H1B DATA ANALYSIS USING HADOOP ECOSYSTEM

**Presented by : Name : AJIT SALGARE**

**STUDENT ID:s171107500151**

**Center: Pune Deccan**

**Abstract**

What is Big Data?

Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques.

Data which are very large in size is called Big Data. Normally we work on data of size MB(Word-Doc ,Excel) or maximum GB(Movies, Codes) but data in Peta bytes i.e. 10^15 byte size is called Big Data.

Hadoop

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs

Hadoop history

[**Hadoop**](http://en.wikipedia.org/wiki/Apache_Hadoop)was created by [**Doug Cutting**](http://en.wikipedia.org/wiki/Doug_Cutting) who had created the [**Apache Lucene**](http://lucene.apache.org/core/)(Text Search),which is origin in Apache Nutch(Open source search Engine).Hadoop is a part of Apache Lucene Project.Actually [**Apache Nutch**](http://nutch.apache.org/) was started in 2002 for working crawler and search system.Nutch Architecture would not  scale up to billions of pages on the web.

In 2003 google had published one Architecture  called [**Google Distributed Filesystem(GFS)**](http://en.wikipedia.org/wiki/Google_File_System),which was solve the storage need for the very large files generated as a part of the web crawl and indexing process.

In 2004 based on GFS architecture Nutch was implementing open source called the **Nutch Distributed Filesystem (NDFS)**.In 2004 google was published Mapreduce,In 2005 Nutch developers had working on [**Mapreduce**](http://en.wikipedia.org/wiki/MapReduce) in Nutch Project.Most of the Algorithms had been ported to run using mapreduce and NDFS.

In February 2006 they moved out of Nutch to form an independent subproject of Lucene called Hadoop.At around the same time, Doug Cutting joined Yahoo!, which provided a dedicated team and the resources to turn Hadoop into a system that ran at web scale. This was demonstrated in February 2008 when Yahoo! announced that its production search index was being generated by a 10,000-core **Hadoop cluster**.

**Why Hadoop is important:**

* **Fast:** In HDFS the data distributed over the cluster and are mapped which helps in faster retrieval. Even the tools to process the data are often on the same servers, thus reducing the processing time. It is able to process terabytes of data in minutes and Peta bytes in hours.
* **Scalable:** Hadoop cluster can be extended by just adding nodes in the cluster.
* **Cost Effective:** Hadoop is open source and uses commodity hardware to store data so it really cost effective as compared to traditional relational database management system.
* **Ability to store and process huge amounts of any kind of data, quickly.** With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.
* **Computing power.** Hadoop's distributed computing model processes big data fast. The more computing nodes you use, the more processing power you have.

**Hadoop ecosystem components used in this project**

**Hive:**

A data warehousing and SQL-like query language that presents data in the form of tables. Hive programming is similar to database programming.

**Pig:**

A platform for manipulating data stored in HDFS that includes a compiler for MapReduce programs and a high-level language called Pig Latin. It provides a way to perform data extractions, transformations and loading, and basic analysis without having to write MapReduce programs.

**Sqoop:**

A connection and transfer mechanism that moves data between Hadoop and relational databases.

**Zookeeper:**

An application that coordinates distributed processing.

**Oozie:**

A Hadoop job scheduler.

**Project Outline**

|  |  |
| --- | --- |
| Title | Big Data Analysis in Hadoop on H1B Data |
| Inputs | H1b Data |
| Data Elements | Sr.no, case status, employer name, soc name, job position, full time position,prevailing wage, year, worksite, longitude, latitude. |
| Purpose | To provide analyzed report to H1B Sponsers to help them to make plans for minimum wage provision and to get skilled workers from respective worksite. |
| Methodology | Agile |

**Acknowledgement**

I wish to thank my master trainer Mr.SandeepAgarwal and my Tech mentor Mrs.Jyoti Mittal for providing complete learning on Big Data and Hadoop and guiding me in accomplishing the objectives of my project.

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

(using pig)

Commands:

group1 = foreach h1b\_final generate $4,$7;

group2 = FILTER group2 BY $0 == 'DATA ENGINEER';

--dump group2;

group3 = group abc1 by $1;

group4 = foreach group3 generate group, COUNT(group2.$0);

group2011 = filter group4 by $0=='2011';

group2012 = filter group4 by $0=='2012';

group2013 = filter group4 by $0=='2013';

group2014 = filter group4 by $0=='2014';

group2015 = filter group4 by $0=='2015';

group2016 = filter group4 by $0=='2016';

data = foreach group2012 generate group2011.$1, group2012.$1, group2013.$1, group2014.$1, group2015.$1, group2016.$1;

percent = foreach data generate CONCAT((chararray)((float)(($1-$0)\*100/$0)),'%'), CONCAT((chararray)((float)(($2-$1)\*100/$1)),'%'), CONCAT((chararray)((float)(($3-$2)\*100/$2)),'%'), CONCAT((chararray)((float)(($4-$3)\*100/$3)),'%'), CONCAT((chararray)((float)(($5-$4)\*100/$4)),'%');

dump percent;

percent1 = foreach percent generate FLATTEN(TOBAG(\*));

dump percent1;

Output:

(77.0%)

(28.0%)

(117.0%)

(79.0%)

(56.0%)

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

(using pig)

group1 = foreach L1 generate $1,$4,$7;

g1 = FILTER group1 by $2 =='2011';

g2011 = group g1 by (job\_title);

count2011 = foreach g2011 generate group, COUNT($1);

--dump count2011;

g2 = FILTER group1 by $2 == '2012';

g2012 = group g2 by (job\_title);

count2012 = foreach g2012 generate group, COUNT($1);

--dump count2012;

g3 = FILTER group1 by $2 == '2013';

g2013 = group g3 by (job\_title);

count2013 = foreach g2013 generate group, COUNT($1);

g4 = FILTER group1 by $2 == '2014';

g2014 = group g4 by (job\_title);

count2014 = foreach g2014 generate group, COUNT($1);

g5 = FILTER group1 by $2 == '2015';

g2015 = group g2 by (job\_title);

count2015 = foreach g2015 generate group, COUNT($1);

g6 = FILTER group1 by $2 == '2016';

g2016 = group g2 by (job\_title);

count2016 = foreach g2016 generate group, COUNT($1);

--describe count2016;

group2 = join count2011 by $0, count2012 by $0, count2013 by $0, count2014 by $0, count2015 by $0, count2016 by $0;

--dump group2;

group3 = foreach group2 generate $0,$1,$3,$5,$7,$9,$11;

--dump group3;

growth = foreach group3 generate $0, (float)(($2-$1)\*100/$1),(float)(($3-$2)\*100/$2),(float)(($4-$3)\*100/$4),(float)(($5-$4)\*100/$4),(float)(($6-$5)\*100/$5);

avg\_growth = foreach growth generate $0, (float)(($1+$2+$3+$4+$5)/5);

--dump avg\_growth;

top5 = order avg\_growth by $1 DESC;

top5\_op = limit top5 5;

dump top5\_op;

Output:

(SENIOR SYSTEMS ANALYST JC60,4229.8)

(SOFTWARE DEVELOPER 2,3382.8)

(MODULE LEAD,3195.2)

(SYSTEMS ANALYST JC65,2969.8)

(LEAD,2507.0)

2 a) Which part of the US has the most Data Engineer jobs for each year?

(using pig)

Command:

group1 = foreach L1 generate $4,$8,$7;

group2 = FILTER group1 BY $0 == 'DATA ENGINEER';

group2011 = FILTER group2 BY $2 == '2011';

group2012 = FILTER group2 BY $2 == '2012';

group2013 = FILTER group2 BY $2 == '2013';

group2014 = FILTER group2 BY $2 == '2014';

group2015 = FILTER group2 BY $2 == '2015';

group2016 = FILTER group2 BY $2 == '2016';

group3\_2011 = group group2011 by ($0,$1,$2);

group3\_2012 = group group2012 by ($0,$1,$2);

group3\_2013 = group group2013 by ($0,$1,$2);

group3\_2014 = group group2014 by ($0,$1,$2);

group3\_2015 = group group2015 by ($0,$1,$2);

group3\_2016 = group group2016 by ($0,$1,$2);

group4\_2011 = foreach group3\_2011 generate group,COUNT(abc2011.$0);

group4\_2012 = foreach group3\_2012 generate group,COUNT(abc2012.$0);

group4\_2013 = foreach group3\_2013 generate group,COUNT(abc2013.$0);

group4\_2014 = foreach group3\_2014 generate group,COUNT(abc2014.$0);

group4\_2015 = foreach group3\_2015 generate group,COUNT(abc2015.$0);

group4\_2016 = foreach group3\_2016 generate group,COUNT(abc2016.$0);

g\_2011 = order group4\_2011 by $1 DESC;

g\_2012 = order group4\_2012 by $1 DESC;

g\_2013 = order group4\_2013 by $1 DESC;

g\_2014 = order group4\_2014 by $1 DESC;

g\_2015 = order group4\_2015 by $1 DESC;

g\_2016 = order group4\_2016 by $1 DESC;

ans\_2011 = LIMIT g\_2011 1;

ans\_2012 = LIMIT g\_2012 1;

ans\_2013 = LIMIT g\_2013 1;

ans\_2014 = LIMIT g\_2014 1;

ans\_2015 = LIMIT g\_2015 1;

ans\_2016 = LIMIT g\_2016 1;

h1b\_ans = UNION ans\_2011,ans\_2012,ans\_2013,ans\_2014,ans\_2015,ans\_2016;

dump h1b\_ans;

Output:

((DATA ENGINEER,SAN FRANCISCO, CALIFORNIA,2011),3)

((DATA ENGINEER,SAN FRANCISCO, CALIFORNIA,2012),7)

((DATA ENGINEER,SAN FRANCISCO, CALIFORNIA,2015),33)

((DATA ENGINEER,MENLO PARK, CALIFORNIA,2013),10)

((DATA ENGINEER,MENLO PARK, CALIFORNIA,2014),13)

((DATA ENGINEER,MENLO PARK, CALIFORNIA,2016),35)

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

Commands:

location = load '/home/shubha/PROJECT/h1b.csv' using CSV\_Storage() as

(s\_no: int,case\_status: chararray, employer\_name: chararray, soc\_name: chararray, job\_title: chararray, full\_time\_position: chararray,prevailing\_wage: int,year: chararray,

worksite: chararray, longitude: double, latitute: double);

location1 = foreach location generate $8, $1, $7;

location2 = filter location1 by $1 == 'CERTIFIED';

location3\_2011 = filter location2 by $2 =='2011';

location3\_2012 = filter location2 by $2 =='2012';

location3\_2013 = filter location2 by $2 =='2013';

location3\_2014 = filter location2 by $2 =='2014';

location3\_2015 = filter location2 by $2 =='2015';

location3\_2016 = filter location2 by $2 =='2016';

location4\_2011 = group location3\_2011 by ($0,$1,$2);

location4\_2012 = group location3\_2012 by ($0,$1,$2);

location4\_2013 = group location3\_2013 by ($0,$1,$2);

location4\_2014 = group location3\_2014 by ($0,$1,$2);

location4\_2015 = group location3\_2015 by ($0,$1,$2);

location4\_2016 = group location3\_2016 by ($0,$1,$2);

location5\_2011 = foreach location4\_2011 generate group, COUNT(location3\_2011.$1);

location5\_2012 = foreach location4\_2012 generate group, COUNT(location3\_2012.$1);

location5\_2013 = foreach location4\_2013 generate group, COUNT(location3\_2013.$1);

location5\_2014 = foreach location4\_2014 generate group, COUNT(location3\_2014.$1);

location5\_2015 = foreach location4\_2015 generate group, COUNT(location3\_2015.$1);

location5\_2016 = foreach location4\_2016 generate group, COUNT(location3\_2016.$1);

desc2011 = order location5\_2011 by $1 desc;

desc2012 = order location5\_2012 by $1 desc;

desc2013 = order location5\_2013 by $1 desc;

desc2014 = order location5\_2014 by $1 desc;

desc2015 = order location5\_2015 by $1 desc;

desc2016 = order location5\_2016 by $1 desc;

limit1 = limit g\_desc2011 5;

limit2 = limit g\_desc2012 5;

limit3 = limit g\_desc2013 5;

limit4 = limit g\_desc2014 5;

limit5 = limit g\_desc2015 5;

limit6 = limit g\_desc2016 5;

top5\_location\_ans = UNION limit1, limit2, limit3, limit4, limit5, limit6;

dump top5\_location\_ans;

Output:

year 2016

(NEW YORK, NEW YORK,CERTIFIED,2012),23737

(HOUSTON, TEXAS,CERTIFIED,2012),9963

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2012),6116

(CHICAGO, ILLINOIS,CERTIFIED,2012),5671

(ATLANTA, GEORGIA,CERTIFIED,2012),5565

year 2015

(NEW YORK, NEW YORK,CERTIFIED,2016),34639

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2016),13836

(HOUSTON, TEXAS,CERTIFIED,2016),13655

(ATLANTA, GEORGIA,CERTIFIED,2016),11678

(CHICAGO, ILLINOIS,CERTIFIED,2016),11064

year 2014

(NEW YORK, NEW YORK,CERTIFIED,2015),31266

(HOUSTON, TEXAS,CERTIFIED,2015),15242

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2015),12594

(ATLANTA, GEORGIA,CERTIFIED,2015),10500

(SAN JOSE, CALIFORNIA,CERTIFIED,2015),9589

year 2013

(NEW YORK, NEW YORK,CERTIFIED,2014),27634

(HOUSTON, TEXAS,CERTIFIED,2014),13360

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2014),9798

(SAN JOSE, CALIFORNIA,CERTIFIED,2014),8223

(ATLANTA, GEORGIA,CERTIFIED,2014),8213

year 2012

(NEW YORK, NEW YORK,CERTIFIED,2013),23537

(HOUSTON, TEXAS,CERTIFIED,2013),11136

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2013),7281

(SAN JOSE, CALIFORNIA,CERTIFIED,2013),6722

(ATLANTA, GEORGIA,CERTIFIED,2013),6377

year 2011

(NEW YORK, NEW YORK,CERTIFIED,2011),23172

(HOUSTON, TEXAS,CERTIFIED,2011),8184

(CHICAGO, ILLINOIS,CERTIFIED,2011),5188

(SAN JOSE, CALIFORNIA,CERTIFIED,2011),4713

(SAN FRANCISCO, CALIFORNIA,CERTIFIED,2011),4711

--------------------------------------------------------------------------------------------

3)Which industry(SOC\_NAME) has the most number of Data Scientist positions? [certified] (using hive)

Ans:

hive> select soc\_name,count(soc\_name) as cnt from h1b\_final where job\_title == 'DATA SCIENTIST' AND case\_status == 'CERTIFIED' group by soc\_name order by cnt desc;

Query ID = hduser\_20171018174038\_7563aac8-553c-42c6-9299-4e1f8c78b41d

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 17:40:47,449 Stage-1 map = 0%, reduce = 0%

2017-10-18 17:41:09,105 Stage-1 map = 17%, reduce = 0%

2017-10-18 17:41:30,173 Stage-1 map = 100%, reduce = 0%

2017-10-18 17:41:36,455 Stage-1 map = 50%, reduce = 0%

2017-10-18 17:41:48,877 Stage-1 map = 70%, reduce = 0%

2017-10-18 17:41:57,034 Stage-1 map = 100%, reduce = 0%

2017-10-18 17:41:58,059 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1310125937\_0001

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 17:42:00,397 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local438169756\_0002

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 1618041213 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 899731710 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

STATISTICIANS 369

COMPUTER AND INFORMATION RESEARCH SCIENTISTS 283

OPERATIONS RESEARCH ANALYSTS 237

Computer and Information Research Scientists 115

COMPUTER OCCUPATIONS, ALL OTHER 113

MATHEMATICIANS 107

Statisticians 92

SOFTWARE DEVELOPERS, APPLICATIONS 64

COMPUTER SYSTEMS ANALYSTS 59

Operations Research Analysts 55

SOFTWARE DEVELOPERS, SYSTEMS SOFTWARE 31

Software Developers, Applications 30

Computer Occupations, All Other 22

Computer Systems Analysts 17

Software Developers, Systems Software 11

Mathematicians 8

NA 8

Database Administrators 7

MATHEMATICAL TECHNICIANS 7

MARKET RESEARCH ANALYSTS AND MARKETING SPECIALISTS 7

DATABASE ADMINISTRATORS 5

ECONOMISTS 5

FINANCIAL SPECIALISTS, ALL OTHER 5

BIOLOGICAL SCIENTISTS, ALL OTHER 5

COMPUTER PROGRAMMERS 4

COMPUTER AND INFORMATION SYSTEMS MANAGERS 3

INDUSTRIAL-ORGANIZATIONAL PSYCHOLOGISTS 3

Biological Scientists, All Other 2

COMPUTER NETWORK ARCHITECTS 2

SALES ENGINEERS 2

Computer Programmers 2

ELECTRICAL ENGINEERS 2

Economists 2

Management Analysts 2

ENGINEERS, ALL OTHER 2

SURVEY RESEARCHERS 2

Survey Researchers 1

SOFTWARE DEVELOPERS, APPLICATIONS, R&D 1

Materials Scientists 1

Market Research Analysts and Marketing Specialists 1

MATERIALS SCIENTISTS 1

Engineers, All Other 1

Computer Software Engineers, Applications 1

COMPUTER SYSTEMS ANALYST 1

COMPUTER & INFORMATION RESEARCH SCIENTISTS 1

CLINICAL DATA MANAGERS 1

CIVIL ENGINEERS 1

ACTUARIES 1

MEDICAL SCIENTISTS, EXCEPT EPIDEMIOLOGISTS 1

MATERIALS ENGINEERS 1

MANAGEMENT ANALYSTS 1

Life Scientists, All Other 1

Financial Analysts 1

Electrical Engineers 1

Credit Analysts 1

Computer and Information Scientists, Research 1

COMPUTER & INFORMATION RESEARCH SCIENTIST 1

BUSINESS INTELLIGENCE ANALYSTS 1

--------------------------------------------------------------------------------------------

4)Which top 5 employers file the most petitions each year? - Case Status - ALL

(using hive)

hive> select year,employer\_name,count(year) as cnt from h1b\_final group by year employer\_name order by cnt desc limit 5;

FAILED: ParseException line 1:74 missing EOF at 'employer\_name' near 'year'

hive> select year,employer\_name,count(year) as cnt from h1b\_final group by year, employer\_name order by cnt desc limit 5;

Query ID = hduser\_20171018175659\_362e0454-cc16-4dff-a8fe-a0291a5c3c5d

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 17:57:01,083 Stage-1 map = 0%, reduce = 0%

2017-10-18 17:57:07,103 Stage-1 map = 33%, reduce = 0%

2017-10-18 17:57:08,105 Stage-1 map = 100%, reduce = 0%

2017-10-18 17:57:14,136 Stage-1 map = 100%, reduce = 50%

2017-10-18 17:57:15,142 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1497641901\_0003

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 17:57:16,624 Stage-2 map = 0%, reduce = 0%

2017-10-18 17:57:18,647 Stage-2 map = 100%, reduce = 0%

2017-10-18 17:57:19,650 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local2035789745\_0004

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 3417504633 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 1799463420 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

Output:

year 2012

(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

year 2013

(INFOSYS LIMITED,2014) 23759

(TATA CONSULTANCY SERVICES LIMITED,2014) 14098

(WIPRO LIMITED,2014) 8365

(DELOITTE CONSULTING LLP,2014) 7017

(ACCENTURE LLP,2014) 5498

--------------------------------------------------------------------------------------------

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

a) for all the applications

Ans:

Year=2011

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2011' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018180710\_02382279-8bc2-4346-82aa-12a39a79ea35

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:07:11,806 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:07:13,811 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:07:17,840 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local155577283\_0005

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:07:19,323 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local1174396054\_0006

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 5216968053 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 2699195130 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 31799 2011

SOFTWARE ENGINEER 12763 2011

COMPUTER PROGRAMMER 8998 2011

SYSTEMS ANALYST 8644 2011

BUSINESS ANALYST 3891 2011

COMPUTER SYSTEMS ANALYST 3698 2011

ASSISTANT PROFESSOR 3467 2011

PHYSICAL THERAPIST 3377 2011

SENIOR SOFTWARE ENGINEER 2935 2011

SENIOR CONSULTANT 2798 2011

------------------------------------------------------------------

YEAR=2012

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2012' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018181223\_f921a74d-9be0-401c-a899-ca78cb18f0c9

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:12:24,639 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:12:26,644 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:12:28,659 Stage-1 map = 50%, reduce = 0%

2017-10-18 18:12:29,664 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1066328847\_0007

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:12:31,199 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local1037328093\_0008

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 7016431473 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 3598926840 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 33066 2012

SOFTWARE ENGINEER 14437 2012

COMPUTER PROGRAMMER 9629 2012

SYSTEMS ANALYST 9296 2012

BUSINESS ANALYST 4752 2012

COMPUTER SYSTEMS ANALYST 4706 2012

SOFTWARE DEVELOPER 3895 2012

PHYSICAL THERAPIST 3871 2012

ASSISTANT PROFESSOR 3801 2012

SENIOR CONSULTANT 3737 2012

-------------------------------------------------------------------------------------------------------------------------

YEAR=2013

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2013' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018181337\_e3163651-bc26-4e8a-abd6-b28fd5822ab4

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:13:39,226 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:13:41,233 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:13:44,246 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local2070857226\_0009

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:13:45,717 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local2121467462\_0010

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 8815894893 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 4498658550 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 33880 2013

SOFTWARE ENGINEER 15680 2013

COMPUTER PROGRAMMER 11271 2013

SYSTEMS ANALYST 8714 2013

TECHNOLOGY LEAD - US 7853 2013

TECHNOLOGY ANALYST - US 7683 2013

BUSINESS ANALYST 5716 2013

COMPUTER SYSTEMS ANALYST 5043 2013

SOFTWARE DEVELOPER 5026 2013

SENIOR CONSULTANT 4326 2013

-------------------------------------------------------------------------------------------------------------------------

YEAR=2014

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2014' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018181459\_a1ca28cc-7962-4cc4-8ea0-aa55cbe9c0e2

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:15:01,039 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:15:04,045 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:15:06,052 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local360855883\_0011

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:15:07,422 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local1335606631\_0012

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 10615358313 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 5398390260 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 43114 2014

SOFTWARE ENGINEER 20500 2014

COMPUTER PROGRAMMER 14950 2014

SYSTEMS ANALYST 10194 2014

SOFTWARE DEVELOPER 7337 2014

BUSINESS ANALYST 7302 2014

COMPUTER SYSTEMS ANALYST 6821 2014

TECHNOLOGY LEAD - US 5057 2014

TECHNOLOGY ANALYST - US 4913 2014

SENIOR CONSULTANT 4898 2014

--------------------------------------------------------------------------------------------------------------------------

YEAR=2015

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2015' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018181607\_92c6f248-a5ee-4220-824d-9e9e56c5c51b

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:16:09,274 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:16:12,281 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:16:15,308 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local874279348\_0013

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:16:16,699 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local531813301\_0014

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 12414821733 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 6298121970 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 53436 2015

SOFTWARE ENGINEER 27259 2015

COMPUTER PROGRAMMER 14054 2015

SYSTEMS ANALYST 12803 2015

SOFTWARE DEVELOPER 10441 2015

BUSINESS ANALYST 8853 2015

TECHNOLOGY LEAD - US 8242 2015

COMPUTER SYSTEMS ANALYST 7918 2015

TECHNOLOGY ANALYST - US 7014 2015

SENIOR SOFTWARE ENGINEER 6013 2015

-------------------------------------------------------------------------------------------------------------------------

YEAR=2016

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2016' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018181722\_3cc312a0-13f5-4ab9-b3b1-0f6a93b3bd9f

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:17:24,000 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:17:28,010 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:17:30,024 Stage-1 map = 100%, reduce = 50%

2017-10-18 18:17:31,029 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local468071267\_0015

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:17:32,378 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local1872084903\_0016

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 14214285153 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 7197853680 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 53743 2016

SOFTWARE ENGINEER 30668 2016

SOFTWARE DEVELOPER 14041 2016

SYSTEMS ANALYST 12314 2016

COMPUTER PROGRAMMER 11668 2016

BUSINESS ANALYST 9167 2016

COMPUTER SYSTEMS ANALYST 6900 2016

SENIOR SOFTWARE ENGINEER 6439 2016

DEVELOPER 6084 2016

TECHNOLOGY LEAD - US 5410 2016

--------------------------------------------------------------------------------------------------------------------------

b) for only certified applications.

YEAR=2011

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2011'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182103\_4af89f03-a78f-4c20-9d4f-9ee2346fb48a

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:21:04,782 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:21:06,784 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:21:08,791 Stage-1 map = 100%, reduce = 50%

2017-10-18 18:21:09,797 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1574230234\_0017

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:21:11,058 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local906853177\_0018

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 16013748573 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 8097585390 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 28806 2011

SOFTWARE ENGINEER 11224 2011

COMPUTER PROGRAMMER 8038 2011

SYSTEMS ANALYST 7850 2011

BUSINESS ANALYST 3444 2011

COMPUTER SYSTEMS ANALYST 3152 2011

ASSISTANT PROFESSOR 3050 2011

PHYSICAL THERAPIST 2911 2011

SENIOR SOFTWARE ENGINEER 2595 2011

SENIOR CONSULTANT 2585 2011

-------------------------------------------------------------------------------------------------------------------------

YEAR=2012

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2012'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182244\_bd7254ae-54c7-4426-9a37-3f0c8c922664

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:22:46,193 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:22:48,199 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:22:51,208 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local561541751\_0019

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:22:52,501 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local190299792\_0020

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 17813211993 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 8997317100 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 29226 2012

SOFTWARE ENGINEER 12273 2012

COMPUTER PROGRAMMER 8483 2012

SYSTEMS ANALYST 8399 2012

BUSINESS ANALYST 4144 2012

COMPUTER SYSTEMS ANALYST 4084 2012

SENIOR CONSULTANT 3420 2012

SOFTWARE DEVELOPER 3290 2012

PHYSICAL THERAPIST 3284 2012

ASSISTANT PROFESSOR 3033 2012

-------------------------------------------------------------------------------------------------------------------------

YEAR=2013

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2013'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182410\_32163baf-dc46-4b17-9068-f2ecfd4d315e

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:24:11,808 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:24:13,812 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:24:15,820 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local985714238\_0021

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:24:17,215 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local643515139\_0022

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 19612675413 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 9897048810 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 29906 2013

SOFTWARE ENGINEER 12973 2013

COMPUTER PROGRAMMER 10202 2013

SYSTEMS ANALYST 7850 2013

TECHNOLOGY LEAD - US 7809 2013

TECHNOLOGY ANALYST - US 7641 2013

BUSINESS ANALYST 4993 2013

COMPUTER SYSTEMS ANALYST 4554 2013

SOFTWARE DEVELOPER 4316 2013

SENIOR CONSULTANT 3996 2013

-------------------------------------------------------------------------------------------------------------------------

YEAR=2014

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2014'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182549\_a3f13e1a-c737-4d40-9b17-8ba6bb37bccc

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:25:50,795 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:25:54,805 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:25:55,812 Stage-1 map = 100%, reduce = 50%

2017-10-18 18:25:56,816 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1697180713\_0023

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:25:58,094 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local1383600582\_0024

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 21412138833 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 10796780520 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 38625 2014

SOFTWARE ENGINEER 17278 2014

COMPUTER PROGRAMMER 13796 2014

SYSTEMS ANALYST 9161 2014

BUSINESS ANALYST 6529 2014

SOFTWARE DEVELOPER 6473 2014

COMPUTER SYSTEMS ANALYST 6204 2014

TECHNOLOGY LEAD - US 5055 2014

TECHNOLOGY ANALYST - US 4911 2014

SENIOR CONSULTANT 4535 2014

-------------------------------------------------------------------------------------------------------------------------

YEAR=2015

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2015'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182722\_deef00d3-c118-4f23-acb6-d36b160c6133

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:27:23,755 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:27:26,760 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:27:29,777 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1896676276\_0025

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:27:31,104 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local271083600\_0026

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 23211602253 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 11696512230 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 48203 2015

SOFTWARE ENGINEER 23352 2015

COMPUTER PROGRAMMER 12971 2015

SYSTEMS ANALYST 11498 2015

SOFTWARE DEVELOPER 9343 2015

TECHNOLOGY LEAD - US 8238 2015

BUSINESS ANALYST 7919 2015

COMPUTER SYSTEMS ANALYST 7234 2015

TECHNOLOGY ANALYST - US 7009 2015

SENIOR SOFTWARE ENGINEER 5324 2015

-------------------------------------------------------------------------------------------------------------------------

YEAR=2016

hive> select job\_title,count(job\_title) as cnt,year from h1b\_final where year =='2016'And case\_status == 'CERTIFIED' group by year,job\_title order by cnt desc limit 10;

Query ID = hduser\_20171018182842\_4f8d0ecd-5c8d-47ee-b20d-ab160b0fe959

Total jobs = 2

Launching Job 1 out of 2

Number of reduce tasks not specified. Estimated from input data size: 2

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:28:43,437 Stage-1 map = 0%, reduce = 0%

2017-10-18 18:28:46,444 Stage-1 map = 100%, reduce = 0%

2017-10-18 18:28:49,459 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_local1321266294\_0027

Launching Job 2 out of 2

Number of reduce tasks determined at compile time: 1

In order to change the average load for a reducer (in bytes):

set hive.exec.reducers.bytes.per.reducer=<number>

In order to limit the maximum number of reducers:

set hive.exec.reducers.max=<number>

In order to set a constant number of reducers:

set mapreduce.job.reduces=<number>

Job running in-process (local Hadoop)

2017-10-18 18:28:50,797 Stage-2 map = 100%, reduce = 100%

Ended Job = job\_local2138515362\_0028

MapReduce Jobs Launched:

Stage-Stage-1: HDFS Read: 25011065673 HDFS Write: 0 SUCCESS

Stage-Stage-2: HDFS Read: 12596243940 HDFS Write: 0 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

PROGRAMMER ANALYST 47964 2016

SOFTWARE ENGINEER 25890 2016

SOFTWARE DEVELOPER 12474 2016

SYSTEMS ANALYST 10986 2016

COMPUTER PROGRAMMER 10528 2016

BUSINESS ANALYST 8175 2016

COMPUTER SYSTEMS ANALYST 6205 2016

DEVELOPER 5912 2016

SETECHNOLOGY LEAD - US 5405 2016

NIOR SOFTWARE ENGINEER 5630 2016

--------------------------------------------------------------------------------------------

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.(using mapreduce)

Program:

package h1b\_Q6

import java.io.\*;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.conf.\*;

import org.apache.hadoop.fs.\*;

import org.apache.hadoop.mapreduce.lib.input.\*;

import org.apache.hadoop.mapreduce.lib.output.\*;

public static class MapClass extends Mapper<LongWritable,Text,Text,Text>

{

public void map(LongWritable key, Text value, Context context)

{

try{

String[] str = value.toString().split("\t");

String year = str[7];

String case\_status = str[1];

context.write(new Text(year),new Text(case\_status));

}

catch(Exception e)

{

System.out.println(e.getMessage());

}

}

}

public static class ReduceClass extends Reducer<Text,Text,Text,Text>

{

public void reduce(Text key, Iterable<Text> values,Context context) throws IOException, InterruptedException

{

long totalcount= 0,certified\_count=0,certified\_withdrawn\_count=0,denied\_count=0,withdrawn\_count=0;

double certified\_AvgPerc=0,certified\_withdrawn\_AvgPerc=0,denied\_AvgPerc=0,withdrawn\_AvgPerc=0;

for (Text T : values)

{

totalcount++;

String case\_status=T.toString();

if(case\_status.equals("CERTIFIED"))

{

certified\_count++;

}

else if(case\_status.equals("CERTIFIED-WITHDRAWN"))

{

certified\_withdrawn\_count++;

}

else if(case\_status.equals("WITHDRAWN"))

{

withdrawn\_count++;

}

else

{

denied\_count++;

}

}

certified\_AvgPerc = ((double)certified\_count/(double)totalcount)\*100;

certified\_withdrawn\_AvgPerc = ((double)certified\_withdrawn\_count/(double)totalcount)\*100;

withdrawn\_AvgPerc = ((double)withdrawn\_count/(double)totalcount)\*100;

denied\_AvgPerc = ((double)denied\_count/(double)totalcount)\*100;

String COUNT=totalcount+"\t"+certified\_count+"\t"+certified\_AvgPerc+"\t"+certified\_withdrawn\_count+"\t"

+certified\_withdrawn\_AvgPerc+"\t"+withdrawn\_count+"\t"+withdrawn\_AvgPerc+"\t"+denied\_count+"\t"+ denied\_AvgPerc;

context.write(key,new Text(COUNT));

}

}

public class percentage\_avg

{

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

Job job = new Job(conf, "H1B DATA");

job.setJarByClass(percentage\_avg.class);

job.setMapperClass(MapClass.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setReducerClass(ReduceClass.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

Output:

2011 358767 307936 85.83175152675692 11596 3.2321813321738064 10105 2.816591269542628 29130 8.119475871526644

2012 415607 352668 84.85612609989725 31118 7.487361858678993 10725 2.5805628875355806 21096 5.0759491538881685

2013 442114 382951 86.61815730784367 35432 8.014222576077664 11590 2.621495813297023 12141 2.7461243027816358

2014 519427 455144 87.62424748809747 36350 6.99809597883822 16034 3.086863024063054 11899 2.2907935090012645

2015 618727 547278 88.45225761927313 41071 6.637984119005635 19455 3.144359305477214 10923 1.7653989562440302

2016 647803 569646 87.93506667922193 47092 7.269493966529948 21890 3.3791137120389996 9175 1.4163256422091284

--------------------------------------------------------------------------------------------

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

(using mapreduces)

Program:

**package** h1b\_Q7;

**import** java.io.\*;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.LongWritable;

**import** org.apache.hadoop.mapreduce.Job;

**import** org.apache.hadoop.mapreduce.Mapper;

**import** org.apache.hadoop.mapreduce.Reducer;

**import** org.apache.hadoop.conf.\*;

**import** org.apache.hadoop.fs.\*;

**import** org.apache.hadoop.mapreduce.lib.input.\*;

**import** org.apache.hadoop.mapreduce.lib.output.\*;

**public** **class** h1b\_data

{

**public** **static** **class** mymap **extends** Mapper<LongWritable,Text, Text,IntWritable>

{

**public** **void** map (LongWritable key,Text val,Context context) **throws** InterruptedException,IOException

{

String[] colum = val.toString().split(",(?=([^\"]\*\"[^\"]\*\")\*[^\"]\*$)");

String year=colum[7];

**int** count =1;

context.write(**new** Text(year), **new** IntWritable(count));

}

}

**public** **static** **class** myreduce **extends** Reducer<Text,IntWritable,Text,IntWritable>

{

**public** **void** reduce (Text key,Iterable<IntWritable>val,Context context)**throws** InterruptedException,IOException

{

**int** count=0;

**for**(IntWritable T:val)

{

count++;

}

context.write(key,**new** IntWritable(count));

}

}

**public** **static** **void** main(String[] args) **throws** Exception

{

Configuration conf = **new** Configuration();

Job job = **new** Job(conf, "h1b\_data");

job.setJarByClass(h1b\_data.**class**);

job.setMapperClass(mymap.**class**);

job.setReducerClass(myreduce.**class**);

job.setMapOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(IntWritable.**class**);

job.setNumReduceTasks(1);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(IntWritable.**class**);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

System.*exit*(job.waitForCompletion(**true**) ? 0 : 1);

}

}

Output:

2011 358767

2012 415607

2013 442114

2014 519427

2015 618727

2016 647803

--------------------------------------------------------------------------------------------

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.] (using hive)

Command:

Select job\_title ,year,Avg(prevailing\_wage) from h1b\_final where year == ‘2011’

AND full\_time\_position == ‘Y’ AND case\_status == ‘CERTIFIED’

group by job\_title,year;

Output:

(DEVELOPER (SOFTWARE SYSTEMS APPLICATIONS),2011,Y) 1.765608E8

(SYSTEMS ENGINEER (DIAGNOSTICS),2011,Y) 9.552608E7

(SENIOR COST CONSULTANT,2011,Y) 8.561904E7

(RADIATION ONCOLOGIST,2011,Y) 7.660264025E7

(IT ADMINSTRATIVE ASSISTANT,2011,Y) 7.1114118E7

(VERIFICATION AND VALIDATION ENGINEER,2011,Y) 6.6867819E7

(SENIOR STAFF TEACHER,2011,Y) 3.3221084E7

(TECHNICAL SOFTWARE CONSULTANT,2011,Y) 1.6386115E7

(PRINCIPAL ARCHITECT,2011,Y) 9925884.705882354

(PHYSICIAN - INTERNIST,2011,Y) 6934388.833333333

(ATTENDING PHYSICIAN,2011,Y) 2745786.7590361447

(TEACHER ASSISTANT/TEACHER AID,2011,Y) 2710300.0

(TEACHER ASSISTANT/,2011,Y) 2710300.0

(FINANCIAL SPECIALIST,2011,Y) 1787969.5972222222

(MANAGER, REPORTING AND BUSINESS ANALYTICS,2011,Y) 1418076.0

(APPLICATIONS ENGINEER,2011,Y) 647993.6202531646

(PHYSICIAN,2011,Y) 558064.8128

(CARDIOLOGIST/INTERVENTIONAL CARDIO(AREA MANAGER, PHARMACEUTICAL PACKAGING,2011,Y) 2.1298784E8

LOGIST,2011,Y) 432740.0

(FURNITURE DESIGNER,2011,Y) 422622.6

(SENIOR LOGISTICS DEVELOPER,2011,Y) 391218.0

(CARDIOLOGIST - NONINVASIVE,2011,Y) 389100.0

(PHYSICIAN/CLINCAL FACULTY,2011,Y) 334700.0

(NEUROSURGERY PHYSICIAN,2011,Y) 334700.0

(SENIOR VICE PRESIDENT, CONTROL DEVICES,2011,Y) 323008.0

(GASTROENTEROLOGIST/ADVANCED ENDOSCOPIST,2011,Y) 318348.0

(INVASIVE CARDIOLOGIST (PHYSICIAN),2011,Y) 317857.0

(INVASIVE, NON-INTERVENTIONAL CARDIOLOGIST,2011,Y) 313418.0

(GENERAL TRAUMA SURGEON,2011,Y) 310472.0

(MARRIAGE FAMILY THERAPIST,2011,Y) 309100.0

(PHYSICIAN/INTENSIVIST,2011,Y) 293251.0

(SPECIAL EDUCATION TEACHER - ELEMENTARY,2011,Y) 289902.9090909091

(PHYSICIAN/HEMATOLOGIST/ONCOLOGIST,2011,Y) 279205.0

(PHYSICIAN / HEMATOLOGIST / ONCOLOGIST,2011,Y) 279205.0

(BILINGUAL ELEMENTARY TEACHER,2011,Y) 277687.8680203046

(NEUROSURGEON,2011,Y) 273642.4166666667

(STAFF ACCOUNTANT,2011,Y) 264885.0208333333

(VICE PRESIDENT, CLINICAL DEVELOPMENT,2011,Y) 260456.66666666666

(ELEMENTARY SCHOOL SPECIAL EDUC. TEACHER,2011,Y) 259710.0

(MANAGING DIRECTOR AND VICE PRESIDENT, NIKE FOUNDAT,2011,Y) 259700.0

(ASSIST.MEMBER, CLINICAL FACULTY/SURGEON,2011,Y) 253800.0

(SR.MEMBER/CLINICAN SCIENTIST PATHWAY, CHAIR OF NEU,2011,Y) 253800.0

(COMPUTER SOFTWARE ENGINEER,2011,Y) 247481.4662420382

(PHYSICIAN OPHTHALMOLOGIST,2011,Y) 245440.0

(DIRECTOR OF LIVER TRANSPLANT SURGICAL PROGRAM,2011,Y) 244026.0

(ORTHOPAEDIC TRAUMATOLOGIST,2011,Y) 243464.0

(PHYSICIAN - NEUROLOGY,2011,Y) 242406.0

(DIRECTOR AND ADVISOR TO THE BOARD OF DIRECTORS,2011,Y) 242216.0

(MANAGING DIRECTOR, CORPORATE SALES,2011,Y) 242216.0

(GENERAL SURGEON PHYSICIAN,2011,Y) 240820.0

(VICE PRESIDENT, MARKETING, COMMUNICATIONS AND GOVE,2011,Y) 240011.0

(PHYSICIAN / NEUROLOGIST,2011,Y) 239096.0

(ORTHOPAEDIC TRAUMA SURGEON,2011,Y) 237411.0

(CHIEF OPERATING OFFICER/CHIEF TECHNICAL OFFICER,2011,Y) 237058.0

(VICE PRESIDENT OF ENGINEERING AND OPERATIONS,2011,Y) 235352.0

(ASSISTANT PROFESSOR IN VASCULAR SURGERY,2011,Y) 235000.0

(HEMATOLOGIST/MEDICAL ONCOLOGIST,2011,Y) 234437.0

(VICE PRESIDENT, GENERAL MANAGER,2011,Y) 234354.0

(VICE PRESIDENT, MARKET ACCESS - GLOBAL DIABETES DI,2011,Y) 234354.0

(CHIEF MARKETING OFFICER - INTERNATIONAL, BASKIN RO,2011,Y) 233563.0

(EXECUTIVE VICE PRESIDENT MEDICAL AFFAIRS,2011,Y) 233563.0

(VICE PRESIDENT, SBU - COMMERCIAL AEROSPACE,2011,Y) 231462.0

(EXECUTIVE VICE PRESIDENT, COMMUNICATION, ADVOCACY ,2011,Y) 231462.0

(PROJECTS IMPLEMENTATION MANAGER,2011,Y) 231462.0

(PRESIDENT AND CHIEF OPERATING OFFICER,2011,Y) 231462.0

(VP & GENERAL MANAGER - SOUTH FLORIDA,2011,Y) 231275.0

(INTERNAL MEDICINE/ GASTROENTEROLOGY PHYSICIAN,2011,Y) 230693.0

(INTERNAL MEDICINE/GASTROENTEROLOGY PHYSICIAN,2011,Y) 230693.0

(EXECUTIVE V. PRESIDENT/GLOBAL CHIEF MARKETING OFFI,2011,Y) 230464.0

(VICE PRESIDENT-SALES & MARKETING,2011,Y) 230464.0

(PRESIDENT, TELEMUNDO,2011,Y) 230464.0

(CFO/COO,2011,Y) 230464.0

(PEDIATRIC RADIOLOGIST- FACULTY,2011,Y) 230069.0

(TRANSPLANT HEPATOLOGIST,2011,Y) 229341.0

(VICE PRESIDENT RESEARCH AND DEVELOPMENT, NEW OPPOR,2011,Y) 229050.0

(PHYSICIAN - DERMATOLOGIST,2011,Y) 229008.0

(EXECUTIVE VICE PRESIDENT, OPERATIONS,2011,Y) 228571.0

(CHIEF EXECUTIVE OFFICER AND MEDICAL DIRECTOR,2011,Y) 228571.0

(TMHPO PHYSICIAN- HEART FAILURE,2011,Y) 228238.0

(PHYSICIAN (RADIOLOGIST),2011,Y) 228165.5

(EXECUTIVE PROJECT DIRECTOR,2011,Y) 226782.0

(ASSOCIATE MEDICAL DIRECTOR / PREHOSPITAL MEDICAL D,2011,Y) 226595.0

(VICE PRESIDENT TRAVEL & TRANSPORTATION,2011,Y) 226325.0

(OPHTHALMOLOGIST/ASSISTANT PROFESSOR OF MEDICINE,2011,Y) 225930.0

(ORTHOPEDIC TRAUMA SURGEON,2011,Y) 225930.0

(DIRECTOR & GENERAL MANAGER, LATIN AMERICA,2011,Y) 225451.0

(VICE PRESIDENT, MARKETING AND CORPORATE COMMUNICAT,2011,Y) 224869.0

(EVP & WEALTH ACCUMULATION AND DIVERSIFICATION EXEC,2011,Y) 224182.0

(PRESIDENT THE AMERICAS,2011,Y) 224099.0

(CHIEF OPERATING OFFICER AND CHIEF FINANCIAL OFFICE,2011,Y) 223746.0

(CEO, OPERATIONS,2011,Y) 223746.0

(VP, TRANSLATIONAL & EXPERIMENTAL MEDICINE HEAD, IN,2011,Y) 223267.0

(PULMONARY, CRITICAL CARE AND SLEEP MEDICINE PHYS.,2011,Y) 222800.0

(PRESIDENT - STEELSCAPE,2011,Y) 222373.0

(PHYSICIAN, INTERNAL MEDICINE & PULMONOLOGY,2011,Y) 222352.0

(MAKO ENERGY CORPORATION,2011,Y) 222269.0

(ONCOLOGY SR. DIRECTOR CLINICAL RESEARCH PHYSICIAN,2011,Y) 222061.0

(PULMONARY / CRITICAL CARE PHYSICIAN,2011,Y) 221146.0

(DIVISIONAL MANAGING DIRECTOR,2011,Y) 220147.0

(PRESIDENT AND CHAIRMAN,2011,Y) 220147.0

(SENIOR VICE PRESIDENT & CHIEF TECHNOLOGY OFFICER,2011,Y) 220022.0

(SENIOR VICE PRESIDENT, GENERAL MANAGER OF BSS BUSI,2011,Y) 220000.0

(INTERVENTIONAL PAIN MANAGEMENT SPECIALIST - PHYSIC,2011,Y) 219773.0

(BONE MARROW TRANSPLANT PHYSICIAN,2011,Y) 219773.0

(PULMONARY/CRITICAL CARE/SLEEP PHYSICIAN,2011,Y) 219461.0

(STAFF SURGEON, GENERAL SURGERY,2011,Y) 218982.0

(EXECUTIVE VICE PRESIDENT/EXECUTIVE CREATIVE DIRECT,2011,Y) 218899.0

(CHIEF OPERATIONS OFFICER OF NORTH AMERICA & ASIA T,2011,Y) 218899.0

(SENIOR VICE PRESIDENT, ENGINEERING SERVICES,2011,Y) 218899.0

(MECHANICAL DIRECTOR - CARDIOVASCULAR INTENSIVE CAR,2011,Y) 218878.0

(MEDICAL DIRECTOR - CARDIOVASCULAR INTENSIVE CARE U,2011,Y) 218878.0

(MEDICAL DIRECTOR-CARDIOVASCULAR INTENSIVE CARE UNI,2011,Y) 218878.0

(STAFF RADIOLOGIST AND NEURORADIOLOGIST,2011,Y) 218566.0

(CRUDE OIL TRADER,2011,Y) 218442.0

(SENIOR DIRECTOR, HEAD OF INVESTORS INDUSTRIES,2011,Y) 218421.0

(VICE PRESIDENT, NORTH AMERICA DENTAL SALES PLANNIN,2011,Y) 218421.0

(CHIEF BUSINESS DEVELOPMENT OFFICER,2011,Y) 218421.0

(SR DIRECTOR, REGULATORY MGMT,2011,Y) 218336.0

(GENERAL SURGEON/DIRECTOR OF GENERAL SURGERY,2011,Y) 218150.0

(VASCULAR SURGEON/CLINICAL TRAINING FACILITY,2011,Y) 218088.0

(PHYSICIAN (GASTROENOLOGY),2011,Y) 218000.0

(STAFF PSYCHIATRIST (SAFETY),2011,Y) 217752.0

(PHYSICIAN (CARDIOLOGY/ CARDIAC ELECTROPHYSIOLOGY),2011,Y) 217693.0

(CARDIOLOGY PHYSICIAN (NON-INTERVENTIONAL),2011,Y) 217048.0

(EXECUTIVE VP, CHIEF INFORMATION & PERFORMANCE OFFI,2011,Y) 216445.0

(EXECUTIVE VP, PRESIDENT OF LATIN AMERICA & THE C,2011,Y) 216445.0

(EXECUTIVE VICE PRESIDENT REAL ESTATE INVESTMENTS,2011,Y) 216445.0

(VP, UNIVISION SPORTS,2011,Y) 216445.0

(SENIOR EXECUTIVE VICE PRESIDENT - MERGER & ACQUISI,2011,Y) 216445.0

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) ? (using mapreduces)

Program:

package Query9;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import java.util.TreeMap;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Q9mapper extends Mapper<LongWritable, Text, Text, Text>{

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException

{

String parts[] =value.toString().split(",");

String status = parts[1];

String Employer =parts[2];

context.write(new Text(Employer), new Text(status));

}

}

public class Q9reducer extends Reducer<Text,Text,NullWritable,Text>

{

private TreeMap<Double, String> topten = new TreeMap<>();

public void reduce(Text key, Iterable<Text> value, Context context){

double total =0;

double successrate=0;

for (Text val:value)

{

String status = val.toString();

if(status.equals("\"CERTIFIED\"") || status.equals("\"CERTIFIED WITHDRAWN\""))

{

total++ ;

successrate++;

}

else

total++;

}

double rate = (successrate/total)\*100;

if(rate >=70 && total >=1000){

String op = key.toString()+ ","+String.format("%.0f",total)+"," + String.format("%f %%",rate);

topten.put(rate, op);

}

}

protected void cleanup(Context context) throws IOException, InterruptedException{

for(String val : topten.values()){

context.write(NullWritable.get(),new Text(val));

}

}

}

public class Q9driver {

public static void main(String[] args) throws Exception

{

Configuration conf =new Configuration();

conf.set("mapreduce.output.textoutputformat.separator", ",");

Job job=Job.getInstance(conf);

job.setJarByClass(Q9driver.class);

//job.setJobName("POSJoin");

job.setMapperClass(Q9mapper.class);

//job.addCacheFile(new Path("AFINN.txt").toUri());

//job.setNumReduceTasks(0);

job.setReducerClass(Q9reducer.class);

job.setOutputKeyClass(NullWritable.class);

job.setOutputValueClass(Text.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job,new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

Output:

THE UNIVERSITY OF IOWA",1569,70.554493 %

"VERINON TECHNOLOGY SOLUTIONS LTD.",1245,70.923695 %

"GENESIS ELDERCARE REHABILITATION SERVICES,1458,71.330590 %

"DEUTSCHE BANK SECURITIES INC.",1170,71.880342 %

"SUNERA TECHNOLOGIES,1464,72.131148 %

"GENERAL HOSPITAL CORPORATION",1429,72.288314 %

"EMORY UNIVERSITY",1680,72.559524 %

"CITIBANK,2173,72.618500 %

"COLUMBIA UNIVERSITY",1841,73.438349 %

"WASHINGTON UNIVERSITY IN ST. LOUIS",1576,73.477157 %

"UNIVERSITY OF MARYLAND COLLEGE PARK",1354,73.485968 %

"MEDTRONIC,1050,73.523810 %

"THE UNIVERSITY OF CHICAGO",1277,73.610023 %

"T-MOBILE USA,1457,73.781743 %

"UNIVERSITY OF UTAH",1069,73.900842 %

"THE UNIV. OF ALA. AT BIRMINGHAM (UAB)",1288,74.145963 %

"NEW YORK UNIVERSITY SCHOOL OF MEDICINE",1126,74.245115 %

"YAHOO! INC.",3348,74.283154 %

"GLOBALFOUNDRIES U.S. INC.",1391,74.694464 %

"GOOGLE INC.",16473,74.989377 %

"UNIVERSITY OF MICHIGAN",2893,75.665399 %

"INTONE NETWORKS INC.",1575,75.936508 %

"UNIVERSITY OF WISCONSIN-MADISON",1115,75.964126 %

"SCHLUMBERGER TECHNOLOGY CORPORATION",2310,76.017316 %

"NORTHWESTERN UNIVERSITY",1439,76.233495 %

"BROADCOM CORPORATION",2862,76.240391 %

"BRIGHAM AND WOMEN'S HOSPITAL",1117,76.454790 %

"SYMANTEC CORPORATION",2290,77.336245 %

"SEARS HOLDINGS MANAGEMENT CORPORATION",1105,77.647059 %

"DUKE UNIVERSITY AND MEDICAL CENTER",1330,77.744361 %

"BLACKROCK FINANCIAL MANAGEMENT,1048,77.767176 %

"MICRON TECHNOLOGY,1934,78.076525 %

"GLOBALLOGIC,1001,78.121878 %

"UNIVERSITY OF FLORIDA",1429,78.796361 %

"IBM CORPORATION",13276,79.255800 %

"HITACHI CONSULTING CORPORATION",2854,79.257183 %

"YALE UNIVERSITY",1852,79.265659 %

"JOHNS HOPKINS UNIVERSITY",1823,80.142622 %

"CUMMINS INC.",4737,80.156217 %

"SMARTPLAY,1447,80.165860 %

"NETAPP,1870,80.213904 %

"HOWARD HUGHES MEDICAL INSTITUTE",1135,80.881057 %

"TEXAS INSTRUMENTS INCORPORATED",1780,81.011236 %

"UNIVERSITY OF CALIFORNIA,6739,81.020923 %

"IBM INDIA PRIVATE LIMITED",34219,81.080686 %

"UNIVERSITY OF PITTSBURGH",1632,81.127451 %

"HORIZON TECHNOLOGIES INC",1731,81.571346 %

"DELL MARKETING L.P.",1532,81.592689 %

"ORACLE FINANCIAL SERVICES SOFTWARE,1532,81.723238 %

"PERSISTENT SYSTEMS,3232,81.775990 %

"QUALCOMM ATHEROS,1274,82.339089 %

"AVANT HEALTHCARE PROFESSIONALS",1006,82.504970 %

"AKAMAI TECHNOLOGIES,1092,82.600733 %

"CITRIX SYSTEMS,1044,82.662835 %

"MASTECH,5235,82.769819 %

"IDEXCEL,1364,82.991202 %

"LINKEDIN CORPORATION",2194,82.999088 %

"USM BUSINESS SYSTEMS,1159,83.002588 %

"IBM INDIA PVT. LTD.",1284,83.099688 %

"APEX TECHNOLOGY SYSTEMS,1060,83.301887 %

"INTRAEDGE,1254,83.413078 %

"QUALCOMM TECHNOLOGIES,6113,83.674137 %

"CHARTER GLOBAL,1190,83.865546 %

"EXPEDIA,1315,83.954373 %

"RJT COMPUQUEST,1662,84.235860 %

"ORACLE AMERICA,7685,84.450228 %

"HCL GLOBAL SYSTEMS INC",3677,84.579821 %

"SIRI INFOSOLUTIONS INC.",1039,84.793070 %

"UNIVERSITY OF WASHINGTON",1187,84.919966 %

"ECLINICALWORKS,1547,85.455721 %

"VIRTUSA CORPORATION",2217,85.611186 %

"EMC CORPORATION",4467,85.627938 %

"CVS RX SERVICES,2735,85.630713 %

"SAPIENT CORPORATION",2237,85.650425 %

"WAL-MART ASSOCIATES,3678,85.807504 %

"COMCAST CABLE COMMUNICATIONS,1214,86.161450 %

"CIBER,2097,86.266094 %

"COGNIZANT TECHNOLOGY SOLUTIONS U.S. CORPORATION",17528,86.729804 %

"UNIVERSITY OF ILLINOIS AT CHICAGO",1131,87.091070 %

"MICROSOFT CORPORATION",25576,87.320144 %

"MORGAN STANLEY & CO. LLC",1669,87.657280 %

"FACEBOOK,4145,87.744270 %

"AT&T SERVICES,1227,87.938060 %

"APOGEE MEDICAL GROUP,1887,88.129306 %

"MARVELL SEMICONDUCTOR,1631,88.228081 %

"CAPITAL ONE SERVICES,2796,88.340486 %

"JPMORGAN CHASE & CO.",7035,88.343994 %

"MICROEXCEL,1227,88.508557 %

"VERIZON DATA SERVICES LLC",1635,88.623853 %

"HP ENTERPRISE SERVICES,1149,88.685814 %

"INTEL CORPORATION",11415,88.760403 %

"SOFTWARE PARADIGMS INTERNATIONAL GROUP,1034,88.781431 %

"BROOKHAVEN NATIONAL LABORATORY",1023,88.856305 %

"BALTIMORE CITY PUBLIC SCHOOLS",1014,88.954635 %

"UST GLOBAL INC",6355,89.000787 %

"L&T TECHNOLOGY SERVICES LIMITED",3722,89.065019 %

"CMC AMERICAS,1157,89.282627 %

"INTUIT INC.",1404,89.316239 %

"THE PENNSYLVANIA STATE UNIVERSITY",1042,89.635317 %

"PYRAMID TECHNOLOGY SOLUTIONS,1796,89.643653 %

"MOUNT SINAI MEDICAL CENTER",1114,89.856373 %

"MANAGEMENT HEALTH SYSTEMS,2000,89.900000 %

"ITECH US,2476,90.064620 %

"SYSTEM SOFT TECHNOLOGIES LLC",3102,90.070922 %

"AMAZON CORPORATE LLC",9026,90.206071 %

"HSBC BANK USA,1111,90.279028 %

"BANK OF AMERICA N.A.",4282,90.331621 %

"ORION SYSTEMS INTEGRATORS,1253,90.343176 %

"WIPRO LIMITED",48117,90.354760 %

"ASTIR IT SOLUTIONS INC.",1955,90.690537 %

"RANDSTAD TECHNOLOGIES,3427,90.779107 %

"CYBERTHINK INC",1618,90.914710 %

"TESLA MOTORS,1441,90.978487 %

"UBER TECHNOLOGIES,1006,91.053678 %

"SATYAM COMPUTER SERVICES LTD.",1694,91.263282 %

"GOLDMAN,3877,91.307712 %

"ITC INFOTECH (USA),1860,91.344086 %

"HEWLETT-PACKARD COMPANY",1639,91.580232 %

"CISCO SYSTEMS,3140,91.687898 %

"CHILDREN'S HOSPITAL CORPORATION",1017,91.740413 %

"TECHNOSOFT CORPORATION",1625,91.815385 %

"INOVANT,1091,91.934005 %

"KPIT INFOSYSTEMS,3115,91.942215 %

"LARSEN & TOUBRO LIMITED",3066,92.009132 %

"MERRILL LYNCH",1873,92.151628 %

"PEOPLE TECH GROUP INC.",1124,92.259786 %

"APPLE INC.",7317,92.319257 %

"A2Z DEVELOPMENT CENTER,1098,92.440801 %

"SYNECHRON,3802,92.766965 %

"FUJITSU AMERICA,5309,92.785835 %

"ADOBE SYSTEMS INCORPORATED",1167,92.802057 %

"CAPGEMINI U.S. LLC",3712,92.860991 %

"TWITTER,1333,92.873218 %

"NIIT TECHNOLOGIES LIMITED",1339,92.905153 %

"PAYPAL,2843,92.965178 %

"EBAY INC.",3464,93.013857 %

"THE BOARD OF TRUSTEES OF THE LELAND STANFORD,1738,93.440736 %

"HEXAWARE TECHNOLOGIES,5479,93.611973 %

"AMERICAN EXPRESS TRAVEL RELATED SERVICES COMPANY,1351,93.634345 %

"SALESFORCE.COM,2250,94.266667 %

"MAYO CLINIC",1772,94.300226 %

"UST GLOBAL INC.",6363,94.389439 %

"POLARIS SOFTWARE LAB (INDIA) LTD.",1326,94.419306 %

"EVEREST CONSULTING GROUP,1651,94.427620 %

"CYMA SYSTEMS INC",1269,94.483846 %

"MEMORIAL SLOAN-KETTERING CANCER CENTER",1080,94.537037 %

"VEDICSOFT SOLUTIONS LLC",1274,94.583987 %

"DALLAS INDEPENDENT SCHOOL DISTRICT",1729,94.621168 %

"SYNTEL CONSULTING INC.",3167,94.695295 %

"LARSEN & TOUBRO INFOTECH LIMITED",17457,95.388669 %

"CAPGEMINI AMERICA INC",16725,95.408072 %

"HEADSTRONG SERVICES LLC",2587,95.477387 %

"EXPERIS US,1642,95.493301 %

"TECH MAHINDRA (AMERICAS),17753,95.617642 %

"DOTCOM TEAM,1125,95.733333 %

"UNIVERSITY OF MINNESOTA",1353,95.787140 %

"THE MATHWORKS,2020,95.792079 %

"TECHDEMOCRACY LLC",1027,95.813048 %

"SATYAM COMPUTER SERVICES LIMITED",2403,95.838535 %

"IGATE TECHNOLOGIES INC.",12564,95.908946 %

"PATNI AMERICAS INC.",3149,95.935218 %

"KPMG LLP",4629,96.025059 %

"ERP ANALYSTS,1788,96.140940 %

"COLLABORATE SOLUTIONS INC",1209,96.195203 %

"CAPGEMINI FINANCIAL SERVICES USA INC",4426,96.475373 %

"BLOOMBERG,2352,96.641156 %

"BIRLASOFT INC",2370,96.666667 %

"PRICEWATERHOUSECOOPERS LLP",2719,96.689960 %

"SMARTSOFT INTERNATIONAL,1212,96.699670 %

"VMWARE,2617,96.713794 %

"LARSEN & TOUBRO TECHNOLOGY SERVICES LTD",1385,96.750903 %

"VEDICSOFT",1169,96.834902 %

"TECH MAHINDRA (AMERICAS) INC.",2102,96.860133 %

"SYNTEL INC",1946,97.019527 %

"CYIENT,1281,97.033568 %

"CSC COVANSYS CORPORATION",2251,97.112394 %

"NATSOFT CORPORATION",1137,97.185576 %

"RITE AID CORP.",1577,97.209892 %

"PRICEWATERHOUSECOOPERS ADVISORY SERVICES LLC",1724,97.215777 %

"NVIDIA CORPORATION",1182,97.292724 %

"PHOTON INFOTECH,1334,97.301349 %

"COMPUTER SCIENCES CORPORATION",1089,97.428834 %

"THE BOSTON CONSULTING GROUP,1352,97.559172 %

"3I INFOTECH,2430,97.572016 %

"INFOSYS TECHNOLOGIES LIMITED",1336,97.604790 %

"SATYAM COMPUTER SERVICES LTD",1622,97.657213 %

"PERFICIENT,1366,97.730600 %

"GENPACT LLC",1046,97.801147 %

"AKVARR INC",1372,97.886297 %

"DELOITTE & TOUCHE LLP",9642,97.925742 %

"PRICEWATERHOUSECOOPERS,2529,97.943851 %

"ERNST & YOUNG U.S. LLP",18232,98.036419 %

"HCL AMERICA,22679,98.042242 %

"MCKINSEY & COMPANY,1097,98.085688 %

"NTT DATA,5548,98.143475 %

"COMPUNNEL SOFTWARE GROUP,3379,98.194732 %

"AVCO CONSULTING INC",1424,98.244382 %

"CGI TECHNOLOGIES AND SOLUTIONS INC.",1995,98.245614 %

"DELOITTE CONSULTING LLP",36742,98.307114 %

"ACCENTURE LLP",33447,98.397465 %

"SAP AMERICA,1456,98.557692 %

"KFORCE INC.",1596,98.558897 %

"DELOITTE TAX LLP",2501,98.560576 %

"MPHASIS CORPORATION",5199,98.576649 %

"DELASOFT,1165,98.626609 %

"AMDOCS INC.",1023,98.631476 %

"RELIABLE SOFTWARE RESOURCES,1992,98.795181 %

"DIASPARK,1673,98.983861 %

"GRANDISON MANAGEMENT,1387,98.990627 %

"TECH MAHINDRA ( AMERICAS),1170,99.059829 %

"MINDTREE LIMITED",4067,99.065650 %

"TATA CONSULTANCY SERVICES LIMITED",64726,99.244508 %

"YASH & LUJAN CONSULTING,1373,99.271668 %

"YASH TECHNOLOGIES,2214,99.277326 %

"INFOSYS LIMITED",130592,99.482357 %

"HTC GLOBAL SERVICES INC.",2632,100.000000 %

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)? (using mapreduces)

Program:

package Query10;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import java.util.TreeMap;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Q10mapper extends Mapper<LongWritable, Text, Text, Text>{

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException

{

String parts[] =value.toString().split(",");

String status = parts[1];

String jobposition =parts[4];

context.write(new Text(jobposition), new Text(status));

}

}

public class Q10reducer extends Reducer<Text,Text,NullWritable,Text>{

private TreeMap<Double, String> topten = new TreeMap<>();

public void reduce(Text key, Iterable<Text> value, Context context) throws IOException, InterruptedException{

double total =0;

double successrate=0;

for (Text val:value)

{

String status = val.toString();

if(status.equals("\"CERTIFIED\"") || status.equals("\"CERTIFIED WITHDRAWN\""))

{

total++ ;

successrate++;

}

else

total++;

}

double rate = (successrate/total)\*100;

if(rate >=70 && total >=1000){

String op = key.toString()+ ","+ String.format("%.0f",total)+ "," +String.format("%f %%",rate);

topten.put(rate, op);

}

}

protected void cleanup(Context context) throws IOException, InterruptedException{

for(String val : topten.descendingMap().values()){

context.write(NullWritable.get(),new Text(val));

}

}

}

public class Q10driver {

public static void main(String[] args) throws Exception

{

Configuration conf =new Configuration();

conf.set("mapreduce.output.textoutputformat.separator", ",");

Job job=Job.getInstance(conf);

job.setJarByClass(Q10driver.class);

//job.setJobName("POSJoin");

job.setMapperClass(Q10mapper.class);

//job.addCacheFile(new Path("AFINN.txt").toUri());

//job.setNumReduceTasks(0);

job.setReducerClass(Q10reducer.class);

job.setOutputKeyClass(NullWritable.class);

job.setOutputValueClass(Text.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job,new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

Output:

"TEST ANALYST - US",1023,100.000000 %

"ASSOCIATE CONSULTANT - US",4392,99.908925 %

"PRODUCTION SUPPORT ANALYST - US",1451,99.862164 %

"SYSTEMS ENGINEER - US",10026,99.840415 %

"CONSULTANT - US",7413,99.797653 %

"TECHNOLOGY LEAD - US",28234,99.762697 %

"TECHNOLOGY ANALYST - US",26028,99.731059 %

"PROJECT MANAGER - US",7002,99.685804 %

"LEAD CONSULTANT - US",3394,99.646435 %

"DEVELOPER USER INTERFACE",5247,99.599771 %

"PRINCIPAL CONSULTANT - US",1349,99.481097 %

"ASSURANCE STAFF",2317,99.050496 %

"COMPUTER PROGRAMMER/CONFIGURER 2",6727,98.914821 %

"COMPUTER SYSTEMS ANALYST 2",4027,98.808046 %

"COMPUTER SYSTEMS ANALYST 3",2168,98.708487 %

"ADVISORY SENIOR",5343,98.633726 %

"COMPUTER PROGRAMMER / CONFIGURER 2",1276,98.510972 %

"COMPUTER PROGRAMMER/CONFIGURER 3",1145,98.427948 %

"BUSINESS TECHNOLOGY ANALYST",1817,98.293891 %

"ADVISORY MANAGER",3204,98.252185 %

"AUDIT SENIOR",1021,98.237023 %

"MODULE LEAD",2128,98.120301 %

"AUDIT ASSISTANT",1203,98.088113 %

"ADVISORY STAFF",2363,97.968684 %

"ERS SENIOR CONSULTANT",2158,97.914736 %

"TAX SENIOR",1795,97.883008 %

"ADVISORY SENIOR ASSOCIATE",1306,97.473201 %

"DEVELOPER",10041,97.340902 %

"ERS CONSULTANT",2068,96.953578 %

"COMPUTER SYSTEMS ENGINEER",3049,96.851427 %

"ASSURANCE SENIOR",1577,96.131896 %

"LEAD CONSULTANT",1759,96.020466 %

"COMPUTER USER SUPPORT SPECIALISTS",1246,95.345104 %

"WEB DEVELOPER",5543,95.273318 %

"PROJECT LEAD",1407,94.669510 %

"SYSTEMS ANALYSTS",1026,94.639376 %

"CONSULTANT",9786,94.277539 %

"MANAGER",4366,94.159414 %

"TECHNICAL ANALYST",2034,93.805310 %

"SENIOR ASSOCIATE",2396,93.405676 %

"SENIOR SYSTEMS ANALYST",4031,93.301910 %

All Other\*",21082,93.259653 %

"SENIOR CONSULTANT",12096,92.402447 %

"SENIOR PROGRAMMER ANALYST",2535,92.071006 %

"BUSINESS SYSTEM ANALYST",3806,91.854966 %

"TECHNICAL CONSULTANT",1541,91.823491 %

"COMPUTER PROGRAMMER",48071,91.516715 %

"INDUSTRIAL DESIGNER",3034,91.397495 %

"NETWORK AND COMPUTER SYSTEMS ADMINISTRATOR",1296,91.126543 %

"PHARMACIST",2827,90.979837 %

"SENIOR BUSINESS ANALYST",1431,90.705800 %

"SYSTEM ADMINISTRATOR",4004,90.509491 %

"SYSTEMS ADMINISTRATOR",2634,90.356872 %

"COMPUTER SYSTEMS ANALYSTS",73669,90.131534 %

"PROJECT MANAGER",8476,90.030675 %

"MANAGEMENT ANALYSTS",9501,90.001053 %

"SENIOR SYSTEMS ANALYST JC60",3068,89.928292 %

"SR. PROGRAMMER ANALYST",1169,89.905902 %

"Computer Systems Analysts",106356,89.675242 %

"COMPUTER OCCUPATIONS,32397,89.532981 %

"SALES ENGINEERS",2168,89.529520 %

"DATABASE ADMINISTRATORS",8118,89.443213 %

"ENGINEER",2113,89.398959 %

"BUSINESS ANALYST",19842,89.396230 %

"Computer Occupations,26326,89.386918 %

"COMPUTER SYSTEMS ANALYST",20191,89.376455 %

"DATABASE ADMINISTRATOR",9780,89.325153 %

"PROGRAMMER ANALYST",128216,89.217414 %

ALL OTHER",99213,89.174806 %

SYSTEMS SOFTWARE",22736,88.929451 %

"NETWORK AND COMPUTER SYSTEMS ADMINISTRATORS",9468,88.857203 %

"SYSTEMS ANALYST",30870,88.697765 %

"COMPUTER PROGRAMMERS",60674,88.678841 %

"NETWORK ADMINISTRATOR",1579,88.663711 %

"SOFTWARE DEVELOPERS,118923,88.600187 %

"DESIGN ENGINEER",2488,88.585209 %

"PHYSICIANS AND SURGEONS,3390,88.554572 %

"SENIOR SOFTWARE DEVELOPER",1642,88.550548 %

"INTERNISTS,1602,88.514357 %

"JAVA DEVELOPER",1169,88.451668 %

Web Developers,2068,88.442940 %

"PHYSICAL THERAPISTS",5471,88.393347 %

"MANAGER JC50",1868,88.383298 %

"Software Developers,118922,88.334370 %

"BUSINESS SYSTEMS ANALYST",4571,88.273901 %

"COMPUTER AND INFORMATION SYSTEMS MANAGERS",4807,88.163095 %

Systems Software",32288,88.162785 %

"SYSTEM ANALYST",2573,88.146133 %

"DATA ANALYST",1440,88.125000 %

"COMPUTER PROGRAMMER ANALYST",6179,88.104871 %

Except Special Educati",6102,88.069485 %

"QA ANALYST",1631,88.044145 %

"PROGRAMMER ANALYST LEVEL 1",2386,88.013412 %

"WEB DEVELOPERS",1431,87.980433 %

All Other",94241,87.944737 %

"Network and Computer Systems Administrators\*",3509,87.888287 %

"Speech-Language Pathologists",1254,87.878788 %

"ELECTRICAL ENGINEERS",5713,87.817259 %

"MECHANICAL ENGINEERS",9943,87.810520 %

"PROGRAM MANAGER",1027,87.731256 %

PC",1075,87.627907 %

"TECHNICAL LEAD",2484,87.560386 %

"PROGRAMMER",2660,87.556391 %

"SENIOR SOFTWARE ENGINEER",1625,87.507692 %

"CHEMISTS",1857,87.506731 %

"ANALYST",7559,87.471888 %

APPLICATIONS",117676,87.454536 %

"PROGRAMMER/ANALYST",3108,87.419562 %

"SOFTWARE ENGINEER",4327,87.404668 %

"Management Analysts",14400,87.388889 %

"SYSTEMS ANALYST JC65",3321,87.383318 %

"Computer Programmers",86556,87.331901 %

EXCEPT SPECIAL",2357,87.314383 %

ALL",1168,87.243151 %

"SENIOR ENGINEER",1037,87.174542 %

",5868,87.116564 %

"Network and Computer Systems Administrators",8268,87.034349 %

"LOGISTICIANS",1511,87.028458 %

"Database Administrators",10159,86.986908 %

GENERAL",3557,86.955299 %

"ACCOUNTANTS AND AUDITORS",6259,86.930820 %

"SYSTEMS ENGINEER",1457,86.890872 %

"SOFTWARE PROGRAMMER",1067,86.691659 %

Applications",107447,86.585945 %

"Credit Analysts",1057,86.565752 %

"INFORMATION SECURITY ANALYSTS",1302,86.482335 %

"STATISTICIANS",3583,86.463857 %

"Mechanical Engineers",14029,86.449497 %

"ENGINEERS,2786,86.396267 %

"PROJECT ENGINEER",1739,86.371478 %

"ARCHITECTURAL AND ENGINEERING MANAGERS",1467,86.366735 %

"MATERIALS ENGINEERS",1127,86.335404 %

"MARKET RESEARCH ANALYSTS AND MARKETING SPECIALISTS",7079,86.297500 %

"MEDICAL TECHNOLOGIST",2569,86.259245 %

Except Special and Career/",1398,86.194564 %

EXCEPT COMPUTER",5706,86.172450 %

"SALES MANAGERS",1092,86.172161 %

"OCCUPATIONAL THERAPIST",1552,86.146907 %

"ASSOCIATE",7081,86.131902 %

"Statisticians",4009,86.106261 %

"Computer Hardware Engineers",1755,86.096866 %

"ELECTRONICS ENGINEERS,7549,86.024639 %

"Family and General Practitioners",1706,85.990621 %

"Physical Therapists",9962,85.976712 %

"OPERATIONS RESEARCH ANALYSTS",6623,85.942926 %

"Sales Engineers",3205,85.897036 %

"PHYSICAL THERAPIST",5968,85.891421 %

"MANAGER,3677,85.749252 %

"Chemists",3459,85.747326 %

"INDUSTRIAL ENGINEERS",3153,85.696162 %

"Architectural and Engineering Managers",2284,85.683012 %

"Computer and Information Systems Managers",7025,85.679715 %

"Pharmacists",3961,85.584448 %

"GRAPHIC DESIGNERS",2157,85.442745 %

"CIVIL ENGINEERS",3525,85.390071 %

"PROCESS ENGINEER",2875,85.356522 %

"Logisticians",2232,85.349462 %

"Operations Research Analysts",8197,85.275101 %

"Materials Engineers",1708,85.245902 %

"SALES ENGINEER",1008,85.218254 %

"FINANCIAL MANAGERS",1535,85.211726 %

INC.",2764,85.202605 %

"Physicians and Surgeons,5156,85.182312 %

"Engineers,3556,85.179978 %

"NETWORK ENGINEER",2300,85.173913 %

"Economists",1225,85.142857 %

"Internists,3623,85.067624 %

Al",1968,85.060976 %

"Computer and Information Research Scientists",1321,85.011355 %

"GENERAL AND OPERATIONS MANAGERS",1671,84.979054 %

"HUMAN RESOURCES SPECIALISTS",1135,84.845815 %

"MARKETING MANAGERS",2744,84.839650 %

"PHARMACISTS",1061,84.731385 %

"Market Research Analysts and Marketing Specialists",11157,84.583669 %

"COMMERCIAL AND INDUSTRIAL DESIGNERS",1252,84.504792 %

"Commercial and Industrial Designers",1845,84.444444 %

Except Special and Care",4588,84.415867 %

"DENTISTS,1166,84.391081 %

"MECHANICAL ENGINEER",3329,84.379694 %

"COMPUTER AND INFORMATION RESEARCH SCIENTISTS",1222,84.369885 %

"Electronics Engineers,12495,84.321729 %

"Fashion Designers",1033,84.123911 %

Preschool,1038,84.104046 %

"SOFTWARE DEVELOPER",2453,84.101101 %

"PRODUCT MANAGER",1239,84.100081 %

"STAFF ACCOUNTANT",1855,84.043127 %

"Dentists,1742,84.041332 %

"ASSOCIATE,1550,83.870968 %

"COMPUTER SYSTEM ANALYST",2239,83.787405 %

"Occupational Therapists",2404,83.735441 %

"Industrial Engineers",5404,83.641747 %

"Food Scientists and Technologists",1033,83.639884 %

General",7168,83.510045 %

"BUSINESS DEVELOPMENT MANAGER",1146,83.507853 %

"Electrical Engineers",11022,83.487570 %

Except Computer",11231,83.474312 %

"MANAGEMENT ANALYST",2829,83.457052 %

NA,4590,83.442266 %

"DIRECTOR,2764,83.393632 %

"Accountants and Auditors",10682,83.383262 %

"Cost Estimators",1233,83.373885 %

Drama,1067,83.317713 %

"PUBLIC RELATIONS SPECIALISTS",1065,83.286385 %

"APPLICATION PROGRAMMER",1582,83.249052 %

EXCEPT SPECIAL AND",3201,83.130272 %

"FINANCIAL ANALYSTS",6859,83.102493 %

"Chemical Engineers",2211,83.084577 %

"Financial Specialists,1126,83.037300 %

"FINANCIAL ANALYST",4542,82.959049 %

"COMPUTER HARDWARE ENGINEERS",1232,82.954545 %

"Civil Engineers",5764,82.876475 %

"MARKETING SPECIALIST",1161,82.859604 %

A MASTECH HOLDINGS,5228,82.746748 %

"SYSTEM ENGINEER",1082,82.624769 %

"APPLICATION DEVELOPER",3841,82.582661 %

"Business Operations Specialists,3030,82.574257 %

"Biomedical Engineers",1408,82.528409 %

All",1706,82.473623 %

"RESEARCH FELLOW",1183,82.417582 %

"Human Resources,1077,82.358403 %

"Industrial Production Managers",1101,82.198002 %

"SAP CONSULTANT",1179,82.103478 %

"MARKET RESEARCH ANALYST",4365,82.061856 %

"IT SPECIALIST",2314,82.022472 %

"BIOCHEMISTS AND BIOPHYSICISTS",1817,82.003302 %

"Sales Managers",2273,81.830180 %

"VICE PRESIDENT,2096,81.774809 %

"ELECTRICAL ENGINEER",1925,81.714286 %

Except Special and Voca",1365,81.684982 %

Postseco",3527,81.655798 %

"RESEARCH ASSISTANT",1210,81.652893 %

"Medical Scientists,2836,81.593794 %

"MEDICAL SCIENTISTS,1479,81.541582 %

"Biological Scientists,2177,81.488287 %

"Information Security Analysts,1869,81.487426 %

"Software Quality Assurance Engineers and Testers",2185,81.418764 %

"GRAPHIC DESIGNER",2567,81.417998 %

EXCEPT LANDSCAPE AND NAVAL",1345,81.412639 %

"Financial Analysts",11895,81.252627 %

"SYSTEM ANALYST JC65",1419,81.183932 %

"Financial Managers",2997,81.114448 %

Except Hydrologists and Geographers",1769,81.062747 %

Postsecondary",26860,81.046165 %

"MANAGING CONSULTANT",2261,81.026095 %

"Computer Support Specialists",1247,80.834002 %

"CHEMICAL ENGINEERS",1139,80.684811 %

"SENIOR SYSTEM ENGINEER",1258,80.524642 %

POSTSECONDARY",15762,80.503743 %

"BIOLOGICAL SCIENTISTS,1777,80.360158 %

"ACCOUNTANT",7148,80.316172 %

"Marketing Managers",5220,80.306513 %

Except Landscape and Naval",2080,80.240385 %

"RESEARCH SCIENTIST",2353,80.195495 %

Applications,1547,80.155139 %

"Physicists",1103,80.054397 %

APPLICATIONS,1012,79.940711 %

"Managers,1433,79.832519 %

"VICE PRESIDENT",1814,79.713341 %

Training,1248,79.647436 %

"Graphic Designers",3983,79.613357 %

"Medical and Health Services Managers",2009,79.542061 %

"Computer Software Engineers,21897,79.458373 %

"General and Operations Managers",3397,79.452458 %

"Architects,1541,79.169371 %

Except Epidemiologists",13927,78.976090 %

"Biochemists and Biophysicists",3641,78.824499 %

"ASSISTANT VICE PRESIDENT",1255,78.565737 %

"PUBLIC RELATIONS SPECIALIST",1082,78.558226 %

"Public Relations Specialists",2020,78.465347 %

"POSTDOCTORAL RESEARCH FELLOW",1060,78.396226 %

Except Special Education",1556,78.341902 %

"ENGINEER,1514,78.071334 %

"Advertising and Promotions Managers",1039,76.997113 %

"RESEARCH ASSOCIATE",6982,76.754512 %

"POSTDOCTORAL ASSOCIATE",2630,76.083650 %

"MARKETING MANAGER",1174,75.809199 %

"Chief Executives",1481,74.409183 %

"POSTDOCTORAL RESEARCH ASSOCIATE",3545,74.104372 %

EXCEPT EPIDEMIOLOGISTS",7869,74.100902 %

R&D",3368,73.871734 %

"POSTDOCTORAL FELLOW",3774,73.688394 %

"Lawyers",1554,73.552124 %

"Market Research Analysts",2971,73.174015 %

"ANALYST,1619,72.205065 %

"Accountants",1811,71.949199 %

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

(using sqoop)

1.mysql –u root –p

2. create database h1b\_data;

3. use h1b\_data;

4.CREATE TABLE q10\_rate(job\_title varchar(100)NOT NULL,total\_no\_of\_appl INT NOT NULL,certifiedANDcertified\_withdrwan\_count INT NOT NULL,

5.desc q10\_rate;

+---------------------------------------+--------------+------+-----+---------+-------+

| field | type | null | key | default | extra |

+---------------------------------------+--------------+------+-----+---------+-------+

| job\_title | varchar(100) | no | | null | |

| total\_no\_of\_appl | int(11) | no | | null | |

| certifiedandcertified\_withdrwan\_count | int(11) | no | | null | |

| success\_rate | float | no | | null | |

+---------------------------------------+--------------+------+-----+---------+-------+

6.Start sqoop…..connect to msql>h1b\_data database

sqoop list-tables --connect jdbc:mysql://localhost/h1b\_data --username root --password 'hduser'

7. export data from hdfs to msql >q10\_rate table

sqoop export --connect jdbc:mysql://localhost/h1b\_data --username 'root' --password 'hduser' --table q10\_rate --export-dir /problem10 --input-fields-terminated-by ',' --mysql-delimiters -m 1;

+------------------------------------------------------------+------------------+---------------------------------------+--------------+

| PRODUCTION SUPPORT LEAD - US | 1301 | 1301 | 100 |

| ASSOCIATE CONSULTANT - US | 4393 | 4390 | 99.9317 |

| SYSTEMS ENGINEER - US | 10036 | 10026 | 99.9004 |

| TEST ENGINEER - US | 2198 | 2195 | 99.8635 |

| PRODUCTION SUPPORT ANALYST - US | 1451 | 1449 | 99.8622 |

| TEST ANALYST - US | 4958 | 4949 | 99.8185 |

| CONSULTANT - US | 7426 | 7412 | 99.8115 |

| TECHNOLOGY LEAD - US | 28350 | 28294 | 99.8025 |

| TECHNICAL TEST LEAD - US | 5374 | 5363 | 99.7953 |

| SENIOR TECHNOLOGY ARCHITECT - US | 1417 | 1414 | 99.7883 |

| TECHNOLOGY ARCHITECT - US | 4707 | 4696 | 99.7663 |

| TECHNOLOGY ANALYST - US | 26055 | 25993 | 99.762 |

| SENIOR PROJECT MANAGER - US | 2774 | 2767 | 99.7477 |

| DEVELOPER USER INTERFACE | 5247 | 5232 | 99.7141 |

| COMPUTER SYSTEMS ANALYST 2 | 4031 | 4019 | 99.7023 |

| SYSTEMS ANALYST - II | 1339 | 1335 | 99.7013 |

| PROJECT MANAGER - III | 1651 | 1646 | 99.6972 |

| PROJECT MANAGER - US | 7046 | 7024 | 99.6878 |

| PROGRAMMER ANALYST - II | 3588 | 3576 | 99.6656 |

| LEAD CONSULTANT - US | 3402 | 3390 | 99.6473 |

| COMPUTER SYSTEMS ANALYST 3 | 2170 | 2161 | 99.5853 |

| COMPUTER PROGRAMMER/CONFIGURER 2 | 6729 | 6700 | 99.569 |

| PROGRAMMER ANALYST - I | 1432 | 1425 | 99.5112 |

| SYSTEMS ANALYST - III | 1006 | 1001 | 99.503 |

| PRINCIPAL CONSULTANT - US | 1352 | 1345 | 99.4822 |

| COMPUTER SPECIALIST/TESTING AND QUALITY ANALYST 2 | 3998 | 3975 | 99.4247 |

| COMPUTER PROGRAMMER/CONFIGURER 3 | 1145 | 1138 | 99.3886 |

| COMPUTER SPECIALIST/SYSTEM SUPPORT AND DEVELOPMENT | 1339 | 1330 | 99.3279 |

| COMPUTER SPECIALIST/SYSTEM SUPPORT AND DEVELOPMENT ADMIN 2 | 1085 | 1077 | 99.2627 |

| DATA WAREHOUSE SPECIALIST | 1631 | 1618 | 99.2029 |