Luca Savarese

CIS 4400

Metro Bikeshare Data Analysis Project

https://github.com/LucaSavarese104/Metro-bikeshare-data

1. Data Requirements:

Data sources:

- **Bikeshare Trip Data:**
  - https://bikeshare.metro.net/wp-content/uploads/2024/10/metro-trips-2024-q3.zip
  - Includes information about individual bike trips, including trip duration, start and end times, start and end stations, passholder types, and bike types.
- **Bikeshare Station Data:**
  - https://bikeshare.metro.net/wp-content/uploads/2024/10/metro-bike-share-stations-2024-10-01.csv
  - Includes information about bike stations, such as station IDs, kiosk names, latitude and longitude, and regional classifications.

Data Characteristics:

- **Data Columns**: trip_id, duration, start_time, end_time,start_station, start_kiosk_name, start_region, start_lat, start_lon, end_station, end_kiosk_name, end_region, end_lat, end_lon, trip_route_category, passholder_type, and bike_type
- **Number of Rows**: 131760

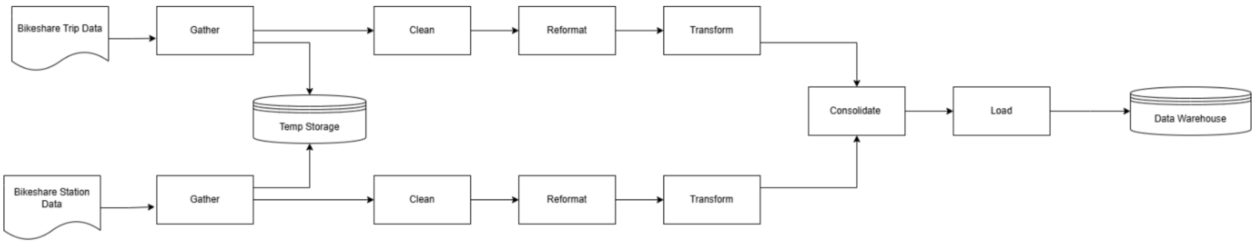
2. Business Requirements:

- **Peak Times Analysis:** Understand when bike usage is highest.
- **Station Demand:** Identify the busiest and least-used stations.
- **Passholder Trends:** Explore how different passholder types contribute to bike usage.

3. Functional Requirements:

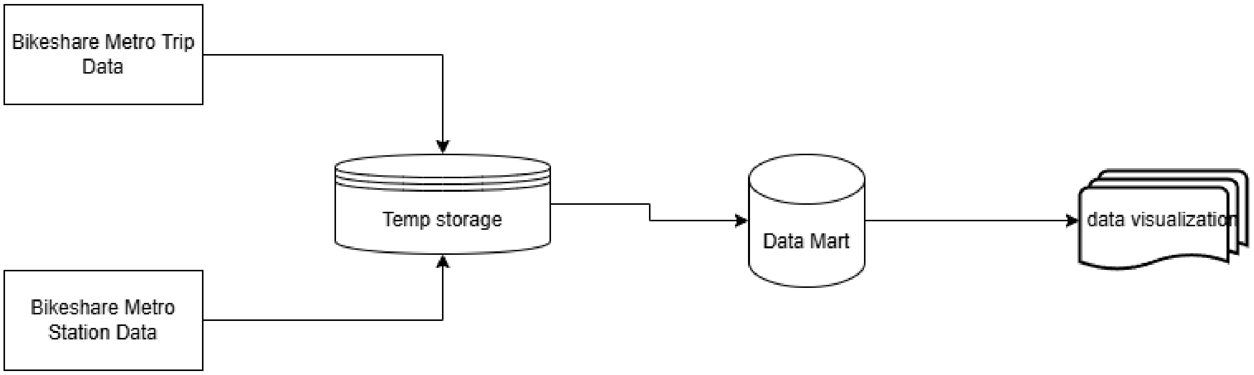
- **Data Loading**: Import data into the analysis tool.
- **Data Cleaning**: Remove missing or inconsistent values.
- **Time-Based Analysis**: Group trips by hour/day to find peak usage times.
- **Station Analysis**: Count trips starting and ending at each station.
- **Passholder Trends:** Explore how different passholder types contribute to bike usage.

4. Information Architecture:



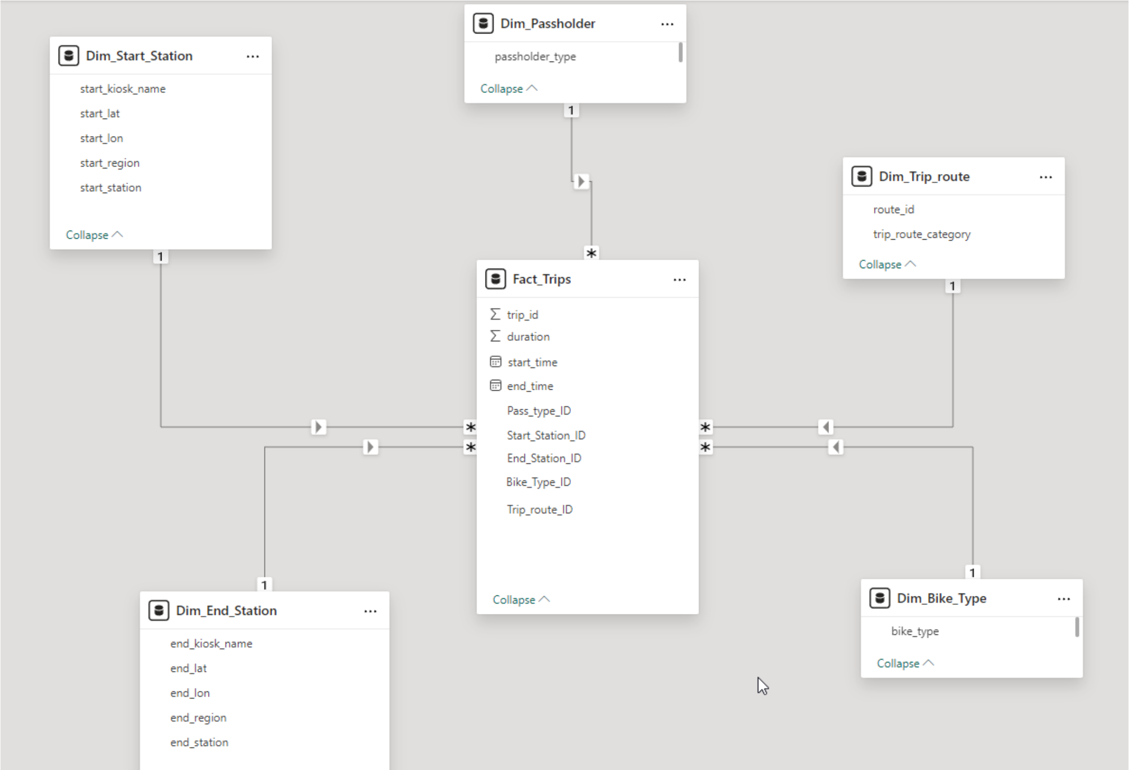
This information architecture provides the general data flow of the Metro Bikeshare System. The whole data cycle begins with collecting trip and station data, then cleaning, reformatting, and transforming to ensure consistency and quality. Data is staged into a temporary space before ultimately being consolidated. Then, the consolidated data is loaded into the data warehouse which will serve as the primary resource for analysis. The data warehouse will power visualizations to allow us to explore different types of analytics such as peak usage times, station demand, and passholder trends.

5. Data Architecture:



The data architecture for the Metro Bikeshare project employs an ETL (Extract, Transform, Load) mechanism for data management and processing. Raw bikeshare trip and station data are consolidated, cleaned, and formatted using Python before being transformed into a dimensional model. The processed data is stored temporarily in a staging area and then uploaded to a centralized data mart optimized for analytical queries and seamless integration with visualization tools. This architecture supports analyses related to peak usage hours, station demand, and trends in passholder numbers.

6. Dimensional Modeling:



Fact table:

Fact_Trips:

```
CREATE TABLE Fact_Trips (

trip_id INT PRIMARY KEY,
duration INT,
start_time DATETIME,
end_time DATETIME,
start_station_id INT,
end_station_id INT,
passholder_type_id INT,
bike_type_id INT,
route_id INT,
FOREIGN KEY (start_station_id) REFERENCES Dim_Start_Station(start_station_id),
FOREIGN KEY (end_station_id) REFERENCES Dim_End_Station(end_station_id),
FOREIGN KEY (passholder_type_id) REFERENCES Dim_Passholder(passholder_type_id),
FOREIGN KEY (bike_type_id) REFERENCES Dim_Bike_Type(bike_type_id),
FOREIGN KEY (route_id) REFERENCES Dim_Trip_Route(route_id)

);
```

Dimension Tables:

Dim_Start_Station:

```
CREATE TABLE Dim_Start_Station (

start_station_id INT PRIMARY KEY,
start_kiosk_name VARCHAR(255),
start_region VARCHAR(255),
start_lat DOUBLE,
start_lon DOUBLE

);
```

Dim_End_Station:

```
CREATE TABLE Dim_End_Station (

end_station_id INT PRIMARY KEY,
end_kiosk_name VARCHAR(255),
end_region VARCHAR(255),
end_lat DOUBLE,
end_lon DOUBLE

);
```

Dim_Passholder:

```
CREATE TABLE Dim_Passholder (

passholder_type_id INT AUTO_INCREMENT PRIMARY KEY,
passholder_type VARCHAR(255)

);
```

Dim_Bike_Type:

```
CREATE TABLE Dim_Bike_Type (

bike_type_id INT AUTO_INCREMENT PRIMARY KEY,

bike_type VARCHAR(255)
);
```

Dim_Trip_Route:

```
CREATE TABLE Dim_Trip_Route (

route_id INT AUTO_INCREMENT PRIMARY KEY,
trip_route_category VARCHAR(255)

);
```