# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial

#### Authors:  ZHICHENG YU, LOUIS TAN, NEVIN YILMAZ(Term paper group four)

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date:12/16/2018

**Lab Tutorial**

ZHICHENG YU ([zyu4@calstatela.edu](mailto:zyu4@calstatela.edu))

Louis Tan ([ltan3@calstatela.edu](mailto:ltan3@calstatela.edu))

Nevin Yilmaz ([nyilmaz@calstatela.edu](mailto:nyilmaz@calstatela.edu) )

12/6/2018

**NYC Parking Tickets Data Analysis using Hive**

**Objectives**

This tutorial is using NYC Parking Tickets dataset from Kaggle.com. You will experience using Hadoop Ambari UI to run HIVEQL and learn data analysis on HIVEQL.

We will begin with downloading the dataset including five csv files, and then pick one file to implement a few HIVE operations, such as:

* Create table Schema
* Select specific columns from table
* Sort the data
* Filter and Group the data

**PRE-REQUISITES**

* Available Ambari UI Hadoop platform
* Tableau or Excel to visualize data

**Platform Spec**

* **IBM Server**
* **16GB RAM**
* **Node: 3**
* **HDFS Version: 2.7.3**
* **Hive Version 1.2.1000**
* **CPU: V4 2.2GHZ**

**OUTLINE**

* What is HIVEQL?
* Step 1: Download the Data
* Step2: Upload the data files
* Step3: Create Hive table to query those parking tickets
* Step4: Create Hive queries to analyze the data
* Step5: Use Tableau or Excel to visualize the data

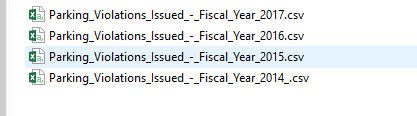
**PART A: HIVEQL In Hadoop Ambari UI**

Step 1: Download the data to local and upload it to the server

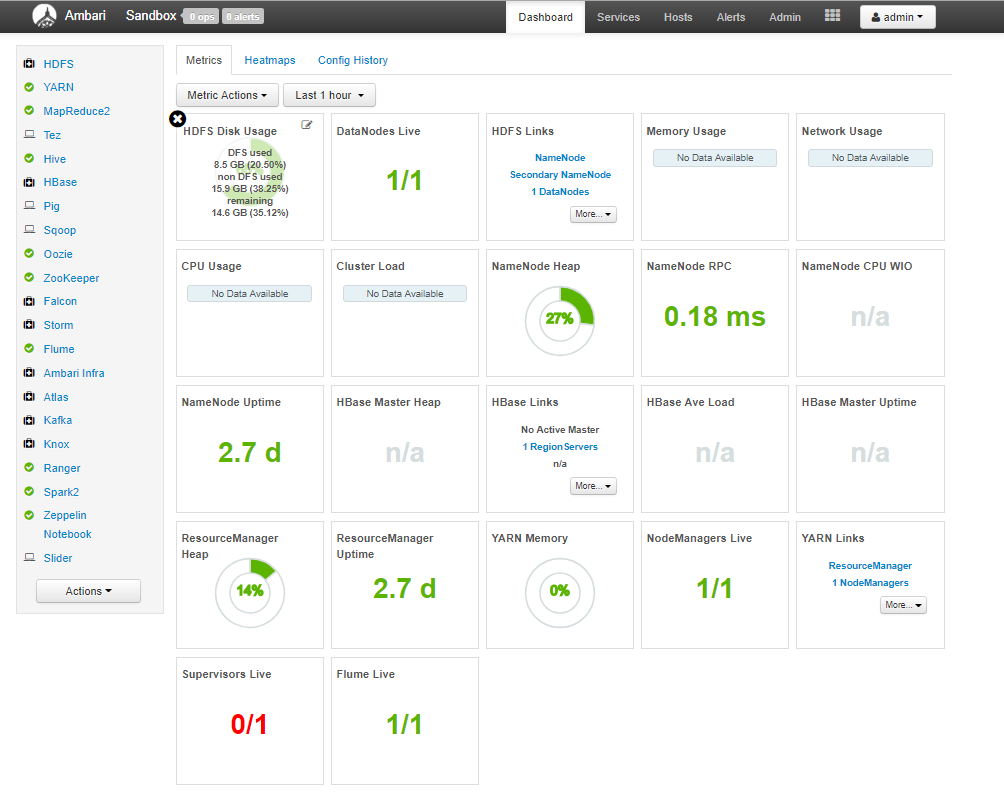
1.Download the parking tickets data using the fi

<https://www.kaggle.com/new-york-city/nyc-parking-tickets>?

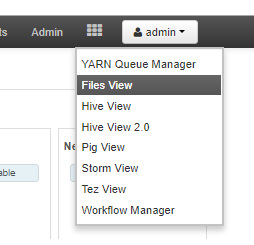
2.Unzip the file



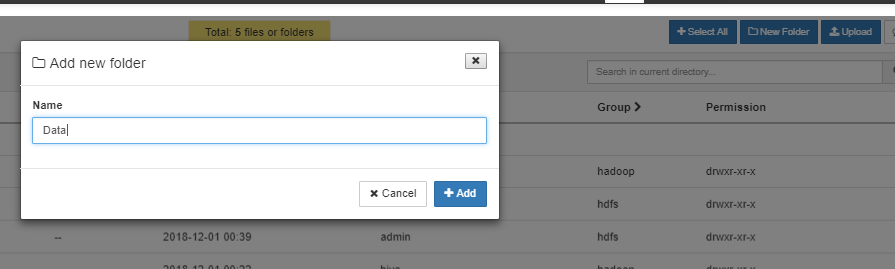
3. UPLOAD THE DATA FILES, open Ambari UI



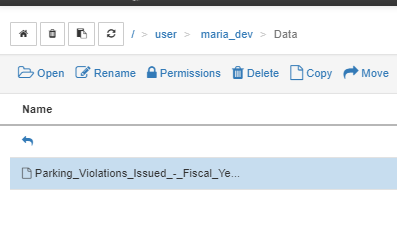
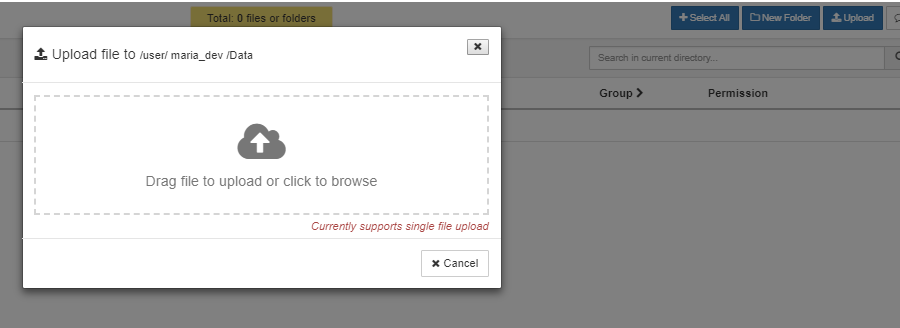
4.Open file view on the right corner of the window

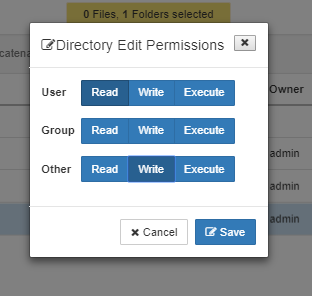
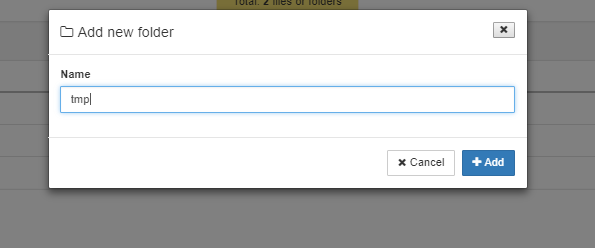


5.Then open path ‘/user/username/’ and Create folder ‘Data’



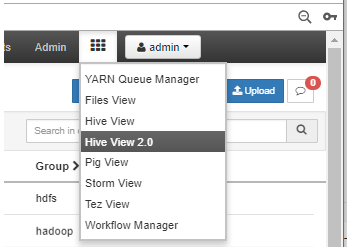
6.Open folder Data and upload dataset csv

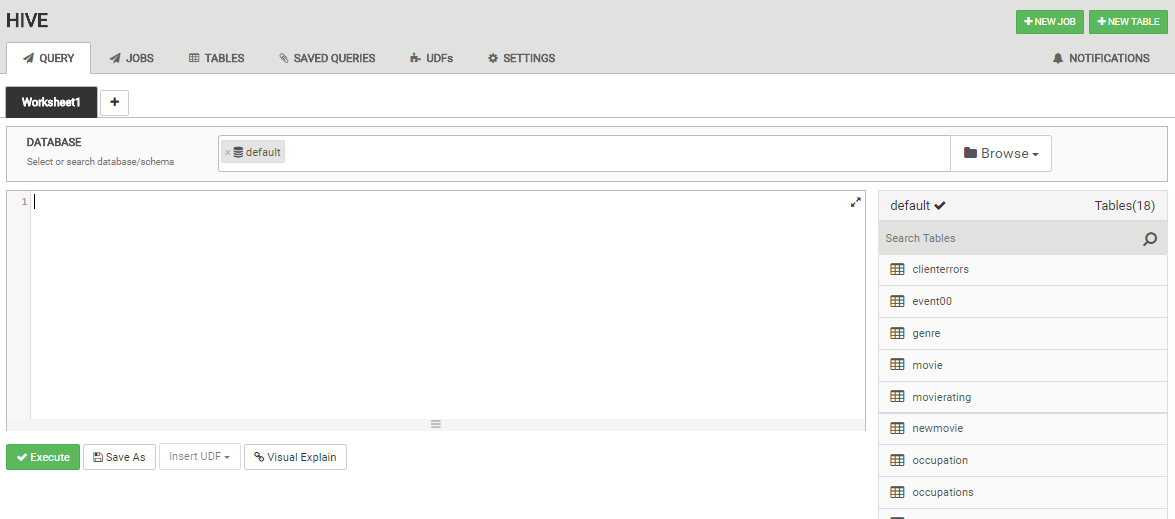


6.Create folder tmp under the path ‘/user/username/’ and change its permission to full access

Step 2: Creating Hive table to Query Parking Tickets Data

1.To perform Hive statement, firstly open the HiveView2.0 in Ambari.





2.Create your own database statement, please replace username with your new user name. Copy and paste the code into HiveView2.0

CREATE database IF NOT EXISTS username

3.Then use select database statement.

USE username

4.To use following code creating external table, which can store the original dataset. Please replace username.

CREATE EXTERNAL TABLE IF NOT EXISTS parking\_2014 (

Summons\_Number int,

Plate\_ID string,

Registration\_State string,

Plate\_Type string,

Issue\_Date string,

Violation\_Code int,

Vehicle\_Body\_Type string,

Vehicle\_Make string,

Issuing\_Agency char(255),

Street\_Code1 int,

Street\_Code2 int,

Street\_Code3 int,

Vehicle\_Expiration\_Date int,

Violation\_Location int,

Violation\_Precinct int,

Issuer\_Precinct int,

Issuer\_Code int,

Issuer\_Command int,

Issuer\_Squad int,

Violation\_Time string,

Time\_First\_Observed string,

Violation\_County string,

Violation\_In\_Front\_Of\_Or\_Opposite char(255),

House\_Number string,

Street\_Name string,

Intersecting\_Street string,

Date\_First\_Observed int,

Law\_Section int,

Sub\_Division string,

Violation\_Legal\_Code string,

Days\_Parking\_In\_Effect string,

From\_Hours\_In\_Effect string,

To\_Hours\_In\_Effect string,

Vehicle\_Color string,

Unregistered\_Vehicle int,

Vehicle\_Year int,

Meter\_Number char(255),

Feet\_From\_Curb int,

Violation\_Post\_Code string,

Violation\_Description string,

No\_Standing\_or\_Stopping\_Violation string,

Hydrant\_Violation string,

Double\_Parking\_Violation string,

Latitude string,

Longitude string,

Community\_Board string,

Community\_Council string,

Census\_Tract string,

BIN string,

BBL string,

NTA string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/data/'

5.Then load data into table

LOAD DATA INPATH '/user/username/data/ Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2014\_.csv' OVERWRITE INTO TABLE parking\_2014;

Step 3: Creating Hive Queries to Analyze Data

1.Copy and paste the following Hive code to HiveView2.0.

CREATE TABLE IF NOT EXISTS SimpleTickets

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/SimpleTickets'

AS

SELECT Registration\_State,

Issue\_Date,

Plate\_Type,

Vehicle\_Body\_Type,

Vehicle\_Make,

Vehicle\_Expiration\_Date,

Violation\_Time,

Violation\_County,

Vehicle\_Color,

Vehicle\_Year

FROM parking\_2014;

2.Create ticketsbyID table to query data

CREATE TABLE IF NOT EXISTS ticketsbyID

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/ticketsbyID'

AS

SELECT Registration\_State ,Plate\_ID,Plate\_Type, COUNT(\*) as Total

FROM SimpleTickets

GROUP BY Registration\_State ,Plate\_ID,Plate\_Type

ORDER BY Total DESC;

3.Create ticketsbyState to query data

CREATE TABLE IF NOT EXISTS ticketsbyState

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/ticketsbyState'

AS

SELECT Registration\_State, COUNT(\*)/9100279 \* 100 AS Per

FROM SimpleTickets

GROUP BY Registration\_State

ORDER BY Per DESC;

4.Create table ticketsbyIssueDate to query data

CREATE TABLE IF NOT EXISTS ticketsbymonth

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/ticketsbymonth'

AS

SELECT split(Issue\_Date ,'[\/]')[0] AS month, COUNT(\*) as Total

FROM SimpleTickets

GROUP BY month

ORDER BY Total DESC;

5.Create table ticketsbymonth to query data

CREATE TABLE IF NOT EXISTS ticketsbymonth

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/username/ticketsbymonth'

AS

SELECT month, SUM(Total) as Total

FROM ticketsbyIssueDate

GROUP BY month

ORDER BY Total DESC;

6.Create table ticketsbylocation

CREATE TABLE IF NOT EXISTS ticketsbylocation

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/ username /tmp/ticketsbylocation'

AS

SELECT House\_Number,

Street\_Name,

Intersecting\_Street, COUNT(\*) as Total

FROM parking 2014

GROUP BY House\_Number,

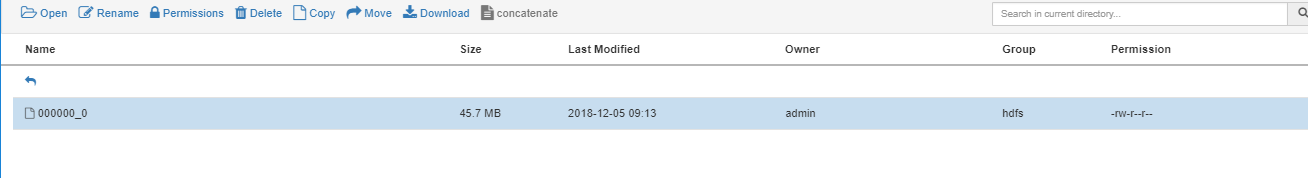
Street\_Name,

Intersecting\_Street

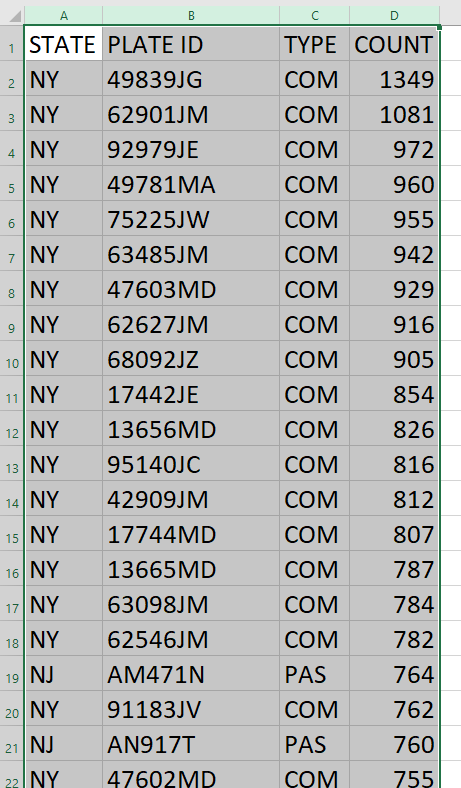
ORDER BY Total DESC;

Step 4: Loading Data into Excel and Visualization

1.Open Ambari into File view under path ‘/user/username/tmp/ticketsbyID/’

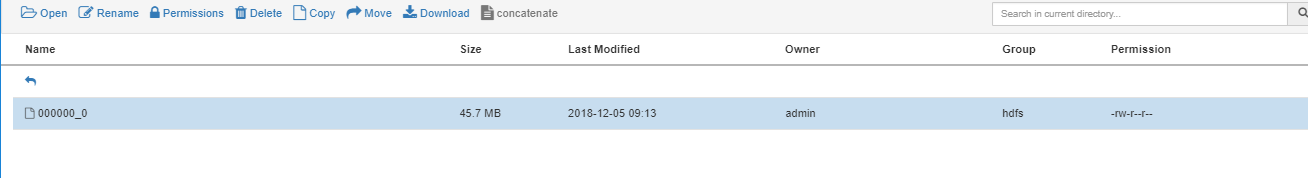
Then select file 000000\_0 and click download button on top.

Open File in Microsoft EXCEL

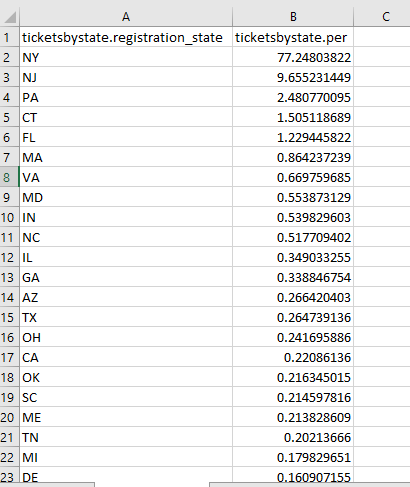
.

2.Open Ambari into File view under path ‘/user/username/tmp/ticketsbyState/’

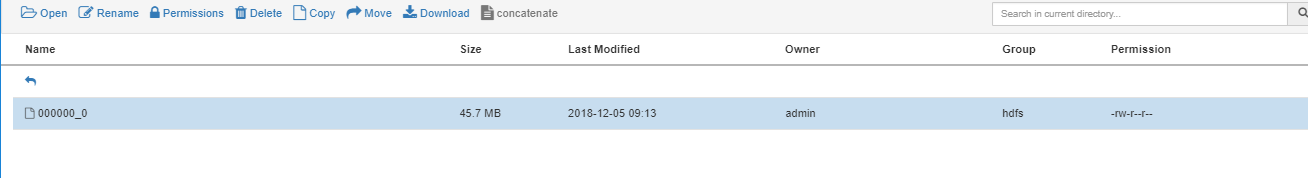
Then select file 000000\_0 and click download button on top.



Open File in Microsoft EXCEL.

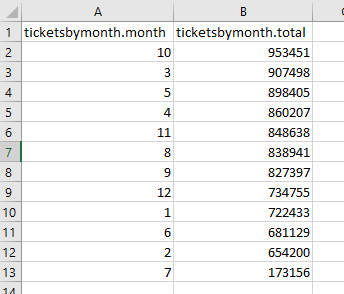


3.Open Ambari into File view under path ‘/user/username/tmp/ticketsbymonth/’

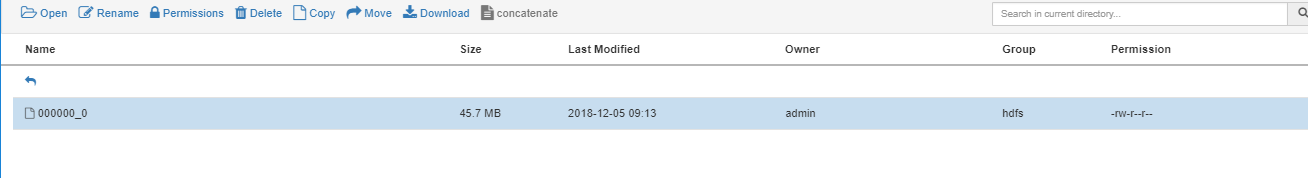


Then select file 000000\_0 and click download button on top.

Open File in Microsoft EXCEL.

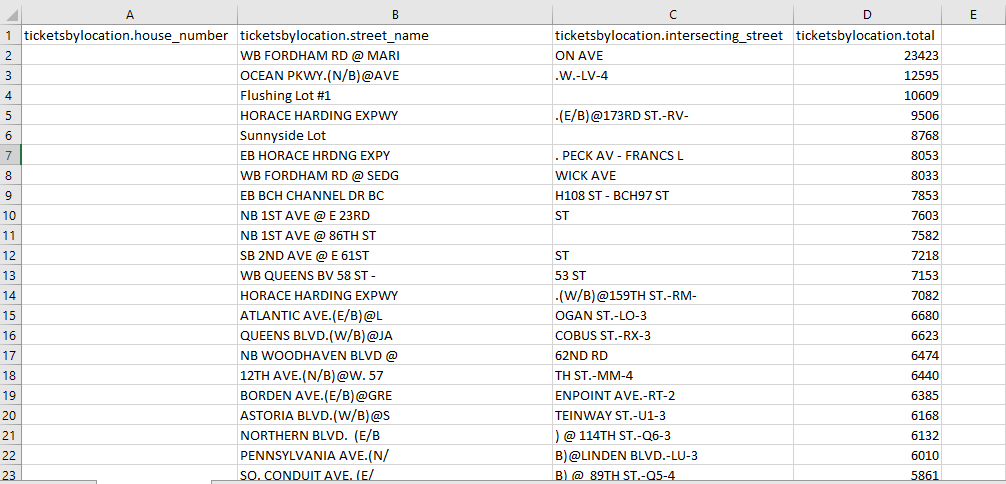


4.Open Ambari into File view under path ‘/user/username/tmp/ticketsbylocation/’



Then select file 000000\_0 and click download button on top.

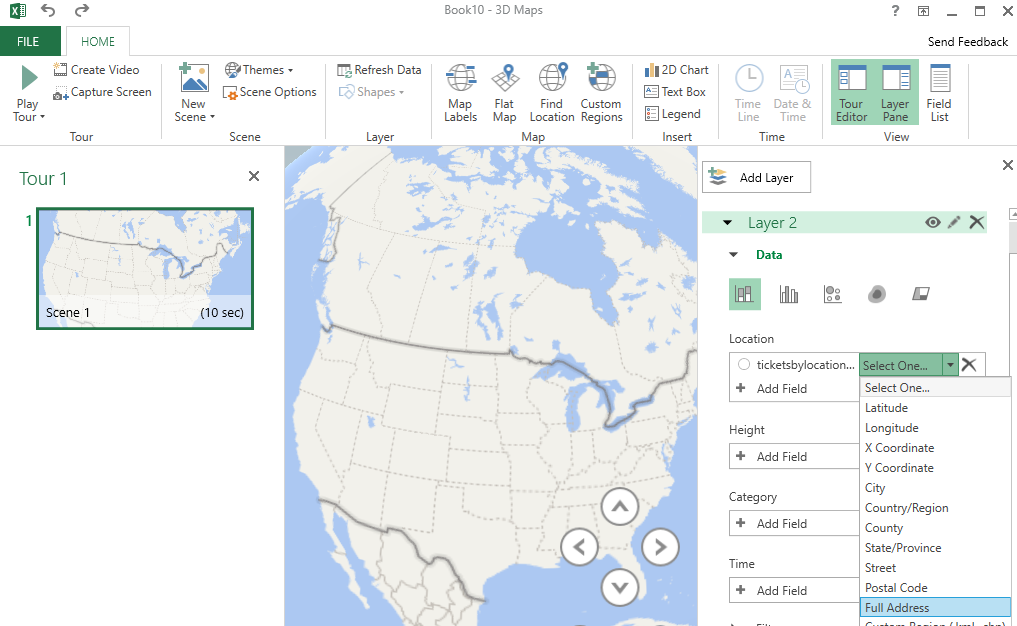
Open File in Microsoft EXCEL.



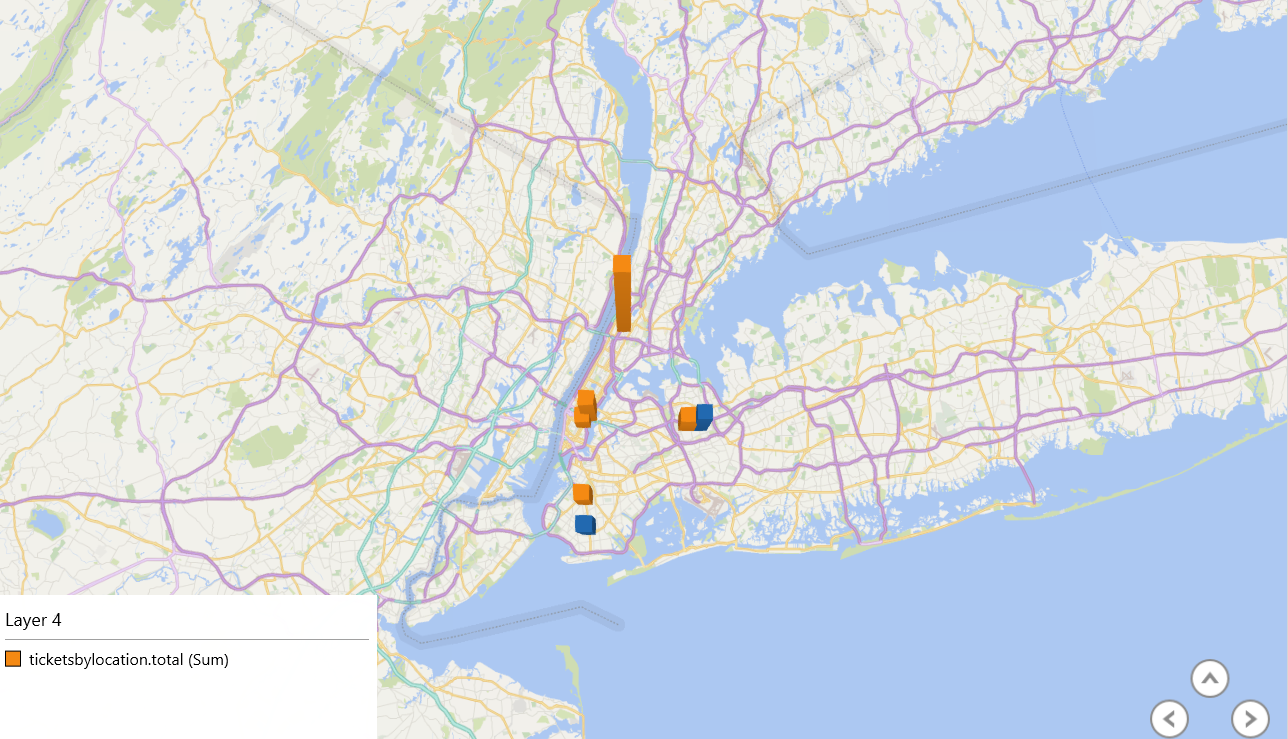
5.Open EXCEL 3D maps and select column street\_name as loation



Select total as height



You will see the 3D map with column as following



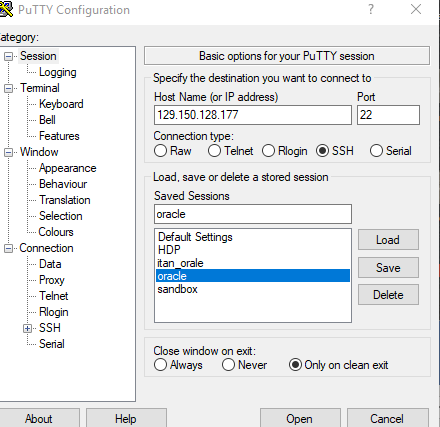
**PART B: Oracle Server**

Step 1: Loading Data into Oracle Server:

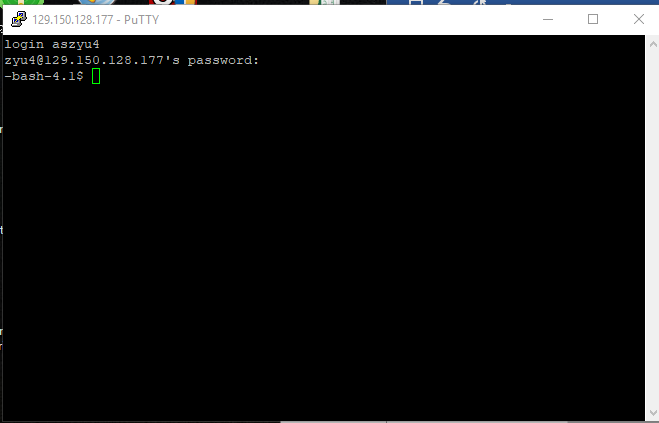
Putty and Psftp download link:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

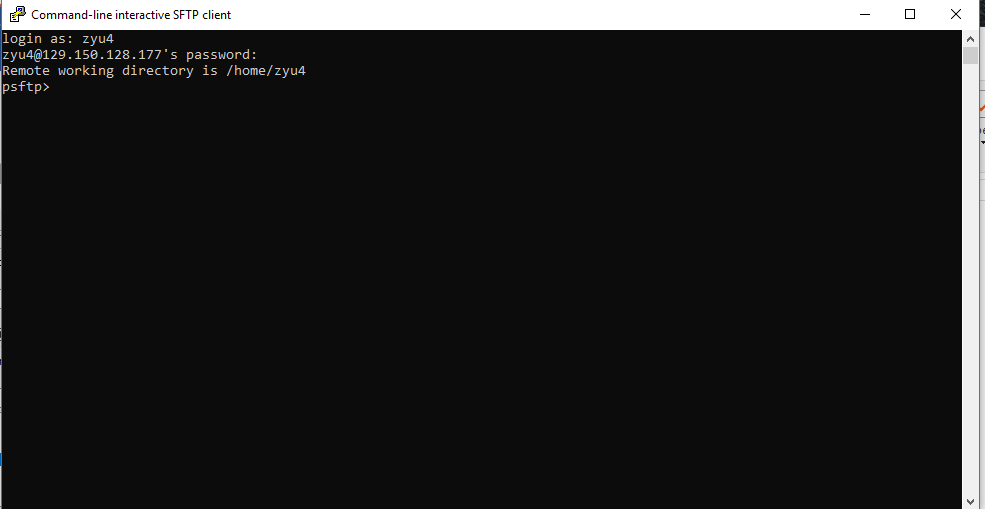
1.Open Putty to connect to oracle server



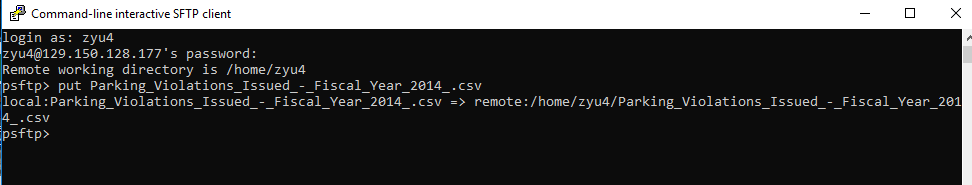
2.Use your own username and password to login



3.Then open SFTP, and use the same username and password as putty to login.



4.Use command ‘Put Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2014\_.csv’ to upload dataset.



Step 2: Analyze data using Hive

1.aConnect to Oracle server with “ssh *username@ip”* with your password.

1b.Connect to beeline with provided beeline command.

1c.After choosing your database with “*USE username*” copy and paste the following Hive code to create the table

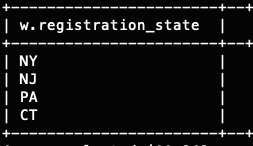
CREATE EXTERNAL TABLE group4\_project (summons\_number INT, PlateID STRING, Registration\_state STRING, Plate\_type STRING, issue\_date STRING, Violation\_code INT, vehicle\_body\_type STRING, Vehicle\_make STRING, issuing\_agenct STRING, street\_code1 INT, street\_code2 INT, street\_code3 INT, vehicle\_exp\_date INT, violation\_location STRING, violation\_precinct INT, issuer\_precinct INT, issuer\_code INT, issuer\_command STRING, issuer\_squad STRING, vioaltion\_time STRING, time\_first\_observed STRING, violation\_country STRING, violation\_in\_front\_of\_or\_opposite STRING, house\_number STRING, street\_name STRING, intersecting\_street STRING, date\_first\_observed INT, law\_section INT, sub\_division STRING, violation\_legal\_code STRING, days\_parking\_in\_effect STRING, from\_hours\_in\_effect STRING, to\_hours\_in\_effect STRING, vehicle\_color STRING, unregistered\_vehicle STRING, vehicle\_year INT, meter\_number STRING, feet\_from\_curb INT, violation\_post\_code STRING, violation\_desc STRING, no\_standing\_or\_stopping\_violation STRING, hydrant\_violation STRING, double\_parking\_violation STRING, latitude INT, longitude INT, community\_board INT, community\_council INT, census\_tract INT, BIN INT, BBL INT, NTA STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' location '/user/nyilmaz/Term\_Project/' tblproperties ("skip.header.line.count"="1");

2. Ngram: Five most common trigrams

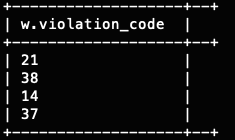
SELECT explode(ngrams(sentences(lower(violation\_desc)), 3, 5)) AS trigram from group4\_project;

3. Show top 4 state occurrences

select w.registration\_state from ( select registration\_state, count(\*) as cnt from group4\_project group by registration\_state order by cnt desc) w limit 4;



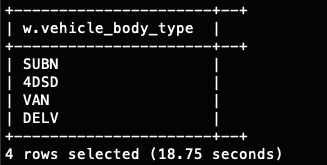
4. Show top 4 most occurred violation codes

select w.violation\_code from ( select violation\_code, count(\*) as cnt from group4\_project group by violation\_code order by cnt desc) w limit 4;

(Violation code descriptions can be found at the github link in DOF\_Parking\_Violation\_Codes.csv) [2]

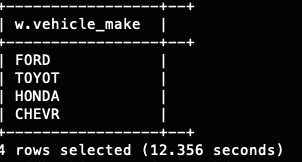
5.Show top 4 vehicle body type

select w.vehicle\_body\_type from ( select vehicle\_body\_type, count(\*) as cnt from group4\_project group by vehicle\_body\_type order by cnt desc) w limit 4;



6. Show top 4 vehicle make

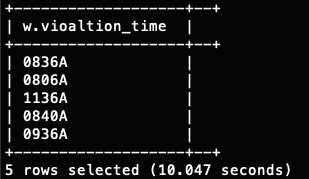
select w.vehicle\_make from (select vehicle\_make, count(\*) as cnt from group4\_project group by vehicle\_make order by cnt desc) w limit 4;



7. Show bottom 20 vehicle make

select w.vehicle\_make from (select vehicle\_make, count(\*) as cnt from group4\_project group by vehicle\_make order by cnt asc) w limit 20;

8. Top 5 violation time

select w.vioaltion\_time from (select vioaltion\_time, count(\*) as cnt from group4\_project group by vioaltion\_time order by cnt desc) w limit 5;

References

* 1. https://www.kaggle.com/new-york-city/nyc-parking-tickets
  2. <https://github.com/nyilmaz1/NYC_parking-Hive.git>
  3. <https://www.dropbox.com/home/cis4560_5200Fall2018?preview=ch12Hive_p2_30.pdf>