NY Dataset Kaggle Link -- https://www.kaggle.com/datasets/nelgiriyewithana/new-york-housing-market

-- Set up Apache Superset VM on Google Cloud Compute Engine

-- create VM in google cloud using scripts

-- Create the apache superset compute engine vm.

gcloud compute instances create superset \

--image-family ubuntu-2004-lts \

--image-project ubuntu-os-cloud \

--machine-type n1-standard-1 \

--description superset \

--zone us-east1-c \

--tags http-server \

--metadata=startup-script='#!/bin/bash

curl -sSL https://storage.googleapis.com/jose-bucket-mac250/script.sh | bash'

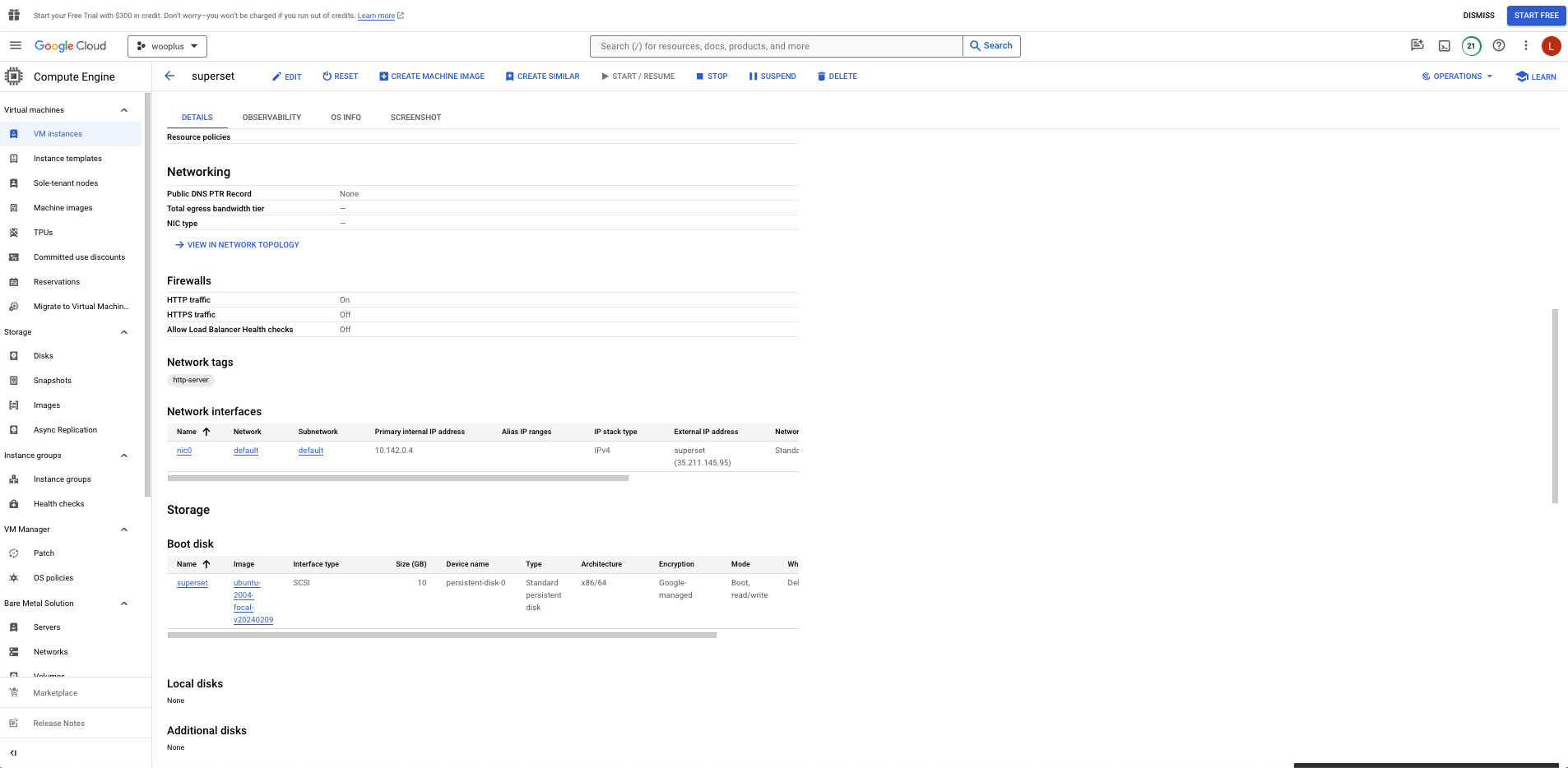


Photo of Superset Compute Engine

-- click connect SSH

sudo systemctl status superset <-- Check superset is running or not

sudo systemctl enable superset

-- Connect Apache Superset to MySQL Cloud SQL instance

IP Address = Database Instance IP Address 34.75.9.212

Database Name = final-aung

PORT = 3306

Username = root

Password = P@$$w0rd

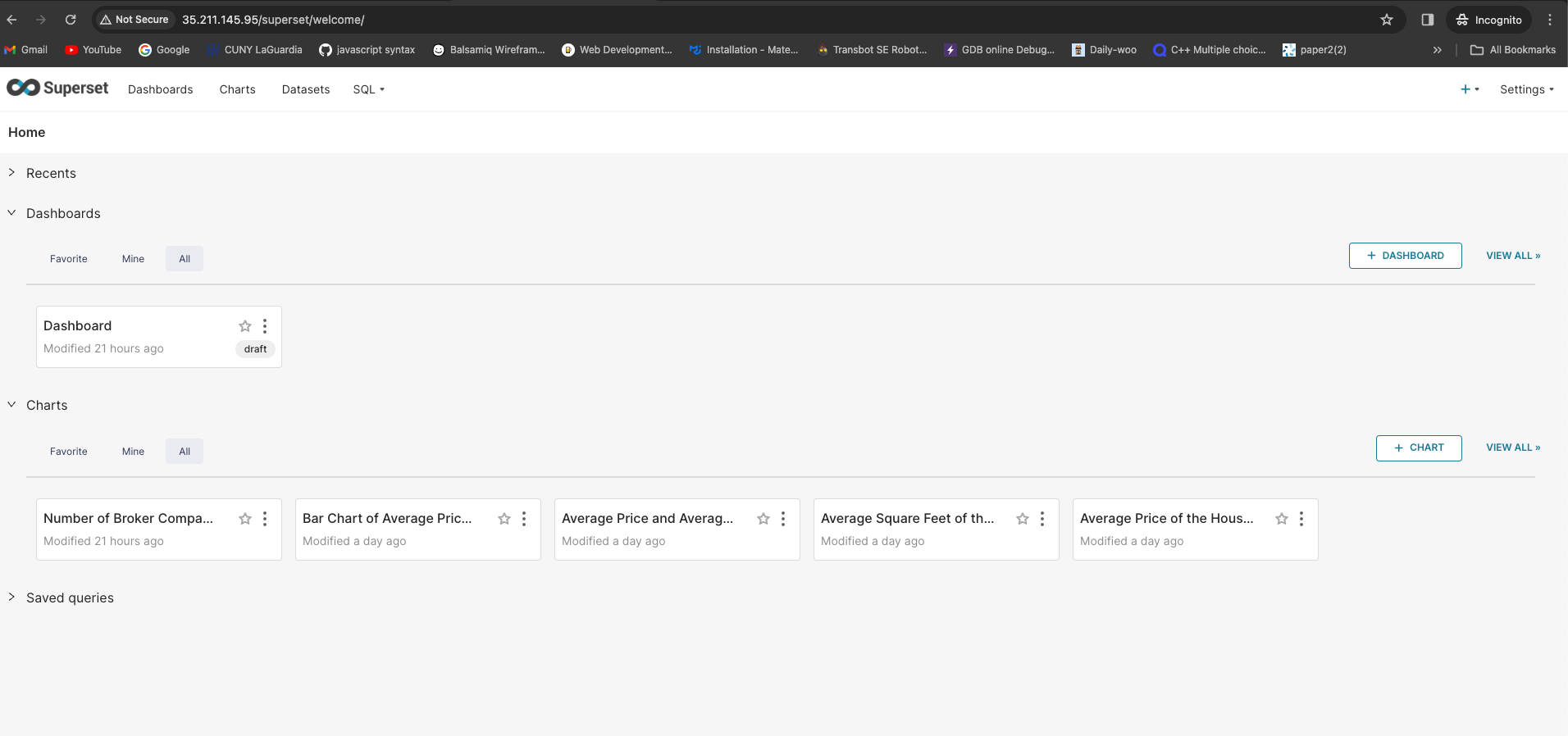


Photo of Web Interface run by Apache Server Hosted on Google Cloud

-- Create a static ip address for your superset vm.

gcloud compute addresses create superset \

--project=glassy-storm-412817 \

--description="Reserved Apache Superset Static IP Address" \

--network-tier=STANDARD \

--region=us-east1

STATIC\_IP=$(gcloud compute addresses describe superset \

--project=glassy-storm-412817 \

--region=us-east1 \

--format='value(address)')

# To see the static ip address echo $STATIC\_IP

echo http://$STATIC\_IP

gcloud compute instances delete-access-config superset \

--project=glassy-storm-412817 \

--zone=us-east1-c \

--access-config-name="External NAT" \

--quiet

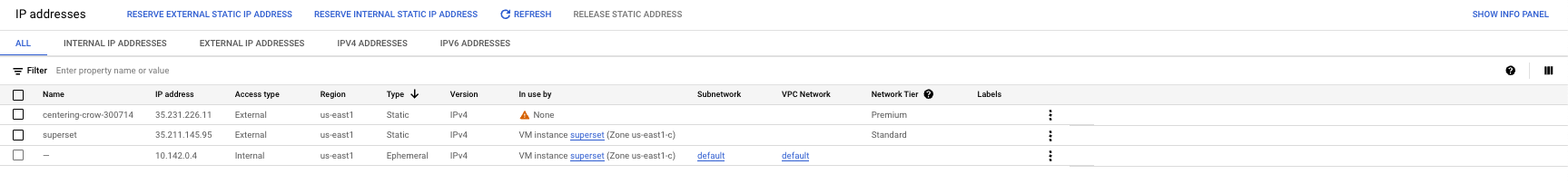
gcloud compute instances add-access-config superset \

--project=glassy-storm-412817 \

--zone=us-east1-c \

--address=$STATIC\_IP \

--network-tier=STANDARD



sadfkllsdf

Photo of reserving a static IP Address to ensure accessibility of the Superset instance on reboots(please zoom in to see clearly)

-- Database Setup

-- !!Go to mysqlsh script directory and run the mysql script file

cd /usr/local/bin

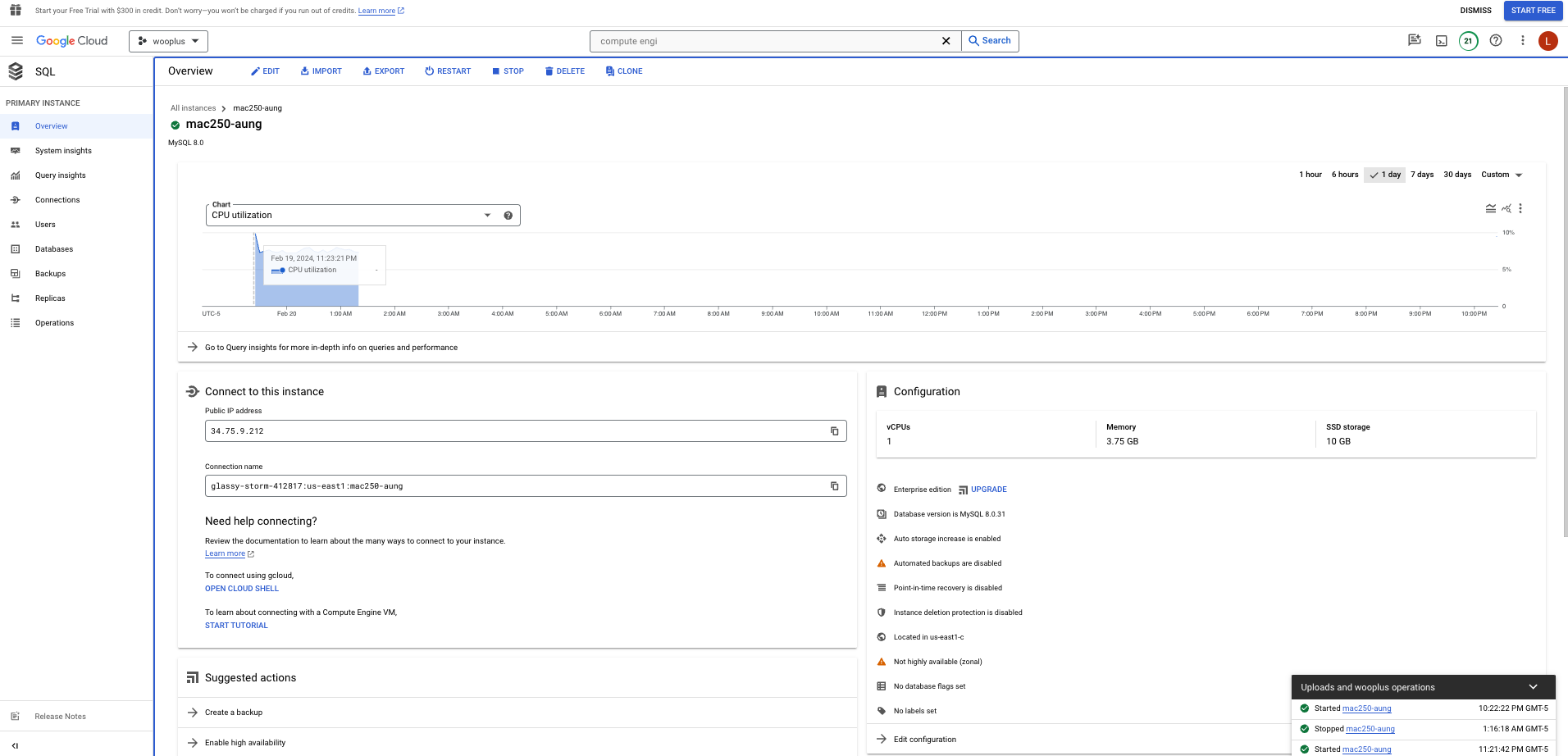
./mysqlsh

-- !!Switch into SQL mode by using \sql and Create the schema and tables

\sql

-- !!Connect to the database server hosted on the Google Cloud on mysql local shell

\c --mysql --user=root --host=34.75.9.212



Creation of MySQL server on Google Cloud

-- !!Create the Database, schema and tables

CREATE DATABASE `final-aung`

USE `final-aung`

CREATE TABLE `final-aung`.`NYHouse` (

`BrokerTitle` varchar(255),

`Type` varchar(255),

`Price` INT,

`Beds` TINYINT,

`Bath` TINYINT,

`PropertySqft` DECIMAL(11,6),

`Address` varchar(255),

`State` varchar(255),

`MainAddress` varchar(255),

`AdministrativeAreaLevel2` varchar(128),

`Locality` varchar(128),

`SubLocality` varchar(128),

`StreetName` varchar(128),

`LongName` varchar(128),

`FormattedAddress` varchar(255),

`Latitude` DECIMAL(11, 8),

`Longitude` DECIMAL(11,8)

);

-- !! change to Javascript Mode

\js

-- !! Import the data from a comma separated values file (csv) JS Mode

util.importTable("/Users/bryan/Desktop/ThirdSemestersCourses/MAC250-Database/NY-House-Dataset.csv" , {

"schema": "final-aung",

"table": "NYHouse",

"dialect": "csv-unix",

"skipRows": 1,

"showProgress": true

})

-- What is the average price of the house in New York ?

SELECT AVG(price) FROM `NYHouse`;

-- What is the average square feet of the house in New York ?

SELECT AVG(PropertySqft) FROM `NYHouse`;

-- Who are the ten top performing brokers ?

SELECT BrokerTitle,COUNT(\*) FROM NYHouse GROUP BY BrokerTitle ORDER BY COUNT(\*) DESC LIMIT 10;

-- Which SubLocality has the highest price ?

SELECT SubLocality, AVG(Price) AS AveragePrice, COUNT(SubLocality) AS NumberOfProperties, AVG(PropertySqft) AS AveragePropertySqft FROM NYHouse GROUP BY SubLocality ORDER BY AVG(Price) DESC;

-- Compare the price and property sq ft in different sublocalities.

SELECT SubLocality, AVG(Price) AS AveragePrice, COUNT(SubLocality) AS NumberOfProperties, AVG(PropertySqft) AS AveragePropertySqft FROM NYHouse GROUP BY SubLocality ORDER BY AveragePropertySqft DESC;

-- Which property type has the highest price in New York? aka List of the average price of the house based on their property type

SELECT Type, AVG(Price) AS AveragePrice, COUNT(SubLocality) AS NumberOfProperties, AVG(PropertySqft) AS AveragePropertySqft FROM NYHouse GROUP BY Type Order By AveragePrice DESC;

-- Which property type has the highest price filtered on major 7 types?

SELECT

CASE

WHEN Type LIKE '%Condo%' THEN 'Condo'

WHEN Type LIKE '%House%' THEN 'House'

WHEN Type LIKE '%Townhouse%' THEN 'Townhouse'

WHEN Type LIKE '%Co-op%' THEN 'Co-op'

WHEN Type LIKE '%Land%' THEN 'Land'

WHEN Type LIKE '%Condop%' THEN 'Condop'

WHEN Type LIKE '%Multi-Family%' THEN 'Multi-family Home'

ELSE 'Other'

END AS PropertyType,

AVG(Price) AS AveragePrice,

COUNT(\*) AS NumberOfProperties,

AVG(PropertySqft) AS AveragePropertySqft

FROM

NYHouse

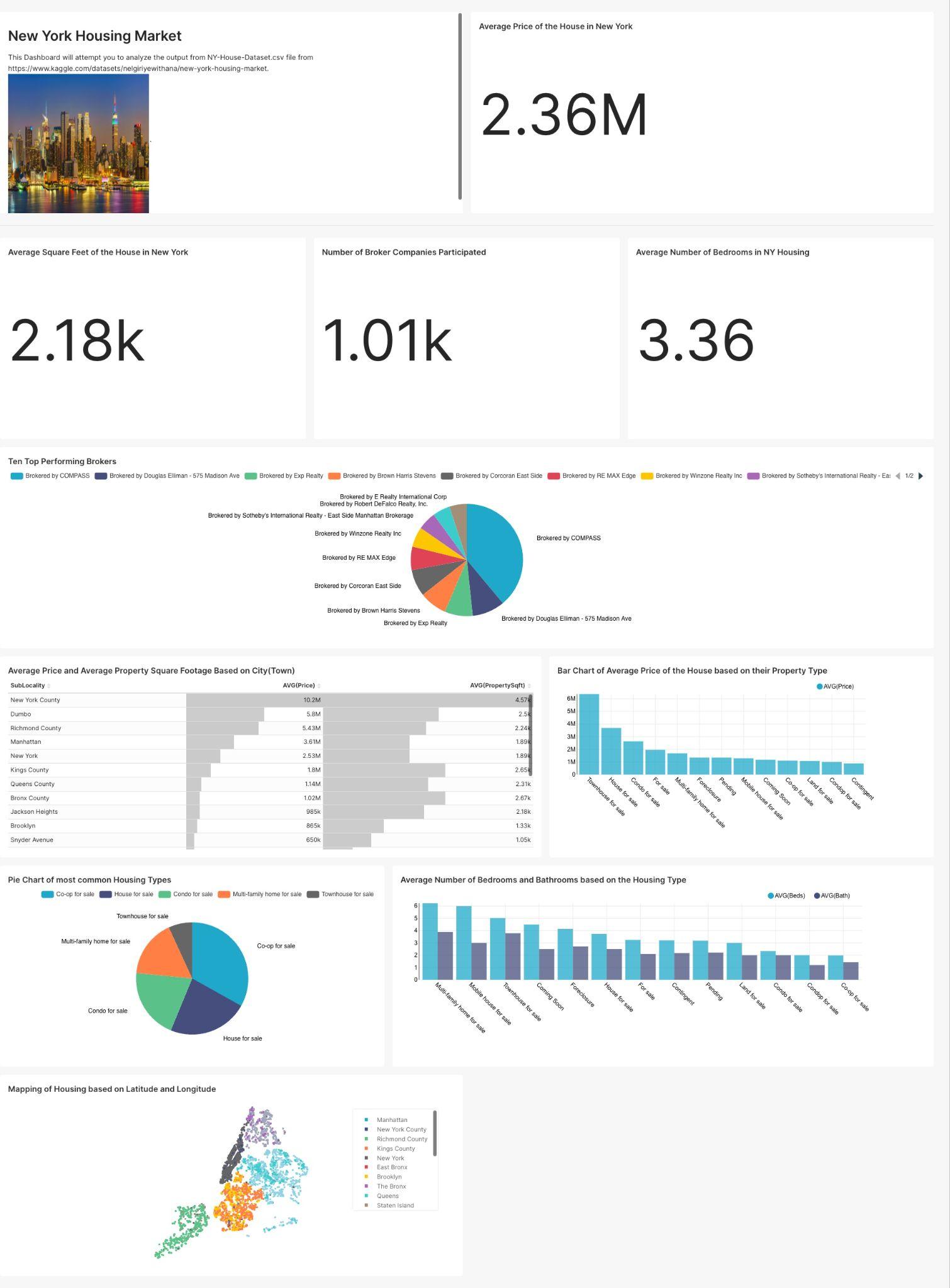
GROUP BY

PropertyType

ORDER BY

AveragePrice;

**Database Visualization**



Dashboard of Analyzing NYHouse Market Data

(<http://35.211.145.95/superset/dashboard/p/6K3zkEq2VXM/>)

**Charts used for Data Visualization**

Description Header - Used for describing resources and datasets utilized for the project

Big Number - Describe the average price of the housing in New York

Big Number - Describe the average Square Feet of the House in New York

Big Number - Describe the total number of Broker Companies Analyzed

Big Number - Describe the average Number of Bedrooms in NY Housing

Pie Chart - Describe the Top Ten Broker Companies Analyzed

Table - Compare the average price and average property square feet of the housing based on SubLocality.

Bar Chart - Describe the average price of the housing based on their property type

Pie Chart - Utilized to display top five trending housing types

Bar Chart - Comparison of the average number of bedrooms and bathrooms depending on housing type

Word Cloud (2D Mapping) - Visualization and differentiation of housing by County and Boroughs, using latitude and longitude.

-- SQL Procedures

DELIMITER |

CREATE PROCEDURE getHousingListAndAveragePrice()

BEGIN

SELECT \* FROM NYHouse;

SELECT AVG(Price) FROM NYHouse;

END |

CALL getHousingListAndAveragePrice();

DELIMITER |

CREATE PROCEDURE getHousingOnSubLocality (param VARCHAR(128))

BEGIN

SELECT BrokerTitle, Type, Price, PropertySqft FROM NYHouse

WHERE SubLocality LIKE param

ORDER BY Price DESC

;

END |

-- Testing with Top 5 Boroughs of New York City

CALL getHousingOnSubLocality('%Brooklyn%');

CALL getHousingOnSubLocality('%Bronx%');

CALL getHousingOnSubLocality('%Queens%');

CALL getHousingOnSubLocality('%Manhattan%');

CALL getHousingOnSubLocality('%Staten Island%');



DELIMITER |

CREATE PROCEDURE getAveragePriceOfPropertyType(param VARCHAR(255))

BEGIN

SELECT Type, AVG(Price) FROM NYHouse

WHERE Type LIKE param

GROUP BY Type;

END |

-- Testing with demanding property type

CALL getAveragePriceOfPropertyType('%condo%'); -- generate average price of condo and condop

CALL getAveragePriceOfPropertyType('%family%');

CALL getAveragePriceOfPropertyType('%house%'); -- generate average price of house, mobile house and town house

CALL getAveragePriceOfPropertyType('%land%');

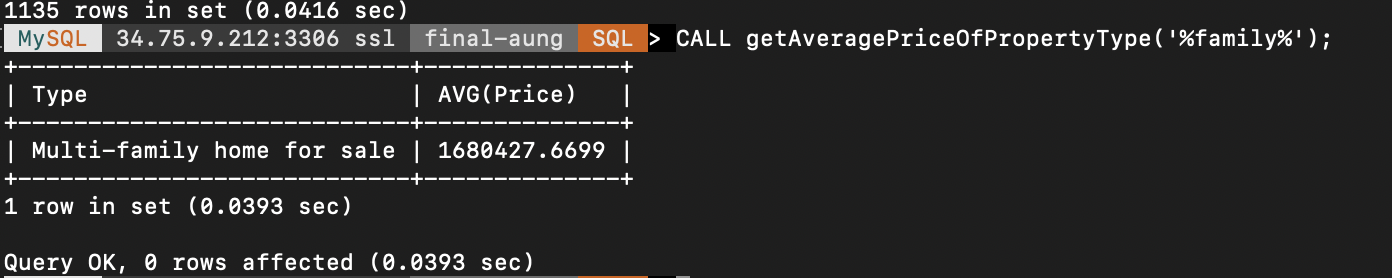


Photo of describing procedural function call

SHOW PROCEDURE STATUS WHERE Db='final-aung'; -- show the procedures created

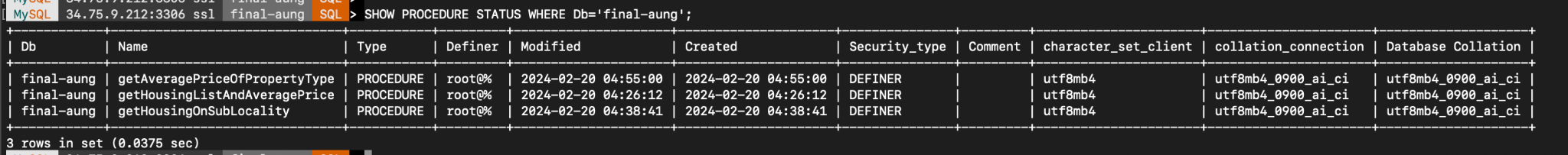


Photo of listing all procedures

-- Create a view

CREATE VIEW NYHousingView AS

SELECT 'mysql' dbms,t.TABLE\_SCHEMA,t.TABLE\_NAME,c.COLUMN\_NAME,c.ORDINAL\_POSITION,c.DATA\_TYPE,c.CHARACTER\_MAXIMUM\_LENGTH,n.CONSTRAINT\_TYPE,k.REFERENCED\_TABLE\_SCHEMA,k.REFERENCED\_TABLE\_NAME,k.REFERENCED\_COLUMN\_NAME FROM INFORMATION\_SCHEMA.TABLES t LEFT JOIN INFORMATION\_SCHEMA.COLUMNS c ON t.TABLE\_SCHEMA=c.TABLE\_SCHEMA AND t.TABLE\_NAME=c.TABLE\_NAME LEFT JOIN INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE k ON c.TABLE\_SCHEMA=k.TABLE\_SCHEMA AND c.TABLE\_NAME=k.TABLE\_NAME AND c.COLUMN\_NAME=k.COLUMN\_NAME LEFT JOIN INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS n ON k.CONSTRAINT\_SCHEMA=n.CONSTRAINT\_SCHEMA AND k.CONSTRAINT\_NAME=n.CONSTRAINT\_NAME AND k.TABLE\_SCHEMA=n.TABLE\_SCHEMA AND k.TABLE\_NAME=n.TABLE\_NAME WHERE t.TABLE\_TYPE='BASE TABLE' AND t.TABLE\_SCHEMA NOT IN('INFORMATION\_SCHEMA','mysql','performance\_schema');

-- Create a view for showing table data structure

CREATE VIEW NYDataStructureView AS

SELECT column\_name, data\_type, character\_maximum\_length

FROM information\_schema.columns

WHERE table\_name = 'NYHouse';

Database Backup

-- Create a directory to store the backup in bin directory

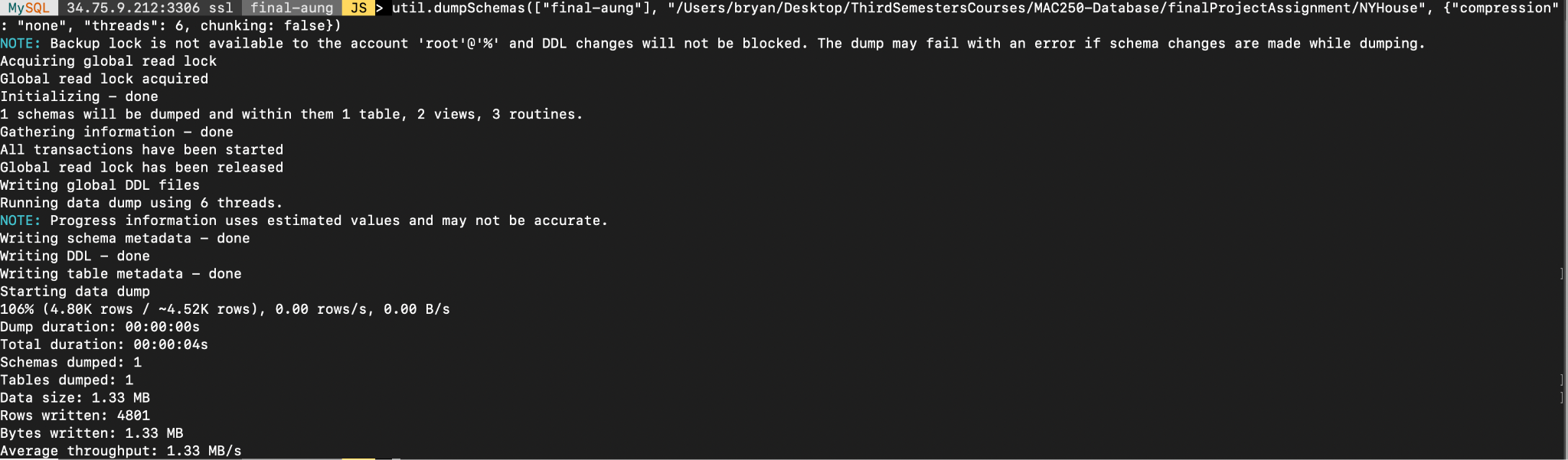
sudo mkdir -p /usr/local/bin/NYHouse

sudo chown -R bryan:staff /usr/local/bin/NYHouse

util.dumpSchemas(["final-aung"], "NYHouse", {"compression": "none", "threads": 6, chunking: false})

- - Store another backup in final project directory

util.dumpSchemas(["final-aung"], "/Users/bryan/Desktop/ThirdSemestersCourses/MAC250-Database/finalProjectAssignment/NYHouse", {"compression": "none", "threads": 6, chunking: false})



Successful Implementation of Database Backup with schema, views and routines.

\