**PROGRAM CODING**

**Solution and Explanation:**

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

**Code:**

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.\*;

**Mapper code**

public class proj1

{

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 pet = str[4];

//String year =(str[7]);

if(str[4].equals("DATA ENGINEER"))

{

context.write(new Text(str[7]),new Text(str[4]));

}

}

catch(Exception e)

{

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

}

}

}

**Reducer Code**

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

{

// private LongWritable result = new LongWritable();

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

{

long count=0;

//String job1 ="";

//String case\_status="";

for (Text val : values)

{

count++;

}

context.write(key, new LongWritable (count));

}

}

**//Driver Code**

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

Configuration conf = new Configuration();

//conf.set("name", "value")

//conf.set("mapreduce.input.fileinputformat.split.minsize", "134217728");

Job job = Job.getInstance(conf, "job Count");

job.setJarByClass(proj1.class);

job.setMapperClass(MapClass.class);

//job.setCombinerClass(ReduceClass.class);

job.setReducerClass(ReduceClass.class);

job.setNumReduceTasks(2);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(LongWritable.class);

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

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

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

}

}

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

Solution: This question has been solved using Pig Latin.

Code:

h1b\_final= load '/user/hive/warehouse/project.db/h1b\_final' using PigStorage('\t') AS (s\_no:int,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:chararray,worksite:chararray,longitute:double,latitute:double);filjob = filter h1b\_final by year=='2011';--dump filjob;grpjt = group filjob by $4;grpcs1 = foreach grpjt generate group, COUNT(filjob.$1);--dump grpcs;filjob = filter h1b\_final by year=='2012';--dump filjob;

grpjt = group filjob by $4;grpcs2 = foreach grpjt generate group, COUNT(filjob.$1);

filjob = filter h1b\_final by year=='2013';

--dump filjob;grpjt = group filjob by $4;grpcs3 = foreach grpjt generate group, COUNT(filjob.$1);filjob = filter h1b\_final by year=='2014';--dump filjob;grpjt = group filjob by $4;

grpcs4 = foreach grpjt generate group, COUNT(filjob.$1);filjob = filter h1b\_final by year=='2015';--dump filjob;grpjt = group filjob by $4;grpcs5 = foreach grpjt generate group, COUNT(filjob.$1);filjob = filter h1b\_final by year=='2016';--dump filjob;

grpjt = group filjob by $4;grpcs6 = foreach grpjt generate group, COUNT(filjob.$1);jngr = join grpcs1 by $0 ,grpcs2 by $0 ,grpcs3 by $0 ,grpcs4 by $0 ,grpcs5 by $0 ,grpcs6 by $0;--dump jngr;jngr = foreach jngr generate $0 , $1 , $3 ,$5 ,$7 ,$9 ,$11;growth = foreach jngr generate $0 , (float)(($2-$1)/$1)\*100, (float)(($3-$2)/$2)\*100,(float)(($4-$3)/$3)\*100, (float)(($5-$4)/$4)\*100, (float)(($6-$5)/$5)\*100;avggw = foreach growth generate $0 , ($1+$2+$3+$4+$5)/5;lm = limit (order avggw by $1 desc) 5;dump lm;

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

Solution: This problem has been solved using the Map Reduce method using Java programming

Code:

import java.io.\*;

import java.util.TreeMap;

import org.apache.hadoop.io.NullWritable;

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.Partitioner;

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 proj2a {

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");

if((str[4].equals("DATA ENGINEER"))&&(str[1].equals("CERTIFIED")))

{

String c = str[4]+"\t"+str[7];

context.write(new Text(str[8]),new Text(c));

}

}

catch(Exception e)

{

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

}

}

}

public static class yearPartitioner extends Partitioner<Text,Text>

{

public int getPartition(Text key, Text values, int numReduceTasks) {

//String b[]="";

String[] b=values.toString().split("\t");

if(b[1].equals("2011"))

{

return 0;

}

else if(b[1].equals("2012"))

{

return 1;

}

else if(b[1].equals("2013"))

{

return 2;

}

else if(b[1].equals("2014"))

{

return 3;

}

else if(b[1].equals("2015"))

{

return 4;

}

else

{

return 5;

}

}

}

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

{

public TreeMap<Long, Text> tm = new TreeMap<Long, Text>();

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

{

long count=0;

//String year="";

//String job="";

String myVal="";

for(Text val:values)

{

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

count++;

myVal = str[1]+"\t"+key+"\t"+str[0];

}

String myValue = myVal+"\t"+count;

tm.put(new Long(count), new Text(myValue));

if(tm.size()>1)

{

tm.remove(tm.firstKey());

}

}

public void cleanup(Context con) throws IOException, InterruptedException

{

for(Text t:tm.descendingMap().values())

{

con.write(NullWritable.get(), t);

}

}

}

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

Configuration conf = new Configuration();

//conf.set("name", "value")

//conf.set("mapreduce.input.fileinputformat.split.minsize", "134217728");

Job job = Job.getInstance(conf, "job Count");

job.setJarByClass(proj2a.class);

job.setMapperClass(MapClass.class);

job.setPartitionerClass(yearPartitioner.class);

//job.setCombinerClass(ReduceClass.class);

job.setReducerClass(ReduceClass.class);

job.setNumReduceTasks(6);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(NullWritable.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);

}

}

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

Solution:

Code:

import java.io.\*;

import java.util.TreeMap;

import org.apache.hadoop.io.NullWritable;

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.Partitioner;

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 proj2b {

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");

if(str[1].equals("CERTIFIED"))

{

String a = str[1]+"\t"+str[7];

context.write(new Text(str[8]),new Text(a));

}

}

catch(Exception e)

{

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

}

}

}

public static class yearPartitioner extends Partitioner<Text,Text>

{

public int getPartition(Text key, Text values, int numReduceTasks) {

//String b[]="";

long c=0;

String[] b=values.toString().split("\t");

c = Long.parseLong(b[1]);

if(c==2011)

{

return 0;

}

else if(c==2012)

{

return 1;

}

else if(c==2013)

{

return 2;

}

else if(c==2014)

{

return 3;

}

else if(c==2015)

{

return 4;

}

else

{

return 5;

}

}

}

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

{

// private LongWritable result = new LongWritable();

public TreeMap<Long, Text> tm = new TreeMap<Long, Text>();

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

{

long count=0;

//String year="";

//String job="";

String myVal="";

for(Text val:values)

{

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

count++;

myVal = str[1]+"\t"+key;

}

String myValue1 = myVal+"\t"+count;

tm.put(new Long(count), new Text(myValue1));

if(tm.size()>5)

{

tm.remove(tm.firstKey());

}

}

public void cleanup(Context con) throws IOException, InterruptedException

{

for(Text t:tm.descendingMap().values())

{

con.write(NullWritable.get(), t);

}

}

}

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

Configuration conf = new Configuration();

//conf.set("name", "value");

//conf.set("mapreduce.input.fileinputformat.split.minsize", "134217728");

Job job = Job.getInstance(conf, "job Count");

job.setJarByClass(proj2b.class);

job.setMapperClass(MapClass.class);

job.setPartitionerClass(yearPartitioner.class);

//job.setCombinerClass(ReduceClass.class);

job.setReducerClass(ReduceClass.class);

job.setNumReduceTasks(6);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(NullWritable.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);

}

}

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

Solution: This query is solved using hive with SQL programming

Code:

select SOC\_NAME,count(case\_status) as ind from h1b\_final where case\_status = 'certified' and job\_title ='Data scientist' group by SOC\_NAME order by ind desc limit 1 ;

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

solution: This query is solved using hive with SQL programming.

Code:

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2011'

group by employer\_name,year order by top desc limit 5;

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2012' group by employer\_name,year order by top desc limit 5;

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2013' group by employer\_name,year order by top desc limit 5;

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2014' group by employer\_name,year order by top desc limit 5;

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2015' group by employer\_name,year order by top desc limit 5;

select employer\_name,year,count(case\_status) as top from h1b\_final where year = '2016' group by employer\_name,year order by top desc limit 5;

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

Solution: This query is solved using hive with SQL programming.

Code:

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

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

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

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

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

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

5) b) Find the most popular top 10 job positions for H1B visa applications for each year? For only certified applications.

Solution: This query is solved using hive with SQL programming.

Code:

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

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

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

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

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

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

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.

Solution: This question has been solved using Pig Latin.

Code:

h1b\_final = load '/user/hive/warehouse/project.db/h1b\_final' using PigStorage('\t') as (sno:int, case\_status:chararray, emp\_name:chararray, soc\_name:chararray, job\_title:chararray, full\_time\_pos:chararray, wage:long, year:chararray, worksite:chararray, longitude:double, lattitude:double);

year\_grp = group h1b\_final by $7;

total = foreach year\_grp generate group, (float)COUNT(h1b\_final.$1) as tot;

--dump total;

grp = group h1b\_final by ($7,$1);

case\_total = foreach grp generate flatten(group), (float)COUNT(h1b\_final.$1) as per;

--dump case\_total;

join\_grp = join total by $0,case\_total by $0;

--dump join\_grp;

final = foreach join\_grp generate $0,$3,$4,ROUND\_TO(($4/$1)\*100,2);

--dump final;

store final into '/home/hduser/Documents/project pig querries/project6';

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

Solution: This problem has been solved using the Map Reduce method using Java programming.

Code:

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 class proj7{

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 app = str[1];

String year =(str[7]);

context.write(new Text(year),new Text(app));

}

catch(Exception e)

{

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

}

}

}

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

{

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

{

long count=0;

//String job1 ="";

//String case\_status="";

for (Text val : values)

{

count++;

}

context.write(key, new LongWritable (count));

}

}

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

Configuration conf = new Configuration();

//conf.set("name", "value")

//conf.set("mapreduce.input.fileinputformat.split.minsize", "134217728");

Job job = Job.getInstance(conf, "app Count");

job.setJarByClass(proj7.class);

job.setMapperClass(MapClass.class);

//job.setCombinerClass(ReduceClass.class);

job.setReducerClass(ReduceClass.class);

job.setNumReduceTasks(2);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(LongWritable.class);

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

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

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

}}

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.]

Code:

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2011' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2011' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2012' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2012' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2013' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2013' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2014' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2014' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2015' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2015' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='Y' and year='2016' group by job\_title,full\_time\_position,year order by average desc;

select job\_title, full\_time\_position, year, avg(prevailing\_wage) as average from h1b\_final where full\_time\_position ='N' and year='2016' group by job\_title,full\_time\_position,year order by average desc;

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)

Solution:

h1b\_final= load '/user/hive/warehouse/project.db/h1b\_final' using PigStorage('\t') AS (s\_no:int,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:double,worksite:chararray,longitute:double,latitute:double);empgrp = GROUP h1b\_final BY $2;--dump empgrp;wordcount = FOREACH empgrp GENERATE group, COUNT(h1b\_final.$1);--dump wordcount;

casefil = filter h1b\_final by case\_status=='CERTIFIED' or case\_status=='CERTIFIED-WITHDRAWN';

case\_status1 = GROUP casefil by $2; wordcount1 = FOREACH case\_status1 GENERATE group, COUNT(casefil.$1);--dump wordcount1;joinb = join wordcount by $0,wordcount1 by $0;--dump joinbag;

out = FOREACH joinb GENERATE $0,(float)$3/$1\*100, $1;--dump out;out1 = filter out by $1>70 and $2>=1000;

dump out1;

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)?

Solution: This question has been solved using Pig Latin.

Code:

h1b\_final= load '/user/hive/warehouse/project.db/h1b\_final' using PigStorage('\t') AS (s\_no:int,case\_status:chararray,employer\_name:chararray,soc\_name:chararray,job\_title:chararray,full\_time\_position:chararray,prevailing\_wage:double,year:double,worksite:chararray,longitute:double,latitute:double);empgrp = GROUP h1b\_final BY $4;--dump empgrp;wordcount = FOREACH empgrp GENERATE group, COUNT(h1b\_final.$1);--dump wordcount;

casefil = filter h1b\_final by case\_status=='CERTIFIED' or case\_status=='CERTIFIED-WITHDRAWN';case\_status1 = GROUP casefil by $4; wordcount1 = FOREACH case\_status1 GENERATE group, COUNT(casefil.$1);--dump wordcount1;

joinb = join wordcount by $0,wordcount1 by $0;--dump joinbag;

out = FOREACH joinb GENERATE $0,(float)$3/$1\*100, $1;--dump out;out1 = filter out by $1>70 and $2>=1000;

--dump out1; store out1 into '/niit/project10';

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

Solution: This query is solved using Sqoop Export command

Code:

create table sq (job VARCHAR(100) NOT NULL, percentage FLOAT NOT NULL,total INT NOT NULL, PRIMARY KEY (job));

sqoop export --connect jdbc:mysql://localhost/project --username root --password '123' --table sq --update-mode allowinsert --update-key job --export-dir /project%2010 --input-fields-terminated-by '\t' ;