



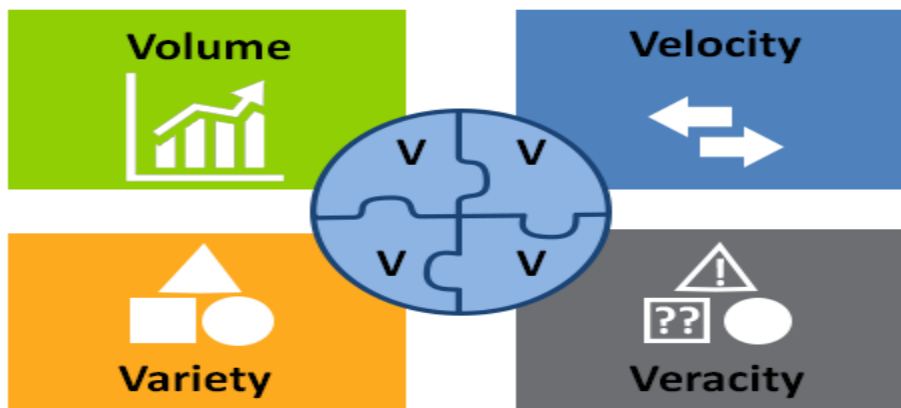
NEHA OHRI -S18000650030





Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. **Big data** challenges include capturing **data**, **data** storage, **data** analysis, search, sharing, transfer, visualization, querying, updating and information private.

4V'S OF Big Data



Volume

It is estimated that, on an average, 2.3 trillion gigabytes of data are generated every day. Forget analyzing, simply capturing such quantities of data is impractical. Most companies in the US have at least 100,000 gigabytes of data stored; and almost all of them will tell you that they aren't collecting enough data.

The right approach is to fight the urge of making your company's server a data dump. Efforts must be made to employ the right software to filter the relevant data.

Variety

Along with quantity, the diversity of data is equally important. The variety in data can be in terms of the devices or the sources of data generation.

Velocity

Not only is the volume of data ever increasing, but the rate of data generation (from the Internet of Things, social media, etc.) is increasing as well.

Veracity

Not all data is good. In fact, unfiltered data is more likely to be bad than good. Although data quality and usability depends largely on the source, you can never rule out junk.

This unreliability of data makes many business heads reluctant to rely on information analysis. That's the wrong approach.

HADOOP



Apache Hadoop is an open-source software framework used for distributed storage and processing of very large data sets. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are a common occurrence and should be automatically handled by the frame.

HIVE

The Hive



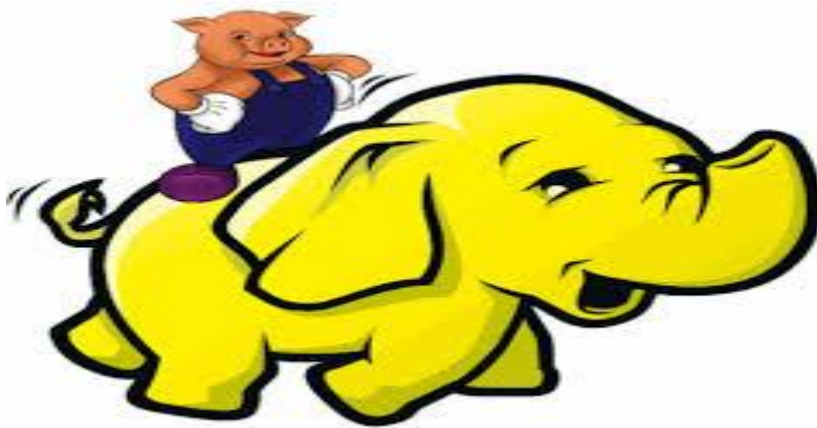
Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.

Hive gives an SQL-like interface to query data stored in various databases and filesystems that integrate with Hadoop.

It is built on top of Hadoop and developed by Facebook. Hive provides a way to query the data using a SQL-like query language called HiveQL (Hive query Language).

Internally, a compiler translates HiveQL statements into MapReduce jobs, which are then submitted to Hadoop framework for execution.

Apache Pig



Apache Pig is a high-level platform for creating programs that run on Apache Hadoop. The language for this platform is called Pig Latin.

Pig can execute its Hadoop jobs in Map Reduce, Apache Tez, or Apache Spark. Pig Latin abstracts the programming from the Java Map Reduce idiom into a notation which makes Map Reduce programming high level, similar to that of SQL for RDBMS.

Pig Latin can be extended using User Defined Functions (UDFs) which the user can write in Java, Python, JavaScript, Ruby or Groovy and then call directly from the language.

MAP REDUCE

Input	Splitting	Mapping	Shuffling	Reducing	Write to file
<div> Hello Mike Hello John John is good Mike is Tall </div>	<div>Hello Mike</div> <div>Hello John</div> <div>John good</div> <div>Mike Tall</div>	<div>Hello , 1 Mike , 1</div> <div>Hello , 1 John , 1</div> <div>John , 1 good , 1</div> <div>Mike , 1 Tall , 1</div>	<div>good , 1</div> <div>Hello , 1 Hello , 1</div> <div>John , 1 John , 1</div> <div>Mike , 1 Mike , 1</div>	<div>good , 1</div> <div>Hello , 2</div> <div>John , 2</div> <div>Mike , 2</div>	<div> good , 1 Hello , 2 John , 2 Mike , 2 </div>

TaskTracker split the file and pass to mapper and mapper converts it into <Key, Value> map. As per above example it uses TextInputFormat to split input file into lines. Mapper split the line into word and uses Text to store word as key and

IntWritable to store 1 as count value. Mapper passes map to OutputCollector, which intern shuffle and sort the map. Combiner is optional which optimize the reducer on node level. Here we are using Reducer as Combiner to combine the output to a single.

SQOOP



Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases.

H1-B CASE STUDY

PROJECT OBJECTIVE--- The H1- B is an employment –based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign national to apply H1-B Visa, an US employer must offer a job and petition for H1-B visa with the US immigration department. This is the most common visa status applied for and held by international students once they complete college/higher education (Maters, Ph.D.) and work in a full-time position.

Hardware Requirements-

6 GB RAM

64Bit OS

Technology Requirements-

- Apache Hadoop
- MapReduce
- Hive
- Pig
- SQOOP

Software Used—

- VMware
- Ubuntu
- Eclipse
- MySQL

Assumptions:

- VMware Workstation – Configurations are set correctly.
- Ubuntu is lying on the Virtual Box and it is powered on
- Hadoop Folder must be extracted and all the services of the Hadoop is running. Configuration to be made in the XML are set.
- Confirmation Box Below that Everything is Set Right.

Datasets Required-

- H1-B CASE Applications Data
- The Dataset has nearly 3 million records.

The Dataset given below—

The columns in the dataset include:

1. **CASE_STATUS:** Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” “Denied,” and “Withdrawn”.

Certified: Employer filed the LCA, which was approved by DOL

Certified Withdrawn: LCA was approved but later withdrawn by employer

Withdrawn: LCA was withdrawn by employer before approval

Denied: LCA was denied by DOL

2. **EMPLOYER_NAME:** Name of employer submitting labour condition application.

3. **SOC_NAME:** The Occupational name associated with the SOC_CODE.

SOC_CODE is the occupational code associated with the job being requested for temporary labour condition, as classified by the Standard Occupational Classification (SOC) System.

4. **JOB_TITLE:** Title of the job

5. **FULL_TIME_POSITION:** Y = Full Time Position; N = Part Time Position

6. **PREVAILING_WAGE:** Prevailing Wage for the job being requested for temporary labour condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer’s minimum requirements for the position.

7. **YEAR:** Year in which the H1B visa petition was filed

8. **WORKSITE:** City and State information of the foreign worker’s intended area of employment

9. **LONGITUDE:** longitude of the Worksite

10. LATITUDE:latitude of the Worksite

Outcome of this Project:

To generate reports and hence,

We will be performing analysis on the H1B visa applicants between the years 2011-2016. After analysing the data, we can derive the following facts.

- 1 a) Is the number of petitions with Data Engineer job title increasing over time?
b) Find top 5 job titles who are having highest avg growth in applications. [ALL]
- 2 a) Which part of the US has the most Data Engineer jobs for each year?
b) find top 5 locations in the US who have got certified visa for each year. [certified]
- 3) Which industry(SOC_NAME) has the most number of Data Scientist positions?
[certified]
- 4) Which top 5 employers file the most petitions each year? - Case Status - ALL
- 5) Find the most popular top 10 job positions for H1B visa applications for each year?
a) for all the applications
b) for only certified applications.
- 6) Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time.
- 7) Create a bar graph to depict the number of applications for each year [All]
- 8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order - [Certified and Certified Withdrawn.]
- 9) Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000)?

10) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than 1000)?

11) Export result for question no 10 to MySQL database.

Create a table to load the h1b applicant's data as shown below:

```
CREATE TABLE h1b_applications (s_no int, case_status string, employer_name string, soc_name string, job_title string, full_time_position string, prevailing_wage bigint, year string, worksitestring, longitude double, latitude double) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
```

```
WITH SERDEPROPERTIES (  
  "separatorChar" = ",",  
  "quoteChar" = "\""   
) STORED AS TEXTFILE;
```

➤ load data local inpath '/home/hduser/Downloads/H1Project/h1b.csv' overwrite into table h1b_applications.

➤ CREATE TABLE h1b_app2(s_no int, case_status string, employer_name string, soc_name string, job_title string, full_time_position string, prevailing_wage bigint, year string, worksitestring, longitude double, latitude double)
row format delimited
fields terminated by '\t'
STORED AS TEXTFILE;

➤ INSERT OVERWRITE TABLE h1b_app2 SELECT regexp_replace (s_no, "\t", ""), regexp_replace (case_status, "\t", ""), regexp_replace (employer_name,

```
"\t", ""), regexp_replace (soc_name, "\t", ""), regexp_replace (job_title, "\t",
""),
regexp_replace (full_time_position, "\t", ""), prevailing_wage,
regexp_replace (year, "\t", ""), regexp_replace (worksite, "\t", ""),
regexp_replace (longitude, "\t", ""), regexp_replace (latitude, "\t", "") FROM
hib_applications where case_status! = "NA";
```

- CREATE TABLE hib_final (s_no int, case_status string, employer_name string, soc_name string, job_title string, full_time_position string, prevailing_wage bigint, year string, worksite string, longitude double, latitude double) row format delimited fields terminated by '\t' STORED AS TEXTFILE;
- INSERT OVERWRITE TABLE hib_final SELECT s_no, case when trim(case_status) = "PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED" then "DENIED" when trim(case_status) = "REJECTED" then "DENIED" when trim(case_status) = "INVALIDATED" then "DENIED" else case_status end, employer_name, soc_name, job_title, full_time_position, case when prevailing_wage is null then 100000 else prevailing_wage end, year, worksite, longitude, latitude FROM hib_app2;

PIG QUESTIONS

1 b) Find top 5 job titles who are having highest average growth in applications.[ALL]

Solution:

```

i(b)--- Find top 5 job titles who are having highest avg growth in applications.[ALL]
quarry---register /usr/local/hive/lib/hive-exec-1.2.1.jar
register /usr/local/hive/lib/hive-common-1.2.1.jar
A = LOAD 'hdfs://localhost:54310/user/hive/warehouse/finalproject.db/h1b_final' USING PigStorage(',') as
(s_no:double,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevall
cleansed= filter A by $7=='2011';
a= group cleansed by $4;
step_a = foreach a generate group,COUNT($1);
describe step a;
cleansed1= filter A by $7=='2012';
b= group cleansed1 by $4;
step_b= foreach b generate group,COUNT($1);
describe step b;
cleansed2= filter A by $7=='2013';
c= group cleansed2 by $4;
step_c= foreach c generate group,COUNT($1);
describe step c;
cleansed3= filter A by $7=='2014';
d= group cleansed3 by $4;
step_d= foreach d generate group,COUNT($1);
describe step d;
cleansed4= filter A by $7=='2015';
e= group cleansed4 by $4;
step_e= foreach e generate group,COUNT($1);
describe step e;
cleansed5= filter A by $7=='2016';
f= group cleansed5 by $4;
step_f= foreach f generate group,COUNT($1);
describe step f;
joined= join step_a by $0,step_b by $0,step_c by $0,step_d by $0,step_e by $0,step_f by $0;
describe joined;
yearwiseapplications= foreach joined generate $0,$1,$3,$5,$7,$9,$11;
progressivegrowth= foreach yearwiseapplications generate $0,ROUND_TO((long)($0-$5)*100/$5,2),ROUND_TO((long)($5-$4)*100/
$4,2),ROUND_TO((long)($4-$3)*100/$3,2),ROUND_TO((long)($3-$2)*100/$2,2),ROUND_TO((long)($2-$1)*100/$1,2);
avgprogressivegrowth= foreach progressivegrowth generate $0,($1+$2+$3+$4+$5)/5;
orderedavggrowth= order avgprogressivegrowth by $1 desc;
answer = limit orderedavggrowth 5;
store answer into 'hdfs://localhost:54310/pig/question1b' using PigStorage(',');

```

```

output---SENIOR SYSTEMS ANALYST JC60,4255.4
SOFTWARE DEVELOPER 2,3480.8
PROJECT MANAGER 3,3233.4
SYSTEMS ANALYST JC65,2985.0
MODULE LEAD,2917.2

```

6) Find the percentage and the count of each case status on total applications for each year. Create a line graph depicting the pattern of All the cases over the period of time?

```

question---(6) Find the percentage and the count of each case status on total applications for each year. Create a line graph
depicting the pattern of All the cases over the period of time?
quarry---register /usr/local/hive/lib/hive-exec-1.2.1.jar
register /usr/local/hive/lib/hive-common-1.2.1.jar
A = LOAD 'hdfs://localhost:54310/user/hive/warehouse/finalproject.db/h1b_final' USING PigStorage(',') as
(s_no:double,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevall
cleansed= filter A by $1 is not null and $1!='NA';
temp= group cleansed by $7;
total= foreach temp generate group,COUNT(cleansed.$0);
cleansed1= filter A by $7 is not null and $7!='NA';
temp1= group cleansed1 by ($7,$1);
yearsoccount= foreach temp1 generate group,group.$0,COUNT($1);
joined= join yearsoccount by $1,total by $0;
ans= foreach joined generate FLATTEN($0),ROUND_TO(((long)$2*100)/((long)$4,2),$2;
store ans into 'hdfs://localhost:54310/pig/question6' using PigStorage(',');

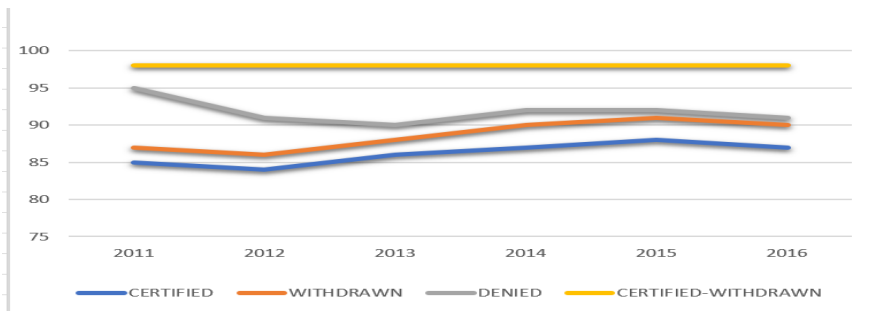
```

```

2011,DENIED,8.0,29130
2011,CERTIFIED,85.0,307936
2011,WITHDRAWN,2.0,10105
2011,CERTIFIED-WITHDRAWN,3.0,11596
2012,DENIED,5.0,21096
2012,CERTIFIED,84.0,352668
2012,WITHDRAWN,2.0,10725
2012,CERTIFIED-WITHDRAWN,7.0,31118
2013,PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED,0.0,15
2013,CERTIFIED-WITHDRAWN,8.0,35432
2013,WITHDRAWN,2.0,11590
2013,CERTIFIED,86.0,382951
2013,DENIED,2.0,12126
2014,WITHDRAWN,3.0,16034
2014,CERTIFIED,87.0,455144
2014,DENIED,2.0,11899
2014,CERTIFIED-WITHDRAWN,6.0,36350
2015,DENIED,1.0,10923
2015,CERTIFIED,88.0,547278
2015,WITHDRAWN,3.0,19455
2015,CERTIFIED-WITHDRAWN,6.0,41071
2016,DENIED,1.0,9175
2016,CERTIFIED,87.0,569646
2016,WITHDRAWN,3.0,21890
2016,CERTIFIED-WITHDRAWN,7.0,47092
hduser@ubuntu:~$

```

YEAR	CERTIFIED	WITHDRAWN	DENIED	CERTIFIED-WITHDRAWN	TOTAL
2011	85	2	8	3	98
2012	84	2	5	7	98
2013	86	2	2	8	98
2014	87	3	2	6	98
2015	88	3	1	6	98
2016	87	3	1	7	98



9) Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total petitions filed 1000 OR more than 1000)?


```

-----
PIG
-----
ques--9 Which are the employers along with the number of petitions who have the success rate more than 70% in petitions. (total
petitions filed 1000 OR more than 1000) ?

quary---register /usr/local/hive/lib/hive-exec-1.2.1.jar
register /usr/local/hive/lib/hive-common-1.2.1.jar
data1 = LOAD 'hdfs://localhost:54310/user/hive/warehouse/finalproject.db/h1b_final' USING PigStorage('\t') as
(s_no:double,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevall
a= filter data1 by $1 is not null and $1!='NA';
temp= group a by $2;
total= foreach temp generate group,COUNT(a.$0);
certified= filter data1 by $1 == 'CERTIFIED';
temp1= group certified by $2;
totalcertified= foreach temp1 generate group,COUNT(certified.$0);
certified_with= filter data1 by $1 == 'CERTIFIED-WITHDRAWN';
temp2= group certified_with by $2;
totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified_with.$0);
joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;
joined= foreach joined generate $0,$1,$3,$5;
intermediateoutput= foreach joined generate $0,(float)($1+$2)*100/($3),$3;
intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000;
finaloutput= order intermediateoutput2 by $1 DESC;
store finaloutput into 'hdfs://localhost:54310/plg/question9' using PigStorage('\t');

output----hadoop fs -cat /plg/question9/p*
INFOSYS LIMITED 99.54055 130592
ACCENTURE LLP 99.39307 33447
TATA CONSULTANCY SERVICES LIMITED 99.337204 64720
HCL AMERICA, INC. 99.26801 22678
RELIABLE SOFTWARE RESOURCES, INC. 99.14658 1992

```

10) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions (total petitions filed 1000 OR more than1000)?

```

question (10) Which are the job positions along with the number of petitions which have the success rate more than 70% in petitions
(total petitions filed 1000 OR more than 1000)?

quary----register /usr/local/hive/lib/hive-exec-1.2.1.jar
register /usr/local/hive/lib/hive-common-1.2.1.jar
a = LOAD 'hdfs://localhost:54310/user/hive/warehouse/finalproject.db/h1b_final' USING PigStorage('\t') as
(s_no:double,case_status:chararray,employer_name:chararray,soc_name:chararray,job_title:chararray,full_time_position:chararray,prevall
a= filter a by $1 is not null and $1!='NA';
temp= group a by $4;
total= foreach temp generate group,COUNT(a.$0);
certified= filter a by $1 == 'CERTIFIED';
temp1= group certified by $4;
totalcertified= foreach temp1 generate group,COUNT(certified.$0);
certified_with= filter a by $1 == 'CERTIFIED-WITHDRAWN';
temp2= group certified_with by $4;
totalcertifiedwithdrawn= foreach temp2 generate group,COUNT(certified_with.$0);
joined= join totalcertified by $0,totalcertifiedwithdrawn by $0,total by $0;
joined= foreach joined generate $0,$1,$3,$5;
intermediateoutput= foreach joined generate $0,(float)($1+$2)*100/($3),$3;
intermediateoutput2= filter intermediateoutput by $1>70 and $2>1000;
finaloutput= order intermediateoutput2 by $1 DESC;
store finaloutput into 'hdfs://localhost:54310/plg/question10' using PigStorage('\t');

output-----COMPUTER PROGRAMMER / CONFIGURER 2 100.0 1276
ASSOCIATE CONSULTANT - US 99.93171 4393
SYSTEMS ENGINEER - US 99.90036 10036
TEST ANALYST - US 99.818474 4958
CONSULTANT - US 99.81147 7426
TECHNOLOGY LEAD - US 99.80247 28350
TECHNICAL TEST LEAD - US 99.79531 5374
TECHNOLOGY ARCHITECT - US 99.766304 4707

```

HIVE

2 b) find top 5 locations in the US who have got certified visa for each year.[certified]

```
Ubuntu 64-bit - VMware Workstation 12 Player (Non-commercial use only)
ques--2(b)find top 5 locations in the US who have got certified visa for each year.[certified]?
quary---2011--select worksite,count(case_status)as temp,year from h1b_final where year ='2011' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;
output---NEW YORK, NEW YORK      23172   2011
HOUSTON, TEXAS      8184      2011
CHICAGO, ILLINOIS    5188      2011
SAN JOSE, CALIFORNIA 4713      2011
SAN FRANCISCO, CALIFORNIA 4711      2011
Time taken: 385.259 seconds, Fetched: 5 row(s)

2012----select worksite,count(case_status)as temp,year from h1b_final where year ='2012' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;

2013----select worksite,count(case_status)as temp,year from h1b_final where year ='2013' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;

2014----select worksite,count(case_status)as temp,year from h1b_final where year ='2014' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;

2015----select worksite,count(case_status)as temp,year from h1b_final where year ='2015' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;

2016----select worksite,count(case_status)as temp,year from h1b_final where year ='2016' and case_status ='CERTIFIED' group by
worksite,year order by temp desc limit 5;
```

5) Find the most popular top 10 job positions for H1B visa applications for each year?
(a) for all the applications

```
ques----5(a) Find the most popular top 10 job positions for H1B visa applications for each year?
a) for all the applications?
b) for only certified applications?
quary---select job_title,year,count(job_title)as temp from h1b_final where year='2011' group by job_title,year order by temp desc
limit 10;
output-----PROGRAMMER ANALYST  2011    31799
SOFTWARE ENGINEER      2011    12763
COMPUTER PROGRAMMER    2011    8998
SYSTEMS ANALYST 2011    8644
BUSINESS ANALYST      2011    3891

2012----select job_title,year,count(job_title)as temp from h1b_final where year='2012' group by job_title,year order by temp desc
limit 10;

2013----select job_title,year,count(job_title)as temp from h1b_final where year='2013' group by job_title,year order by temp desc
limit 10;

2014----select job_title,year,count(job_title)as temp from h1b_final where year='2014' group by job_title,year order by temp desc
limit 10;

2015----select job_title,year,count(job_title)as temp from h1b_final where year='2015' group by job_title,year order by temp desc
limit 10;

2016----select job_title,year,count(job_title)as temp from h1b_final where year='2016' group by job_title,year order by temp desc
limit 10;
```


5 (b) for only certified applications?

```

S(b)---for only certified applications?
quary----5(b)-----select job_title,year,count(job_title)as temp from h1b_final where case_status='CERTIFIED' group by job_title,year
order by temp desc limit 10;

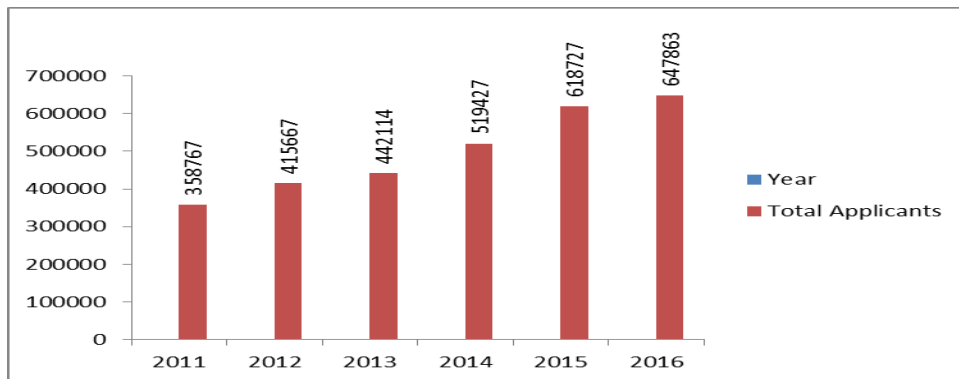
output----
PROGRAMMER ANALYST      2015      48203
PROGRAMMER ANALYST      2016      47964
PROGRAMMER ANALYST      2014      38625
PROGRAMMER ANALYST      2013      29906
PROGRAMMER ANALYST      2012      29226
PROGRAMMER ANALYST      2011      28806
SOFTWARE ENGINEER       2016      25890
SOFTWARE ENGINEER       2015      23352
SOFTWARE ENGINEER       2014      17278
COMPUTER PROGRAMMER     2014      13796
  
```

7) Create a bar graph to depict the number of applications for each year [All]

```

question----(7) Create a bar graph to depict the number of applications for each year [All]?
quary-----select year,count(*) from h1b_final group by year order by year;

output-----
2011      358767
2012      415607
2013      442114
2014      519427
2015      618727
2016      647863
  
```



8) Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in descending order - [Certified and Certified Withdrawn.]

```

HIVE
ques--(8)Find the average Prevailing Wage for each Job for each Year (take part time and full time separate). Arrange the output in
descending order - [Certified and Certified Withdrawn.]?

quary--1st...2011... select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where
full_time_position='Y' and year='2011' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by
job_title,full_time_position,year order by average desc;

output----FRANCHISE DEVELOPMENT ANALYST Y      2011      43306.0
MARKET RESEARCH ANALYST (ACCOUNT MANAGER) Y      2011      43306.0
PRODUCT RESEARCH ANALYST, DAYLIGHTING Y      2011      43306.0
LOTUS NOTES/DOMINO DEVELOPER Y      2011      43306.0
TELEVISION/ NEWS VIDEO EDITOR Y      2011      43306.0
MARKETING/SALES ANALYST Y      2011      43306.0

2nd...2011...select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2011' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

ARM MANAGER N      2011      42931.0
ESTIMATOR N      2011      42924.0
ASSISTANT RESEARCH SCIENTIST N      2011      42916.90476190476
QUALITY CONTROL CHEMIST N      2011      42910.0
IMPORT & CONTRACT SPECIALIST N      2011      42889.0

2012----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='Y' and
year='2012' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2012--select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2012' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2013----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='Y' and
year='2013' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2013---select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2013' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

```

```

2013----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2013' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2014----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='Y' and
year='2014' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2014----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2014' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2015----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='Y' and
year='2015' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2015----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2015' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2016----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='Y' and
year='2016' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

2016----select job_title,full_time_position,year,avg(prevaling_wage) as average from hib_final where full_time_position='N' and
year='2016' and case_status in ('CERTIFIED','CERTIFIED-WITHDRAWN') group by job_title,full_time_position,year order by average desc;

```

MAPREDUCE

Question 2(a) – Which part of the US has the most Data Engineer Job for each year?

```

hduser@ubuntu:~$ hadoop fs -cat /finalproject/outputmapreduce1/p**
SEATTLE, WASHINGTON      2011,20
SAN FRANCISCO, CALIFORNIA      2011,4
SAN MATEO, CALIFORNIA      2011,3
WALTHAM, MASSACHUSETTS      2011,2
TALLAHASSEE, FLORIDA      2011,1
SEATTLE, WASHINGTON      2012,30
SAN FRANCISCO, CALIFORNIA      2012,10
PONTIAC, MICHIGAN      2012,3
SAN MATEO, CALIFORNIA      2012,2
WOODLAND HILLS, CALIFORNIA      2012,1
SEATTLE, WASHINGTON      2013,46
SAN FRANCISCO, CALIFORNIA      2013,17
MENLO PARK, CALIFORNIA      2013,12
NEW YORK, NEW YORK      2013,6
ATLANTA, GEORGIA      2013,5
SEATTLE, WASHINGTON      2014,45
SAN FRANCISCO, CALIFORNIA      2014,34
MENLO PARK, CALIFORNIA      2014,21
NEW YORK, NEW YORK      2014,18
MOUNTAIN VIEW, CALIFORNIA      2014,13
SEATTLE, WASHINGTON      2015,61
NEW YORK, NEW YORK      2015,41

```

Question -(3) Which industry (SOC_NAME) has the most number of Data Scientist Position?[Certified]

```

hduser@ubuntu:~$ hadoop jar /home/hduser/Documents/DataScientist.jar DataScientist /user/hive/warehouse/finalproject.db/hib_final /ft
nalproject/DScientoutput
hduser@ubuntu:~$ hadoop fs -cat /finalproject/DScientoutput/p*
STATISTICIANS,572
COMPUTER AND INFORMATION RESEARCH SCIENTISTS,419
OPERATIONS RESEARCH ANALYSTS,380
Computer and Information Research Scientists,181
COMPUTER OCCUPATIONS, ALL OTHER,160
hduser@ubuntu:~$

```

Question 4—Which top 5 employer file the most petitions each year Case Status—ALL??


```

hduser@ubuntu:~$ hadoop fs -cat /finalproject/dataengi2ka4/p*
TATA CONSULTANCY SERVICES LIMITED                2011,5416
MICROSOFT CORPORATION 2011,4253
DELOITTE CONSULTING LLP 2011,3621
WIPRO LIMITED 2011,3028
COGNIZANT TECHNOLOGY SOLUTIONS U.S. CORPORATION 2011,2721
INFOSYS LIMITED 2012,15818
WIPRO LIMITED 2012,7182
TATA CONSULTANCY SERVICES LIMITED                2012,6735
DELOITTE CONSULTING LLP 2012,4727
IBM INDIA PRIVATE LIMITED 2012,4074
INFOSYS LIMITED 2013,32223
TATA CONSULTANCY SERVICES LIMITED                2013,8790
WIPRO LIMITED 2013,6734
DELOITTE CONSULTING LLP 2013,6124
ACCENTURE LLP 2013,4994
INFOSYS LIMITED 2014,23759
TATA CONSULTANCY SERVICES LIMITED                2014,14098
WIPRO LIMITED 2014,8365
DELOITTE CONSULTING LLP 2014,7017
ACCENTURE LLP 2014,5498
INFOSYS LIMITED 2015,33245
TATA CONSULTANCY SERVICES LIMITED                2015,16553
WIPRO LIMITED 2015,12201
IBM INDIA PRIVATE LIMITED 2015,10693
ACCENTURE LLP 2015,9605
INFOSYS LIMITED 2016,25352
CAPGEMINI AMERICA INC 2016,16725
TATA CONSULTANCY SERVICES LIMITED                2016,13134
WIPRO LIMITED 2016,10607
IBM INDIA PRIVATE LIMITED 2016,9787

```

Question 1-(a) Is the number of petitions with Data Engineer job title increasing over time?

```

hduser@ubuntu:~$ hadoop fs -cat /finalproject/output4/p*
2011 1
2012 100
2013 50
2014 33
2015 25
2016 20
hduser@ubuntu:~$

```

11) Export result for question no 10 to MySQL database.

Solution:

Quary `--sqoop export --connect jdbc:mysql://localhost/h1b_final --username root --password linux --table question11 --update-mode allowinsert --export --dir /pig/question10/p* --input-fields-terminated-by '\t'`;

Output show--- `mysql -u root -p -e 'select * from h1b_final.question11'`;

```
Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hduser@ubuntu: $ mysql -u root -p -e 'select * from hib_final.question11';
Enter password:
```

job_title	success_rate	petitions
SOFTWARE DEVELOPERS, APPLICATIONS	92.9707	1195
POSTDOCTORAL FELLOW	94.8581	7857
RESEARCH FELLOW	96.3551	5981
SENIOR HARDWARE ENGINEER	94.8578	1653
QA ENGINEER	94.8291	2224
APPLICATIONS DEVELOPER	96.3458	3366
COMPUTER PROGRAMMER / CONFIGURER 2	100	1276
COMPUTER SYSTEM ANALYST	92.7525	3753
SENIOR SOFTWARE DEVELOPER	96.3166	10208
PROGRAMMER ANALYST	96.1279	249038
SENIOR PROGRAMMER ANALYST	96.1274	5810
ASSISTANT RESEARCH SCIENTIST	96.1015	1103
SENIOR ASSOCIATE	96.017	3540
SOFTWARE QUALITY ASSURANCE ENGINEER	95.9756	4920
SENIOR MANAGER	95.9694	1439
PHYSICIAN IN A POST GRADUATE TRAINING PROGRAM	95.9521	2421
SYSTEMS ANALYST	95.9477	61965
QUALITY ASSURANCE ANALYST	95.9459	7326
TECHNICAL ARCHITECT	95.9422	2908
PROJECT LEAD	95.9374	2363
SOFTWARE ENGINEER III	95.9337	1328
SOFTWARE ANALYST	95.8955	1072

CONCLUSION:

Therefore from the given dataset H1-B applicants within the years 2011-2016, which contained around 3 billion that is 3lakh records, we have done a complete analysis on various factors which has predictive nature. This analysis on the H1-B VISA data set has resolved around finding top Occupation, States, Employers, Industries the contribute to highest number of H1-B VISA. This clearly indicates the H1-B VISA filings which has a high correlation with the employer's rate.
