Uber data Engineering project

@DHRUV KUMAR
FOLLOW ME ON

Project Summary

In this project, I successfully built a data engineering pipeline using **Pandas**, **Google Cloud**, **Mage**, **BigQuery** and **LookerStudio**. The key steps included:

- **Data Cleaning & Transformation:** Removed duplicates and created structured dimensions such as date, time, day, month, pick hours, etc.. using Pandas.

- **Data Modeling:** Designed a data modeling structure for dimensional analysis.

- **Cloud Integration:** Set up an instance on Google Cloud and used Mage to build an automated data pipeline.

- **Data Export & Processing:** Exported the transformed data to BigQuery using the configure credentials .

- **Data Merging & Visualization:** Merged multiple tables and imported the final dataset into **Looker Studio** for visualization and insights.

This project streamlined data processing, storage, and visualization, making it more efficient and scalable.

Transform Data using pandas

Uber data cleaning

```
import pandas as pd #Import pandas df = pd.read_csv('uber_data.csv') #read
csv file via pandas df.head() #fetch the first five rows df.info() #information
all the dataset
```

pickup and dropoff date are not correct data type so we can use the pandas datatime fuction.

```
df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'])
df['tpep_dropoff_datetime'] = pd.to_datetime(df['tpep_dropoff_datetime'])
```

```
df.info()

datetime_dim =
df[['tpep_pickup_datetime','tpep_dropoff_datetime']].drop_duplicates()
.reset_index(drop=True)
datetime_dim['pick_hour'] =
datetime_dim['tpep_pickup_datetime'].dt.hour
datetime_dim['pick_day'] = datetime_dim['tpep_pickup_datetime'].dt.day
datetime_dim['pick_month'] =
datetime_dim['tpep_pickup_datetime'].dt.month
datetime_dim['pick_year'] =
datetime_dim['tpep_pickup_datetime'].dt.year
datetime_dim['pick_weekday'] =
datetime_dim['tpep_pickup_datetime'].dt.weekday
datetime_dim['drop_hour'] =
datetime_dim['tpep_dropoff_datetime'].dt.hour
datetime_dim['drop_day'] =
datetime_dim['tpep_dropoff_datetime'].dt.day
datetime_dim['drop_month'] =
datetime_dim['tpep_dropoff_datetime'].dt.month
datetime_dim['drop_year'] =
datetime_dim['tpep_dropoff_datetime'].dt.year
datetime_dim['drop_weekday'] =
datetime_dim['tpep_dropoff_datetime'].dt.weekday
datetime_dim['datetime_id'] = datetime_dim.index

datetime_dim[['datetime_id','tpep_pickup_datetime','pick_hour','pick_d
ay','pick_month','pick_year','pick_weekday','tpep_dropoff_datetime',
```

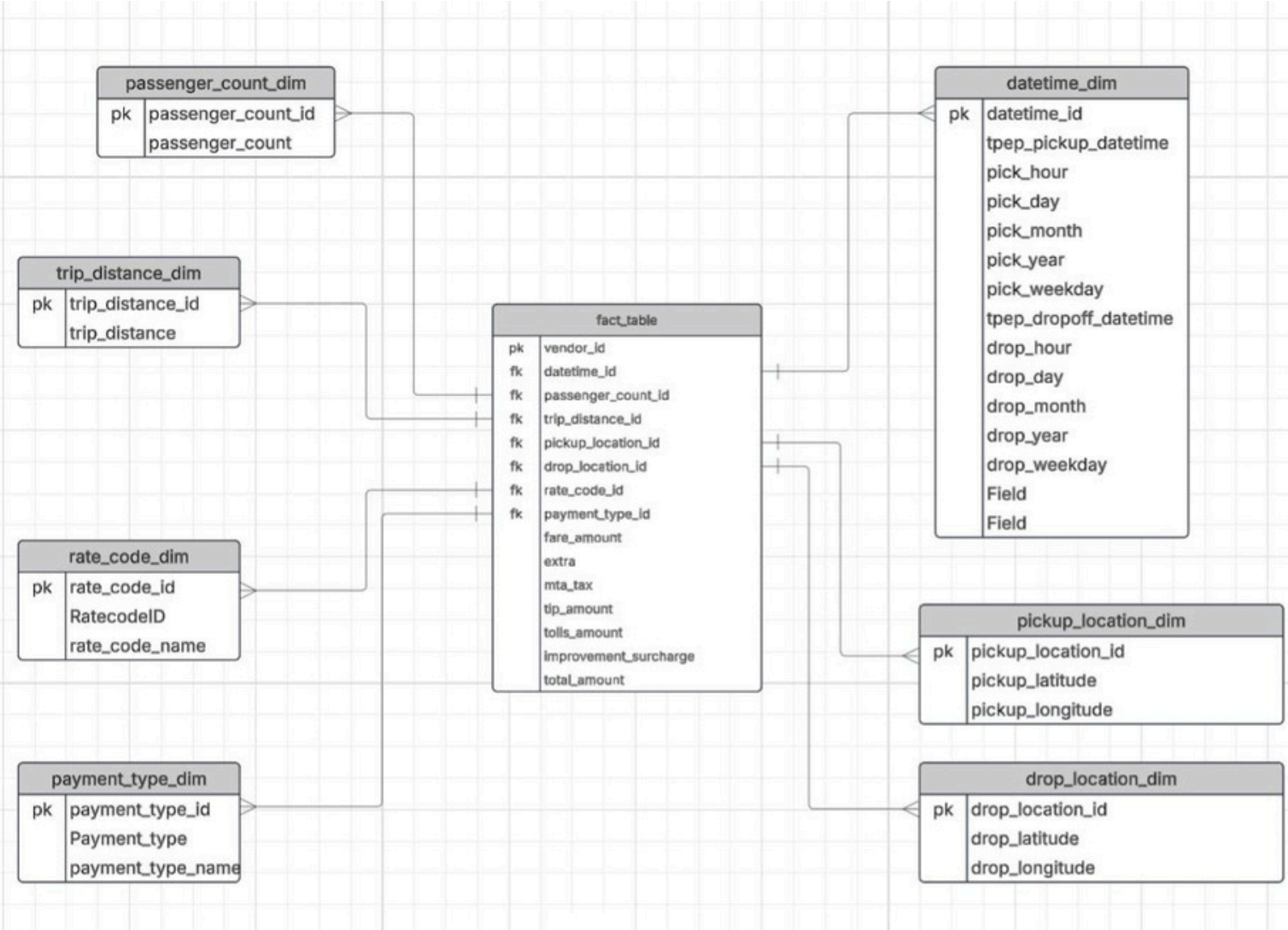
```
'drop_hour','drop_day','drop_month','drop_year','drop_weekday']]

passenger_count_dim =
df[['passenger_count']].drop_duplicates().reset_index(drop=True)
passenger_count_dim['passenger_count_id'] = passenger_count_dim.index
passenger_count_dim =
passenger_count_dim[['passenger_count_id','passenger_count']]

trip_distance_dim =
df[['trip_distance']].drop_duplicates().reset_index(drop=True)
trip_distance_dim['trip_distance_id'] = trip_distance_dim.index
trip_distance_dim =
trip_distance_dim[['trip_distance_id','trip_distance']]

passenger_count_dim

rate_code_type = {
    1: 'Standard rate',
    2: 'JFK',
    3: 'Newark',
    4: 'Nassau or Westchester',
    5: 'Negotiated fare',
    6: 'Group ride'
}
rate_code_dim =
df[['RatecodeID']].drop_duplicates().reset_index(drop=True)
rate_code_dim['rate_code_id'] = rate_code_dim.index
rate_code_dim['rate_code_name'] =
rate_code_dim['RatecodeID'].map(rate_code_type)
rate_code_dim =
rate_code_dim[['rate_code_id','RatecodeID','rate_code_name']]

rate_code_dim

pickup_location_dim =
df[['pickup_longitude','pickup_latitude']].drop_duplicates().reset_ind
ex(drop=True)
pickup_location_dim['Pickup_location_id'] = pickup_location_dim.index
pickup_location_dim =
pickup_location_dim[['Pickup_location_id','pickup_latitude','pickup_lo
ngitude']]

dropoff_location_dim =
df[['dropoff_longitude','dropoff_latitude']].drop_duplicates().reset_i
ndex(drop=True)
dropoff_location_dim['drop_location_id'] = dropoff_location_dim.index
dropoff_location_dim =
dropoff_location_dim[['drop_location_id','dropoff_longitude','dropoff_
latitude']]
```

```
payment_type_name = {
    1: "Credit card", 2:
    "cash",    3:    "No
    charge",          4:
    "Dispute",        5:
    "Unknown",        6:
    "Voided trip"
}
```

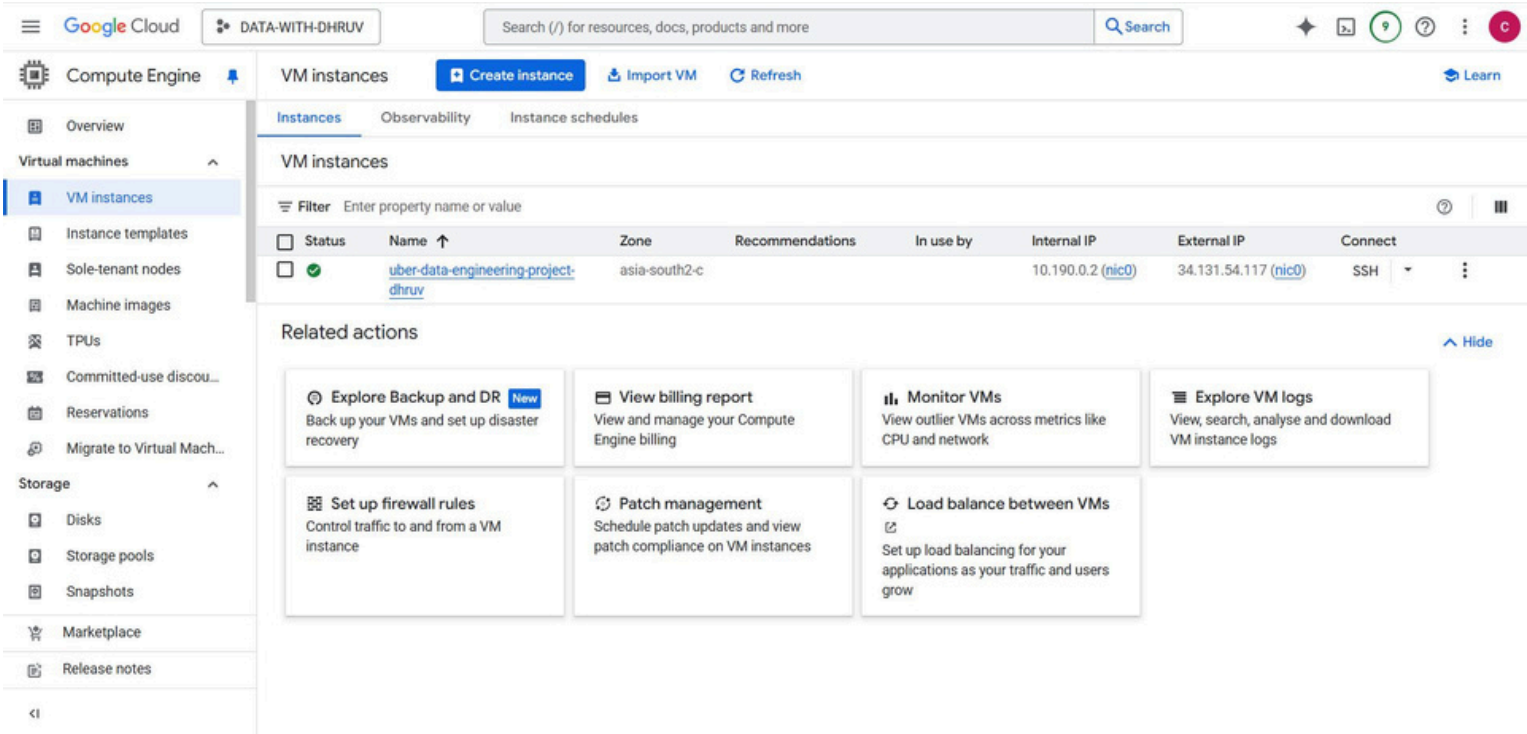
```
payment_type_dim =
df[['payment_type']].drop_duplicates().reset_index(drop=True)
payment_type_dim['payment_type_id'] = payment_type_dim.index
payment_type_dim['payment_type_name'] =
payment_type_dim['payment_type'].map(payment_type_name)
payment_type_dim =
payment_type_dim[['payment_type_id','payment_type','payment_type_name'
]]
payment_type_dim
```

```
fact_table = df.merge(passenger_count_dim, on='passenger_count')\

                    .merge(trip_distance_dim, on='trip_distance')\
                    .merge(rate_code_dim, on='RatecodeID')\
                    .merge(pickup_location_dim,
on=['pickup_longitude','pickup_latitude'])\
                      .merge(dropoff_location_dim,
      on=['dropoff_longitude','dropoff_latitude'])\
                    .merge(datetime_dim,
  on=['tpep_pickup_datetime','tpep_dropoff_datetime'])\
                    .merge(payment_type_dim, on='payment_type')\
                    [['VendorID','datetime_id','passenger_count_id',

'trip_distance_id','rate_code_id','store_and_fwd_flag','Pickup_locatio
n_id','drop_location_id','payment_type_id',

'fare_amount','extra','mta_tax','tip_amount','tolls_amount','improveme
nt_surcharge','total_amount']]

fact_table
```

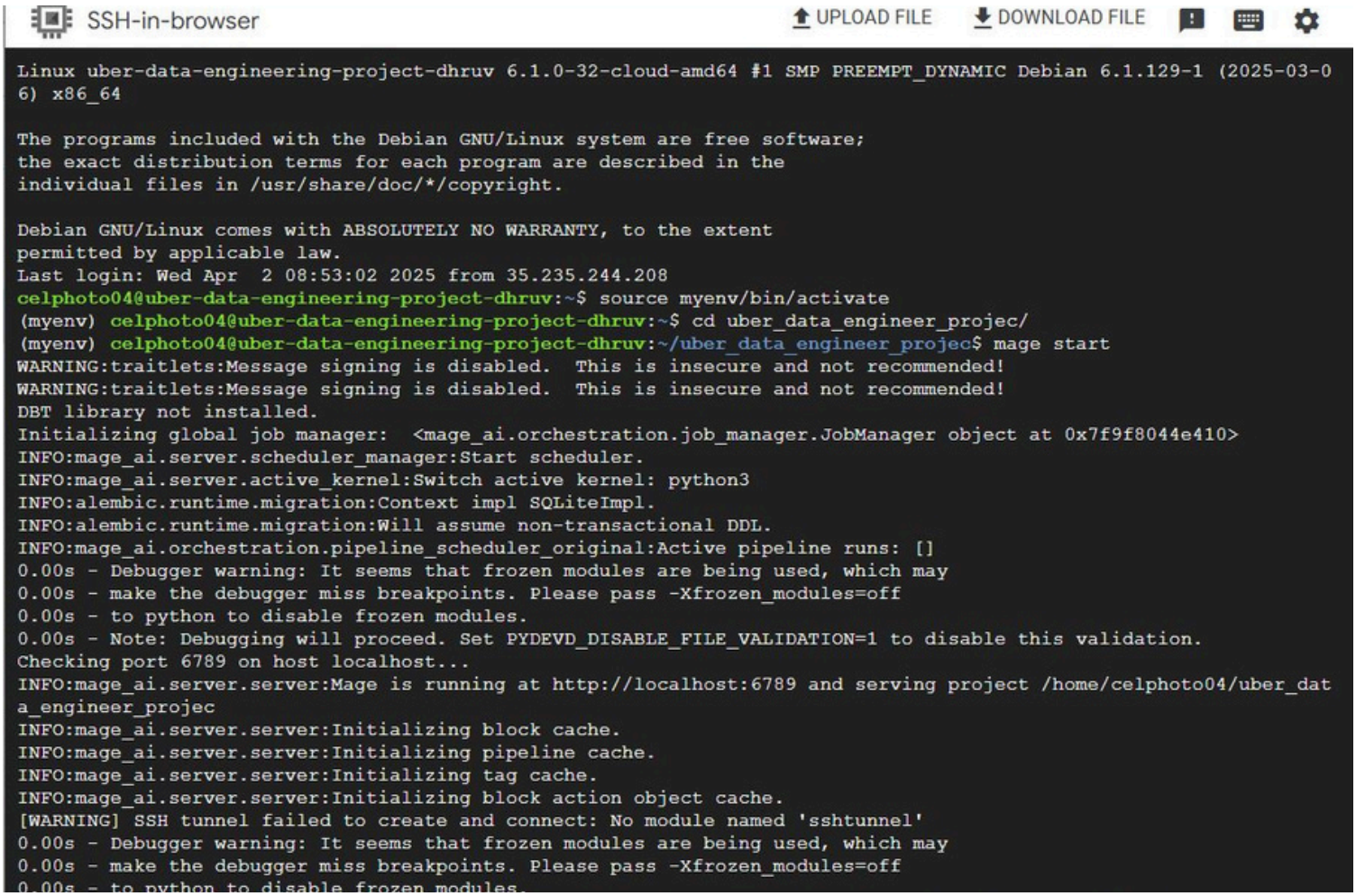
Data Modeling



- Set up an instance

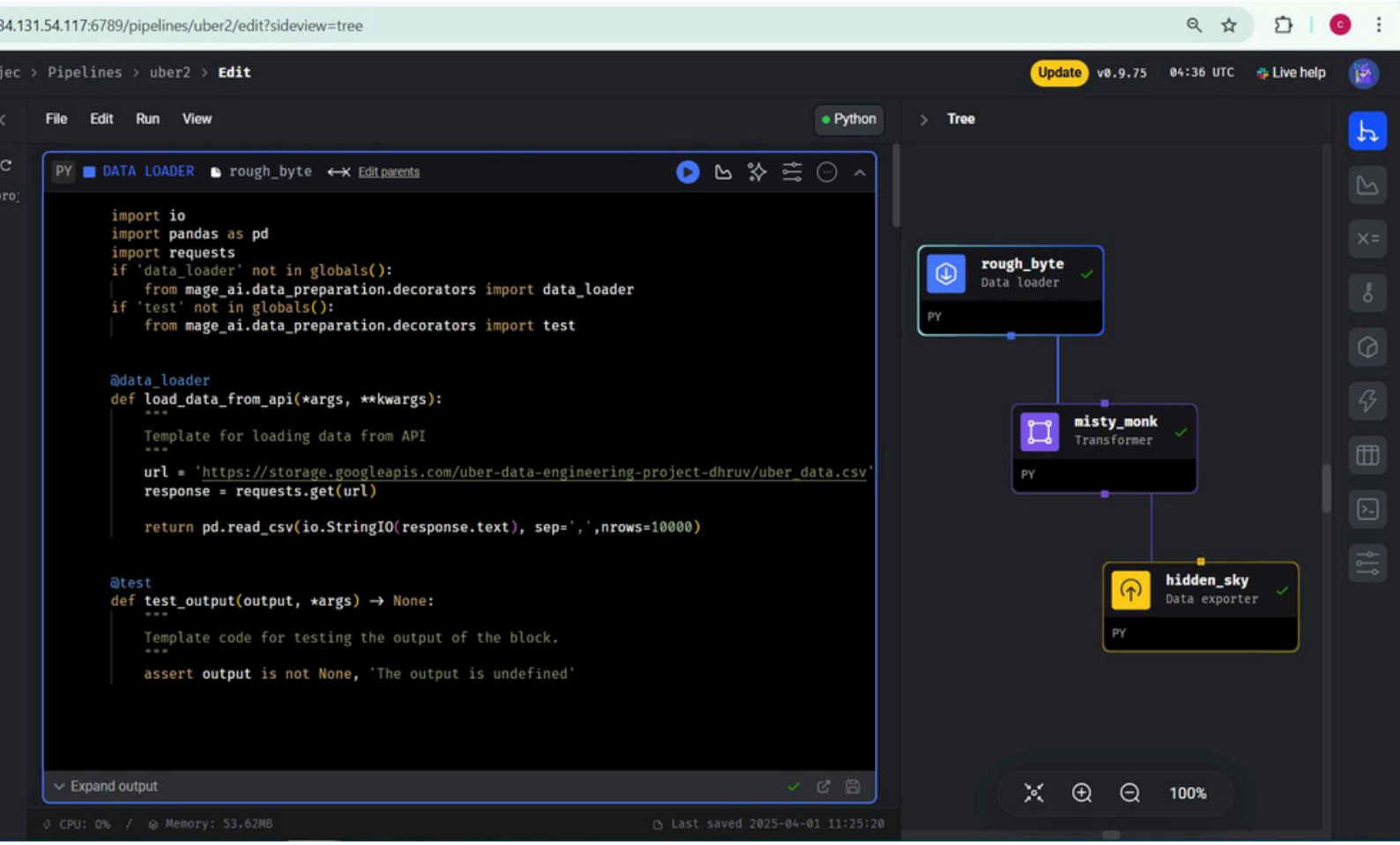


- VM Connect to Mage

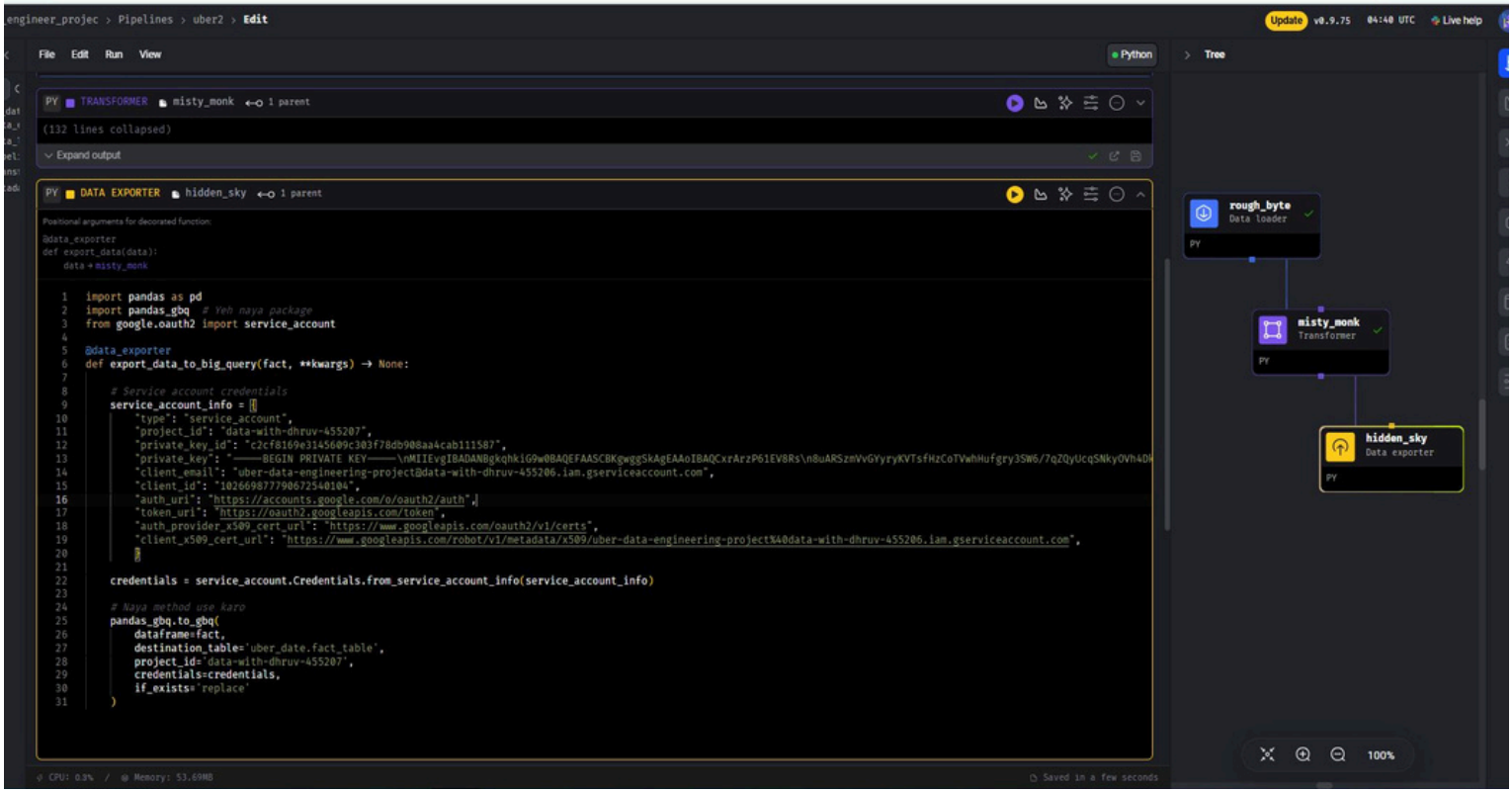


Mage to build an automated data pipeline.

- Created Data Loader File to Load Data

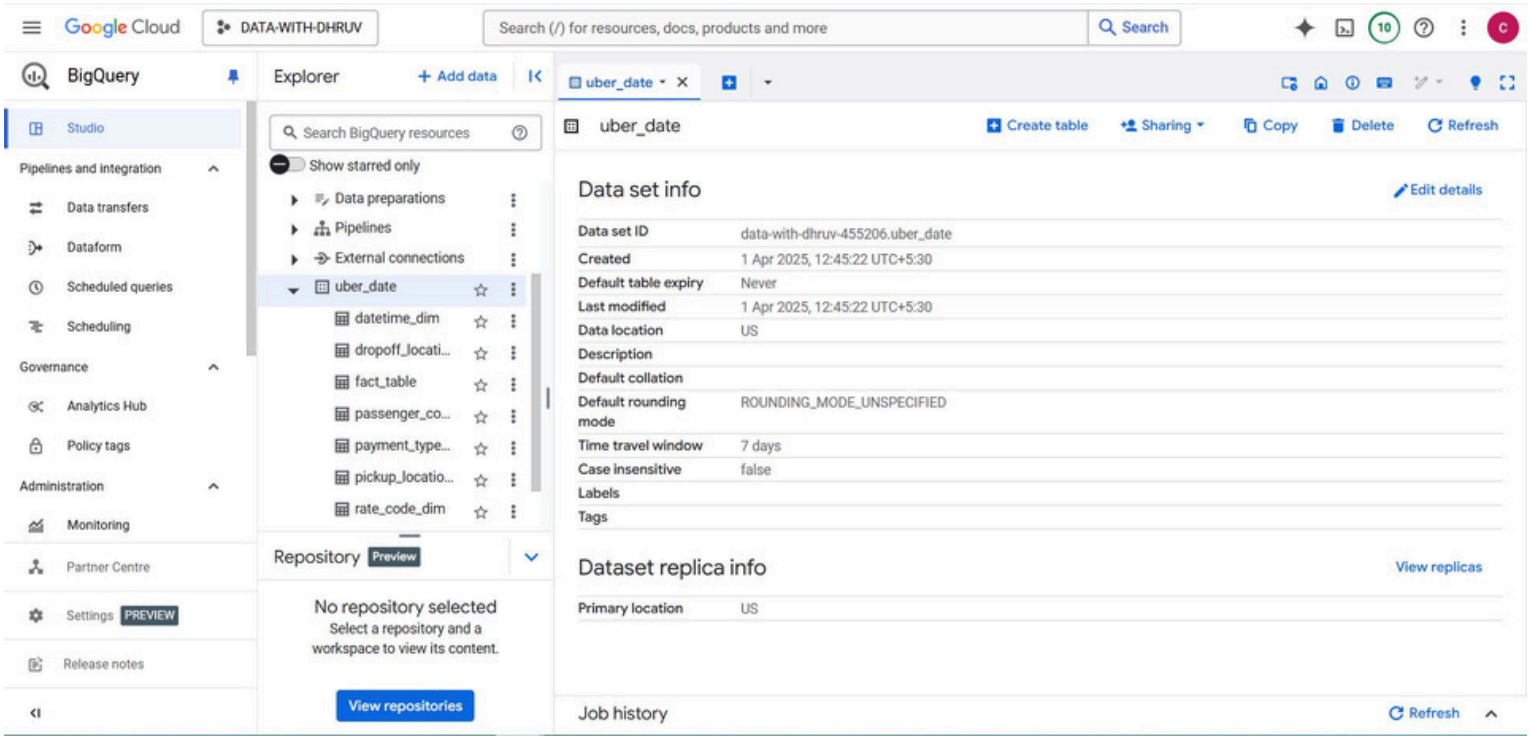


- Data Export to BigQuery using the configure credentials .



python™

- Data Tables



- Merged multiple tables

console.cloud.google.com/bigquery?invt=AbtvLQ&project=data-with-dhruv-455206&ws=!1m7!1m6!12m5!1m3!1sdata-with-dhruv-455206!2sus-central1!3scf7

uber_A... ics

uber_Analytics | Run | Open in | More | Save query | Download | Share | Schedule

```
CREATE OR REPLACE TABLE `data-with-dhruv-455206.uber_date.uber_analytics` AS (
SELECT
f.VendorID,
d.tpep_dropoff_datetime,
d.tpep_pickup_datetime,
p.passenger_count,
t.trip_distance,
r.rate_code_name,
pl.pickup_latitude,
pl.pickup_longitude,
dl.dropoff_latitude,
dl.dropoff_longitude,
py.payment_type_name,
f.fare_amount,
f.extra,
f.mta_tax,
f.tip_amount,
f.tolls_amount,
f.improvement_surcharge,
FROM
`data-with-dhruv-455206.uber_date.fact_table` f
join `data-with-dhruv-455206.uber_date.datetime_dim` d
on f.datetime_id = d.datetime_id
join `data-with-dhruv-455206.uber_date.passenger_count_dim` p
on f.passenger_count_id = p.passenger_count_id
join `data-with-dhruv-455206.uber_date.trip_distance_dim` t
on f.trip_distance_id = t.trip_distance_id
join `data-with-dhruv-455206.uber_date.rate_code_dim` r
on f.rate_code_id = r.rate_code_id
join `data-with-dhruv-455206.uber_date.pickup_location_dim` pl
on f.Pickup_location_id = pl.Pickup_location_id
join `data-with-dhruv-455206.uber_date.dropoff_location_dim` dl
on f.drop_location_id = dl.drop_location_id
join `data-with-dhruv-455206.uber_date.payment_type_dim` py
on f.payment_type_id = py.payment_type_id);
```

Uber

Vendor ID

Payment Type

Rate Co...

Trip Distance

0 45.68

Clean filter

Download repor

Summery

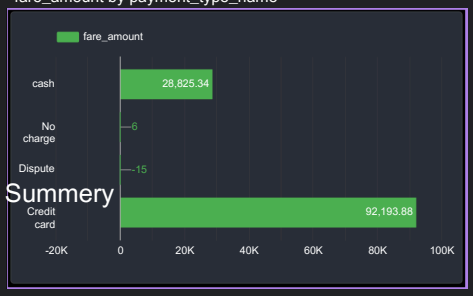
Card

Total Revenue	Total Tolls	Total Trip Amount	Avg. Fare Amount	Avg. Fare Amount	Avg. Trip Amount
150.3K	2.9K	18.3K	12.1	12.1	1.8

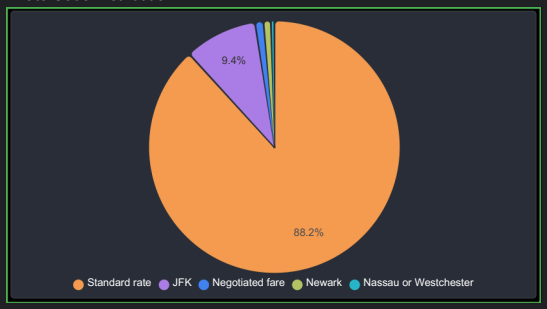
Summery

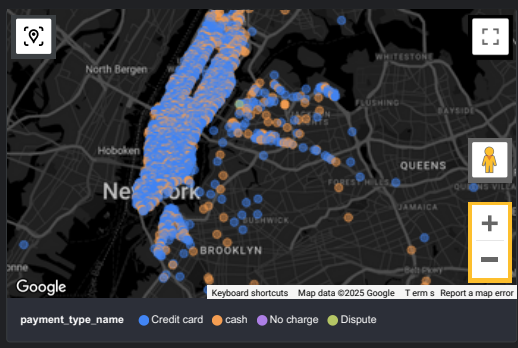
Chart

fare_amount by payment_type_name

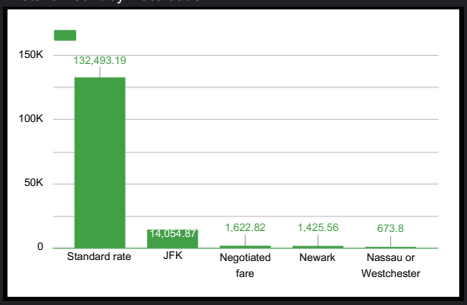


Rate Code Distribution

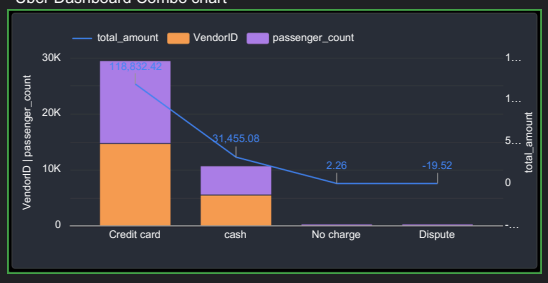


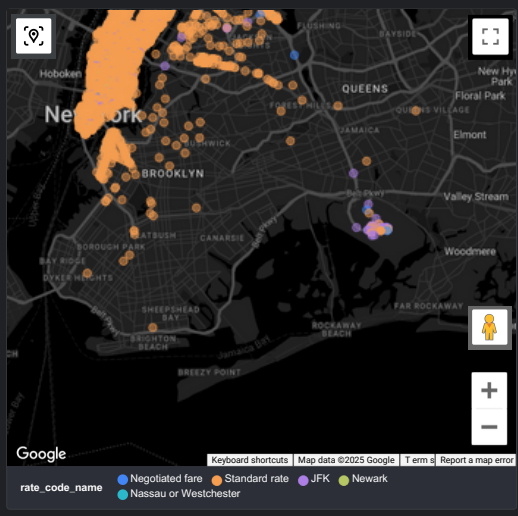


Total amount by Rate code



Uber Dashboard Combo chart





Uber Dashboard Table

	Rate Code	Fare amount	T. Amount
1.	Standard rate	106.8K	132.5K
2.	JFK	11.2K	14.1K
3.	Negotiated fare	1.3K	1.6K
4.	Newark	1K	1.4K

1 - 5 / 5 < >

Profit & Loss by Rate Code