***HIVE Analysis and Representation of the UFO Sightings across USA***

**Objective**

The main objective here is to analyze and visualize UFO sightings using HIVE. The entire procedure in brief is as shown:

• Download UFO data to the local systems in Bluemix BigInsights.

• Then we upload it to HDFS using AMBARI.

• We manipulate and analyze UFO data in HDFS using HiveQL.

• We also visualize the result of various queries in Excel for a better understanding.

**Introduction**

The National UFO Reporting Center is in Seattle, WA. The Center's primary function over the past two decades has been to receive, record, and to the greatest degree possible, corroborate and document reports from individuals who have been witness to unusual, possibly UFO-related events. Here, in this project we have extracted and analyzed some of the data updated in this website regarding various ufo sightings.

**Prerequisites**

All that we need to write scripts and execute queries is already provisioned with the IBM Bluemix cluster. To export the analyzed data to Microsoft Excel, we must meet the following requirements:

• We must have Microsoft Excel 2010,2013 or 2016 installed.

• We must have Microsoft Hive ODBC Driver to import data from Hive into Excel. Select either the 32bit or 64-bit version based on your

**UFO sightings related data Loaded into BigInsights**

The ufo related data is downloaded overall from the official website using the link shown:

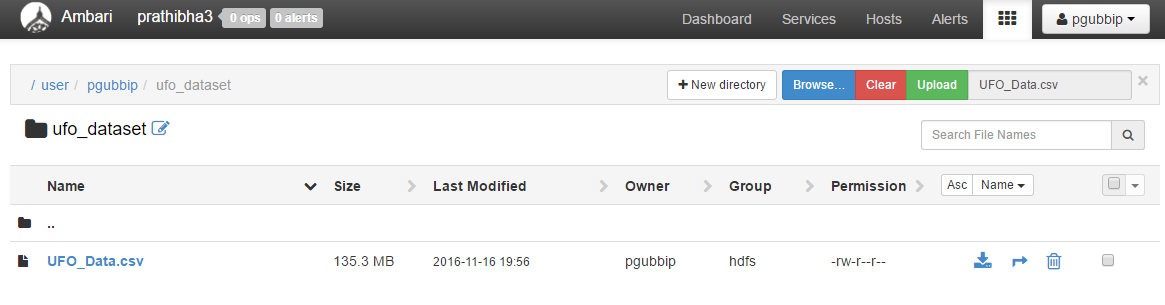
# http://www.nuforc.org/webreports/ndxevent.html

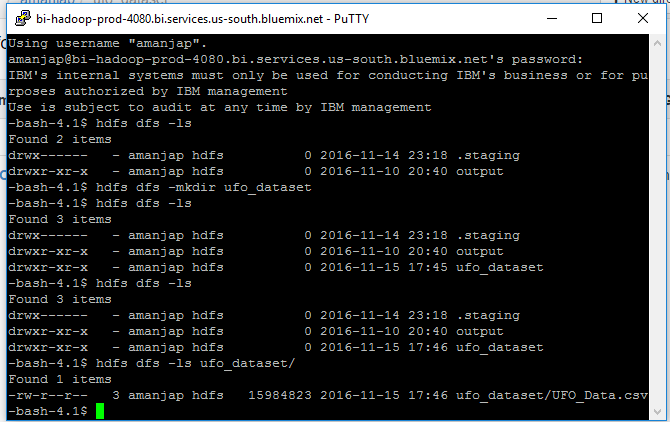
The ufo data present in HTML files are converted into a single CSV file. Here, UFO\_Data.csv is the input file which we need to upload to HDFS of the Hadoop cluster. A directory “ufo\_dataset” is created to place the dataset in it. UFO\_Data.csv is uploaded into to the directory ufo\_dataset in IBM Bluemix using Ambari as shown below:

$ hdfs dfs -ls

$ hdfs dfs -mkdir ufo\_dataset

$ hdfs dfs -ls ufo\_dataset/





**Creating Hive tables to Query UFO data**

Open the hive shell environment as follows:

$ hive

The following Hive statement creates an external table that allows Hive to query data stored in HDFS. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file. The Hive statement below creates a new table, named ufo\_data by describing the fields within the files, the delimiter (comma) between fields, and the location of the file in Azure Blob Storage. This will allow you to create Hive queries over your data.

$ DROP TABLE IF EXISTS ufo\_data;

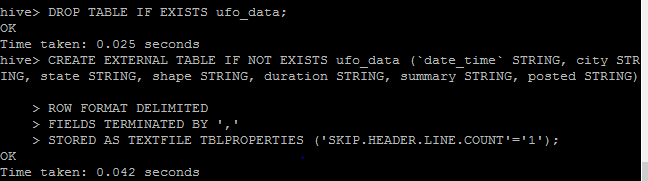
$ CREATE EXTERNAL TABLE IF NOT EXISTS ufo\_data (`date\_time` STRING, city STRING, state

STRING, shape STRING, duration STRING, summary STRING, posted STRING)

ROW FORMAT DELIMITED

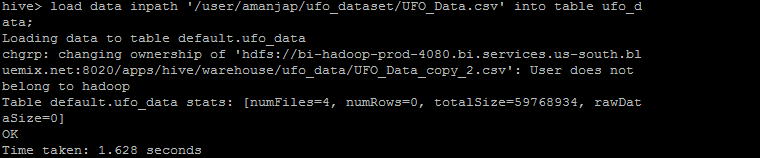
FIELDS TERMINATED BY ','

STORED AS TEXTFILE TBLPROPERTIES ('SKIP.HEADER.LINE.COUNT'='1');



The data is then loaded into the table we just created as shown:

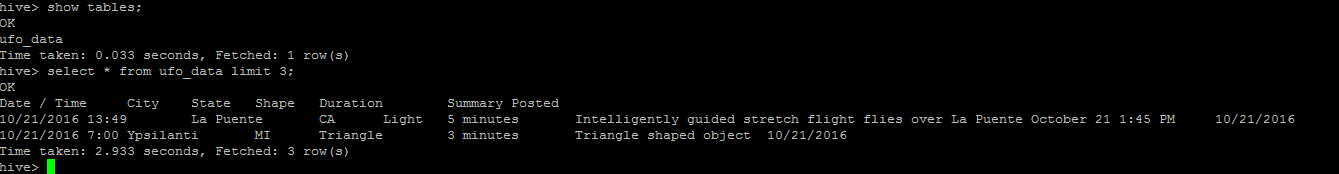
$ LOAD DATA INPATH '/user/amanjap/ufo\_dataset/UFO\_Data.csv' INTO TABLE ufo\_data;



Then, in the hive shell, we need to check if the table “ufo\_data” present along with the data we loaded by the following commands:

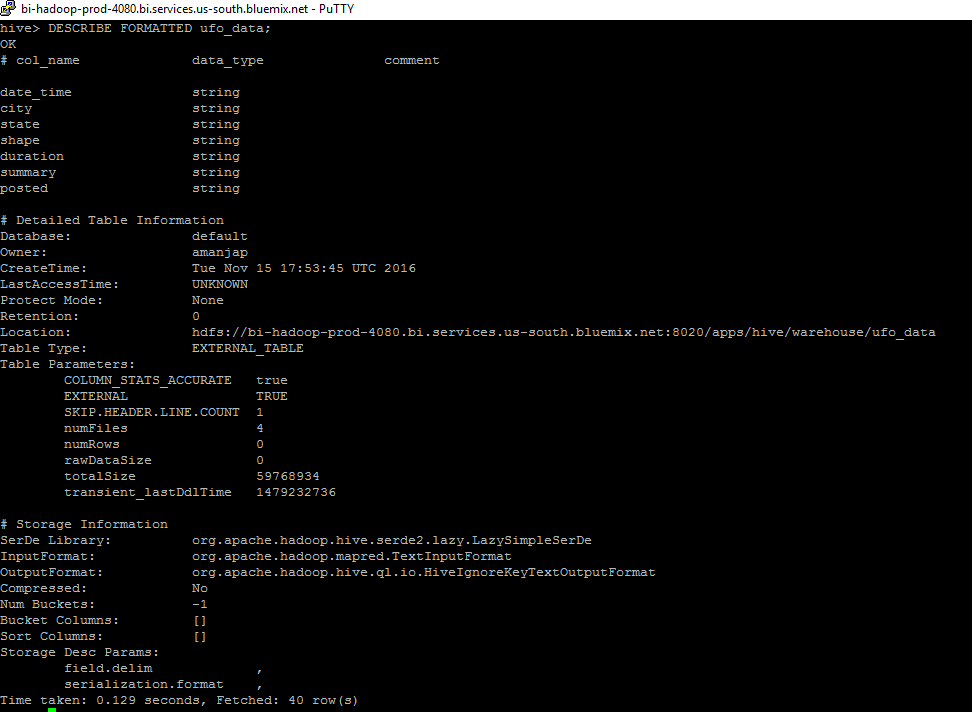
$ SHOW TABLES;

$ SELECT \* FROM ufo\_data LIMIT 3;



The detailed description of the table format, ufo\_data is displayed by the following command:

$ DESCRIBE FORMATTED ufo\_data;



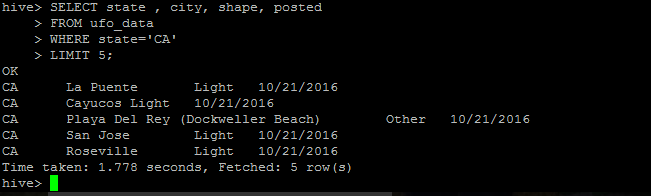
**Creating Hive Queries to Analyze Data**

We can now query the table as shown below. We have queried here to display the entities state, city, shape and posted from the table ufo\_data associated with the state entity being equal to “CA “.

$ SELECT state, city, shape, posted

FROM ufo\_data

WHERE state=”CA” LIMIT 5;



**Creating more tables by keeping ufo\_data as the base table**

We now create more tables keeping the base table ufo\_data as reference as shown below :

TABLE1:

$ DROP TABLE IF EXISTS ufoperyear;

$ CREATE EXTERNAL TABLE IF NOT EXISTS ufoperyear (year INT, shape STRING, ufo\_cnt INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/amanjap/ufo\_dataset/ufoperyear/';

$ INSERT OVERWRITE TABLE ufoperyear

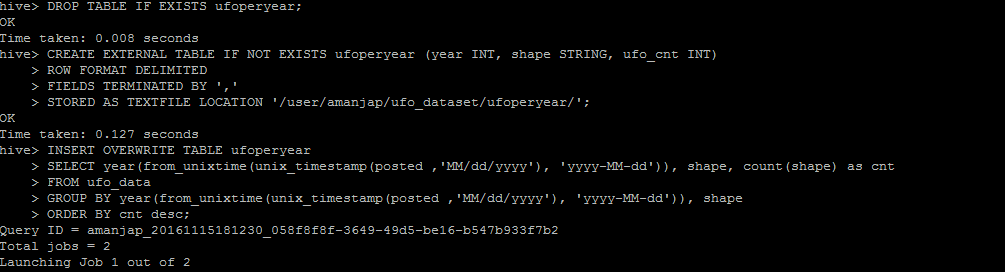
SELECT year (from\_unixtime

(unix\_timestamp (posted,'MM/dd/yyyy'), 'yyyy-MM-dd')), shape, count(shape) as cnt

FROM ufo\_data

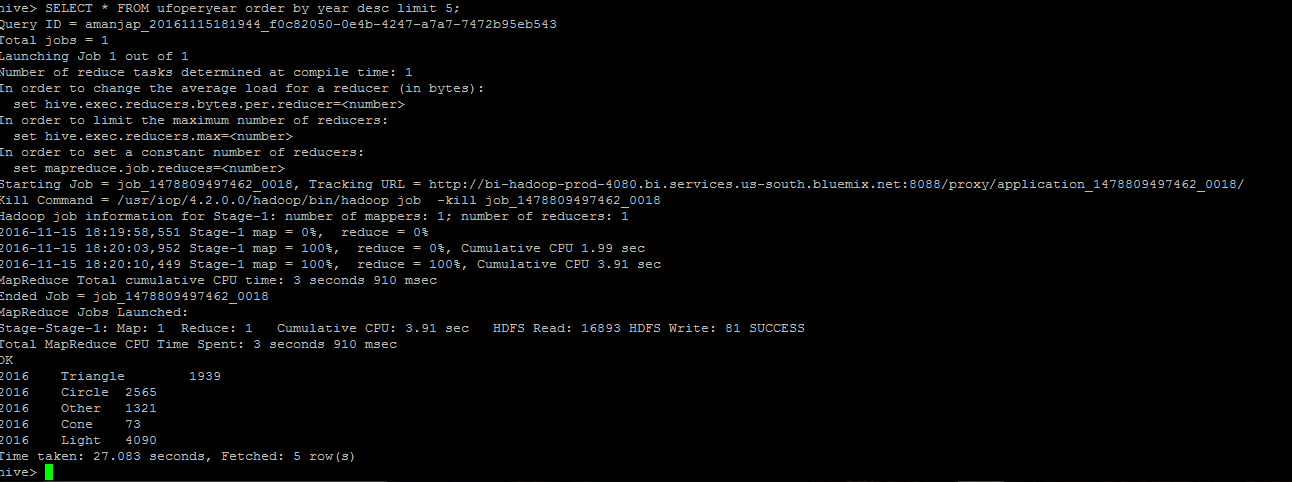
GROUP BY year (from\_unixtime (unix\_timestamp(posted ,'MM/dd/yyyy'), 'yyyy-MM-dd')), shape

ORDER BY cnt DESC;



The created table is now queried as shown below:

$ SELECT \* FROM ufoperyear ORDER BY year DESC LIMIT 5;



$ SELECT \* FROM ufoperyear WHERE shape IN ('Light', 'Circle') AND year > 2010;

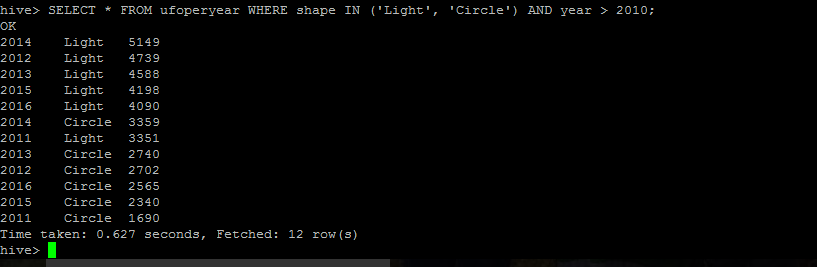


TABLE2:

Here we create table ufo\_duration\_analysis where we segregate the ufopostings according to the duration for which it was visible. If it was visible only for a few seconds, we put it under the category HARDLY VISIBLE, if it was visible for a few minutes, we put it under the category VISIBLE and if it was visible for a few hours, we put it under the category PRMONENTLY VISIBLE. Anything other than that is put under the category, AMBIGOUS.

$ CREATE TABLE ufo\_duration\_analysis

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/amanjap/ufo\_dataset/'

AS

SELECT duration, state, date\_time,

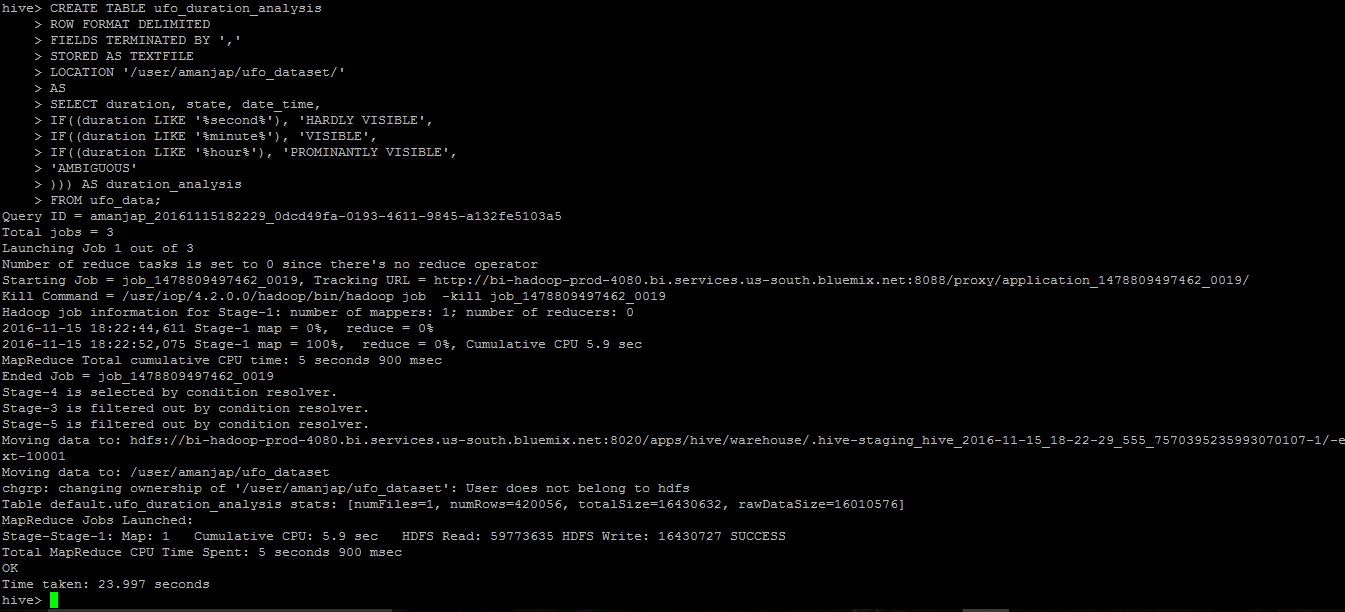
IF((duration LIKE '%second%'), 'HARDLY VISIBLE',

IF((duration LIKE '%minute%'), 'VISIBLE',

IF((duration LIKE '%hour%'), 'PROMINANTLY VISIBLE',

'AMBIGUOUS'))) AS duration\_analysis

FROM ufo\_data;



The table ufo\_duration\_analysis is queried as shown below:

$ SELECT \* FROM ufo\_duration\_analysis LIMIT 10;

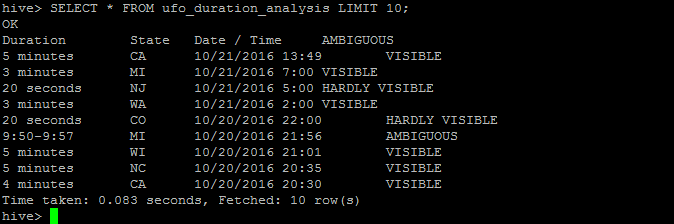


TABLE3:

This table is created to filter out all the records related to the shape “LIGHT”, so that the records of this category can be queried and analyzed separately.

$ CREATE TABLE UFO\_byshapeLight

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

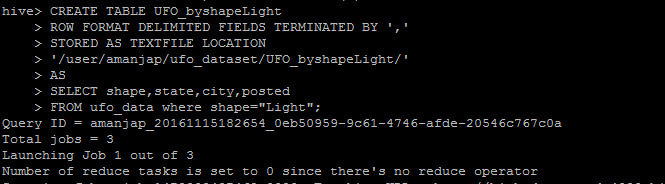
STORED AS TEXTFILE LOCATION

'/user/amanjap/ufo\_dataset/UFO\_byshapeLight/'

AS

SELECT shape,state,city,posted

FROM ufo\_data where shape="Light";



The table UFO\_byshapeLight is queried as shown below:

$ Select \* from UFO\_byshapeLight limit 5;

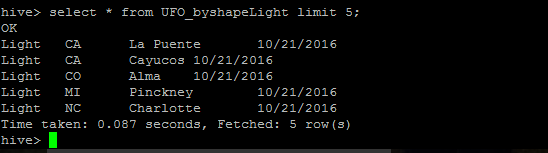


TABLE4:

This table is created to filter out all the records related to the shape “DIAMOND ”, so that the records of this category can be queried and analyzed separately .

$ CREATE TABLE UFO\_byshapeDiamond

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

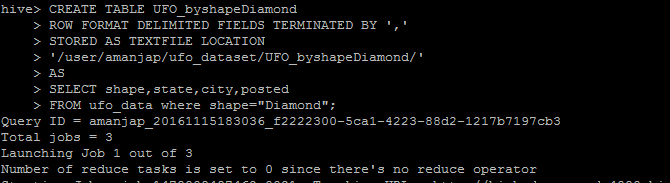
STORED AS TEXTFILE LOCATION

'/user/amanjap/ufo\_dataset/UFO\_byshapeDiamond/'

AS

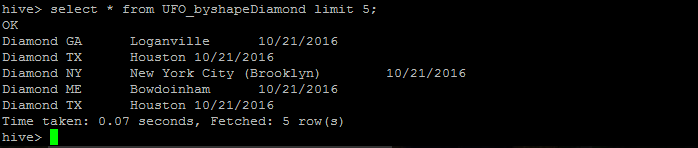
SELECT shape,state,city,posted

FROM ufo\_data where shape="Diamond";



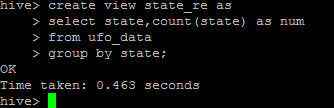
The table UFO\_byshapeLight is queried as shown below:

$ Select \* from UFO\_byshapeDiamond limit 5;



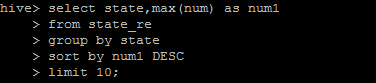
A view is created to analyze data based on the state and total number of UFO’s appeared in that state.

|  |
| --- |
| $ drop view if exists state\_re;  create view state\_re as  select state,count(state) as num  from ufo\_data  group by state; |

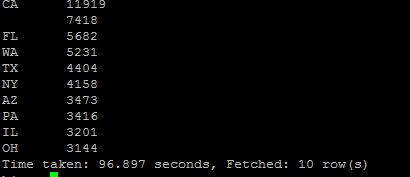


Query to find in which top ten states the UFO’s have appeared mostly

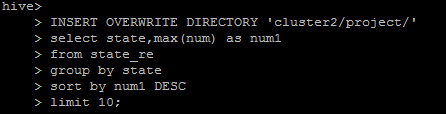
|  |
| --- |
| $ select state,max(num) as num1  from state\_re  group by state  sort by num1 DESC  limit 10; |

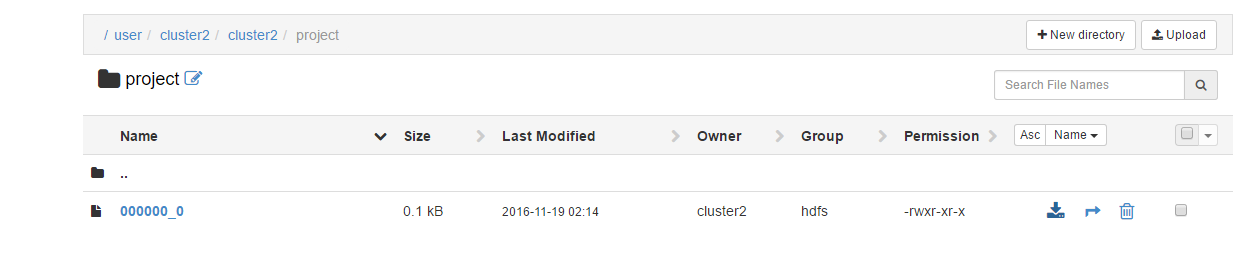


Output:



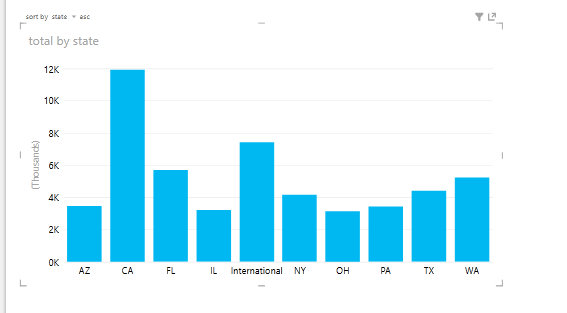
Query to generate sightings for the above query using power view and 3D maps.



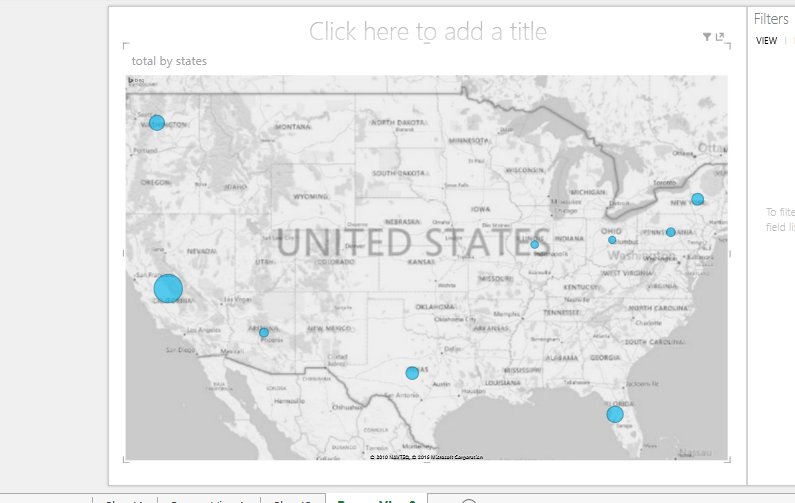


The file is generated in ambari with name 000000\_0, import the following file into Excel in order to analyze the output and generate graphs, 3D-maps as follows;

Bar Graph:

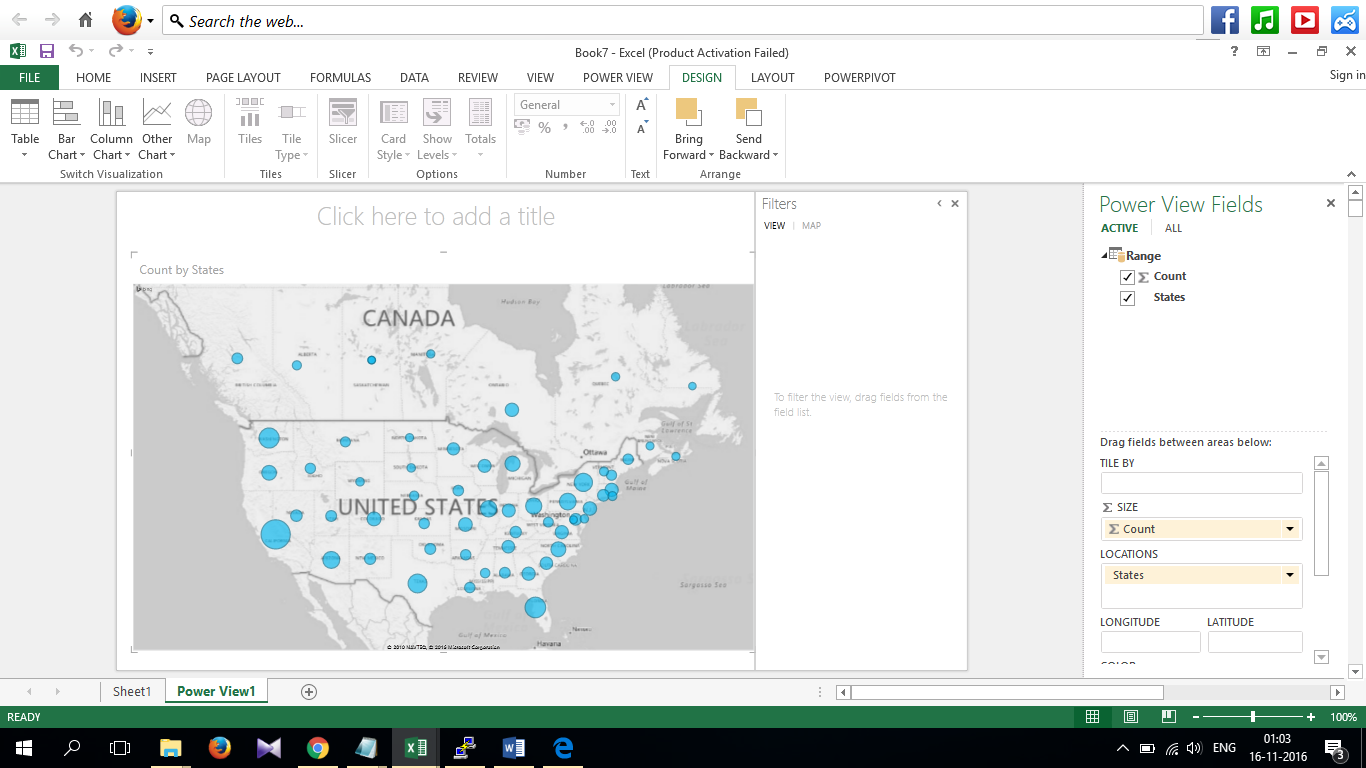


3D-map:



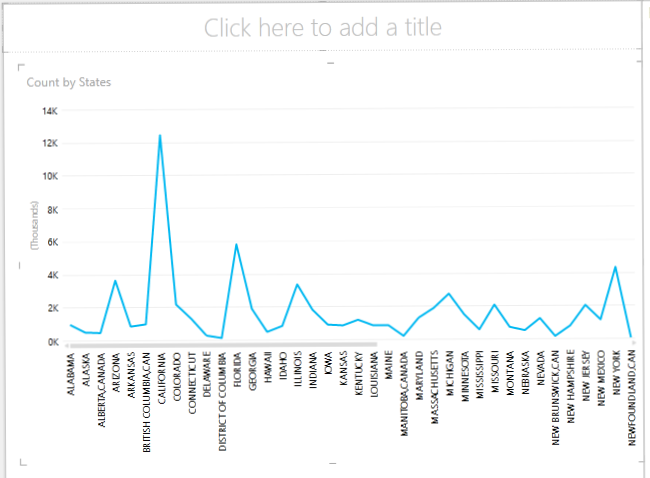
The sighting of all states and total number of UFO’s appeared is as follows:

3D-map



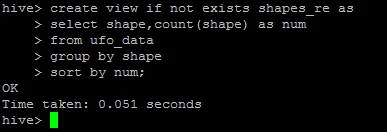
Graph:

This curve shows the peak edge for state California where maximum number of UFO’s have appeared.



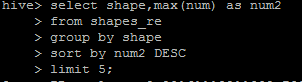
A view is created to analyze data based on the state and total number of UFO’s appeared in that particular shape.

|  |
| --- |
| $ drop view shapes\_re;  create view if not exists shapes\_re as  select shape,count(shape) as num  from ufo\_data  group by shape  sort by num; |

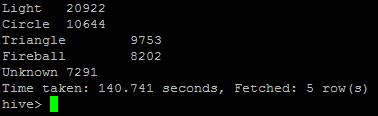


Query to show five UFO shapes appeared maximum and number of times it had appeared.

|  |
| --- |
| $ select shape,max(num) as num2  from shapes\_re  group by shape  sort by num2 DESC  limit 5; |

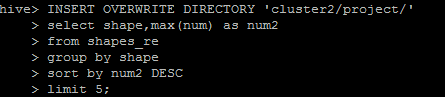


Output:

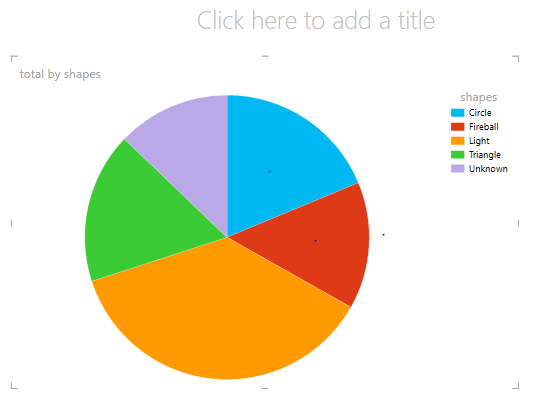


Query to generate sightings for the above query using power view to analyze output.

|  |
| --- |
| $ INSERT OVERWRITE DIRECTORY 'cluster2/project/'  select shape,max(num) as num2  from shapes\_re  group by shape  sort by num2 DESC  limit 5; |



PIE Chart for the 5 shapes that have appeared maximum times:

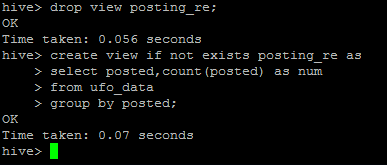


Bar Graph for all shapes and total number of times it had appeared:



View to retrieve five dates on which maximum postings are made by people based on UFO sightings appeared.

|  |
| --- |
| $ drop view posting\_re;  create view if not exists posting\_re as  select posted,count(posted) as num  from ufo\_data  group by posted; |

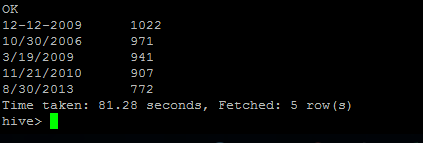


Query to retrieve five dates on which maximum postings are made by people based on UFO sightings appeared.

|  |
| --- |
| hive>select \* from posting\_re  sort by num DESC  limit 5; |

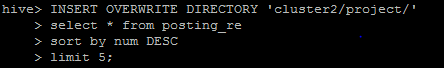
C:\Users\USER\Desktop\project\5postings.PNG

Output:



Query to generate sightings for the above query using power view to analyze output.

|  |
| --- |
| $ INSERT OVERWRITE DIRECTORY 'cluster2/project/'  select \* from posting\_re  sort by num DESC  limit 5; |



Graph: Maximum 5 postings

