**Table of Contents**

1. [Project Overview](#project-overview)
2. [Public Data Source](#public-data-source)
3. [Tasks](#tasks)
   1. [Data Extraction and Initial Exploration](#data-extraction-and-initial-exploration)
   2. [Schema Design](#schema-design)
   3. [Data Transformation and Loading (ETL)](#data-transformation-and-loading-etl)
   4. [Data Analysis](#data-analysis)
   5. [Additional Task (Optional)](#additional-task-optional)
4. [Submission](#submission)

**Project Overview**

The goal of this project is to demonstrate the ability to work with data from a public data source, transform it, and perform analysis using various SQL operations. The project involves creating a simple data warehouse schema, performing ETL processes, and generating insights from the data.

**Public Data Source**

The data used in this project is the New York City Taxi and Limousine Commission (TLC) Trip Record Data for January 2023. The data can be downloaded from the following link: [TLC Trip Record Data](https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page).

**Tasks**

**1. Data Extraction and Initial Exploration**

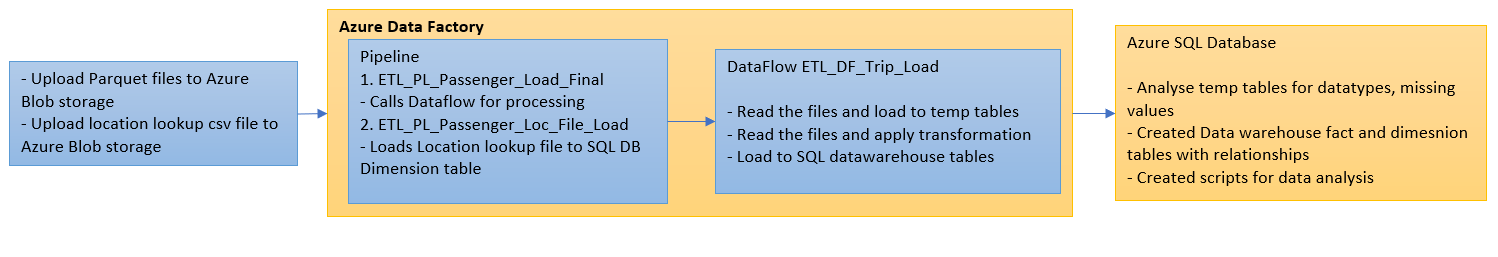
* **Download the TLC trip record data for January 2023.**
* **Load the data into Microsoft SQL Server or SQLite.**
* **Perform initial data exploration to understand the dataset. Document any observations about the data, such as data types, missing values, and distributions.**

**Documentation:**

**Tech stack:**

|  |  |
| --- | --- |
| Storage | Azure Blob Storage |
| ETL | Azure Data Factory |
| Database | Azure SQL DB |

**ETL Architecture:**

****

1. **Data Download:**

Below source files are downloaded.

<https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-01.parquet>

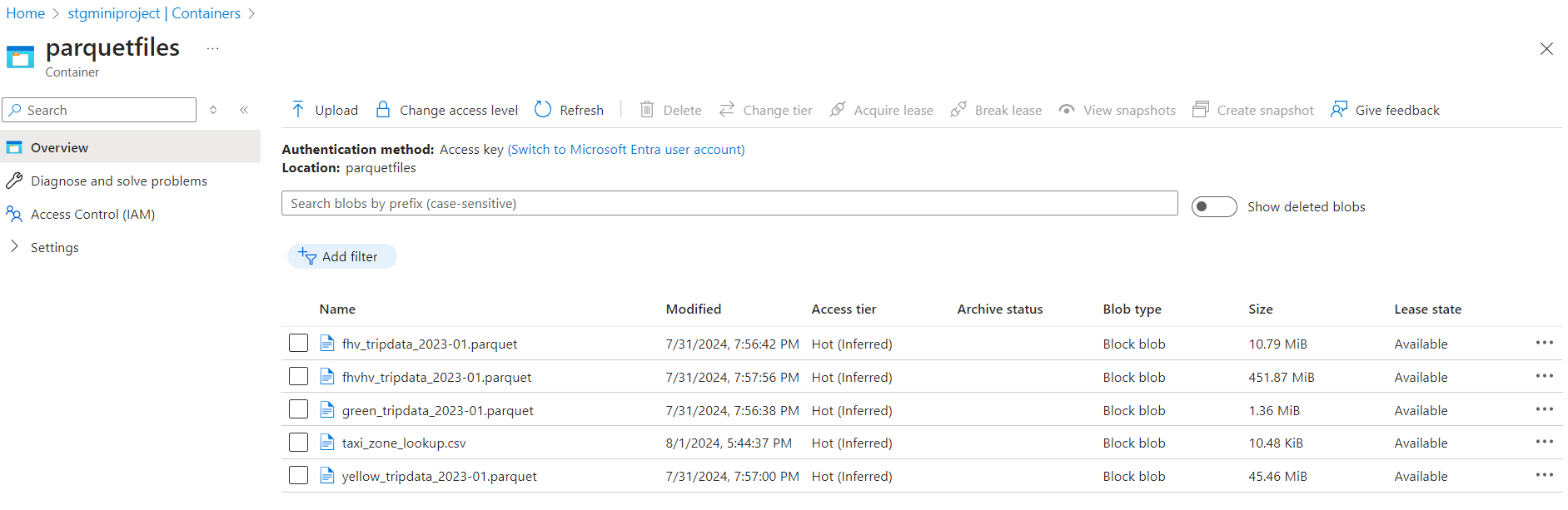
<https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2023-01.parquet>

<https://d37ci6vzurychx.cloudfront.net/trip-data/fhv_tripdata_2023-01.parquet>

<https://d37ci6vzurychx.cloudfront.net/trip-data/fhvhv_tripdata_2023-01.parquet>

<https://d37ci6vzurychx.cloudfront.net/misc/taxi_zone_lookup.csv>

Azure Blob Storage File upload:



1. **Temp Data Loading:**

|  |  |
| --- | --- |
| Source File | Table |
| yellow\_tripdata\_2023-01.parquet | dbo.YelloTripData |
| green\_tripdata\_2023-01.parquet | dbo.GreenTripData |
| fhv\_tripdata\_2023-01.parquet | dbo.HVTripData |
| fhvhv\_tripdata\_2023-01.parquet | dbo.FHHVTripData |

1. **Initial Exploration:**

**Table DDLS :**

****

Below are the observations which could impact the final analysis. Areas of verification are missing values, datatype consistency for key columns and null or invalid values for key columns ( Passenger count, pickup and drop off times, amounts and location IDs )

* Tables were auto created from the files
  + Data Types:
    - pickup\_datetime column with datetime datatype.
    - dropoff\_datetime with datetime datatype.
    - datatypes were consistent across the tables for tripdistance, amounts.
  + Missing Values:

HVTripData:

* + - Location IDs missing
    - Missing columns required for final model such as amount, passenger count, trip distance

greentripdata:

* Passengercount is null for many records
* Few records with older date eg: year 2009, are present in the file

YellowTripData:

* Future date records, ie, Feb 1st 2023 data is present
* passenger count is null for many records

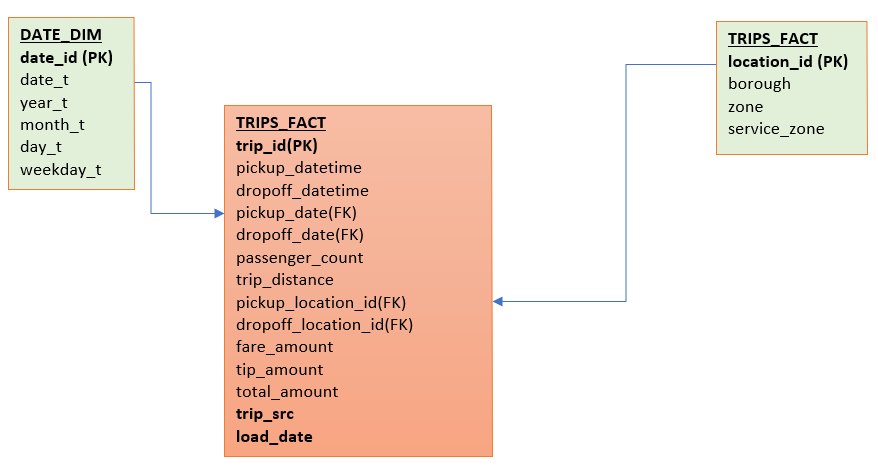
**2. Schema Design**

Design a star schema for a data warehouse with the following components:

Modified the below columns for modelling:

* Added audit columns Trip\_src ( to differentiate the taxi source file ) and load\_date ( date of data load ) to the trips\_fact model table
* Added new derived columns pickup\_date and dropoff\_date to store date from source columns pickup\_datetime and dropoff\_datetime in order to match the datatype with date\_dim dimension table date column.
* Added Unique constraint to Date\_t column in date\_dim table inorder to make it a foreign key
* Modified column names in date\_dim tables by adding ‘\_t’ as the column names were keywords.
* **Fact Table: trips\_fact**
  + trip\_id (Primary Key)
  + pickup\_datetime
  + dropoff\_datetime
  + pickup\_date(Foreign key)
  + dropoff\_date(Foreign key)
  + passenger\_count
  + trip\_distance
  + pickup\_location\_id(Foreign key)
  + dropoff\_location\_id(Foreign key)
  + fare\_amount
  + tip\_amount
  + total\_amount
  + trip\_src
  + load\_date
* **Dimension Tables:**
  + **date\_dim (Date Dimension)**
    - date\_id (Primary Key)
    - date\_t (Unique )
    - year\_t
    - month\_t
    - day\_t
    - weekday\_t
  + **location\_dim (Location Dimension)**
    - location\_id (Primary Key)
    - borough
    - zone
    - service\_zone

Model:



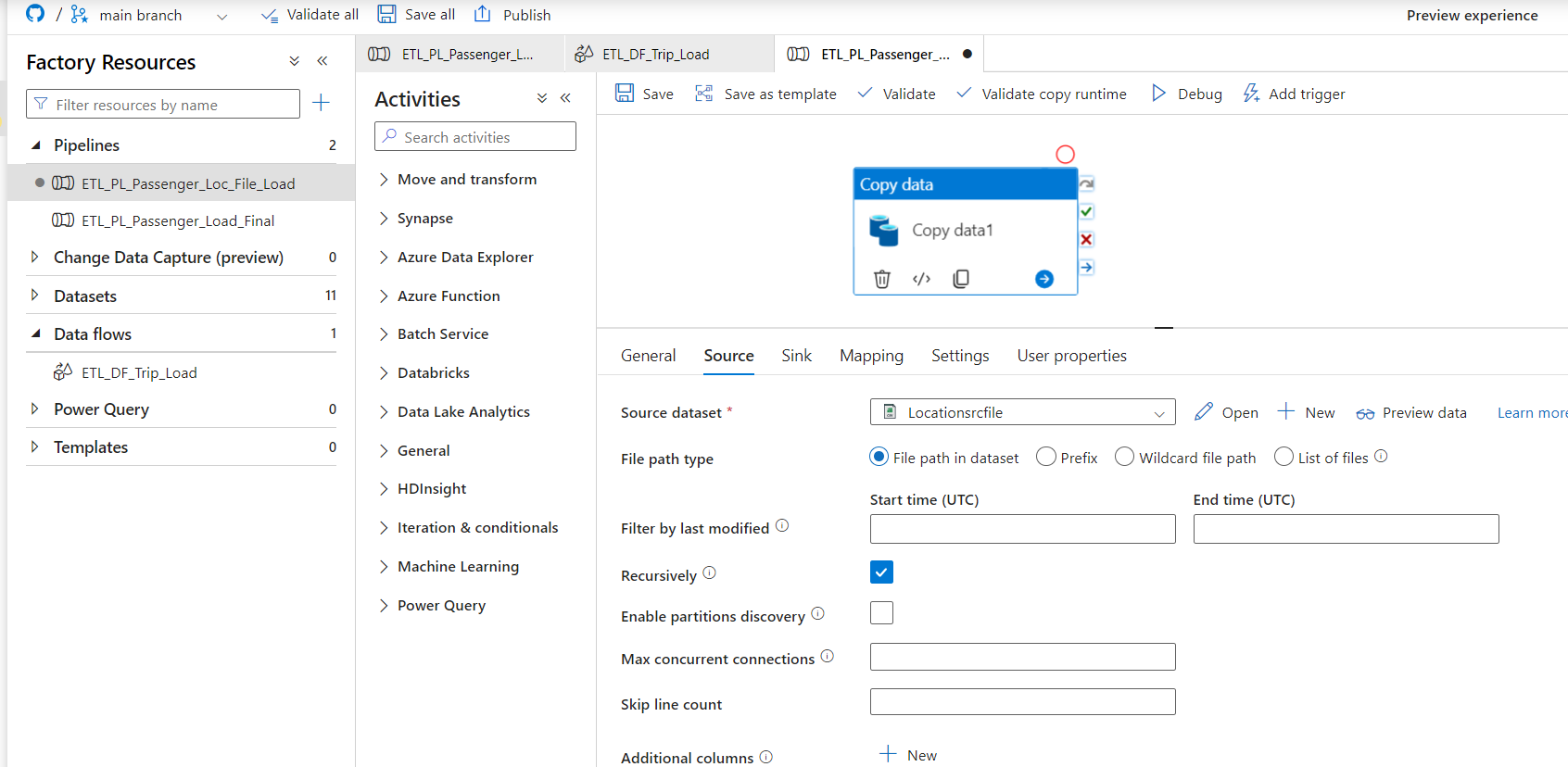
**3. Data Transformation and Loading (ETL)**

* **Write scripts to transform the raw data into the designed schema.**
* **Load the transformed data into the fact and dimension tables.**
* **Ensure that all foreign key relationships are maintained.**

**Documentation:**

1. **ETL Pipelines:**

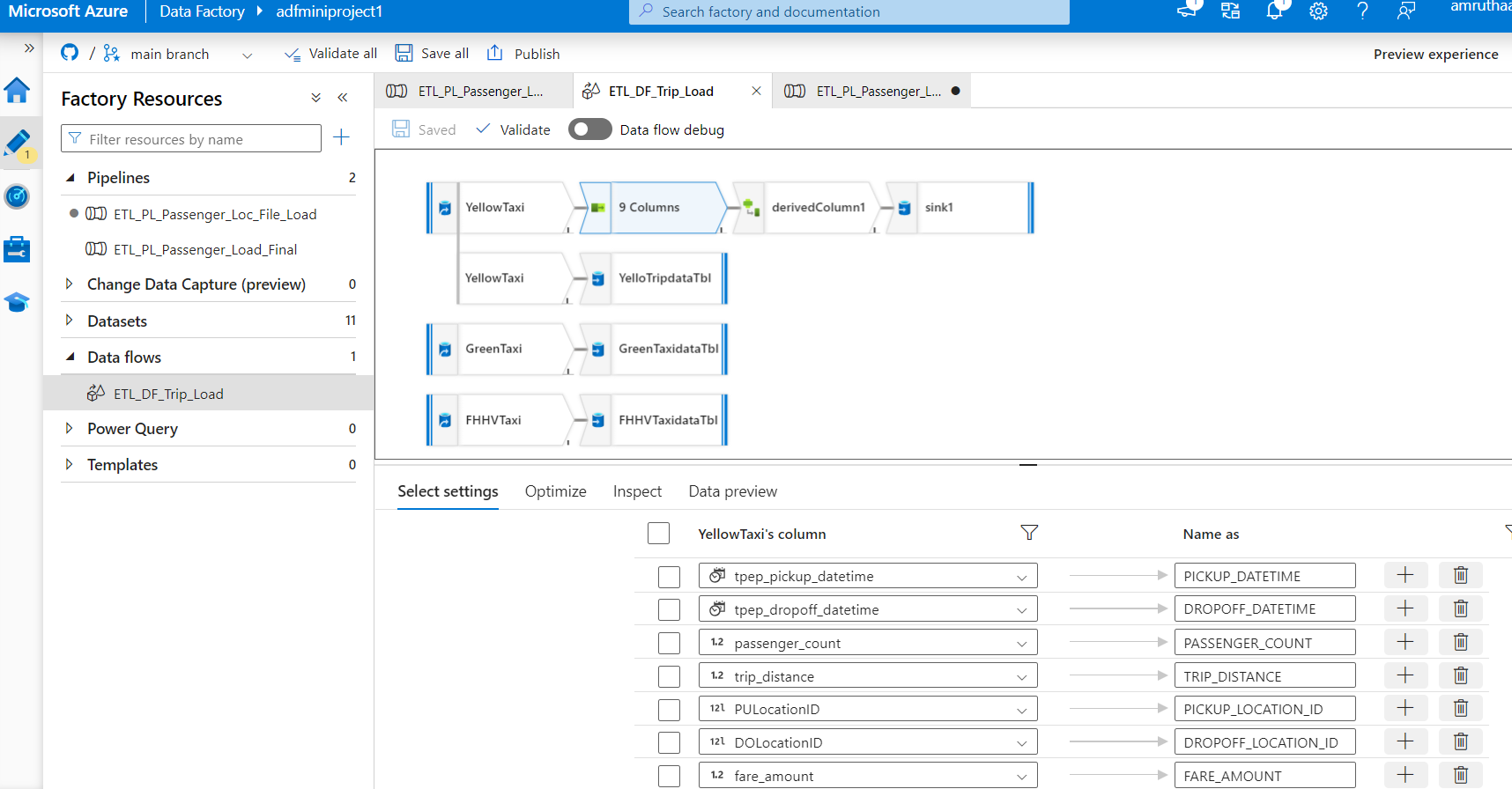
* ETL\_PL\_Passenger\_Loc\_File\_Load

****

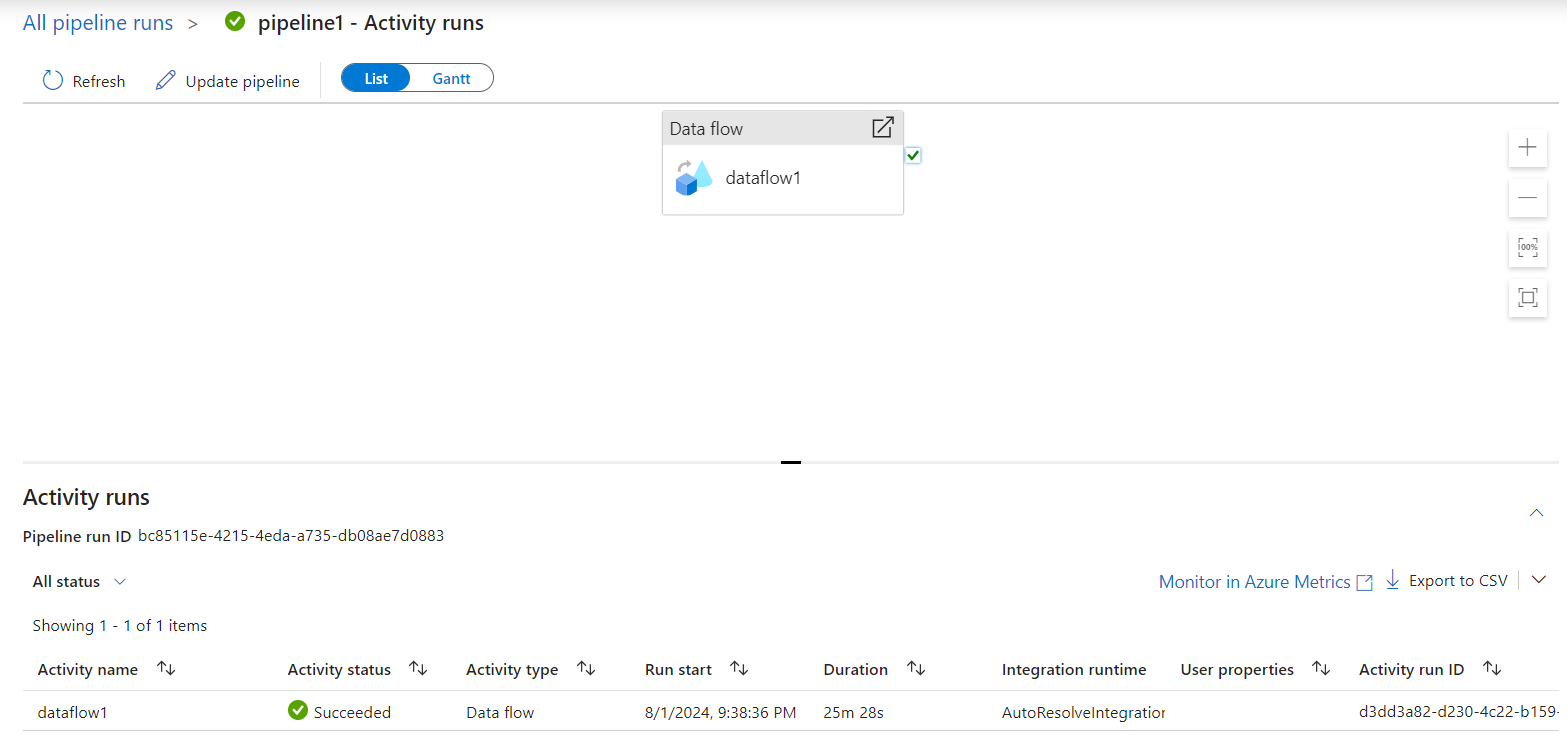
* Date\_Dim table load (using manually generated calendar data and insert SQL script )

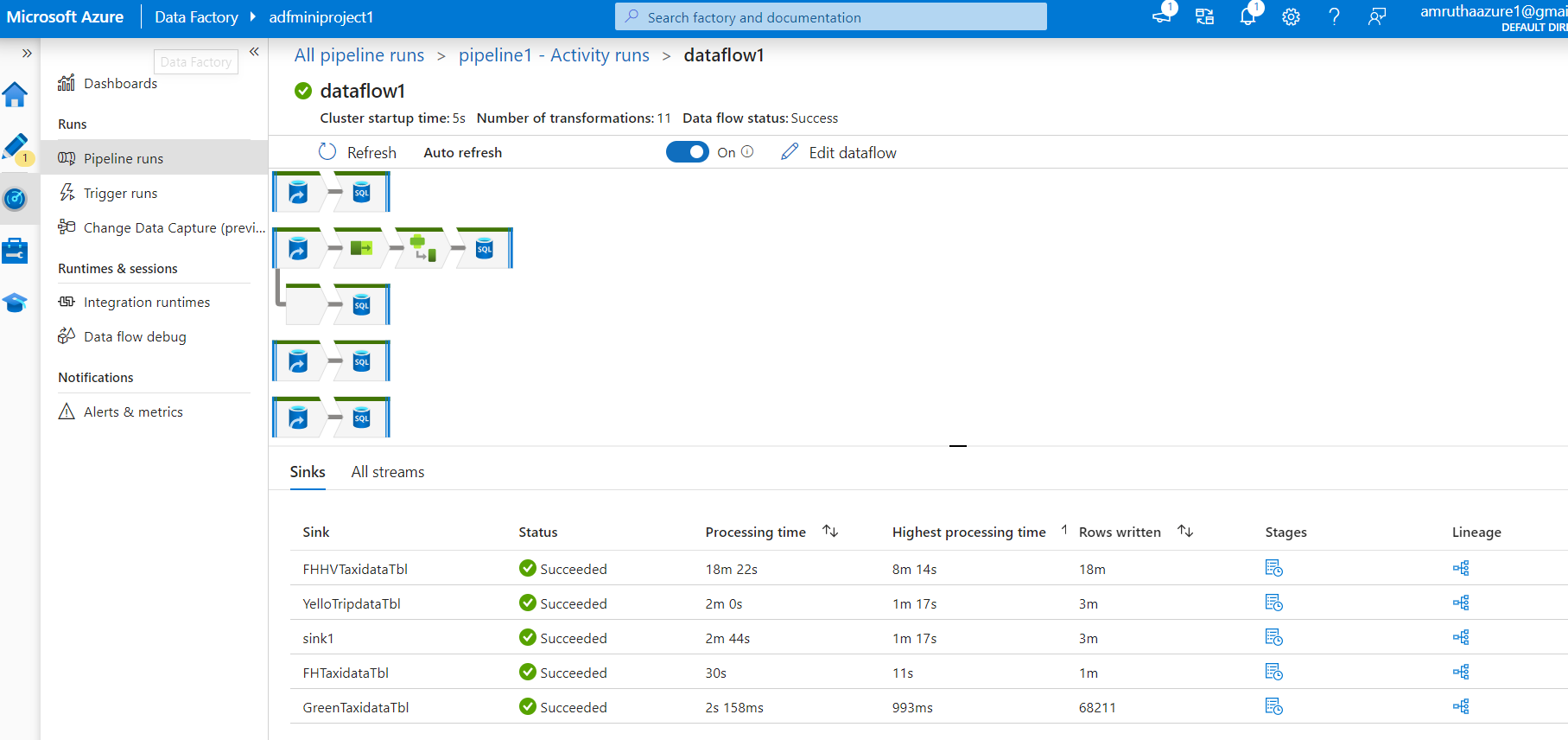
****

* ETL\_DF\_Trip\_Load



Execution Status and Record count:

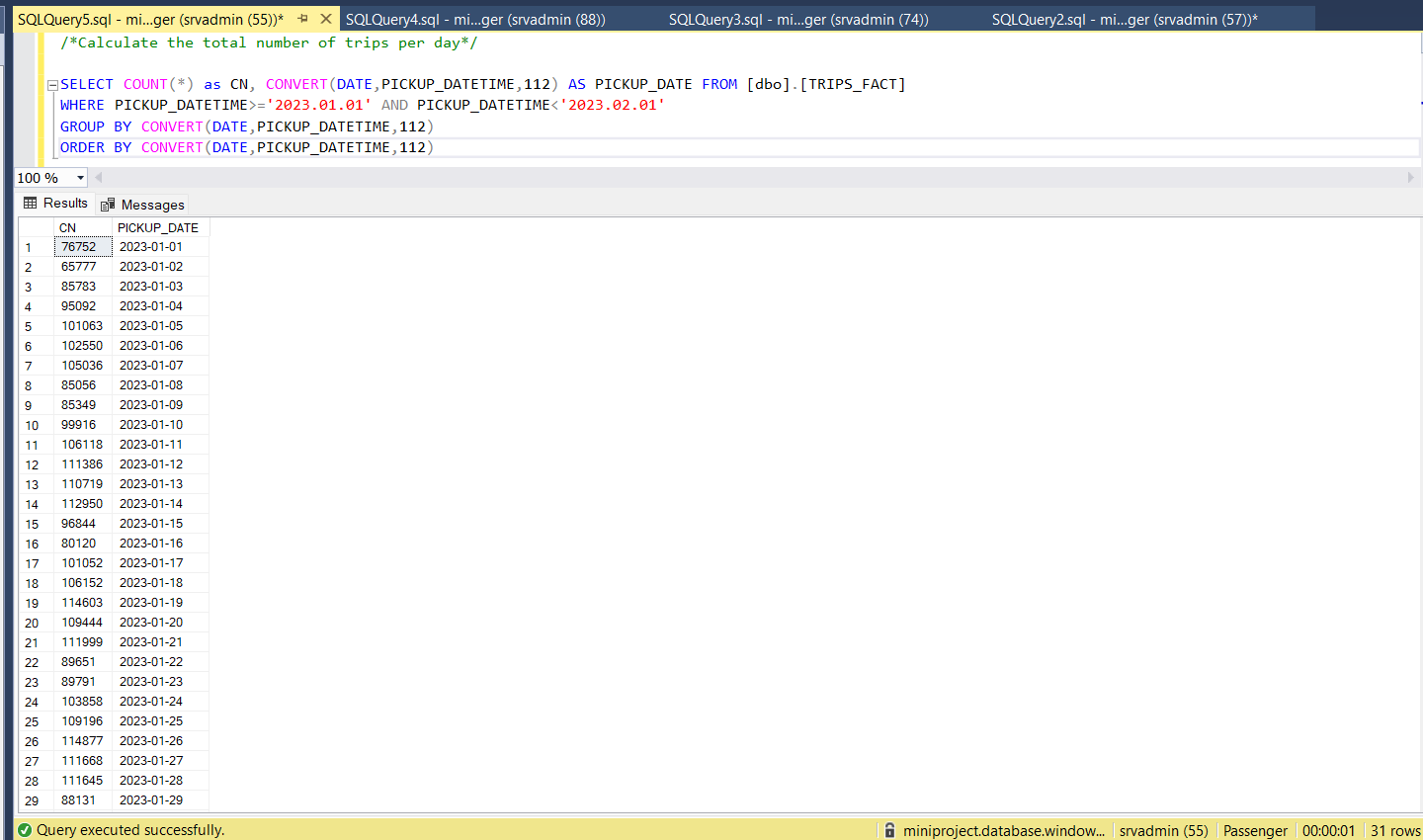




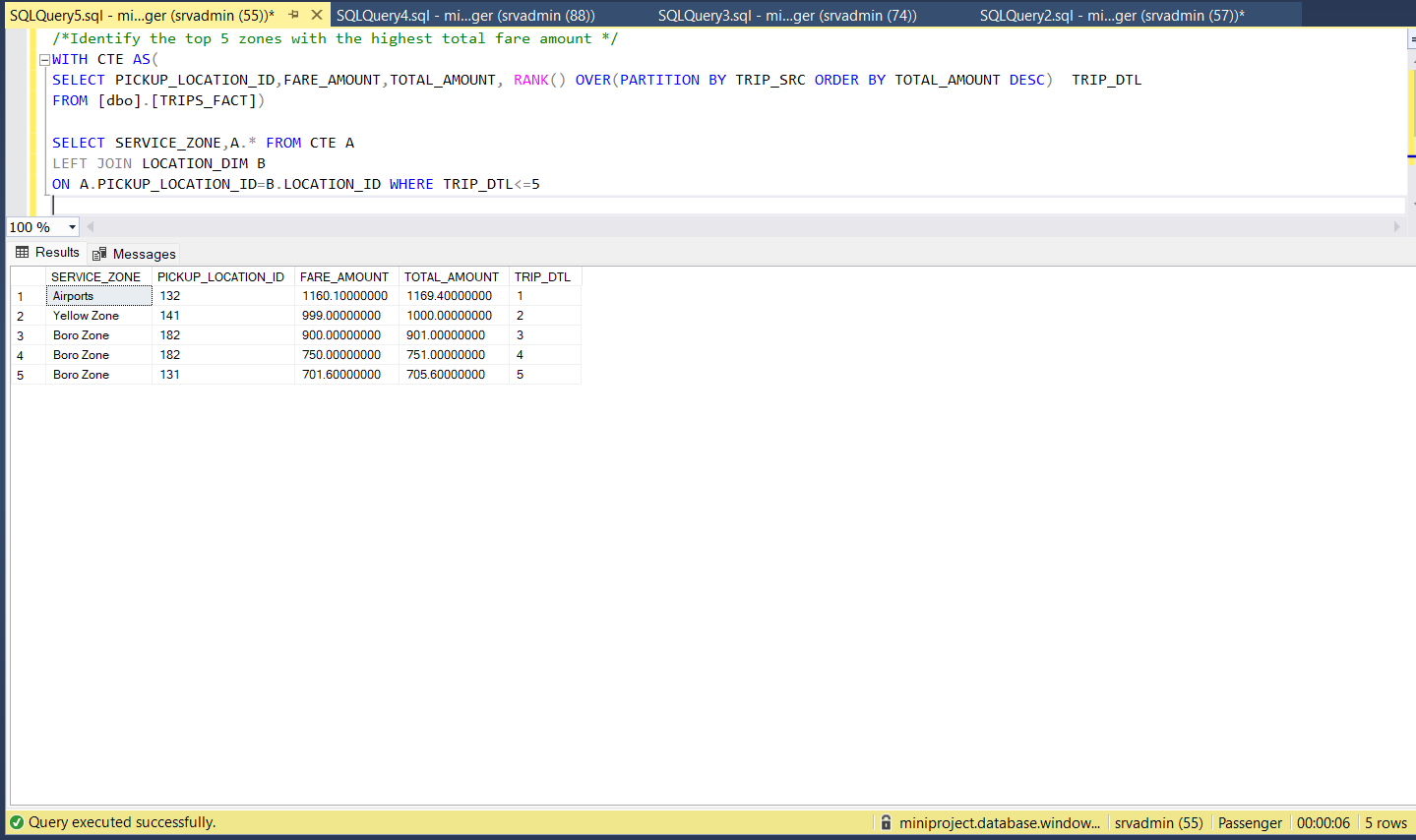
**4. Data Analysis**

Perform the following analyses using SQL queries:

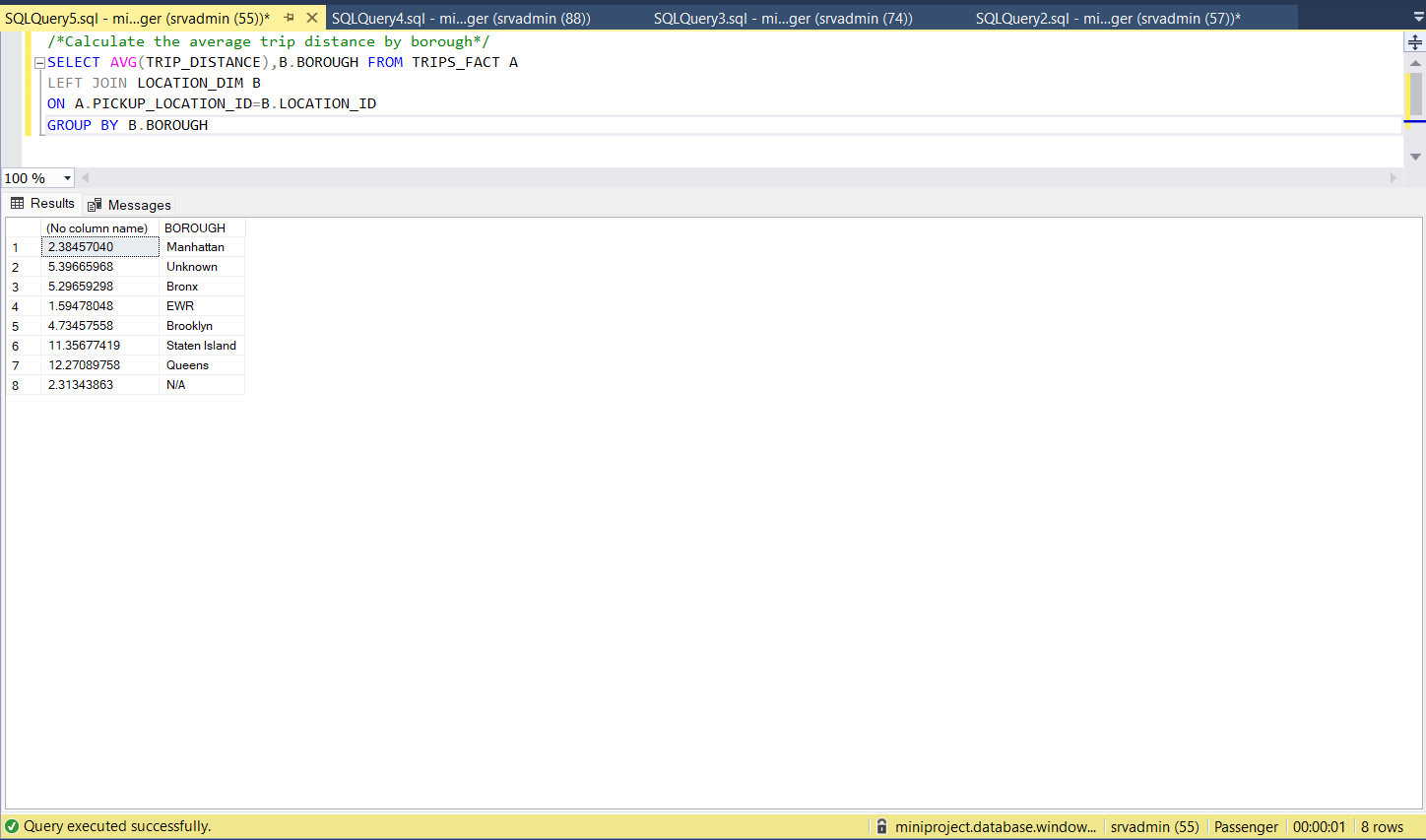
* **Calculate the total number of trips per day.**



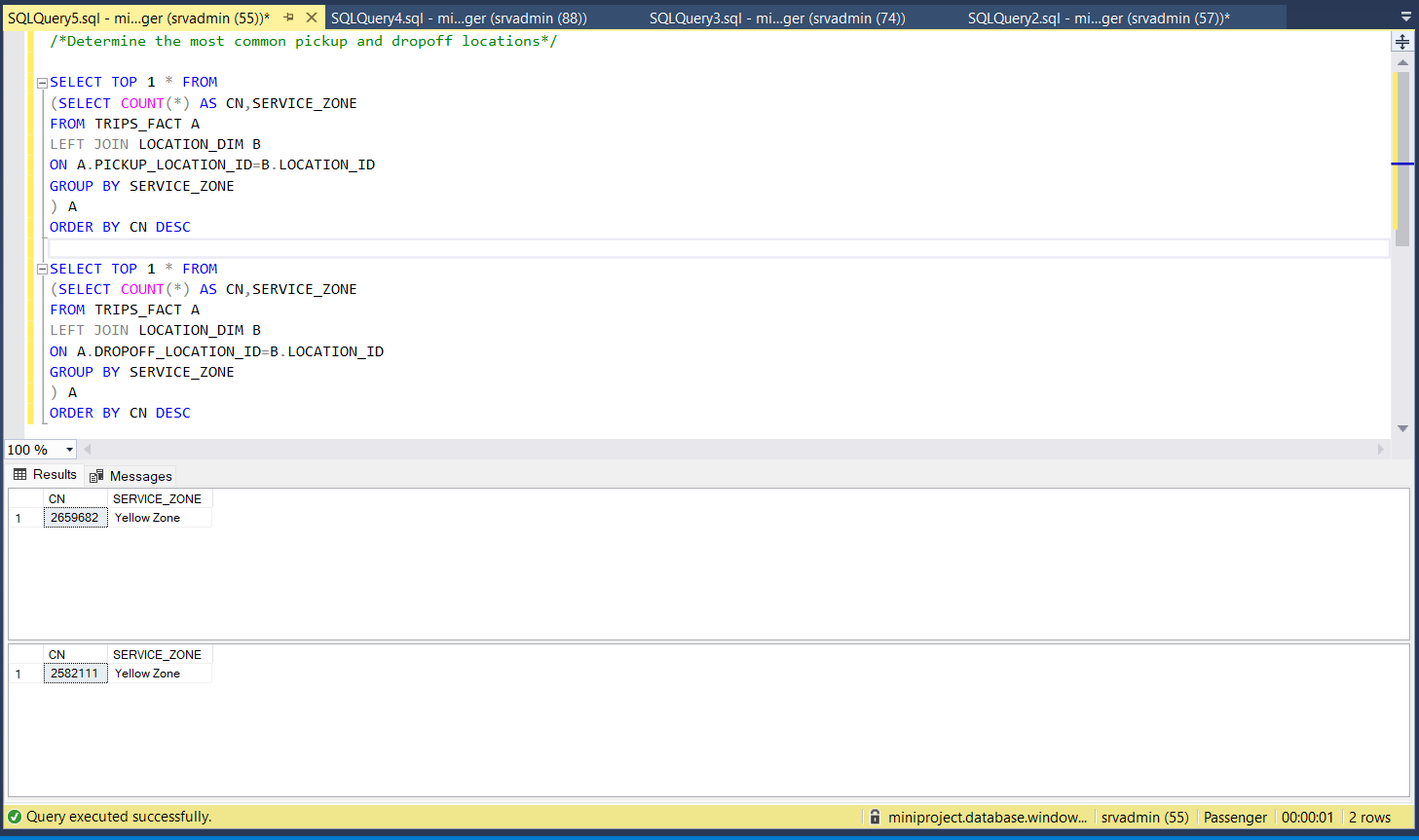
* **Identify the top 5 zones with the highest total fare amount.**



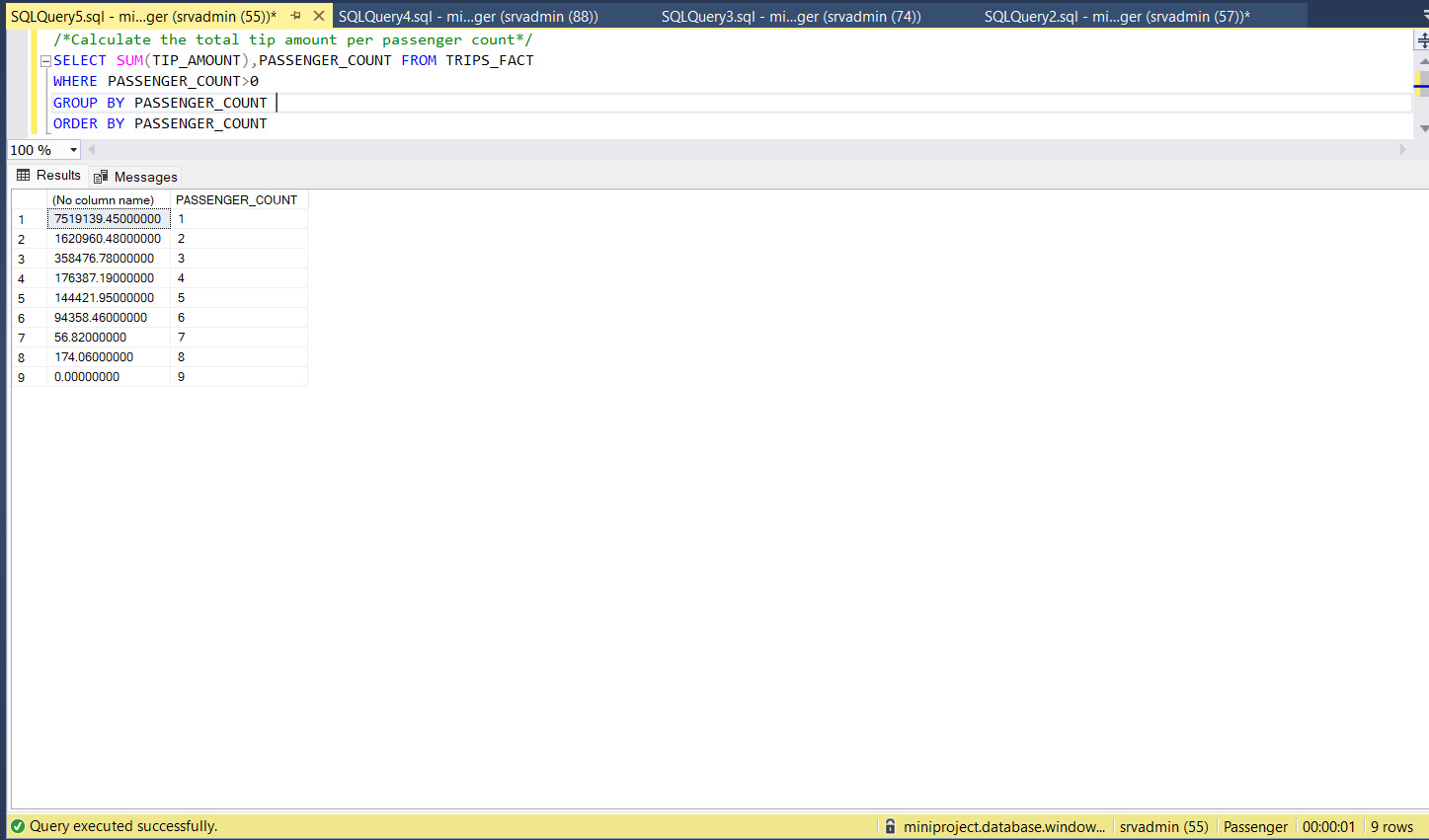
* **Calculate the average trip distance by borough.**



* **Determine the most common pickup and dropoff locations.**



* **Calculate the total tip amount per passenger count.**



SQL Queries:

****

**Submission**

Provide a GitHub repository with the following:

* **Scripts for data extraction, transformation, and loading.**
* **SQL queries used for data analysis.**
* **Documentation of your process, including any assumptions made and challenges encountered.**
* **(Optional) Dashboard files and screenshots.**

GitHub URL:

<https://github.com/amruthabhadran/MiniProject.git>

Enhancements:

* 1. Transformations:
* Remove the records where passenger count is 0 or null to load into model table
* Remove the archival or future date records to load into model table
* Assign default values to NULL data for Location IDs and Passenger counts.
  1. Data Load:
* Currently only YellowTaxiData is loaded to final table. Replicate the code to add remaining source data as well
  1. Monitoring and controls:
* Add error logs to the data load
* Add validation scripts before data load
* Add notifications on failure to send email to pre-defined recepients
* Add Trigger to schedule the execution
  1. Feasibility POC:
* Read the source files directly from URL based on month