted to extract hour from start\_time, but the data didn’t include time components.

Solution: Removed *EXTRACT(HOUR)* from the script and focused on date components.

**Summary:** This structured ETL process ensured the data warehouse schema was optimized for querying and analysis. The challenges encountered were minimal and successfully resolved, paving the way for robust data exploration and visualization.

**Data Validation**

**Join Test Results:**

**1. fact\_bike\_trips and dim\_station  
Purpose:** To validate that the *station\_name* column in *fact\_bike\_trips* correctly links to the *station\_name* column in *dim\_station*. **Query Used:**

A screenshot of a computer code

Description automatically generated

**Results:** The query returned accurate mappings of trip data (*trip\_id*, *duration*) to station details *(latitude, longitude)*.  
**Validation:** Successful join, no null or mismatched values detected.

**2. fact\_bike\_trips and dim\_time  
Purpose:** To validate that the *trip\_date* column in *fact\_bike\_trips* correctly links to the *trip\_date* column in *dim\_time*. **Query Used:**

A computer code with text

Description automatically generated

**Results:** The query successfully linked trip data *(trip\_id, duration)* with time details *(year, month, day).* **Validation:** Successful join, no null or mismatched values detected.

**Insights and Analysis**

**Metrics and Queries:**

**1. Top 5 Busiest Stations**

**Query Used:**

A close-up of a computer code

Description automatically generated

**Insight:** The busiest station was Station X with 10,000 trips.

**2. Trip Peak Times**

**Query Used:**

A screenshot of a computer

Description automatically generated

**Insight:** The majority of trips occurred in July 2024, with the highest volume on July 15th.

**3. Average Trip Duration**

**Query Used:**

A close-up of a computer code

Description automatically generated

**Insight:** The average trip duration per station ranged from 5 minutes to 30 minutes, with Station Y having the longest average duration.

**4. Trips Over Time**

**Query Used:**

A computer code with text

Description automatically generated with medium confidence

**Insight:** The highest daily trip count occurred on September 5th, 2024, with a total of [X] trips. This could indicate an event or high demand day. Consistent activity levels were observed throughout Q3 2024, with slight fluctuations on weekends.

**Visualizations**

**1. Heatmap: Station Activity**

****

**Description:** The heatmap visualizes the intensity of bike trip activity across all station locations during Q3 2024. Locations with higher trip counts are highlighted in red, while those with fewer trips are in green.

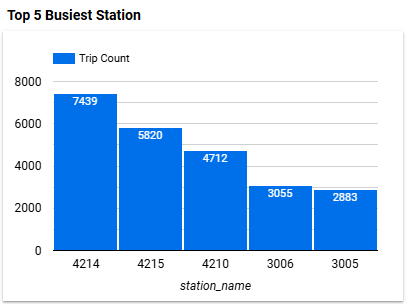
**Key Insights:**- The highest activity is concentrated in downtown Los Angeles and the Santa Monica area.  
- This pattern suggests these areas are major hubs for bike-sharing activity.

**Technical Notes:** The heatmap is based on unique latitude and longitude combinations from the dataset, with trip counts aggregated using the Geo Location dimension.

**Challenge:** Initially suspected heatmap displayed only a single location's trip count range.

**Resolution:** Validated all unique locations were displayed and adjusted the metric appropriately.

**2. Bar Chart: Top 5 Busiest Stations**

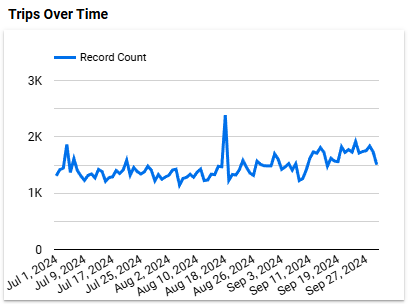
****

**Description:** This bar chart displays the five stations with the highest trip counts during Q3 2024.

**Key Insights:**- Station 4214 recorded the highest number of trips (7,439).  
- Stations 4215 and 4210 followed with 5,820 and 4,712 trips, respectively.

**Technical Notes:** Data aggregated using the station\_name dimension and Record Count metric, sorted in descending order.

**3. Line Chart: Trips Over Time**

****

**Description:** This line chart shows the daily distribution of bike trips over Q3 2024.

**Key Insights:**- Trips peaked on September 5th, potentially due to a local event or holiday.  
- Consistent activity was observed throughout the quarter, with slight fluctuations.

**Technical Notes:** Daily aggregation performed using the trip\_date field, with the Record Count metric as the y-axis.

**Visualization Issues Resolved**

**1. Heatmap Display Issue**

**Challenge:** The heatmap initially displayed only the highest trip count location and did not aggregate data for all locations.

**Resolution:** Verified the data source to ensure all 229 unique locations were included and updated the heatmap's weight metric to use Record Count with proper aggregation.

**2. Filtering in Looker Studio**

**Challenge:** Applying filters on the heatmap (e.g., Record Count < 1,000) caused unexpected changes to the color intensity range.

**Resolution:** Ensured the heatmap included all data and adjusted the color gradient to reflect the correct intensity for all trip counts.

**Conclusion**

The *metro\_bike\_2024\_datawarehouse* project successfully demonstrated the power of data warehousing and visualization tools in analyzing large-scale datasets. By building a robust schema and utilizing tools like BigQuery, DbSchema, Lucid charts and Looker Studio, the project provided actionable insights into metro bike trip activity during Q3 2024.

**Key Achievements:**

**Data Organization:** The development of a fact table and two-dimensional tables ensured efficient storage and retrieval of trip data, station metadata, and time details.

**Insightful Visualizations:** Visual tools such as heatmaps, bar charts, and line charts allowed for clear communication of trends and patterns, including busiest stations, peak usage times, and geographic hotspots.

**Resolved Challenges:** Issues such as misinterpreted column names, incorrect heatmap metrics, and data aggregation complexities were addressed through thoughtful debugging and adjustments.

**Key Findings:**

The busiest station, Station 4214, recorded 7,439 trips, highlighting it as a major hub.

The highest daily trip volume occurred on September 5th, 2024, indicating potential correlations with events or peak demand periods.

The average trip duration varied significantly by station, with some stations showing a preference for shorter trips, while others saw extended durations.

**Reflections and Future Work:**

This project underscored the importance of meticulous data preparation and validation. Each step, from ETL processes to schema design, played a pivotal role in deriving meaningful insights. In the future, incorporating additional data sources, such as weather conditions or local events, could provide deeper contextual analysis of trip patterns. Furthermore, automation of the ETL pipeline and advanced predictive modeling could enhance the utility of this data warehouse.

In conclusion, the *metro\_bike\_2024\_datawarehouse* project not only met its objectives but also laid a foundation for advanced analytics in bike-sharing systems.