Title

Big Data Project

Author: Aditya Verma

Analyzing data on International Visas

Date: 11/29/2015

Guidance: Prof Jose Perez-Carballo

# Abstract

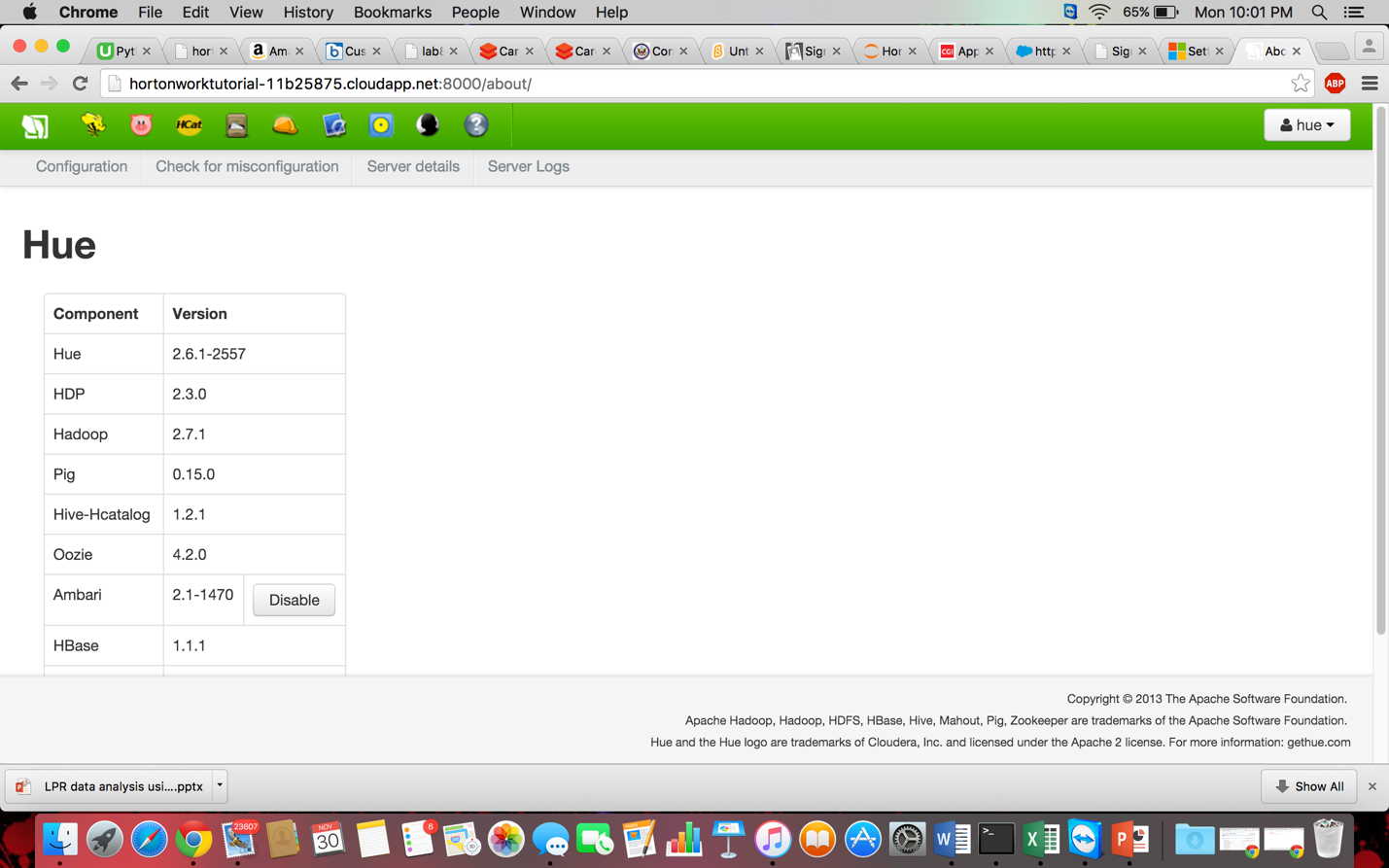
My paper is on analyzing large sets of data which are based on visa information of all the international students which come to united states for study purpose. These data are taken from the website <https://nces.ed.gov/ipeds/datacenter/> which is real data. Based on this data we have analyzed the number of students which come to united states every year and how each student helps or contribute in the economy of the country and in its revenue. This revenue is based on each state and the contribution of the students in each state. These revenues are basically from the students tuition fee and living expenses generated by the universities in each state of America. The data analysis will show that which state has the highest number of international students and which state generates highest revenue corresponding to the international students they have every year. These data sets are generally unstructured and present in very large numbers. This data is analyzed using especial algorithms which are specifically designed for these kind of large data sets.

# Introduction

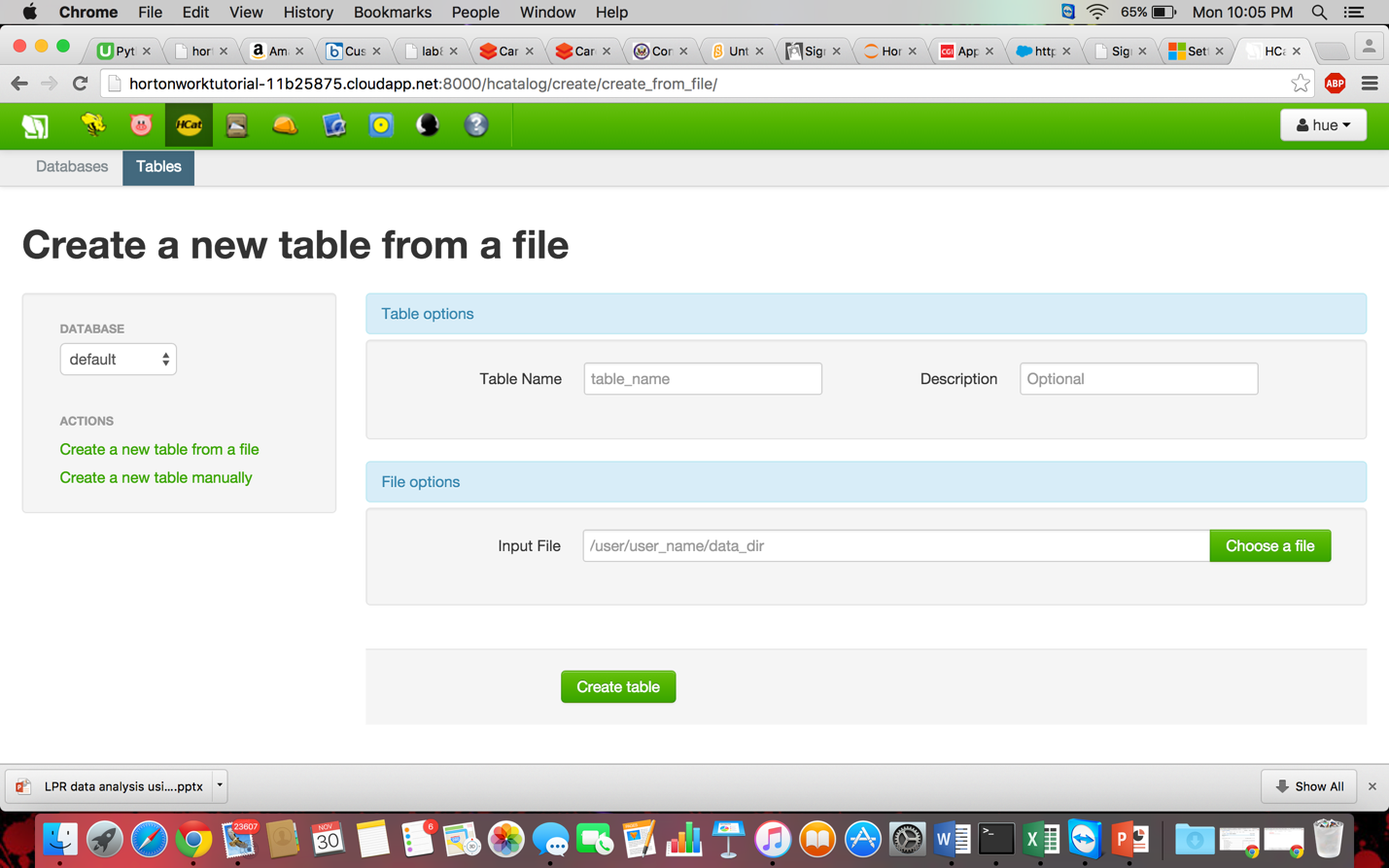
As data in the world has been increasing at a very fast pace. This large amount is given a term called **big data**. Big IT cum business companies like Facebook, twitter, Instagram generate hundreds of tera bytes of data. Facebook being the largest among them. This large amount of data is used in lot of real time analysis know as sentiment analysis which is generally based on the type of choices one makes while choosing for a particular product or a type of thing they follow or choose. This kind of analysis is basically done by the companies like amazon and twitter which are driven mostly by the type of people you follow and the products you tend to choose or search once you are log into your amazon account. Hence to analyze such amount of data called big data and to develop algorithms, we use use term HDFS. HDFS stands for HADOOP DISTRIBUTED FILE SYSTEM. We used this system to store relatively large amount of data in the Hadoop file system which can be retrieved from anywhere by login in to your local Hadoop file system. In this project we have shown some analysis using some query languages.

# Processing Data:

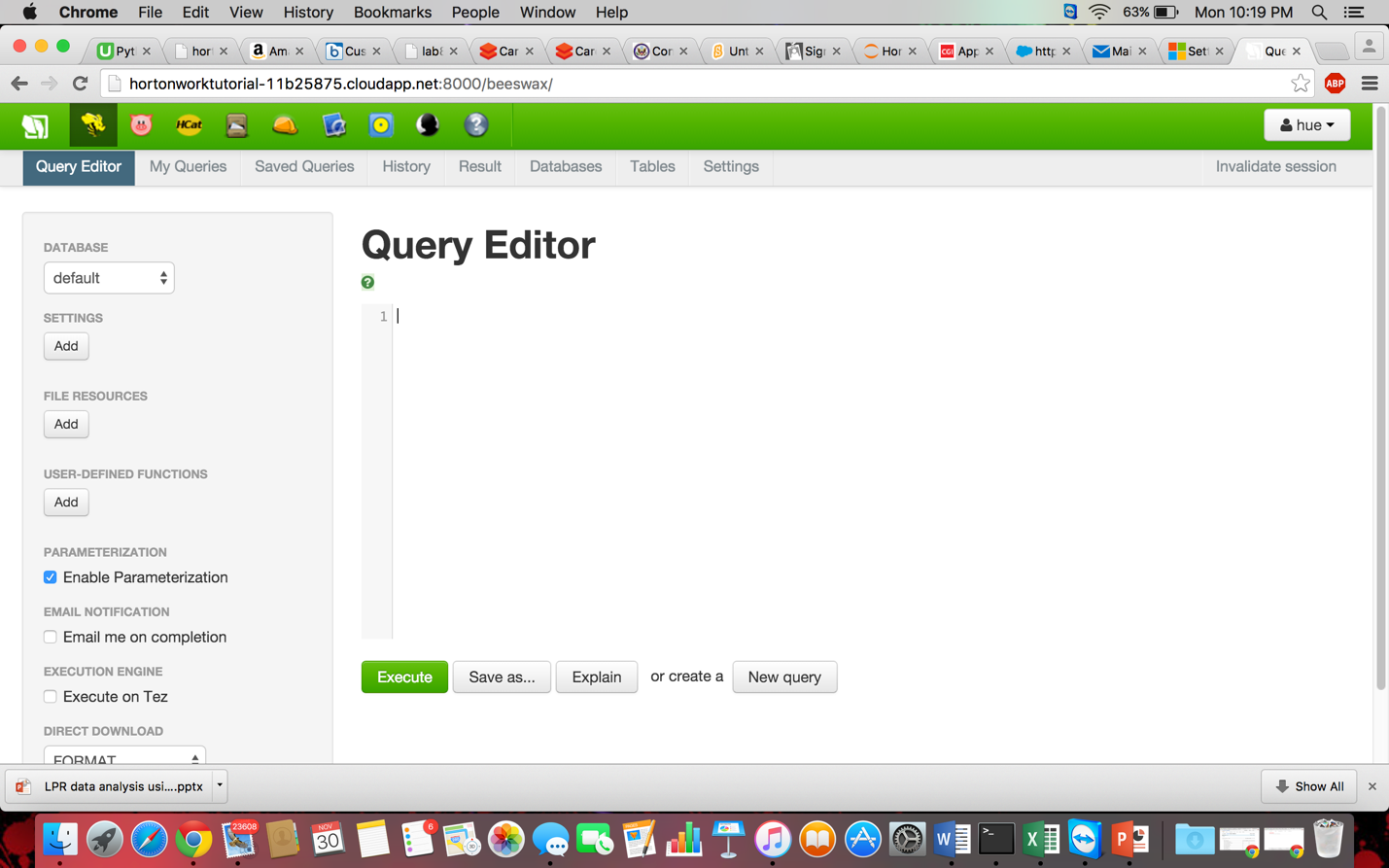
My Big data project is based on analyzing data based on revenue generated by the international students in each state. We calculate what and how much revenue is generated in each state and how much they contribute to the economy of each state. Here we have used a combination of big data query languages like pig, hive, Pyhive for the purpose of analysis. The problem using the normal SQL languages in such hive amount data is that, it Is out of scope or range of the language and the language is not efficient enough to generate the output, hence to use languages which are not sequential in nature because the data is unstructured and needs to be given the output in the form of structure.

Steps involved in the project: The real time data is first downloaded form the website <https://nces.ed.gov/ipeds/datacenter/> into a .zip format data being of very large size then it is unzipped it and saved it into a .csv format called ( comma separated format ). This data is then uploaded to Github or Hue which is on order to retrieve or make tables of the data that data. 

In this you can create a new table form a file using HUE which is available on the port number 8000 of your HortonWorks Hadoop system. Which can be opened in the browser window by giving using the URL : <http://hortonworktutorial-11b25875.cloudapp.net:8000/hcatalog/create/create_from_file/>



This link directly opens the page where you can create HIVE tables uploading the CSV file from Local system. This procedure take up some time in order to create proper formatted tables.

This is the Query Editor of the Hue where one can fire the respective queries related to the hive table which we formed in the last step. After this we save the hive queries in the SQL format with the format .SQL which will be run the python code. This can be done by importing the required libraries in order to run **input**.**sql** files on the python platform. Here we use a special combination of languages called **Pyhive** which is a combination of scripting language like python and query and language like Hive. As python is a powerful and flexible language and integrating this language which the hive query langue will increase the ability of the language to a great extent, Thus incurring all the properties of both the language.

In this we generally write codes on **Jupyter** which is a python code editor. This editor is generally opened by accessing your port number 8889, this port number

is generally preceded by the user name of my Hadoop cluster. Shown below is the **jupyter** window where I wrote the **Pyhive** code. The following generates the output. Now as depicted in the pic if you see the query used. Its says cmd =”hive –f ‘input.sql’”. on the Conn I gave the connection to the localhost mentioning the port number and the username which connects to the hive port. After the execution of this query we have a table generated at the HUE which is the port number 8000 , from there we are able to download the format of that file in the form of comma separted file.

These are the queries used to make table form the selected table depending upon the preferences of the data which we need to fetch or make the output from.

**QUERIES USED TO GENERATE TABLES FROM THE EXISTING TABLES**

1. make 2 tables from 2 csv files college\_Exp\_2013 and enrollments\_2013

2. CREATE TABLE Get\_the\_Fees row format delimited fields terminated by ',' STORED AS RCFile

AS select unitid, institution\_name, drvic2013\_rv\_tuition\_and\_fees\_2013\_14,

(drvic2013\_rv\_total\_price\_for\_out\_of\_state\_students\_living\_on\_campus\_2013\_14 - drvic2013\_rv\_tuition\_and\_fees\_2013\_14) AS Living\_Cost\_ON\_Campus, drvic2013\_rv\_total\_price\_for\_out\_of\_state\_students\_living\_off\_campus\_not\_with\_family\_2013\_14 AS Living\_Cost\_Off\_Campus\_Without\_Family,

drvic2013\_rv\_total\_price\_for\_out\_of\_state\_students\_living\_off\_campus\_with\_family\_2013\_14 AS Living\_Cost\_Off\_Campus\_With\_Family,

hd2013\_state\_abbreviation AS State, hd2013\_level\_of\_institution AS Level\_of\_institution

from college\_exp\_2013;

3. CREATE TABLE Fees row format delimited fields terminated by ',' STORED AS RCFile

AS select Get\_the\_fees.unitid, enrollments\_2013.institution\_name, Get\_the\_fees.state, enrollments\_2013.drvef2013\_total\_enrollment,

drvic2013\_rv\_tuition\_and\_fees\_2013\_14, Living\_Cost\_Off\_Campus\_Without\_Family, Living\_Cost\_Off\_Campus\_With\_Family

from Get\_the\_fees JOIN enrollments\_2013 on (Get\_the\_fees.institution\_name = enrollments\_2013.institution\_name);

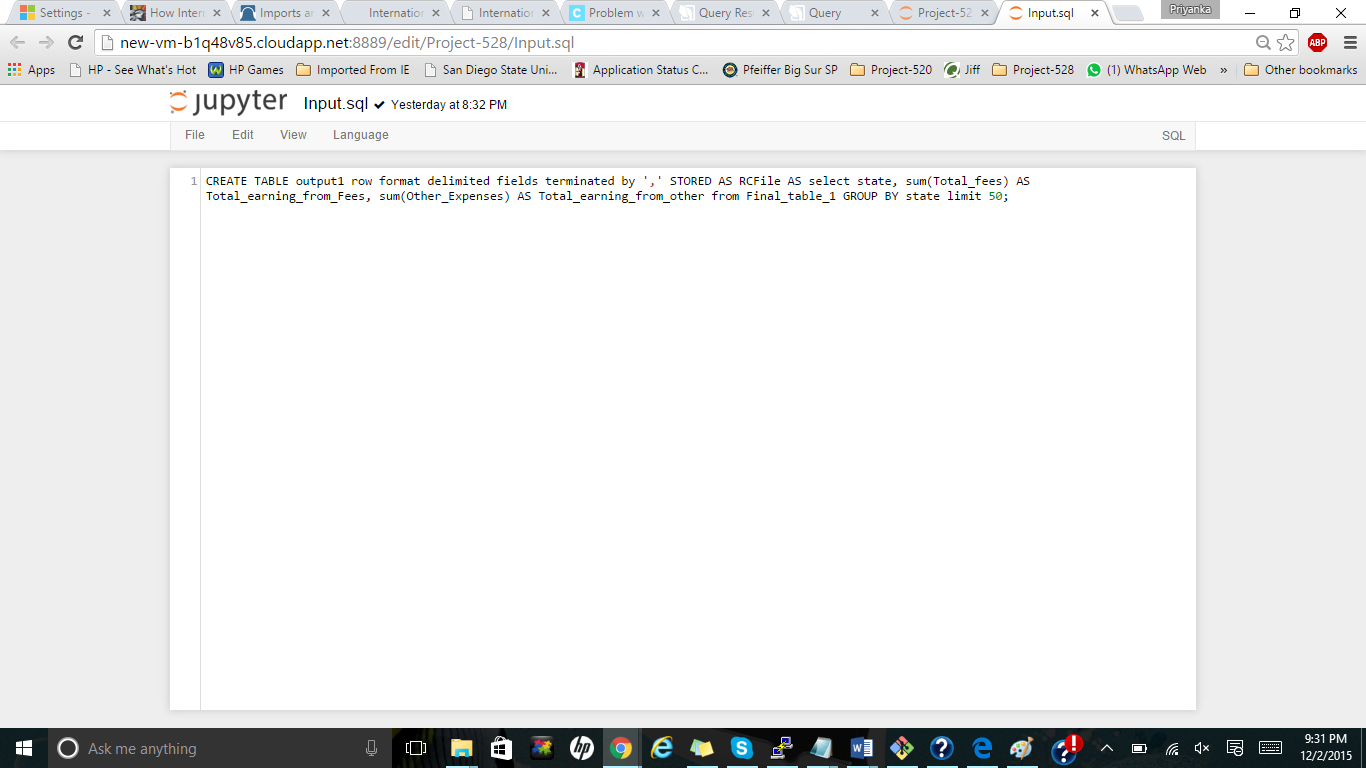
4. CREATE TABLE Final\_Table\_1 row format delimited fields terminated by ',' STORED AS RCFile

AS select institution\_name,State, (drvef2013\_total\_enrollment \* drvic2013\_rv\_tuition\_and\_fees\_2013\_14) AS Total\_fees,

(Living\_Cost\_on\_Campus + Living\_Cost\_Off\_Campus\_Without\_Family + Living\_Cost\_Off\_Campus\_With\_Family)

AS Other\_Expenses from fees;

5. select state, sum(Total\_fees) AS Total\_earning\_from\_Fees, sum(Other\_Expenses) AS Total\_Earning\_from\_Other from Final\_table\_1 GROUP BY state; --Run this in input.sql file in python



The above input.sql code will run on the python code on the jupyter which will give the final output as discusses in the hue and download the .csv file

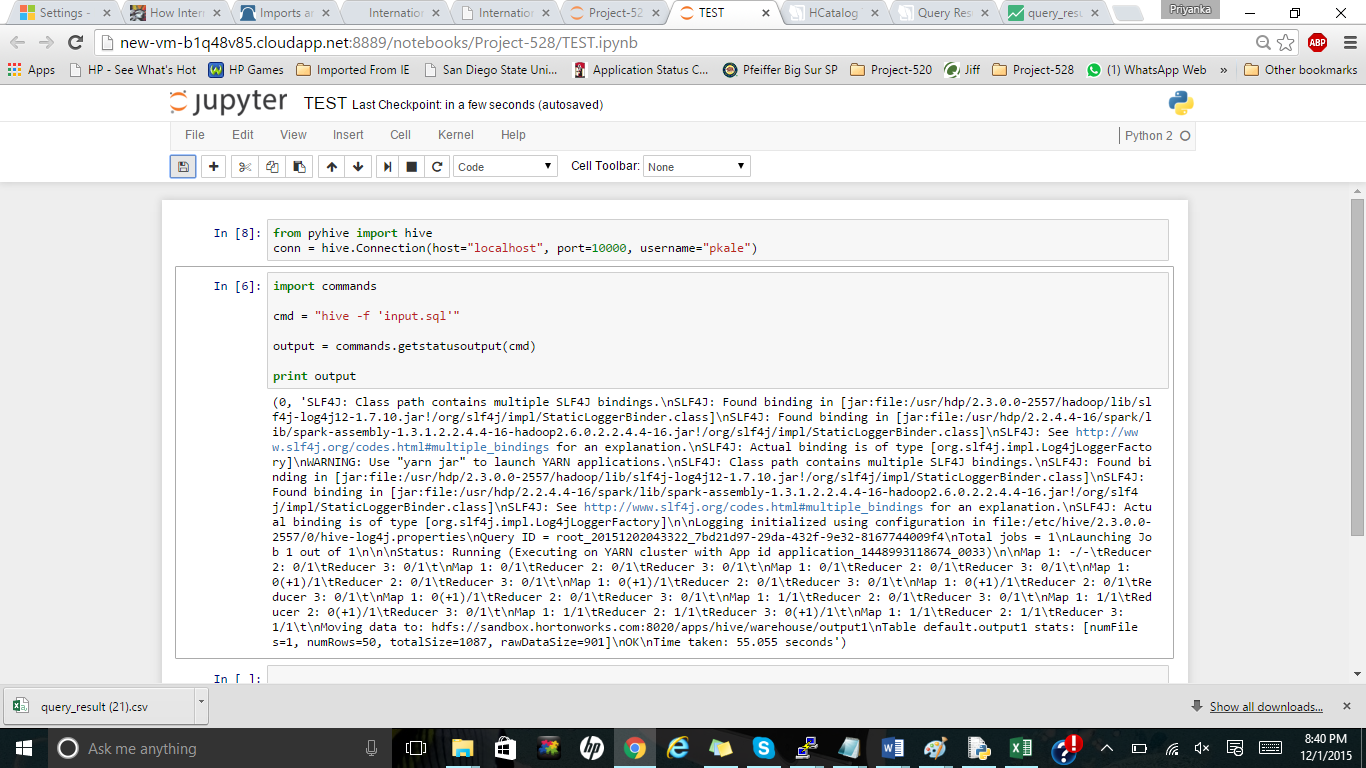
This is the pyhive command we write in order to make create the table automatically in the Hue where we make table and save in the .csv format for further analyses

The Pyhive code:

‘’”

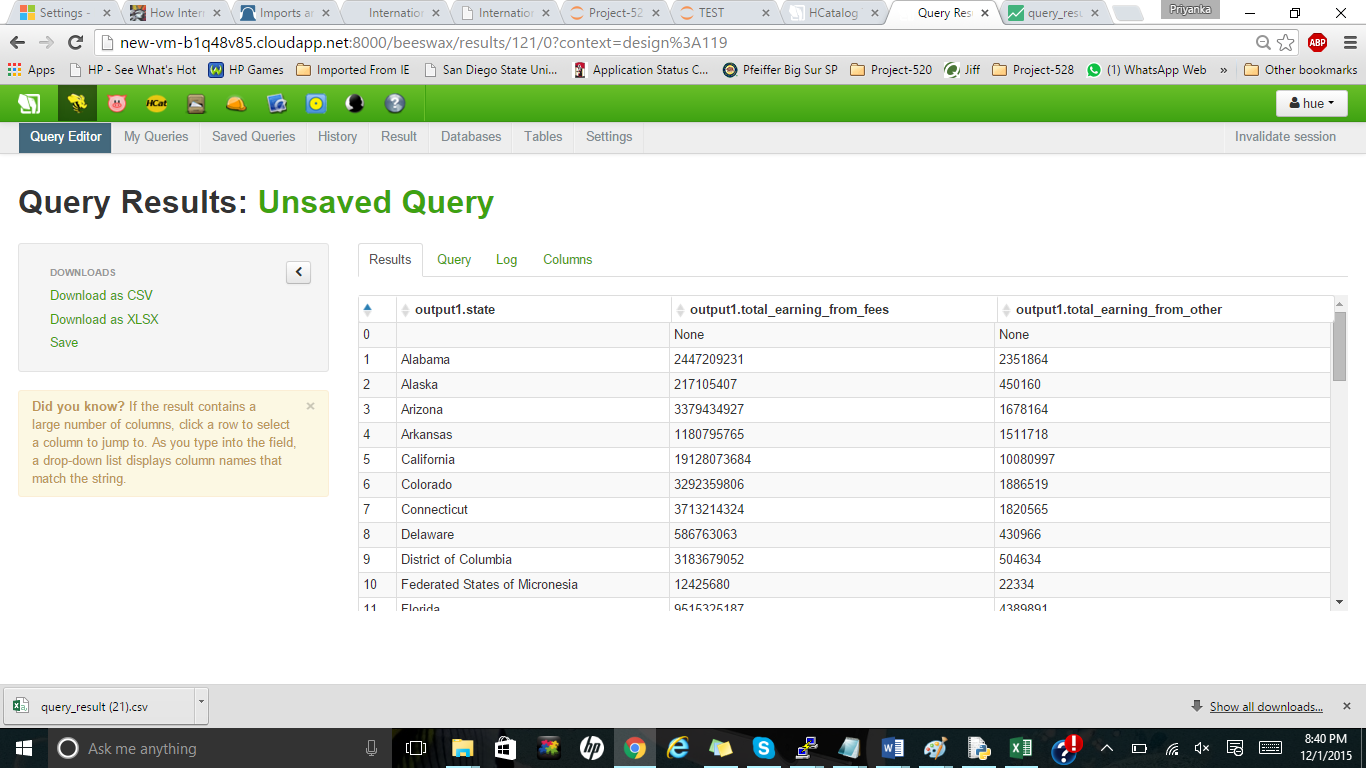
**from** pyhive **import** hive  
conn= hive.Connection(host=**"localhost"**,port=10000,username=**"Aditya"**)  
  
**import** commands  
  
cmd = **"hive -f 'input.sql'"**output = commands.getstatusoutput(cmd)  
  
print output

“”



The INPUT.SQL FILE

This is the output of the comma separated file as shown below. This is achieved

Giving the statement select from \* ‘the table name or the column one want to get the results. This output is saved as **.csv** format on the computer. 

Now comes the visualization of the code, this CSV generated file is then uploaded in the visualization tool to have the overview of the areas which get affected by the following. This gives us the coordinates of the place and we can take actions depending upon the results or develop our reports based on the visualization. Below is the visualization of the CSV file we generated

