# CIS5200 Term Project Tutorial

#### Authors: Dalyapraz Dauletbak; Ketki Vyas; Wenji Hu; Ritika Joshi

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

#### Date: 10/12/2018

**Lab Tutorial**

Dalyapraz Dauletbak, Wenji Hu, Ketki Vyas, Ritika Joshi

([dmanato@calstatela.edu](mailto:dmanato@calstatela.edu), [whu4@calstatela.edu](mailto:whu4@calstatela.edu), [kvyas2@calstatela.edu](mailto:kvyas2@calstatela.edu), [rjoshi5@calstatela.edu](mailto:rjoshi5@calstatela.edu))

10/12/2018

**Waze Traffic Patterns**

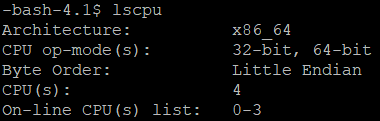
**Objectives**

In this hands-on lab, you will learn how to:

* Upload dataset to HDFS
* Create table on beeline
* Clean data using RegExp in queries
* Create summary tables that have only required data to build visualizations.
* Export results of summary tables into csv files
* Build visualizations using Power BI and Excel.

**Platform Spec**

* Hadoop Cluster version: Hadoop 2.7.1.2.4.2.0-258
* Hive version: Hive 1.2.1000.2.4.2.0-258
* Number of nodes: 5
* HDFS Capacity: 147 GB
* OCPUs: 10
* Memory: 150GB
* Storage: 678GB
* CPU speed: 2195.084 MHz
* CPU Model Name: Intel(R) Xeon(R) CPU E5-2699C v4 @ 2.20GHz



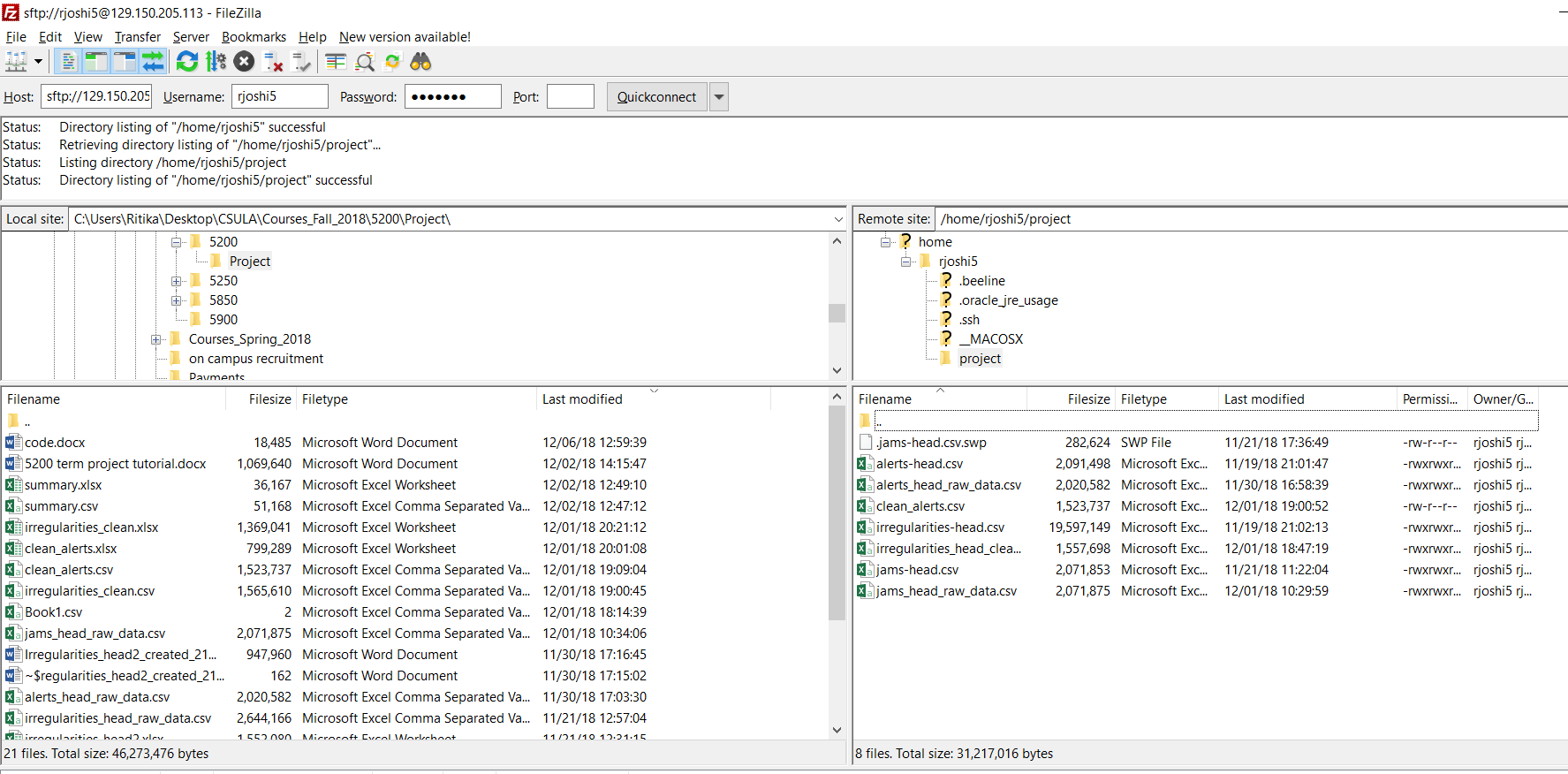




Step 1: Download the dataset in your local machine and then upload it in Hadoop server using FileZilla

Since the data set we used is not public, there is no link to download it. However, such data can be shared according to the agreement with Waze only to authorized partners registered for CCP program with Waze. Note that we have been shared the dataset from City of Los Angeles in the part of the agreement with California State University.

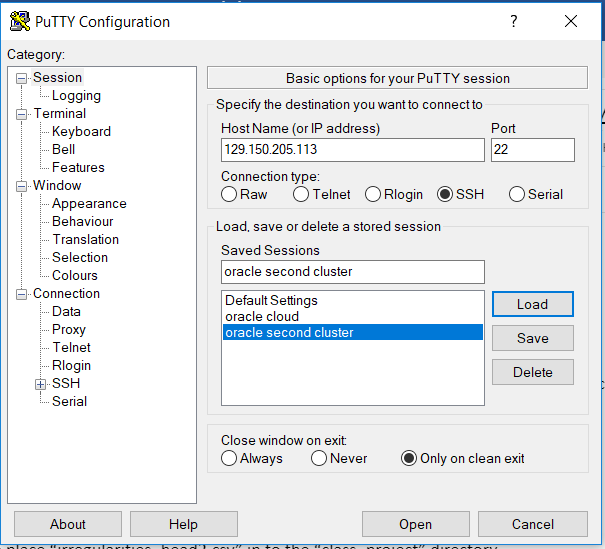
Once you have data set, you can upload it to Hadoop server using FileZilla as shown below.

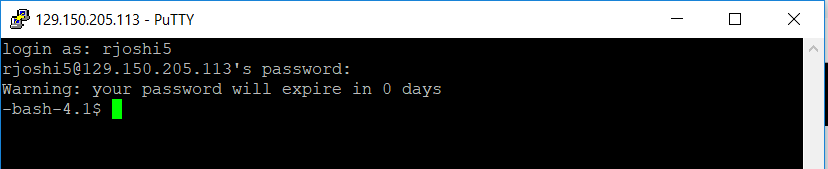


Step2: Upload data into HDFS

Login to the server and place all files into HDFS following below steps.

1. Connect to Hadoop cluster using putty and provide username/password



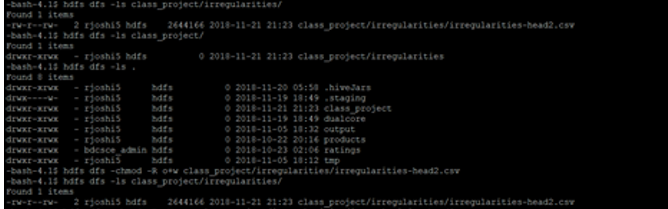
Using HDFS commands create a dir “class\_project” and a sub-dir “irregularities” in HDFS.

hdfs dfs -mkdir class\_project

hdfs dfs -mkdir class\_project/irregularities

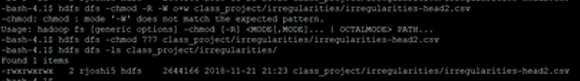
1. Move file place “irregularities\_head2.csv” in to the “class\_project” directory.

hdfs dfs -put irregularities\_head2.csv class\_project/irregularities/



1. Change permissions.

hdfs dfs -chmod -R 777 .



1. Repeat similar steps for second file ‘jams\_head\_raw\_data.csv”. Create new sub-folder ‘jams’ and place the file followed by providing permissions.

hdfs dfs -put jams\_head\_raw\_data.csv ./class\_project/jams

hdfs dfs -chmod 777 ./class\_project/jams

hdfs dfs -chmod 777 ./class\_project/jams/



1. Follow the same steps for “alerts\_head\_raw\_data.csv“ file. Create a sub-dir “alerts” inside “class\_project” and put “alerts\_head\_raw\_data.csv“ inside sub-dir “alerts”. Also give required permissions.

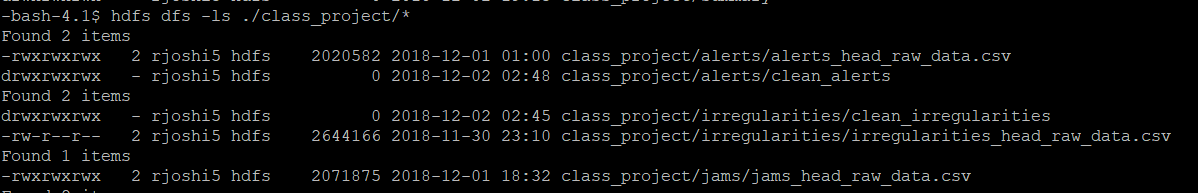
hdfs dfs -mkdir ./class\_project/alerts

hdfs dfs -chmod 777 ./class\_project/\*

hdfs dfs -put alerts\_head\_raw\_data.csv ./class\_project/alerts/

hdfs dfs -chmod 777 ./class\_project/alerts/\*

hdfs dfs -ls ./class\_project/\*



Step 3: Connect to hive

This step is to Open beeline CLI (Command Line Shell Interface) that is equivalent to hive CLI environment for table operations.

1. Connect to beeline

-bash-4.1$ beeline

WARNING: Use "yarn jar" to launch YARN applications.

Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive

beeline> !connect jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce\_admin

Connecting to jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive

Enter password for jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive:

1. Use your database with your username to separate your tables with other users.

use rjoshi5;

show tables;

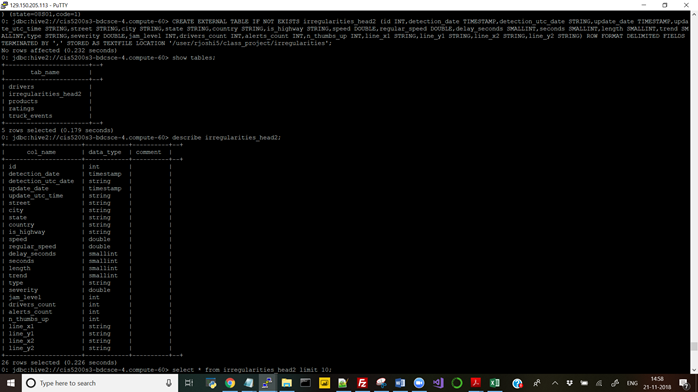


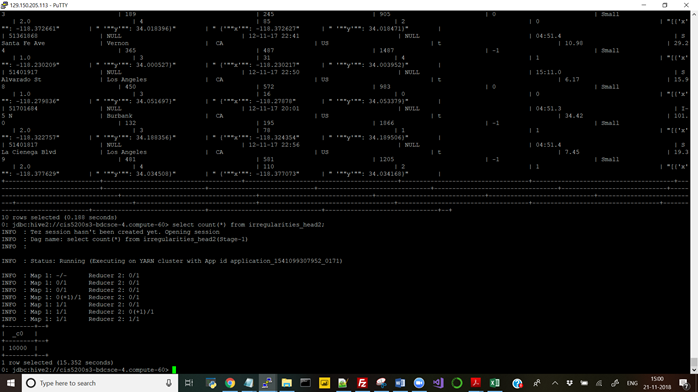
1. Create an external table for irregularities.

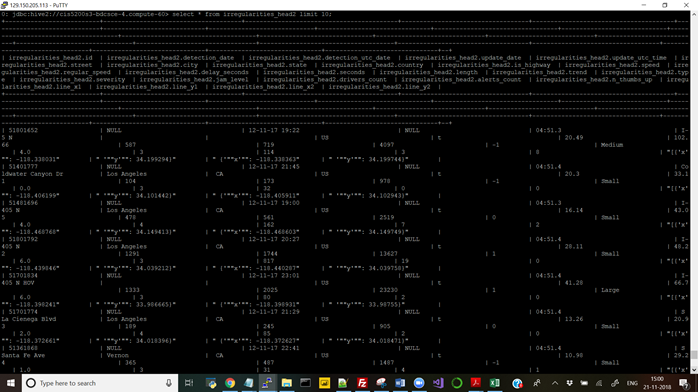
CREATE EXTERNAL TABLE IF NOT EXISTS irregularities\_head\_raw\_data(id INT,detection\_date STRING,detection\_utc\_date STRING,update\_date STRING,update\_utc\_time STRING,street STRING,city STRING,state STRING,country STRING,is\_highway STRING,speed DOUBLE,regular\_speed DOUBLE,delay\_seconds SMALLINT,seconds SMALLINT,length SMALLINT,trend SMALLINT,type STRING,severity DOUBLE,jam\_level INT,drivers\_count INT,alerts\_count INT,n\_thumbs\_up INT,line\_x1 STRING,line\_y1 STRING,line\_x2 STRING,line\_y2 STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/rjoshi5/class\_project/irregularities';







1. Create new table and insert cleaned data into “irregularities\_head\_clean”.

bash-4.1$ hdfs dfs -mkdir /user/rjoshi5/class\_project/irregularities/clean\_irregularities/

-bash-4.1$ hdfs dfs -chmod 777 /user/rjoshi5/class\_project/irregularities/clean\_irregularities/

CREATE EXTERNAL TABLE IF NOT EXISTS irregularities\_head\_clean

( id INT,

weekday STRING,

data\_time STRING,

street STRING,

city STRING,

state STRING,

country STRING,

is\_highway STRING,

speed DOUBLE,

regular\_speed DOUBLE,

delay\_seconds SMALLINT,

seconds SMALLINT,

length SMALLINT,

trend SMALLINT,

type STRING,

severity DOUBLE,

jam\_level INT,

drivers\_count INT,

alerts\_count INT,

n\_thumbs\_up INT,

x1 STRING,

y1 STRING,

x2 STRING,

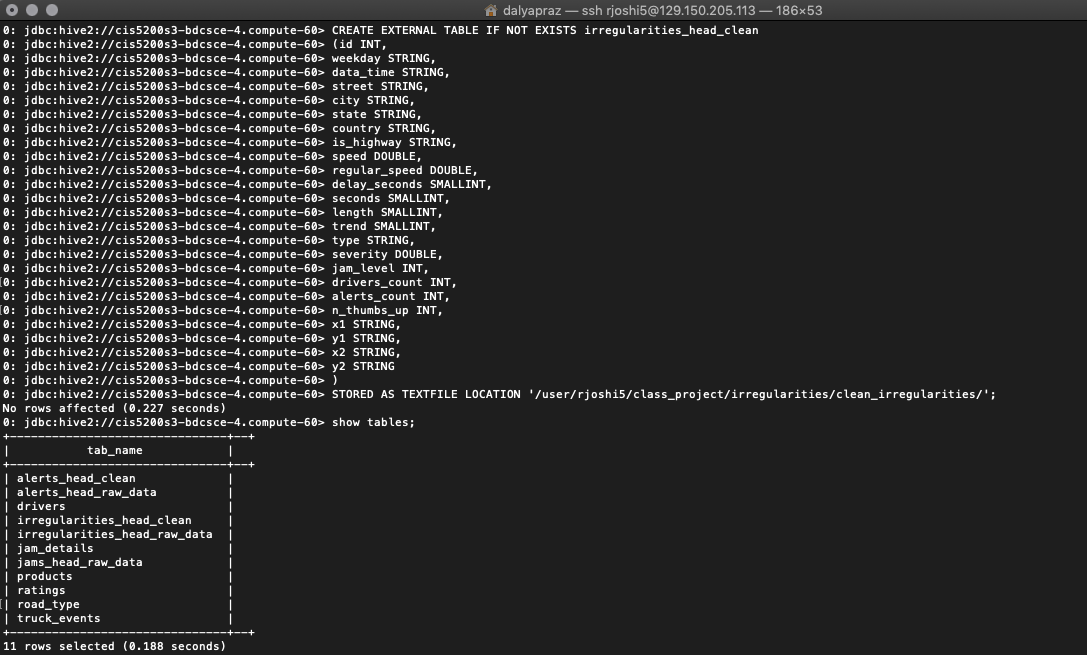
y2 STRING

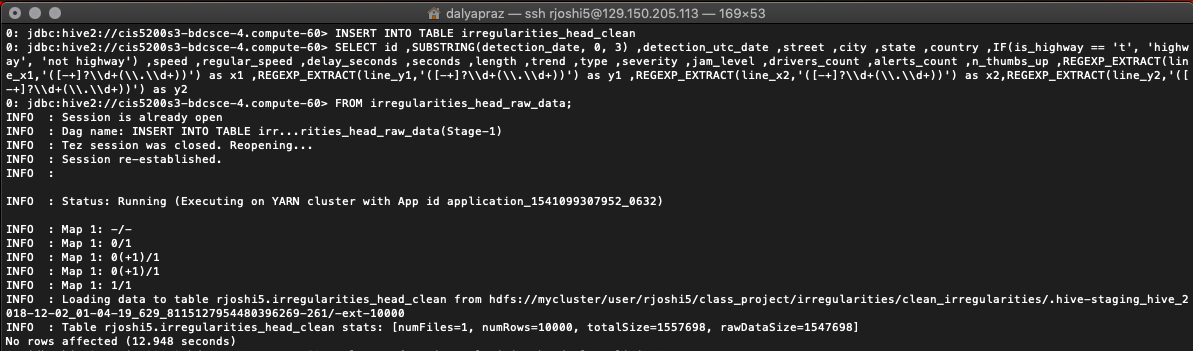
)

STORED AS TEXTFILE LOCATION '/user/rjoshi5/class\_project/irregularities/clean\_irregularities/';

INSERT INTO TABLE irregularities\_head\_clean

SELECT id ,SUBSTRING(detection\_date, 0, 3) ,detection\_utc\_date ,street ,city ,state ,country ,IF(is\_highway == 't', 'highway', 'not highway') ,speed ,regular\_speed ,delay\_seconds ,seconds ,length ,trend ,type ,severity ,jam\_level ,drivers\_count ,alerts\_count ,n\_thumbs\_up ,REGEXP\_EXTRACT(line\_x1,'([-+]?\\d+(\\.\\d+))') as x1 ,REGEXP\_EXTRACT(line\_y1,'([-+]?\\d+(\\.\\d+))') as y1 ,REGEXP\_EXTRACT(line\_x2,'([-+]?\\d+(\\.\\d+))') as x2,REGEXP\_EXTRACT(line\_y2,'([-+]?\\d+(\\.\\d+))') as y2 FROM irregularities\_head\_raw\_data;



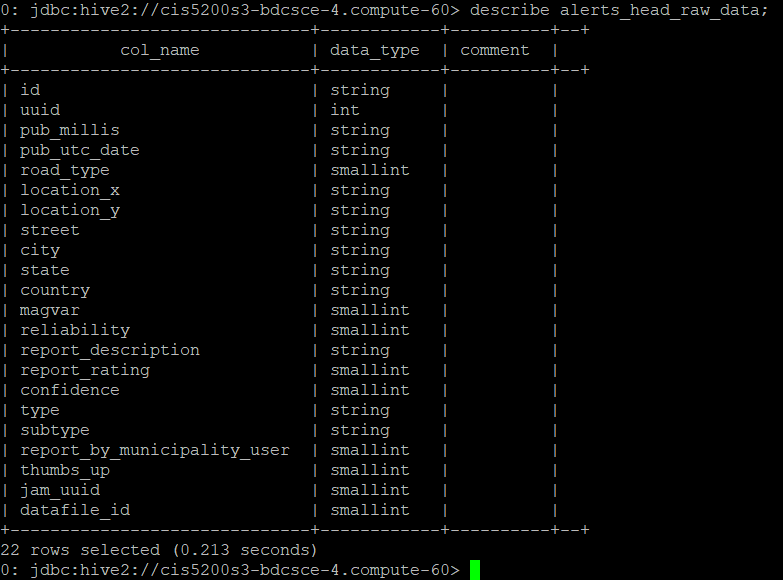


1. Create another table “alerts\_head\_raw\_data”. To use alerts file.

CREATE EXTERNAL TABLE IF NOT EXISTS alerts\_head\_raw\_data(id STRING,uuid INT,pub\_millis STRING,pub\_utc\_date STRING,road\_type SMALLINT,location\_x STRING,location\_y STRING,street STRING,city STRING,state STRING,country STRING,magvar SMALLINT,reliability SMALLINT,report\_description STRING,report\_rating SMALLINT,confidence SMALLINT,type STRING,subtype STRING,report\_by\_municipality\_user SMALLINT,thumbs\_up SMALLINT,jam\_uuid SMALLINT,datafile\_id SMALLINT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/user/rjoshi5/class\_project/alerts';

Check the table structure using describe:

describe alerts\_head\_raw\_data;



1. Create additional table with name for road types “road\_type” and insert data manually.

CREATE TABLE road\_type (value SMALLINT,name STRING);

INSERT INTO road\_type (value, name) VALUES (1, 'Streets');

INSERT INTO road\_type (value, name) VALUES (2, 'Primary Street');

INSERT INTO road\_type (value, name) VALUES (3, 'Freeways');

INSERT INTO road\_type (value, name) VALUES (4, 'Ramps');

INSERT INTO road\_type (value, name) VALUES (5, 'Trails');

INSERT INTO road\_type (value, name) VALUES (6, 'Primary');

INSERT INTO road\_type (value, name) VALUES (7, 'Secondary');

INSERT INTO road\_type (value, name) VALUES (8, '4X4 Trails');

INSERT INTO road\_type (value, name) VALUES (9, 'Walkway');

INSERT INTO road\_type (value, name) VALUES (10, 'Pedestrian');

INSERT INTO road\_type (value, name) VALUES (11, 'Exit');

INSERT INTO road\_type (value, name) VALUES (12, '?');

INSERT INTO road\_type (value, name) VALUES (13, '?');

INSERT INTO road\_type (value, name) VALUES (14, '4X4 Trails');

INSERT INTO road\_type (value, name) VALUES (15, 'Ferry crossing');

INSERT INTO road\_type (value, name) VALUES (16, 'Stairway');

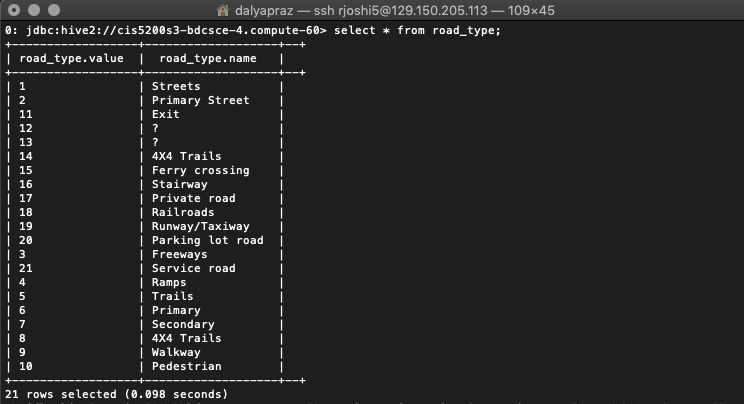
INSERT INTO road\_type (value, name) VALUES (17, 'Private road');

INSERT INTO road\_type (value, name) VALUES (18, 'Railroads');

INSERT INTO road\_type (value, name) VALUES (19, 'Runway/Taxiway');

INSERT INTO road\_type (value, name) VALUES (20, 'Parking lot road');

INSERT INTO road\_type (value, name) VALUES (21, 'Service road');



1. Create new table and insert cleaned data “alerts\_head\_clean ”.

-bash-4.1$ hdfs dfs -mkdir /user/rjoshi5/class\_project/alerts/clean\_alerts/

-bash-4.1$ hdfs dfs -chmod 777 /user/rjoshi5/class\_project/alerts/clean\_alerts/

CREATE EXTERNAL TABLE IF NOT EXISTS alerts\_head\_clean

( id STRING,

date\_time STRING,

road\_type STRING,

location\_x STRING,

location\_y STRING,

street STRING,

city STRING,

state STRING,

country STRING,

reliability SMALLINT,

report\_description STRING,

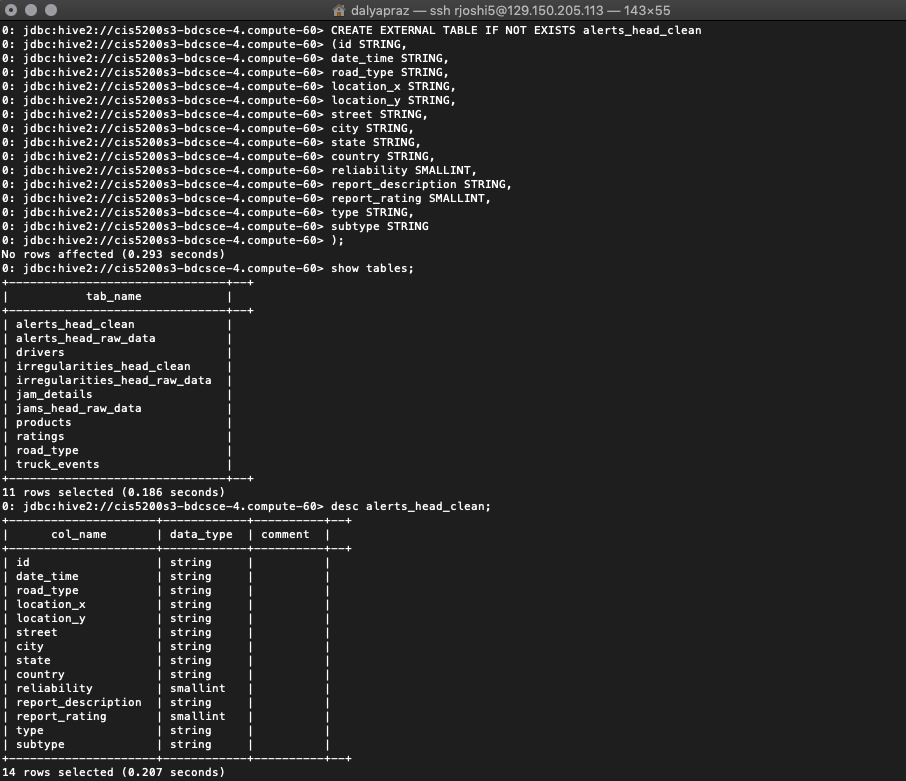
report\_rating SMALLINT,

type STRING,

subtype STRING

)

STORED AS TEXTFILE LOCATION '/user/rjoshi5/class\_project/alerts/clean\_alerts/';



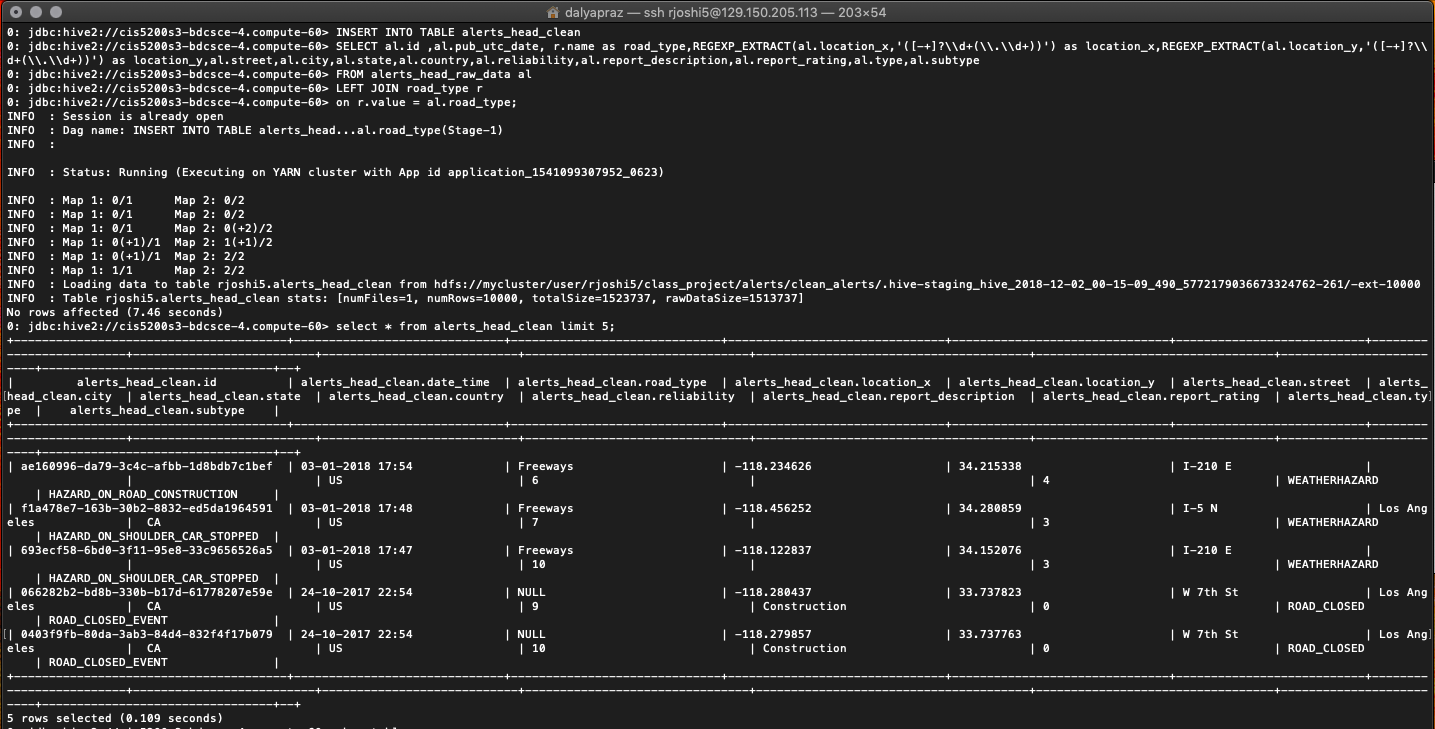
1. INSERT INTO TABLE alerts\_head\_clean

SELECT al.id ,al.pub\_utc\_date, r.name as road\_type,REGEXP\_EXTRACT(al.location\_x,'([-+]?\\d+(\\.\\d+))') as location\_x,REGEXP\_EXTRACT(al.location\_y,'([-+]?\\d+(\\.\\d+))') as location\_y,al.street,al.city,al.state,al.country,al.reliability,al.report\_description,al.report\_rating,al.type,al.subtype

FROM alerts\_head\_raw\_data al

LEFT JOIN road\_type r

on r.value = al.road\_type;



1. Create a summary table “alerts\_sum” from “alerts\_head\_clean ” using HiveQL.

CREATE EXTERNAL TABLE IF NOT EXISTS alerts\_sum

( no\_of\_alerts SMALLINT,

date\_ STRING,

time\_ STRING,

city STRING,

road\_type STRING,

type STRING,

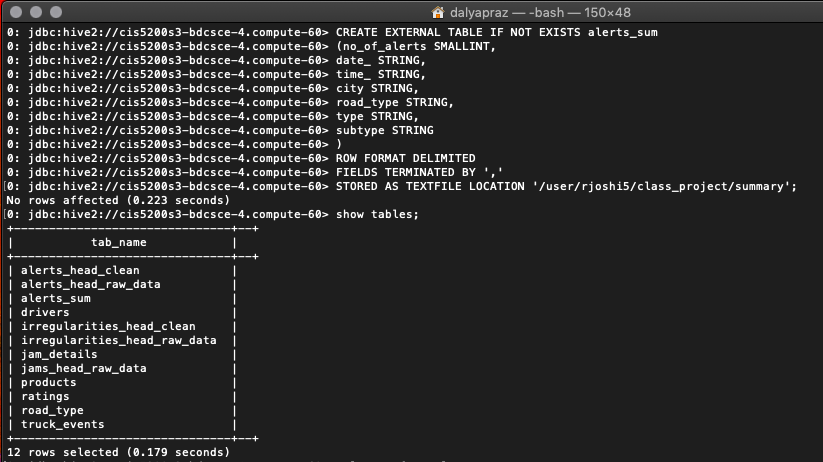
subtype STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/rjoshi5/class\_project/summary';



1. Insert data into summary table name “alerts\_sum”

INSERT INTO TABLE alerts\_sum

SELECT count(\*) as no\_of\_alerts,SUBSTRING(date\_time, 0, 10), SUBSTRING(date\_time, 12, 5), city,road\_type, type, subtype

FROM alerts\_head\_clean

GROUP BY date\_time, city, road\_type, type, subtype;



Step 4: Download the output files into local machine

After the Hive tables are created, fetch csv files from output location to Hadoop folder and download it to your local machine using FileZilla.

1. Rename and Download output of cleaned tables to your local computer.

-bash-4.1$ hdfs dfs -cat /user/rjoshi5/class\_project/irregularities/clean\_irregularities/00\* | wc –l



-bash-4.1$ hdfs dfs -cat /user/rjoshi5/class\_project/alerts/clean\_alerts/00\* | wc –l



-bash-4.1$ hdfs dfs -get /user/rjoshi5/class\_project/irregularities/clean\_irregularities/\*

-bash-4.1$ ls -l

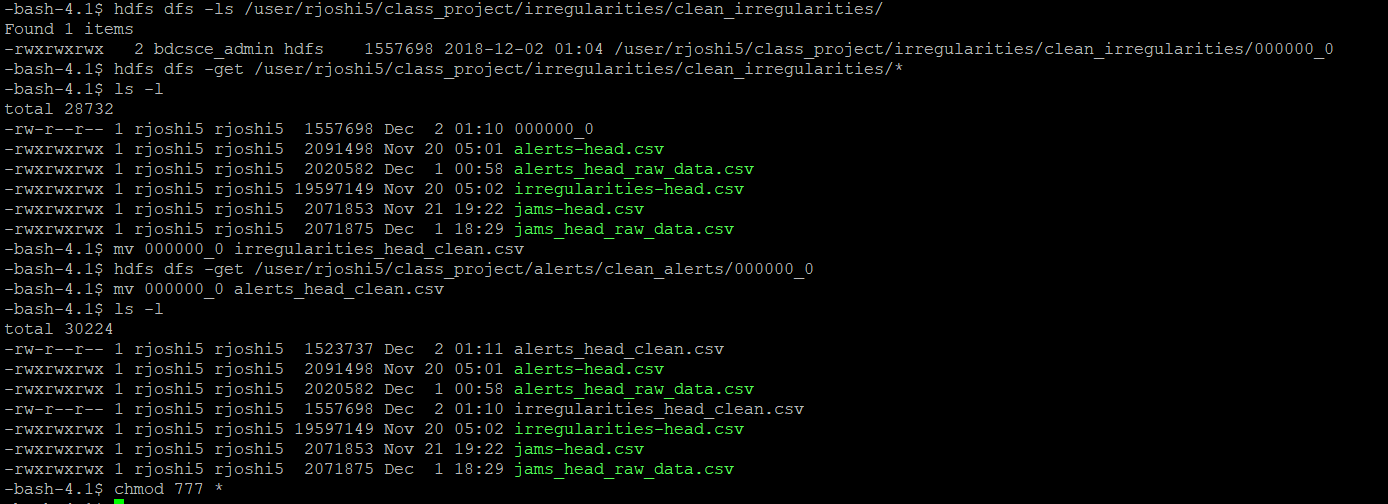
-bash-4.1$ mv 000000\_0 irregularities\_head\_clean.csv

-bash-4.1$ hdfs dfs -get /user/rjoshi5/class\_project/alerts/clean\_alerts/000000\_0

-bash-4.1$ mv 000000\_0 alerts\_head\_clean.csv

-bash-4.1$ ls -l

-bash-4.1$ chmod 777 \*

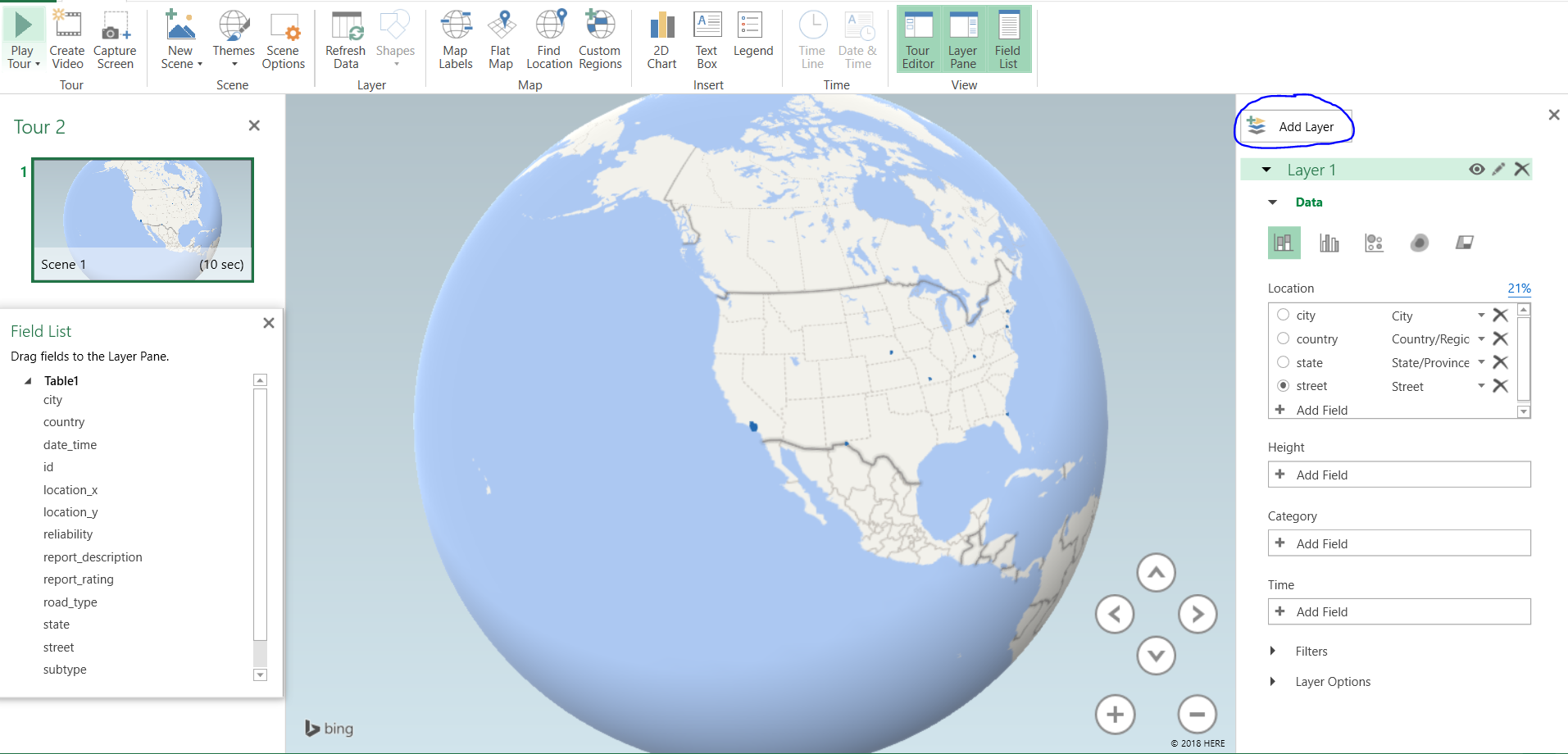


Step 5: Visualization

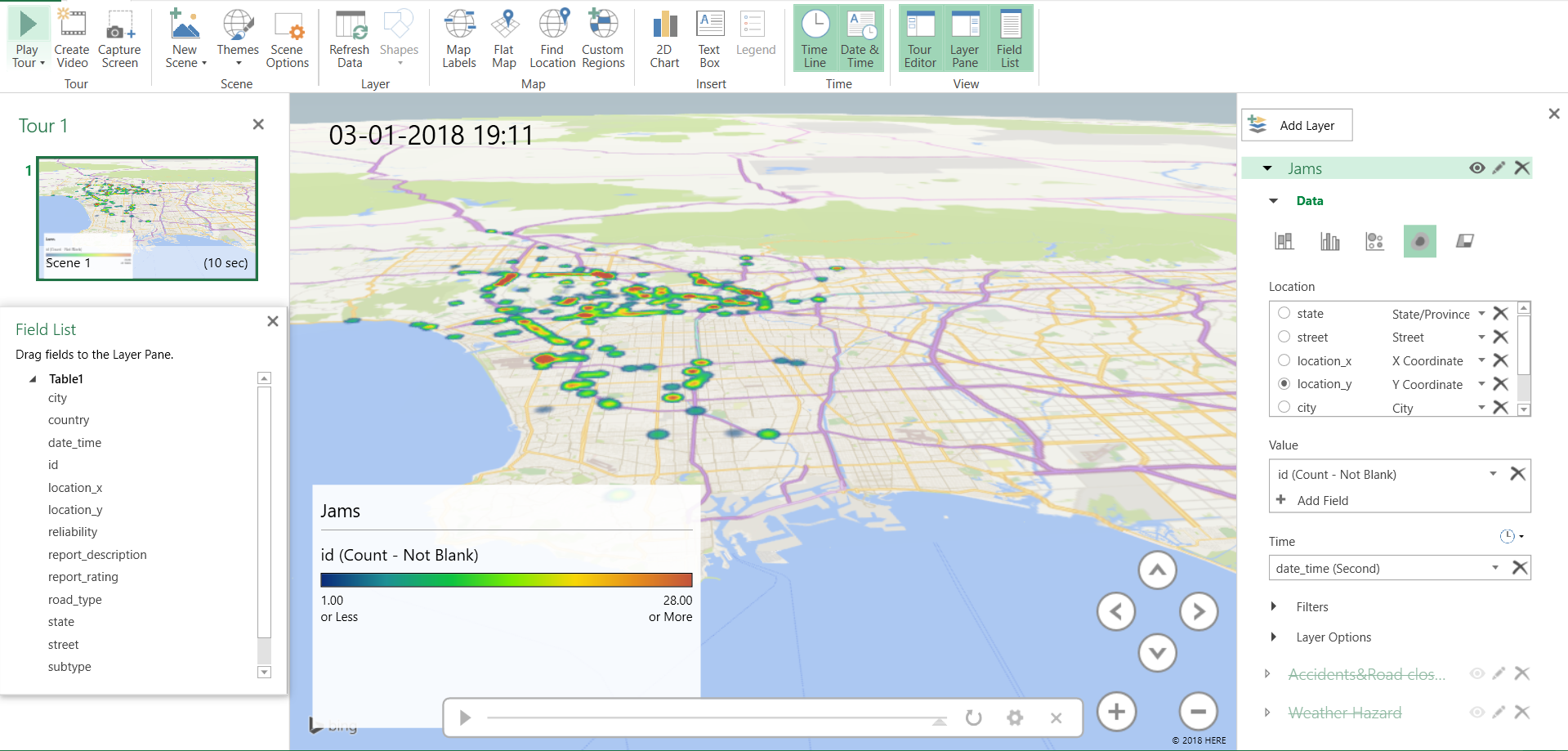
This step is to create visualization from cleaned files “alerts\_head\_clean.csv” and “irregularities\_head\_clean.csv” using excel 3D map and power BI. Alerts and Accidents on 3D map:

1. To visualize alerts and accidents of alerts\_head\_clean.csv on map, convert csv file to excel. Select all columns and then click on 3D Maps. Click on ‘Add Layers’ to add three layers for ‘Jams’,’Accidents&Road closures’ and ‘Weather’.

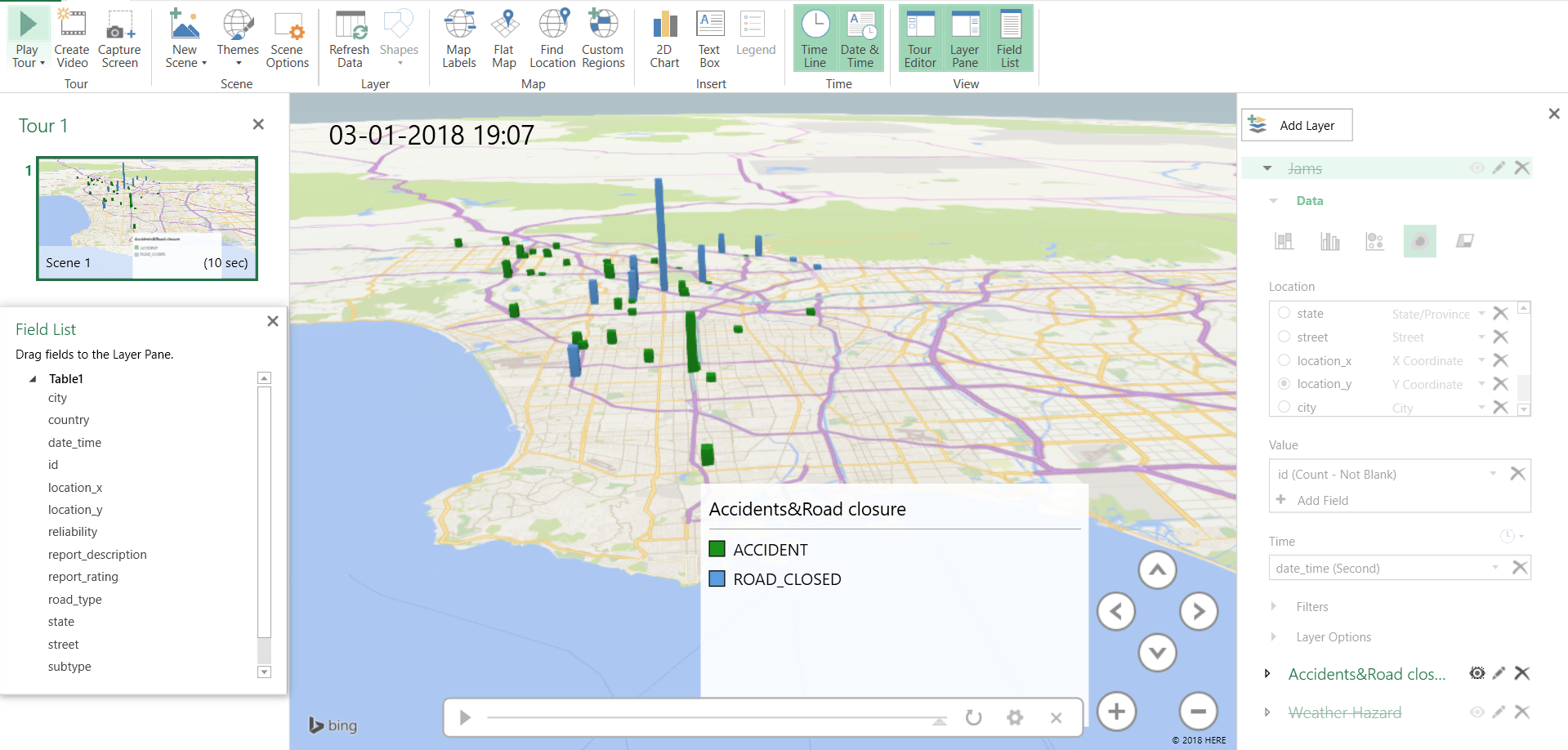
Follow below screenshots for the configuration used for each layer individually.



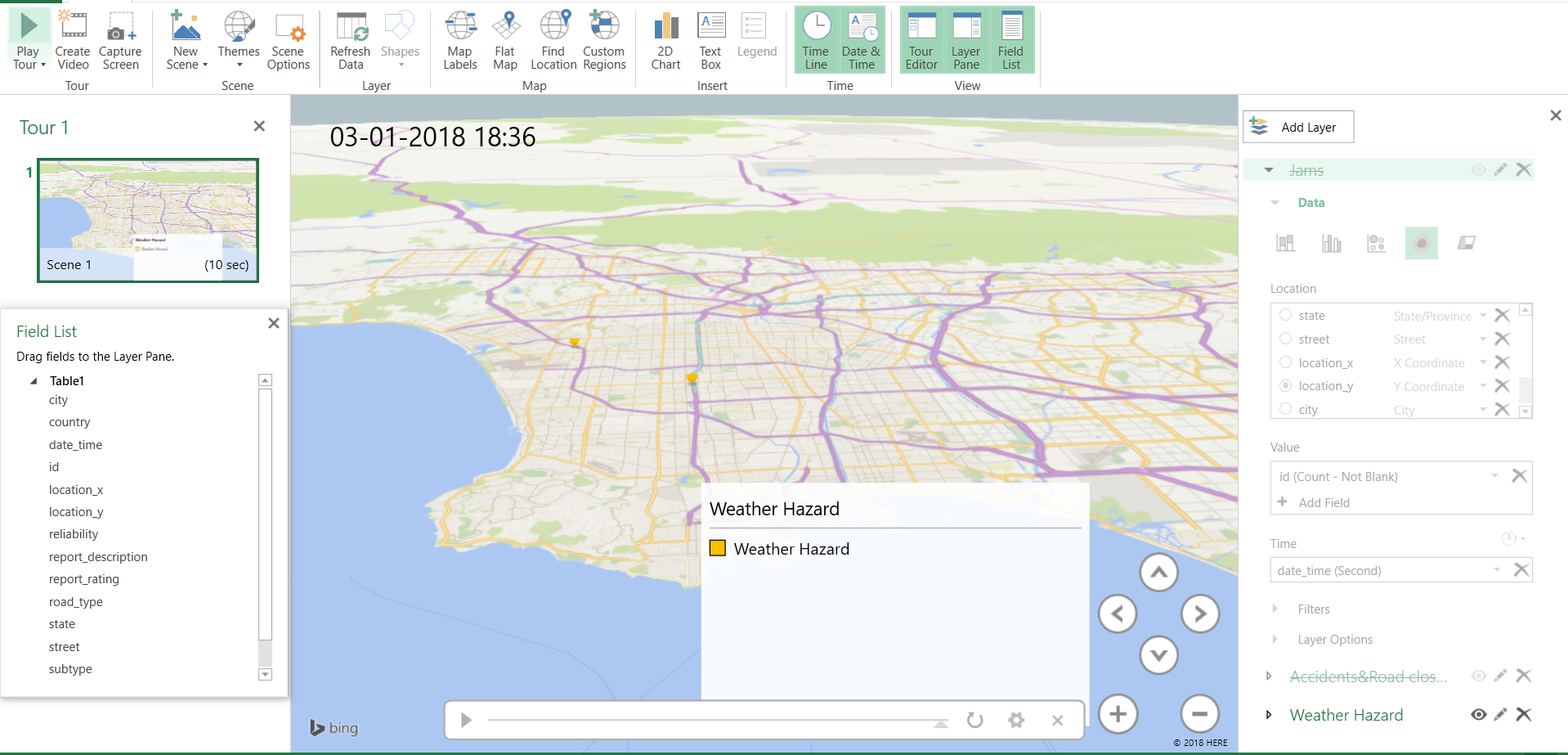
Jams



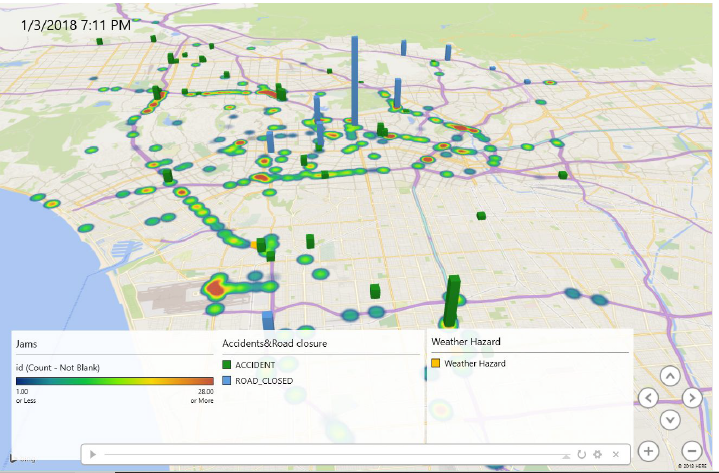
Accidents&Road closures



Weather

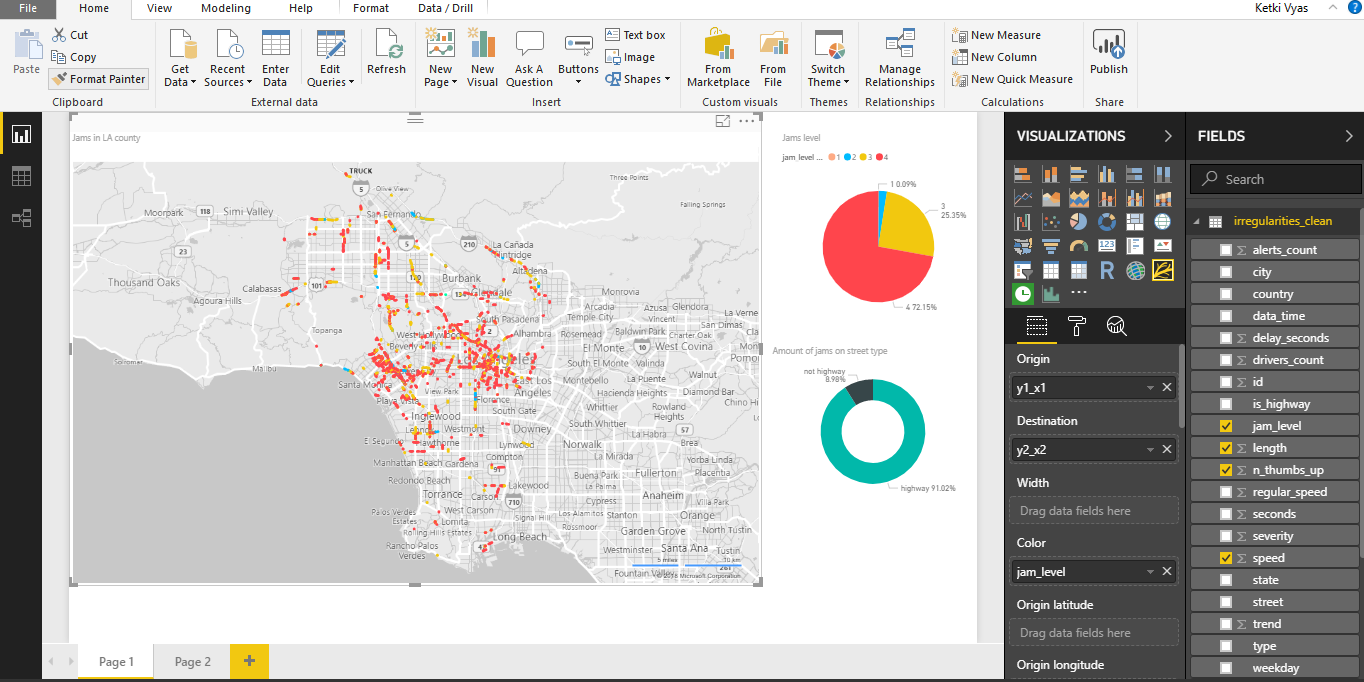


After adding all of the layers, map will display below visual.

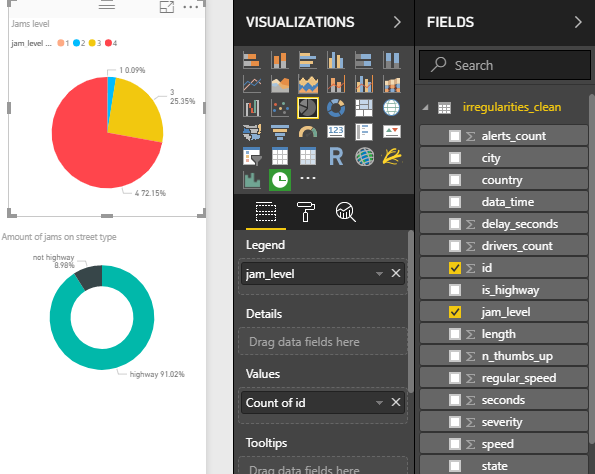


Traffic Jams Dashboard:

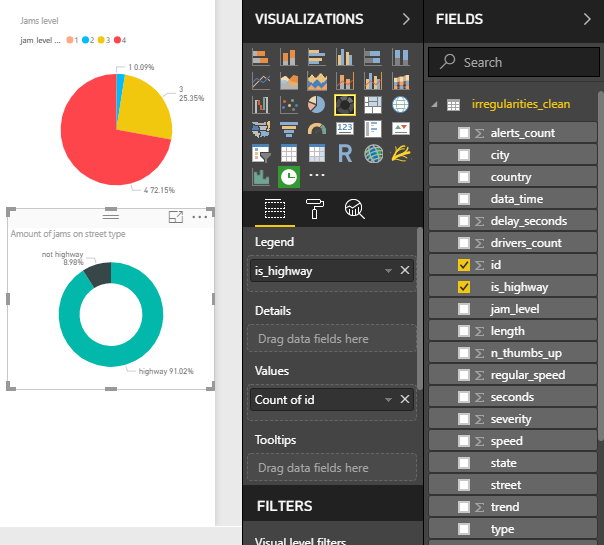
1. To visualize traffic jams of irregularities\_clean.csv on map. Use “Flow map” and select fields shown in the screenshot.



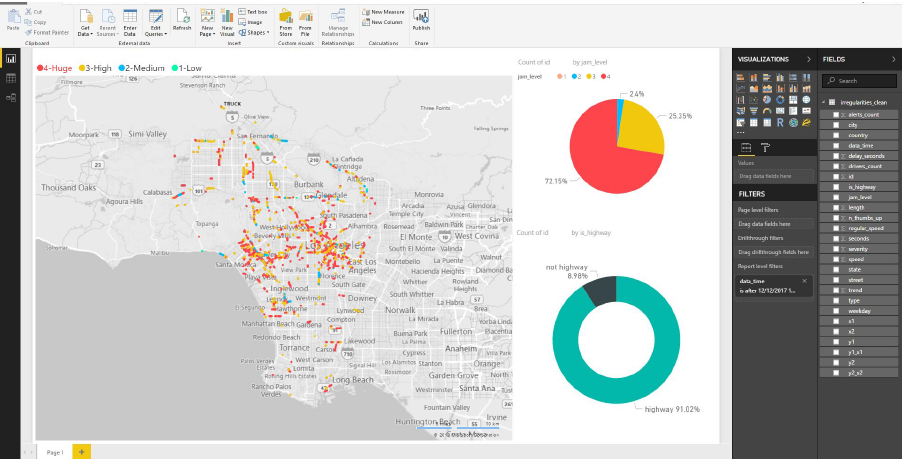
1. To visualize traffic jams levels, use Pie chart and select following fields as shown in the screenshot below.



1. To show the amount of Jams according to street type select donut chart and the fields as shown in the below screenshot.

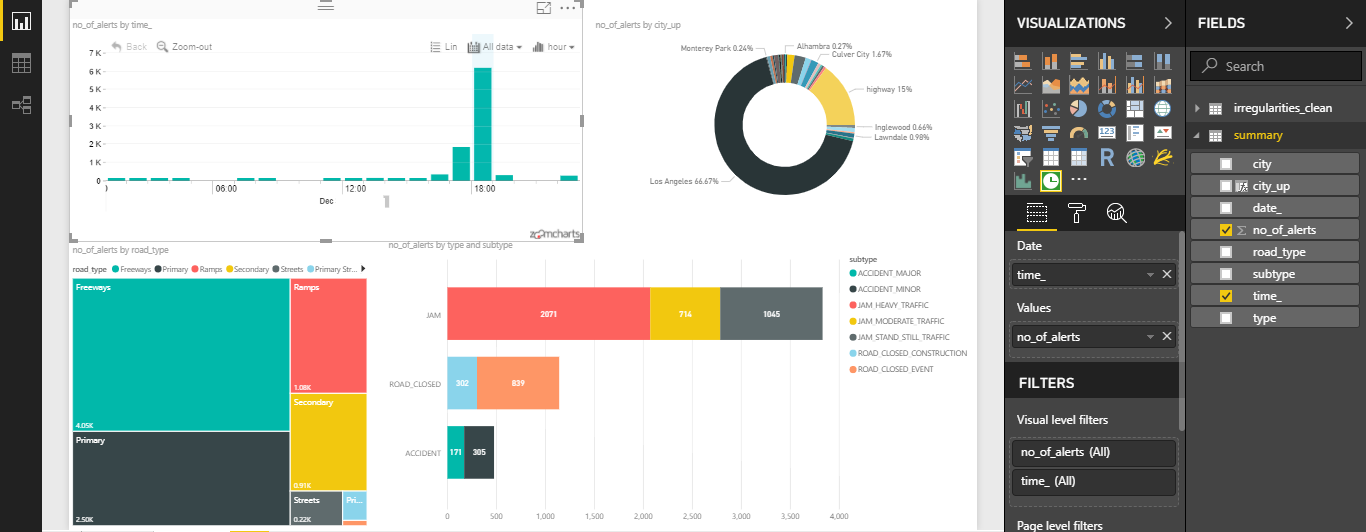


1. Final Dashboard

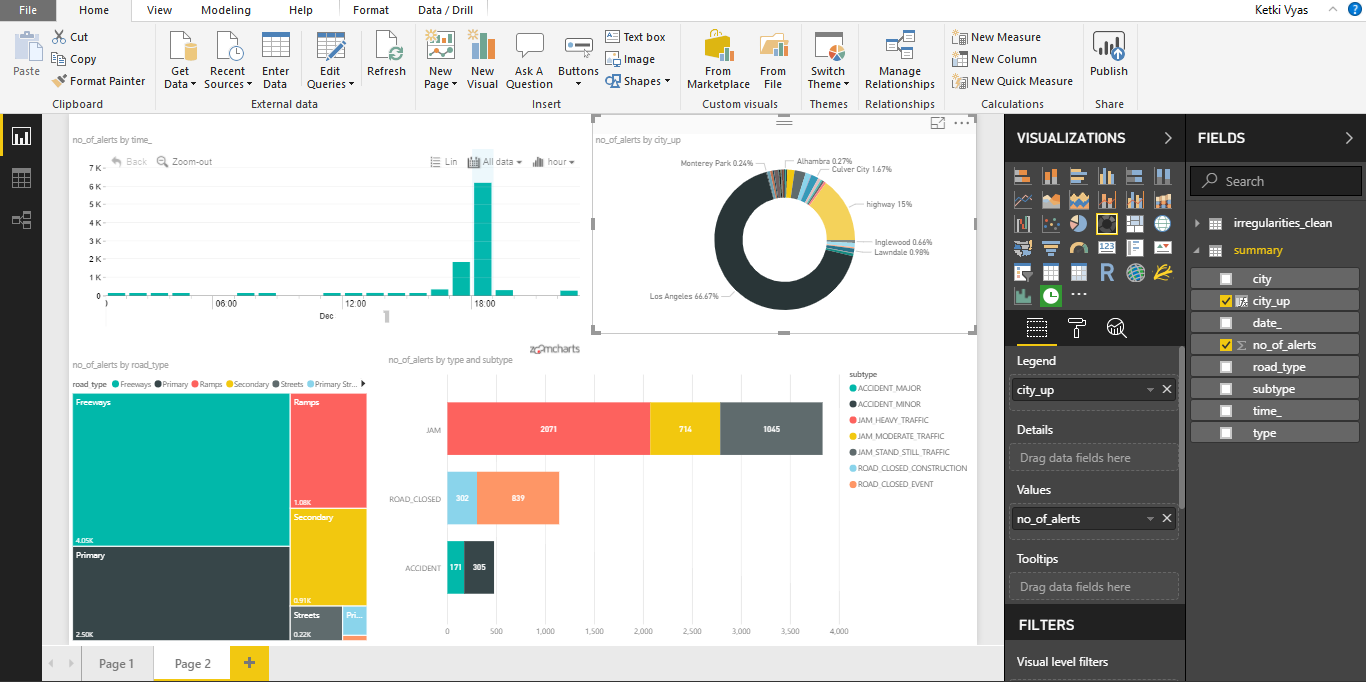


Alerts in LA County Dashboard:

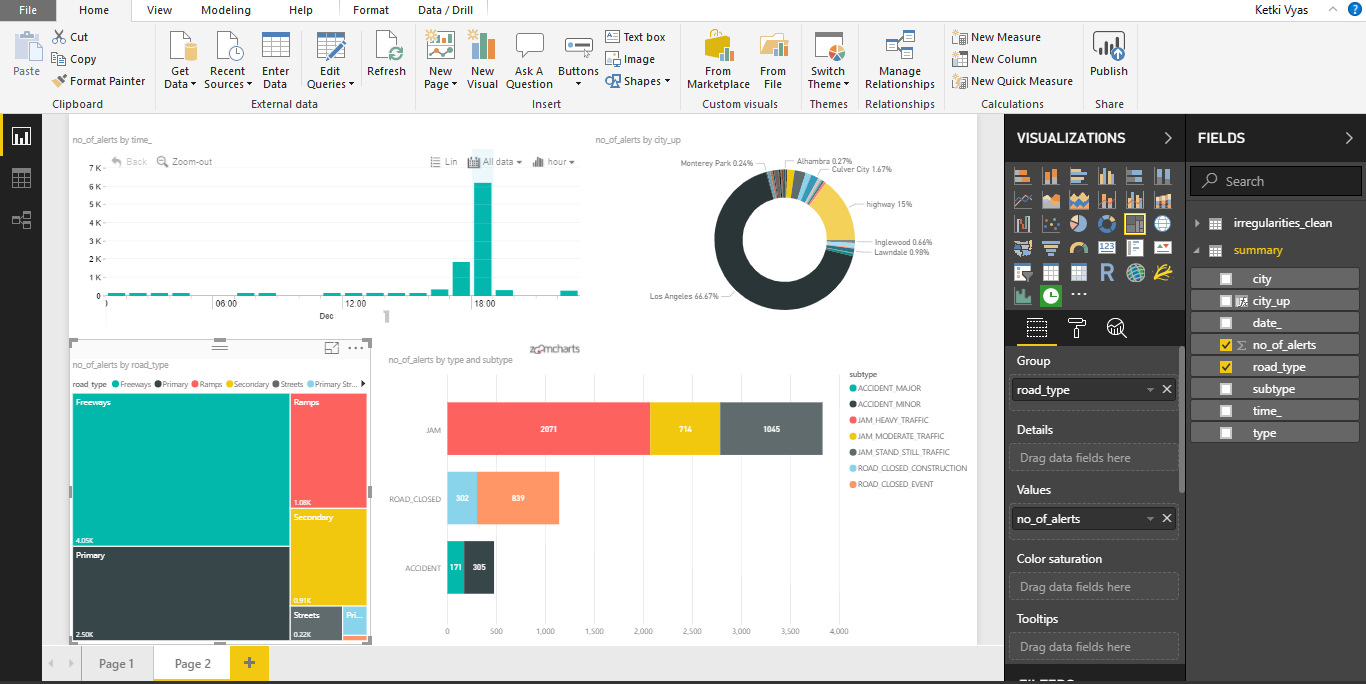
1. To visualize number of alerts by time using “alerts\_head\_clean.csv”. Select “Drilled down column chart for time-based chart” and select following columns shown in the screenshot.



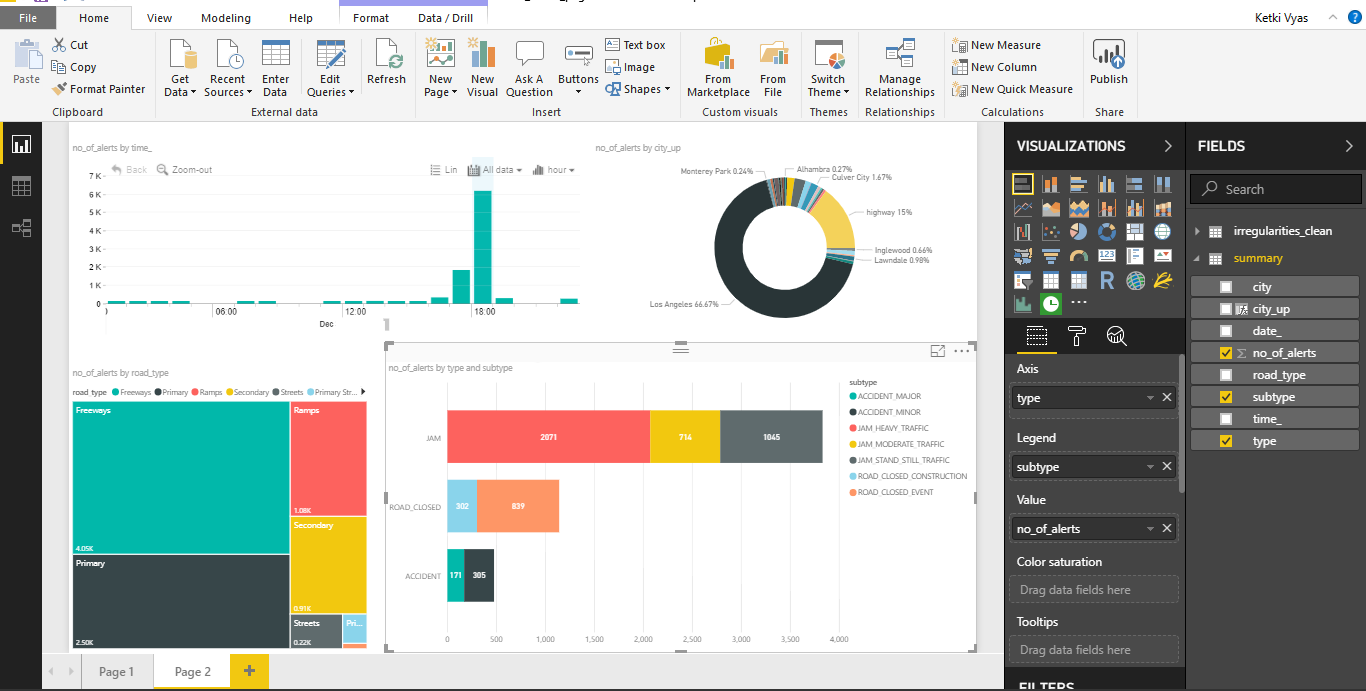
1. To show number of alerts by city select “Donut chart” and select columns as shown in the screenshot.



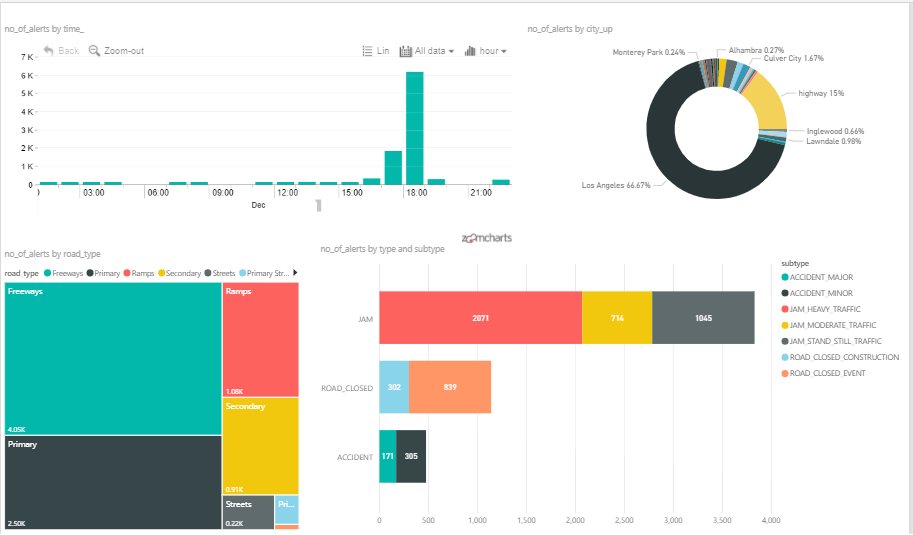
1. To show number of alerts by road type select Tree-map and columns as shown in the screenshots.



1. To show number of alerts by type and subtype select “Stacked Bar chart” and columns shown in the bellow screenshot.



1. Final Alerts Dashboard.



References

* 1. URL of Data Source- Data was received over email. Schema was used using

<https://github.com/CityOfLosAngeles/aqueduct/blob/master/waze/schema-waze.sql>

* 1. URL of your Github- <https://github.com/rjoshi5/waze_traffic_patterns_LosAngeles>
  2. URL of References