# Uber Data Analytics Project Report

## Reason for doing this project:

The goal of this project was to gain hands-on experience with modern data engineering tools and cloud platforms by building a complete data pipeline using a real-world dataset. As a graduate student transitioning into the data industry, I wanted to understand how raw data can be ingested, transformed, stored, and visualized at scale using current industry practices. This project not only allowed me to apply theoretical concepts like dimensional modeling but also strengthened my practical skills in tools like Mage, Google Cloud, and BigQuery.

## Dataset:

The dataset simulates New York City Yellow/Green Taxi trip records, with fields including:

* Pickup/Dropoff timestamps and locations
* Trip distances and passenger counts
* Rate codes and payment types
* Fare-related financial metrics

A sample file was stored in Google Cloud Storage for pipeline development and testing.

Link for the dataset:

* Original Data Source - <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
* Data Dictionary - <https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf>

## Tools and Technologies Used:

|  |  |
| --- | --- |
| Component | Tool Used |
| Data Storage | Google Cloud Storage |
| Compute Engine | Google VM (Compute Engine) |
| Orchestration | Mage (open-source) |
| Programming | Python (Pandas) |
| Data Warehouse | Google BigQuery |
| Visualization | Looker Studio |

## Data Modelling:

The raw dataset was structured into a dimensional schema:

1. Fact Table: Trip-level transactional data (e.g., fare, tip, vendor ID)
2. Dimension Tables: Date/time, rate codes, passenger count, payment type, pickup/dropoff locations, and distance.

This model enhances querying performance and supports scalable dashboard creation.

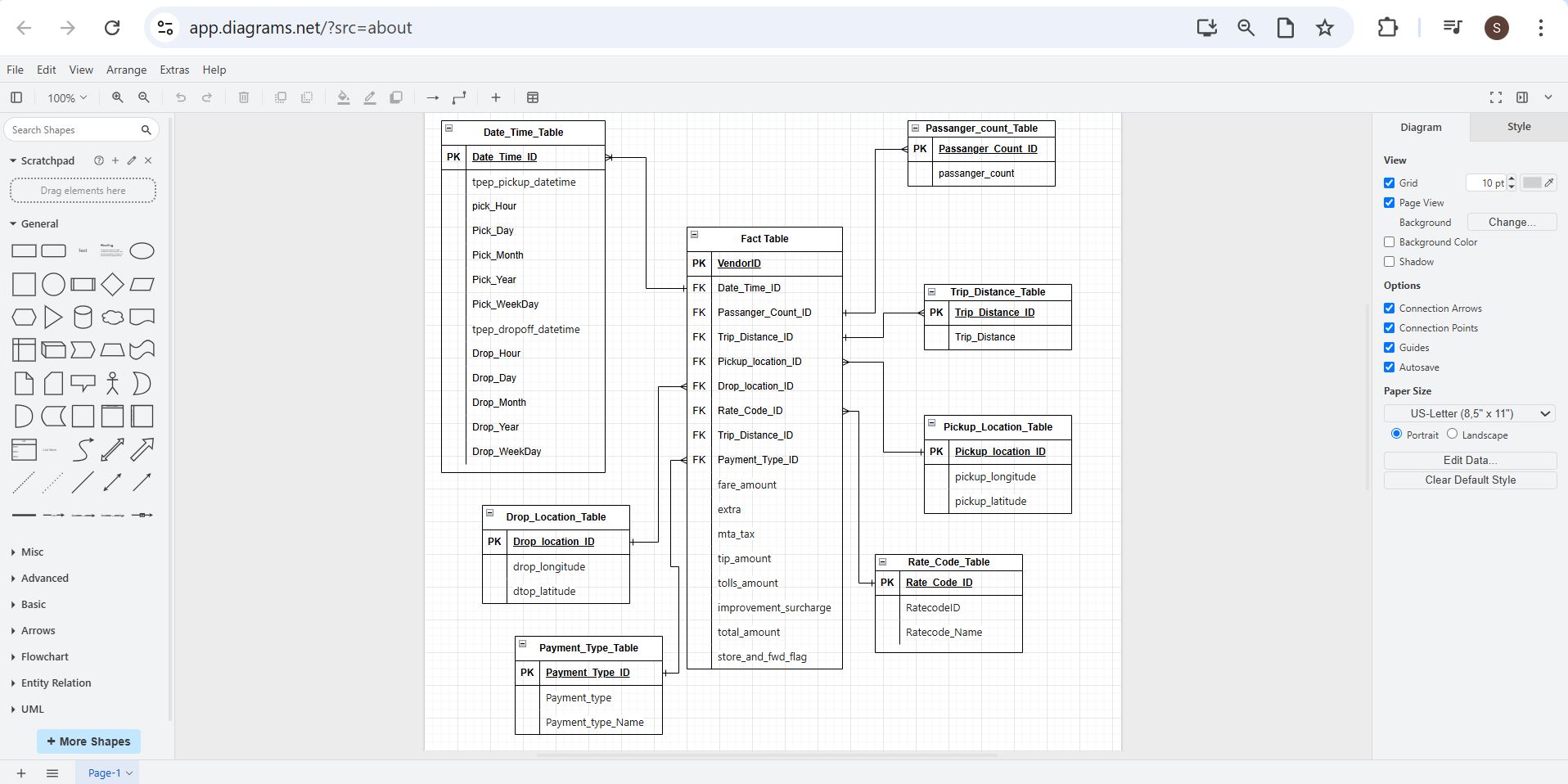


Figure 1: Star Schema Screenshot

## Data Transformation:

Key transformation tasks included:

1. Type conversion (e.g., object → datetime)
2. Date breakdown into hour, day, month, and weekday
3. Mapping ID codes to descriptive labels (e.g., payment type names)
4. Removing duplicates and creating surrogate keys
5. Structuring data for BigQuery ingestion

All transformations were performed inside Mage using modular Python blocks. Used Pandas before for transformation of the sample data.

## Cloud Deployment Steps:

* 1. Uploaded raw CSV to Google Cloud Storage (publicly accessible)
  2. Deployed Mage on a Google VM (Compute Engine)
  3. Built ETL pipeline using Mage’s loader, transformer, and exporter blocks.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Mage Loader code block

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: Mage Loader Output

A screen shot of a computer

AI-generated content may be incorrect.

Figure 4: Mage Transformer Code Block

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: Mage Transformer Output Block

A computer screen shot of a computer code

AI-generated content may be incorrect.

Figure 6: Mage Exporter Code Block

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7: Mage Exporter Output Block

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8: Uber Data ETL Pipeline

* 1. Loaded final tables into BigQuery

A screenshot of a computer

AI-generated content may be incorrect.

Figure 9 : Screenshots of Table loaded in BigQuery

* 1. Built dashboard on Looker Studio connected to BigQuery.

A screenshot of a computer

AI-generated content may be incorrect.Dashboard link : https://lookerstudio.google.com/s/ry5lrPQj4No

Figure 10 : Looker Dashboard Screenshot

## Errors Faced:

* + 1. Protobuf Version Mismatch:

A screenshot of a computer

AI-generated content may be incorrect.

**Error Message**: Detected mismatched Protobuf Gencode/Runtime major versions  
  
**Fix:**  
Align Protobuf versions:  
pip install --upgrade protobuf==6.30.2

* + 1. Missing 'db-dtypes' Package

A screenshot of a computer program

AI-generated content may be incorrect.

Error Message: ValueError: Please install the 'db-dtypes' package to use this function.  
  
Fix:  
Install the required package:  
pip install db-dtypes