PROJECT2

GLOBAL EMPLOYABILITY

&

EDUCATIONAL ANALYSIS

Project Author: - Sandeep Kumar Bhandoria

OBJECTIVE

To come up with relevant data for new companies from IT Sector, new matrimonial site, a global Educational status around the world for government and Immigrants working in our Country to know an overall status of country.

TECHNOLOGY USED

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data. It enables parallel processing on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

Apache HIVE is a data warehouse built on top of Hadoop for reading, writing, and managing large datasets residing in HDFS using SQL (HIVEQL).

Apache SQOOP is a tool designed for importing and exporting data from and to a RDBMS databases.

Pig is a scripting language for creating program that run on top of Apache Hadoop.

Apache HDFS (Hadoop Distributed File System) is a distributed file system designed to run on commodity hardware.

SOFTWARE USED

1) VIRTUAL BOX
2) ECLIPSE
3) UBUNTU
4) CLOUDERA

DATA WE USE

1. **Census_Records.json** -
   This is the primary source of our data.

2. **Pension_amt** -

A table in MySQL database containing the amount as pension given to the Senior Citizens.

3. **CensusData -**

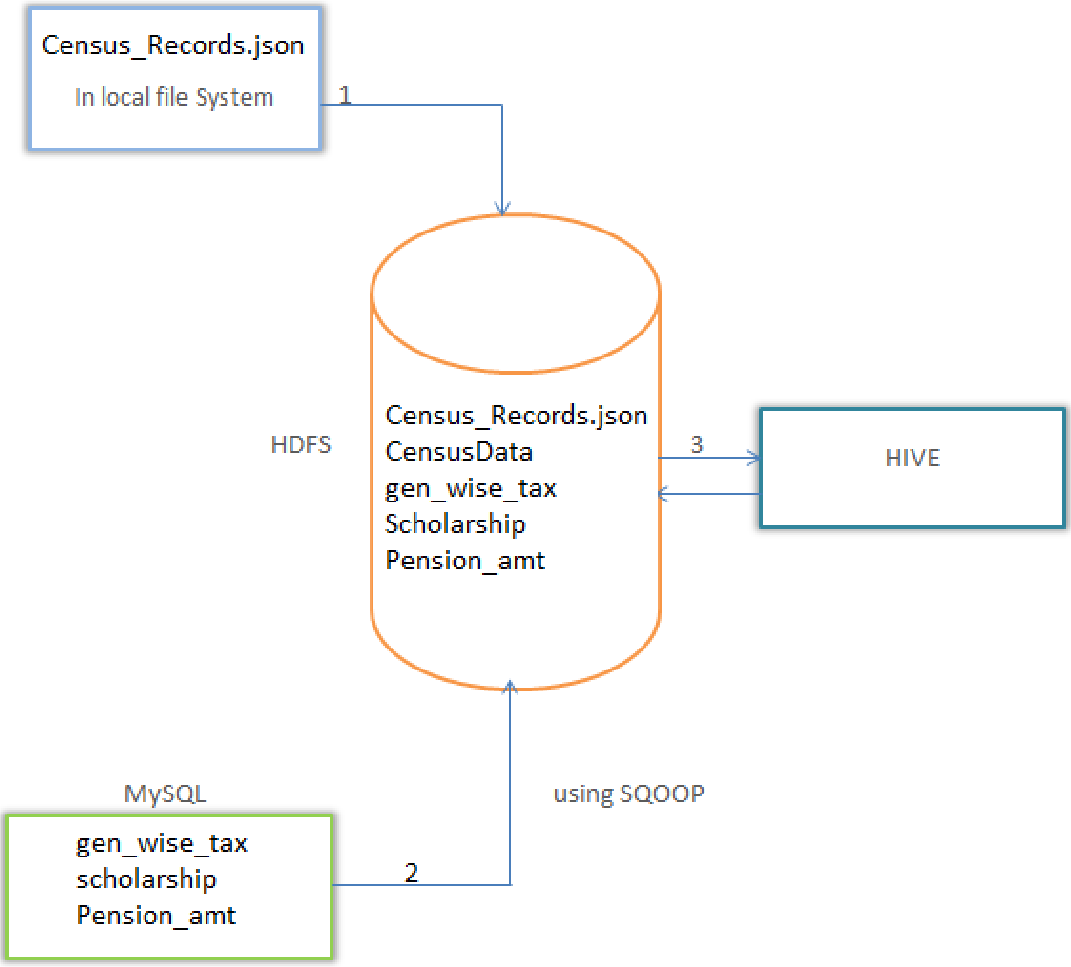
   It's a copy of primary Data Census_Records.json to work with Map Reduce application.

4. **Scholarship -**

   A table in MySQL database containing the amount to scholarship give to the peoples.

5. **Gen_wise_tax -**

   A table in MySQL that contains the percentage of tax is applicable to male and female depending on other income.



PROJECT DESCRIPTION

Project comes up with 4 category of task Education, Finance, Social, Planning and some miscellaneous. Each category will be targeting different fields such as Government, Customer, and Employability etc.

We have used all the possible technology such as Map Reduce, SQOOP, HIVE, MySQL and PIG for different tasks.

EDUCATION

TASK 1: Education Ratio between Male and Female

USE CASE:

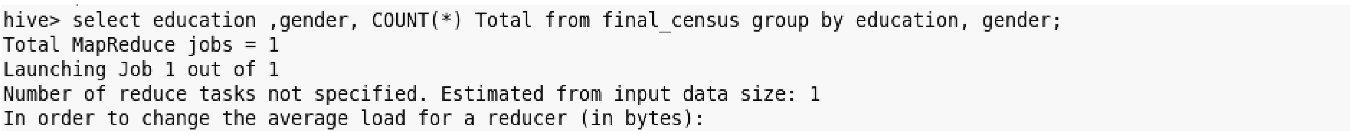
Comparative Literacy statistics on country

Literacy rate for India in 2011 is 74% as compared to other neighbouring country Myanmar, Sri Lanka and China. The main factors for low literacy rate are usefulness of education and availability of school in vicinity in rural areas.

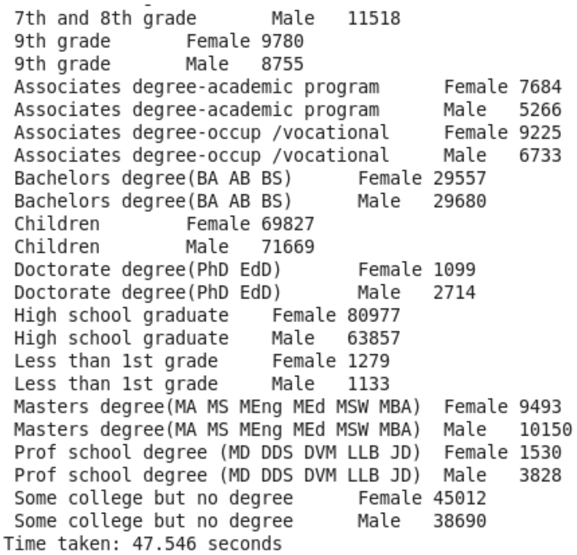
But within last 4 year India has indicated rising literacy in most of their states which make the literacy rate 90% in 2015.

A recent change in the government has done so much help in rural sectors. And a recent analysis is required to calculate the male female educated count for 2016.

Using Hive

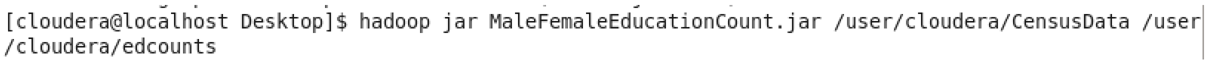


```
hive> select education ,gender, COUNT(*) Total from final_census group by education, gender;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
```

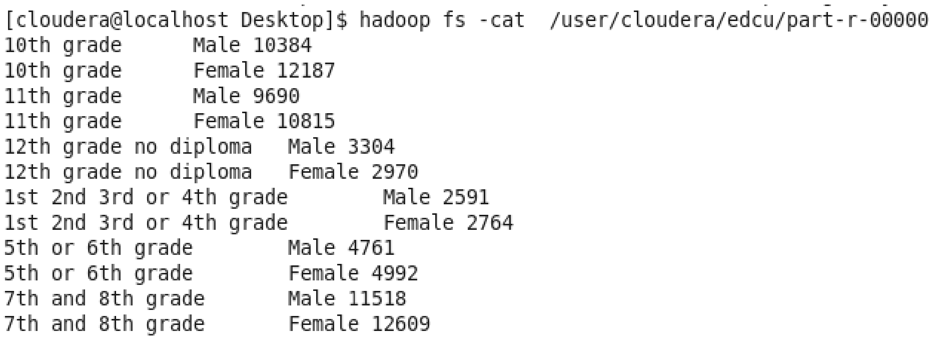


```
7th and 8th grade       Male    11518
9th grade       Female 9780
9th grade       Male    8755
Associates degree-academic program      Female 7684
Associates degree-academic program      Male    5266
Associates degree-occup /vocational     Female 9225
Associates degree-occup /vocational     Male    6733
Bachelors degree(BA AB BS)      Female 29557
Bachelors degree(BA AB BS)      Male    29680
Children        Female 69827
Children        Male    71669
Doctorate degree(PhD EdD)       Female 1099
Doctorate degree(PhD EdD)       Male    2714
High school graduate    Female 80977
High school graduate    Male    63857
Less than 1st grade     Female 1279
Less than 1st grade     Male    1133
Masters degree(MA MS MEng MEd MSW MBA)  Female 9493
Masters degree(MA MS MEng MEd MSW MBA)  Male    10150
Prof school degree (MD DDS DVM LLB JD)  Female 1530
Prof school degree (MD DDS DVM LLB JD)  Male    3828
Some college but no degree      Female 45012
Some college but no degree      Male    38690
Time taken: 47.546 seconds
```

Using Basic Map Reduce in Java



```
[cloudera@localhost Desktop]$ hadoop jar MaleFemaleEducationCount.jar /user/cloudera/CensusData /user
/cloudera/edcounts
```



```
[cloudera@localhost Desktop]$ hadoop fs -cat  /user/cloudera/edcu/part-r-00000
10th grade      Male 10384
10th grade      Female 12187
11th grade      Male 9690
11th grade      Female 10815
12th grade no diploma   Male 3304
12th grade no diploma   Female 2970
1st 2nd 3rd or 4th grade        Male 2591
1st 2nd 3rd or 4th grade        Female 2764
5th or 6th grade        Male 4761
5th or 6th grade        Female 4992
7th and 8th grade       Male 11518
7th and 8th grade       Female 12609
```

TASK 2: Employed/Unemployed based on education and their experience.

USE CASE:

1) **Literacy and Level of Education**

   Even after the literacy, it's important to know at what level of education people are at.

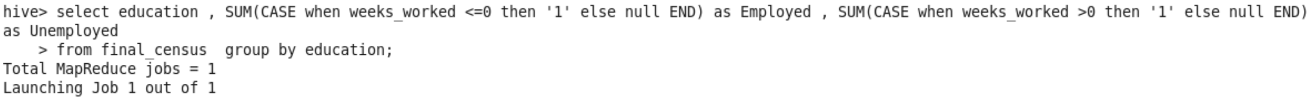
2) **A survey is required to know**

   a) Which stream in graduation helps people get a job?

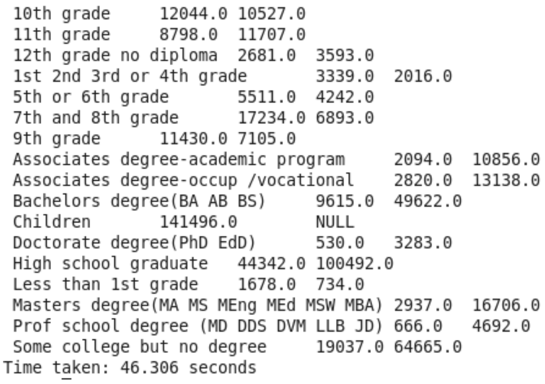
   b) How many students are already employed when they are in college?

   c) How 10 & 12 Grade students are working to earn money.

Using Hive



```
hive> select education , SUM(CASE when weeks_worked <=0 then '1' else null END) as Employed , SUM(CASE when weeks_worked >0 then '1' else null END)
as Unemployed
    > from final_census  group by education;
Total MapReduce jobs = 1
Launching Job 1 out of 1
```



```
 10th grade     12044.0 10527.0
 11th grade     8798.0  11707.0
 12th grade no diploma  2681.0  3593.0
 1st 2nd 3rd or 4th grade       3339.0  2016.0
 5th or 6th grade       5511.0  4242.0
 7th and 8th grade      17234.0 6893.0
 9th grade      11430.0 7105.0
 Associates degree-academic program     2094.0  10856.0
 Associates degree-occup /vocational    2820.0  13138.0
 Bachelors degree(BA AB BS)     9615.0  49622.0
 Children       141496.0        NULL
 Doctorate degree(PhD EdD)      530.0   3283.0
 High school graduate   44342.0 100492.0
 Less than 1st grade    1678.0  734.0
 Masters degree(MA MS MEng MEd MSW MBA) 2937.0  16706.0
 Prof school degree (MD DDS DVM LLB JD) 666.0   4692.0
 Some college but no degree     19037.0 64665.0
Time taken: 46.306 seconds
```

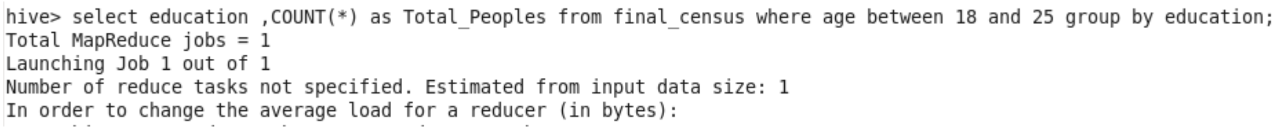
TASK 3: Total Number of people based on Education for a specific age range

USE CASE:

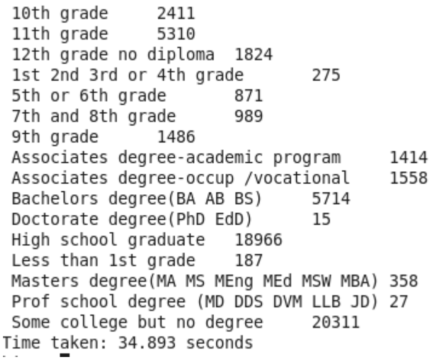
1) In specific range of age people are doing what education?

2) Within specific age range how many are 10th and 12th grade and how many are doing graduation?

Using HIVE

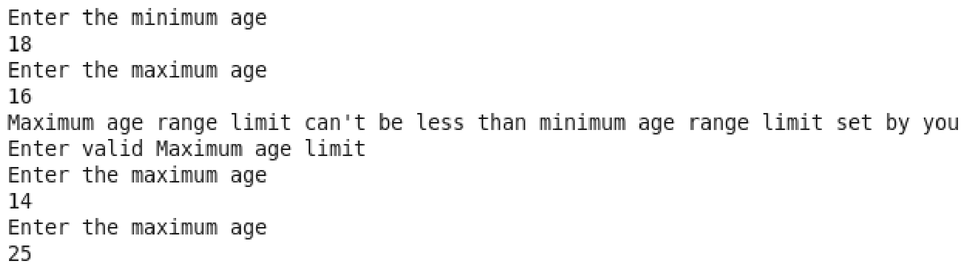


```
hive> select education ,COUNT(*) as Total_Peoples from final_census where age between 18 and 25 group by education;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
```

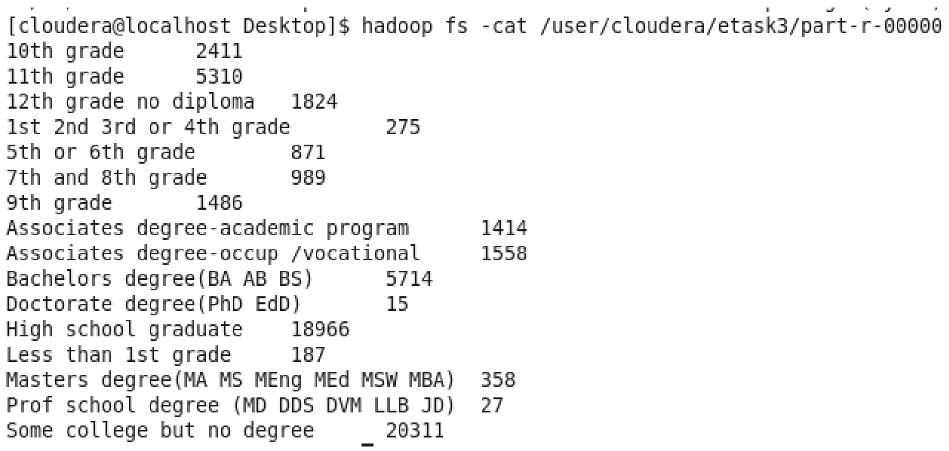


```
 10th grade     2411
 11th grade     5310
 12th grade no diploma  1824
 1st 2nd 3rd or 4th grade       275
 5th or 6th grade       871
 7th and 8th grade      989
 9th grade      1486
 Associates degree-academic program     1414
 Associates degree-occup /vocational    1558
 Bachelors degree(BA AB BS)     5714
 Doctorate degree(PhD EdD)      15
 High school graduate   18966
 Less than 1st grade    187
 Masters degree(MA MS MEng MEd MSW MBA) 358
 Prof school degree (MD DDS DVM LLB JD) 27
 Some college but no degree     20311
Time taken: 34.893 seconds
```

Using Advance MapReduce



```
Enter the minimum age
18
Enter the maximum age
16
Maximum age range limit can't be less than minimum age range limit set by you
Enter valid Maximum age limit
Enter the maximum age
14
Enter the maximum age
25
```



```
[cloudera@localhost Desktop]$ hadoop fs -cat /user/cloudera/etask3/part-r-00000
10th grade      2411
11th grade      5310
12th grade no diploma   1824
1st 2nd 3rd or 4th grade        275
5th or 6th grade        871
7th and 8th grade       989
9th grade       1486
Associates degree-academic program      1414
Associates degree-occup /vocational     1558
Bachelors degree(BA AB BS)      5714
Doctorate degree(PhD EdD)       15
High school graduate    18966
Less than 1st grade     187
Masters degree(MA MS MEng MEd MSW MBA)  358
Prof school degree (MD DDS DVM LLB JD)  27
Some college but no degree      20311
```

Finance

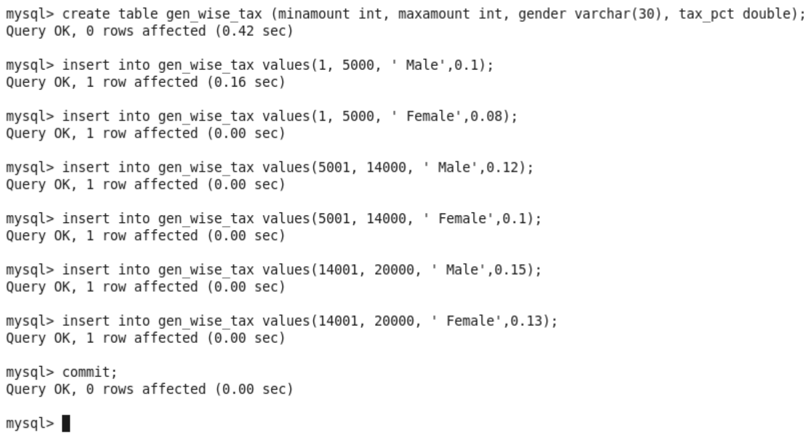
TASK 4: Total Tax Analysis Gender Wise

SCENARIO:

We here are calculating the amount of tax applicable for Male and Female.

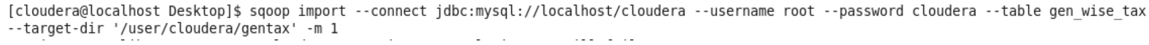
Steps Involved

a) Create a table in MySQL database for gender wise tax amount based on their income. Insert into data accordingly.



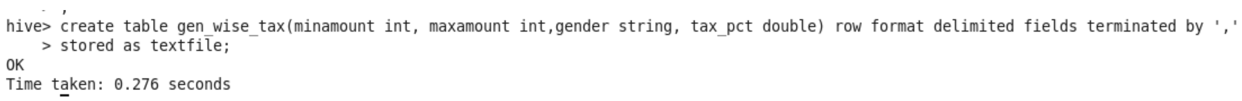
```
mysql> create table gen_wise_tax (minamount int, maxamount int, gender varchar(30), tax_pct double);
Query OK, 0 rows affected (0.42 sec)

mysql> insert into gen_wise_tax values(1, 5000, ' Male',0.1);
Query OK, 1 row affected (0.16 sec)

mysql> insert into gen_wise_tax values(1, 5000, ' Female',0.08);
Query OK, 1 row affected (0.00 sec)

mysql> insert into gen_wise_tax values(5001, 14000, ' Male',0.12);
Query OK, 1 row affected (0.00 sec)

mysql> insert into gen_wise_tax values(5001, 14000, ' Female',0.1);
Query OK, 1 row affected (0.00 sec)

mysql> insert into gen_wise_tax values(14001, 20000, ' Male',0.15);
Query OK, 1 row affected (0.00 sec)

mysql> insert into gen_wise_tax values(14001, 20000, ' Female',0.13);
Query OK, 1 row affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

b) Import gen_wise_tax table created above into HDFS using SQOOP.



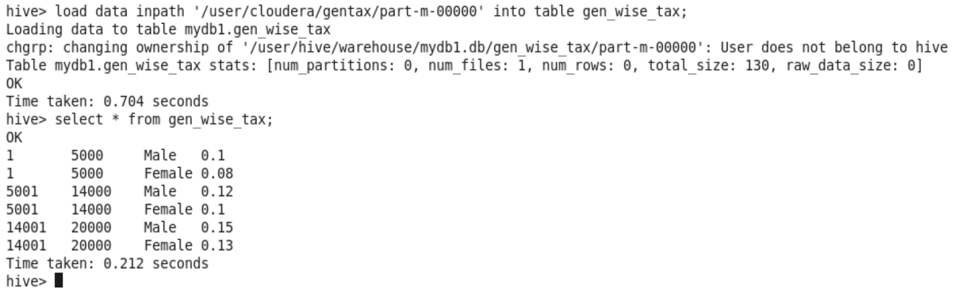
```
[cloudera@localhost Desktop]$ sqoop import --connect jdbc:mysql://localhost/cloudera --username root --password cloudera --table gen_wise_tax
--target-dir '/user/cloudera/gentax' -m 1
```

c) Create a table in HIVE named gen_wise_tax.



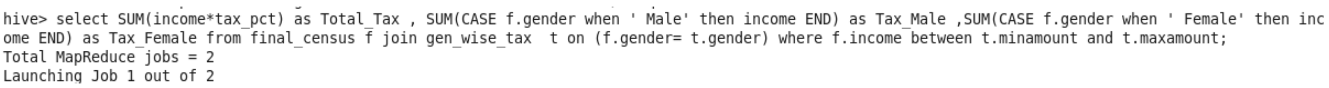
```
hive> create table gen_wise_tax(minamount int, maxamount int,gender string, tax_pct double) row format delimited fields terminated by ','
    > stored as textfile;
OK
Time taken: 0.276 seconds
```

d) Load the data into HIVE table using the file imported from SQOOP.



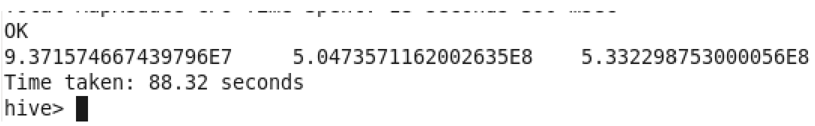
```
hive> load data inpath '/user/cloudera/gentax/part-m-00000' into table gen_wise_tax;
Loading data to table mydb1.gen_wise_tax
chgrp: changing ownership of '/user/hive/warehouse/mydb1.db/gen_wise_tax/part-m-00000': User does not belong to hive
Table mydb1.gen_wise_tax stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 130, raw_data_size: 0]
OK
Time taken: 0.704 seconds
hive> select * from gen_wise_tax;
OK
1       5000     Male   0.1
1       5000     Female 0.08
5001    14000    Male   0.12
5001    14000    Female 0.1
14001   20000    Male   0.15
14001   20000    Female 0.13
Time taken: 0.212 seconds
hive> █
```

e) HIVE query to analyse the tax based on gender and their income.



```
hive> select SUM(income*tax_pct) as Total_Tax , SUM(CASE f.gender when ' Male' then income END) as Tax_Male ,SUM(CASE f.gender when ' Female' then inc
ome END) as Tax_Female from final_census f join gen_wise_tax  t on (f.gender= t.gender) where f.income between t.minamount and t.maxamount;
Total MapReduce jobs = 2
Launching Job 1 out of 2
```

f) Output from HIVE



```
OK
9.371574667439796E7     5.0473571162002635E8    5.332298753000056E8
Time taken: 88.32 seconds
hive>
```

TASK 5: Per Capita Income (PCI) analysis Gender Wise

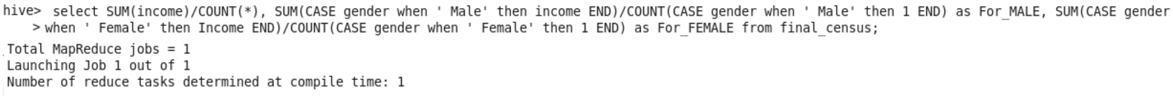
USE CASE:

1) Total Earning of different cities across the country per person.

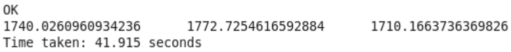
2) For a specific rural area wise male-female earnings.

Using HIVE

1) Gender Wise Per Capita Income (PCI)

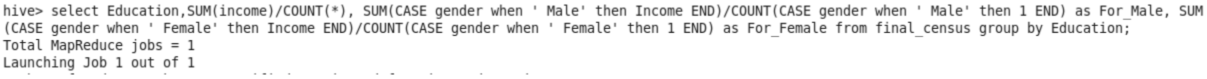


```
hive>  select SUM(income)/COUNT(*), SUM(CASE gender when ' Male' then income END)/COUNT(CASE gender when ' Male' then 1 END) as For_MALE, SUM(CASE gender
    > when ' Female' then Income END)/COUNT(CASE gender when ' Female' then 1 END) as For_FEMALE from final_census;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
```

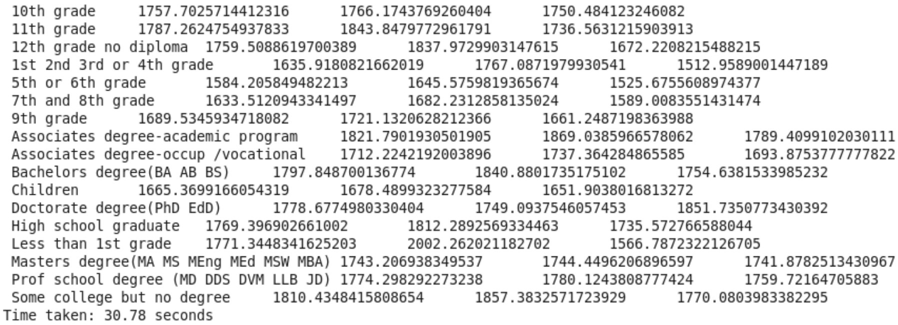


```
OK
1740.0260960934236      1772.7254616592884      1710.1663736369826
Time taken: 41.915 seconds
```

2) Education and Gender Wise Per Capita Income (PCI)



```
hive> select Education,SUM(income)/COUNT(*), SUM(CASE gender when ' Male' then Income END)/COUNT(CASE gender when ' Male' then 1 END) as For_Male, SUM
(CASE gender when ' Female' then Income END)/COUNT(CASE gender when ' Female' then 1 END) as For_Female from final_census group by Education;
Total MapReduce jobs = 1
Launching Job 1 out of 1
```



```
 10th grade     1757.7025714412316      1766.1743769260404      1750.484123246082
 11th grade     1787.2624754937833      1843.8479772961791      1736.5631215903913
 12th grade no diploma  1759.5088619700389      1837.9729903147615      1672.2208215488215
 1st 2nd 3rd or 4th grade       1635.9180821662019      1767.0871979930541      1512.9589001447189
 5th or 6th grade       1584.205849482213       1645.5759819365674      1525.6755608974377
 7th and 8th grade      1633.5120943341497      1682.2312858135024      1589.0083551431474
 9th grade      1689.5345934718082      1721.1320628212366      1661.2487198363988
 Associates degree-academic program     1821.7901930501905      1869.0385966578062      1789.409910203011
 Associates degree-occup /vocational    1712.2242192003896      1737.364284865585       1693.8753777777822
 Bachelors degree(BA AB BS)     1797.848700136774       1840.8801735175102      1754.6381533985232
 Children       1665.3699166054319      1678.4899323277584      1651.9038016813272
 Doctorate degree(PhD EdD)      1778.6774980330404      1749.0937546057453      1851.7350773430392
 High school graduate   1769.396902661002       1812.2892569334463      1735.572766588044
 Less than 1st grade    1771.3448341625203      2002.262021182702       1566.7872322126705
 Masters degree(MA MS MEng MEd MSW MBA) 1743.206938349537       1744.4496206896597      1741.8782513430967
 Prof school degree (MD DDS DVM LLB JD) 1774.298292273238       1780.1243808777424      1759.72164705883
 Some college but no degree     1810.4348415808654      1857.3832571723929      1770.0803983382295
Time taken: 30.78 seconds
```

SOCIAL

TASK 6: Total amount dispensed on pension in x years(s)

Scenario:

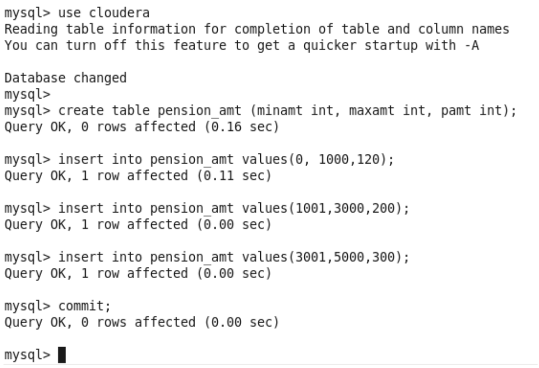
In this we are analysing the amount of pension given to the senior citizen in coming next years.

USE CASE:

1) Tax analysis gender wise and their income wise

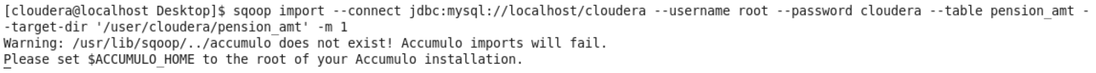
Using HIVE

a) Create a table in MySQL for pension amount give to senior citizen based on their income.



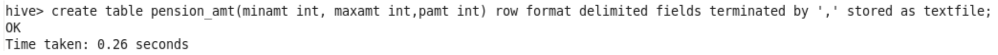
```
mysql> use cloudera
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
mysql> create table pension_amt (minamt int, maxamt int, pamt int);
Query OK, 0 rows affected (0.16 sec)

mysql> insert into pension_amt values(0, 1000,120);
Query OK, 1 row affected (0.11 sec)

mysql> insert into pension_amt values(1001,3000,200);
Query OK, 1 row affected (0.00 sec)

mysql> insert into pension_amt values(3001,5000,300);
Query OK, 1 row affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

b) Importing pension_amt table created above into the HDFS using SQOOP



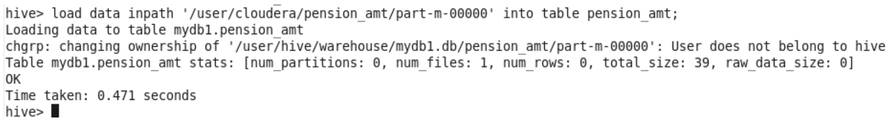
```
[cloudera@localhost Desktop]$ sqoop import --connect jdbc:mysql://localhost/cloudera --username root --password cloudera --table pension_amt -
-target-dir '/user/cloudera/pension_amt' -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

c) Create a table pension_amt in HIVE with the following columns minamt, maxamt and pamt.



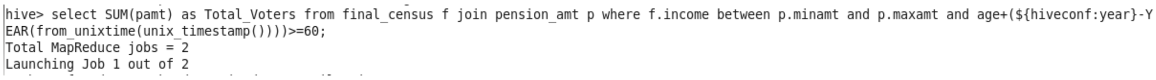
```
hive> create table pension_amt(minamt int, maxamt int,pamt int) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.26 seconds
```

d) Load the HIVE table with the data present in HDFS imported using SQOOP.



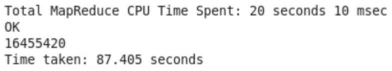
```
hive> load data inpath '/user/cloudera/pension_amt/part-m-00000' into table pension_amt;
Loading data to table mydb1.pension_amt
chgrp: changing ownership of '/user/hive/warehouse/mydb1.db/pension_amt/part-m-00000': User does not belong to hive
Table mydb1.pension_amt stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 39, raw_data_size: 0]
OK
Time taken: 0.471 seconds
hive> █
```

e) Write a HIVE QUERY to generate the total pension given to senior citizen.



```
hive> select SUM(pamt) as Total_Voters from final_census f join pension_amt p where f.income between p.minamt and p.maxamt and age+(${hiveconf:year}-Y
EAR(from_unixtime(unix_timestamp())))>=60;
Total MapReduce jobs = 2
Launching Job 1 out of 2
```

f) Output



```
Total MapReduce CPU Time Spent: 20 seconds 10 msec
OK
16455420
Time taken: 87.405 seconds
```

Using Advance MapReduce

User Input: User must enter a year in a format like 2018.

Custom Input Format Key: Income

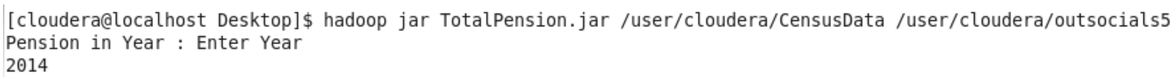
Custom Input Format Value: Age

Output: IntWritable (Total Pension Given by Government)

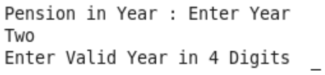
Other Concepts: Distributed Cache, Map-Side Join

Validation:

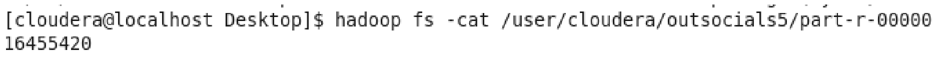
1. User can’t pass a string when asked to enter a year, so it must be a number.
2. Year in number are checked for its length. They must have 5 digits.
3. In case of violation of the above rule an Error Message is displayed to enter a valid year.



```
[cloudera@localhost Desktop]$ hadoop jar TotalPension.jar /user/cloudera/CensusData /user/cloudera/outsocials5
Pension in Year : Enter Year
2014
```



```
Pension in Year : Enter Year
Two
Enter Valid Year in 4 Digits  _
```



```
[cloudera@localhost Desktop]$ hadoop fs -cat /user/cloudera/outsocials5/part-r-00000
16455420
```

USE CASE:

Feasibility Pension policy for Senior Citizen and Divorced or Widow Females

Government is planning to form a new policy for divorced or widowed females. They plan to give the scholarship to unmarried daughter of divorced or widowed female. Government also need to check the feasibility of new pension plan for senior citizen.

TASK 7: Amount dispensed on scholarship

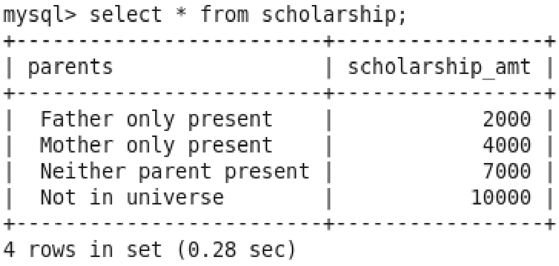
TECHNOLOGY USED:

1) PIG
2) MySQL
3) SQOOP

STEPS INVOLVED:

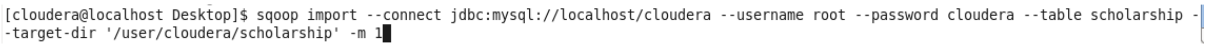
Using PIG:

a) Create a table in MySQL for scholarship.



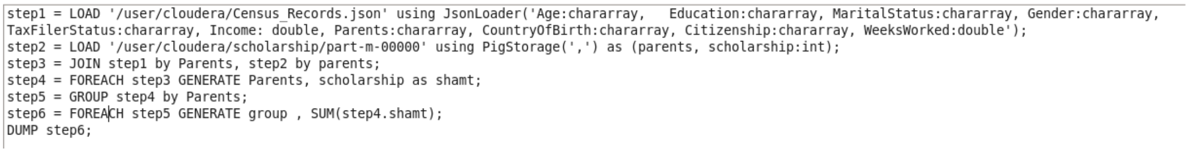
```
mysql> select * from scholarship;
+------------------------+-----------------+
| parents                | scholarship_amt |
+------------------------+-----------------+
|  Father only present   |            2000 |
|  Mother only present   |            4000 |
|  Neither parent present |           7000 |
|  Not in universe       |           10000 |
+------------------------+-----------------+
4 rows in set (0.28 sec)
```

b) Import the scholarship table using SQOOP.



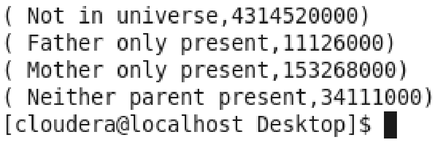
```
[cloudera@localhost Desktop]$ sqoop import --connect jdbc:mysql://localhost/cloudera --username root --password cloudera --table scholarship -
-target-dir '/user/cloudera/scholarship' -m 1
```

c) Write a PIG script.



```
step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray, Gender:chararray,
TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:double');
step2 = LOAD '/user/cloudera/scholarship/part-m-00000' using PigStorage(',') as (parents, scholarship:int);
step3 = JOIN step1 by Parents, step2 by parents;
step4 = FOREACH step3 GENERATE Parents, scholarship as shamt;
step5 = GROUP step4 by Parents;
step6 = FOREACH step5 GENERATE group , SUM(step4.shamt);
DUMP step6;
```

d) Output



```
( Not in universe,4314520000)
( Father only present,11126000)
( Mother only present,153268000)
( Neither parent present,34111000)
[cloudera@localhost Desktop]$
```

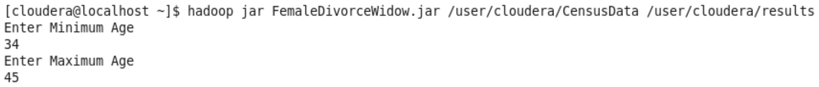
TASK 8: Employable Female widowed and divorced

TECHNOLOGY & SOFTWARE USED:

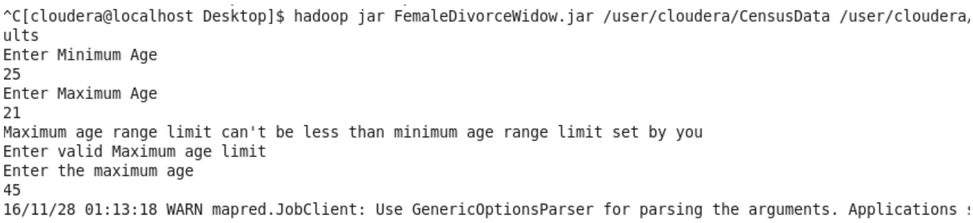
1) MapReduce In JAVA
2) Eclipse

STEPS INVOLVED:

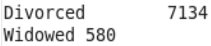
Using MapReduce



```
[cloudera@localhost ~]$ hadoop jar FemaleDivorceWidow.jar /user/cloudera/CensusData /user/cloudera/results
Enter Minimum Age
34
Enter Maximum Age
45
```



```
^C[cloudera@localhost Desktop]$ hadoop jar FemaleDivorceWidow.jar /user/cloudera/CensusData /user/cloudera,
ults
Enter Minimum Age
25
Enter Maximum Age
21
Maximum age range limit can't be less than minimum age range limit set by you
Enter valid Maximum age limit
Enter the maximum age
45
16/11/28 01:13:18 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications :
```



```
Divorced        7134
Widowed 580
```

PLANNING

SCENARIO:

India has a multi-party political system. The national parties are BJP (Bharatiya Janata Party),

INC (Indian National Congress), NCP (Nationalist Congress Party) and more.

A new political party is planning to come in for the election. And to stand ground in the election against the parties we have in India. They want to analyse the number of voters they can target this current and in the next coming year.

They have to come up with best portfolio for election and for this they need to know the number of voter, senior citizen they have in their local area, city and country.

Plus the

Total number of male and female voters is there?

What local and foreign people ratio is?

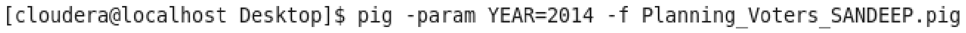
TASK 9 Total Voters in x year(s)

TECHNOLOGY USED:

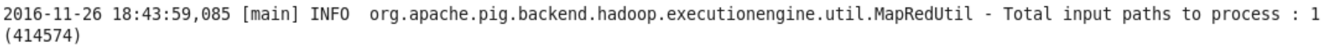
1) PIG
2) HIVE
3) ADVANCE MAP REDUCE

Using PIG

```
step1 = LOAD '/user/cloudera/CensusData' using PigStorage(',') as (age : int , education , marital_status , gender , tax_fil_status , income: double ,
parents , country_birth , citizenship , weeks_worked );
step2 = FILTER step1 by age + ($YEAR-GetYear(CurrentTime()))>=18;
step3 = FOREACH step2 GENERATE 1 as one, age;
step4 = GROUP step3 by one;step5 = FOREACH step4 GENERATE COUNT(step3.age) as TOTAL_VOTERS;
DUMP step5;
```

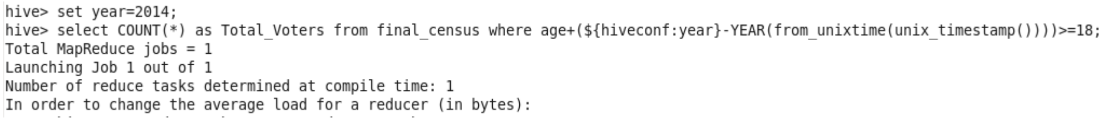


```
[cloudera@localhost Desktop]$ pig -param YEAR=2014 -f Planning_Voters_SANDEEP.pig
```

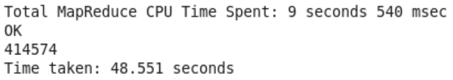


```
2016-11-26 18:43:59,085 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(414574)
```

Using HIVE



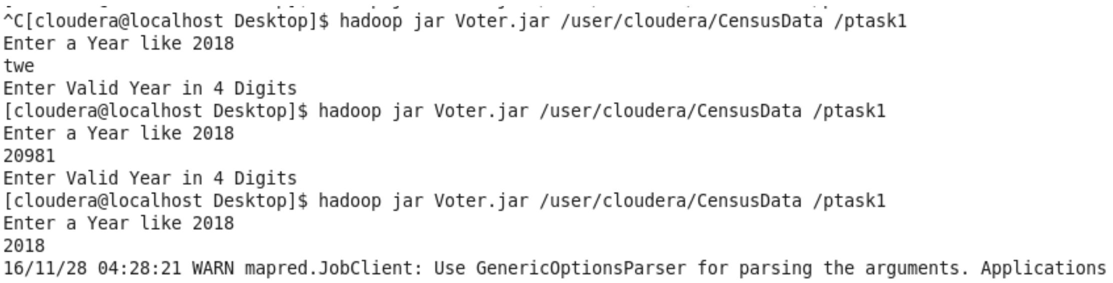
```
hive> set year=2014;
hive> select COUNT(*) as Total_Voters from final_census where age+(${hiveconf:year}-YEAR(from_unixtime(unix_timestamp())))>=18;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
```



```
Total MapReduce CPU Time Spent: 9 seconds 540 msec
OK
414574
Time taken: 48.551 seconds
```

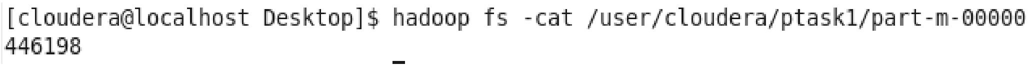
Using Advance Map Reduce

1) Validation



```
^C[cloudera@localhost Desktop]$ hadoop jar Voter.jar /user/cloudera/CensusData /ptask1
Enter a Year like 2018
twe
Enter Valid Year in 4 Digits
[cloudera@localhost Desktop]$ hadoop jar Voter.jar /user/cloudera/CensusData /ptask1
Enter a Year like 2018
20981
Enter Valid Year in 4 Digits
[cloudera@localhost Desktop]$ hadoop jar Voter.jar /user/cloudera/CensusData /ptask1
Enter a Year like 2018
2018
16/11/28 04:28:21 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications
```

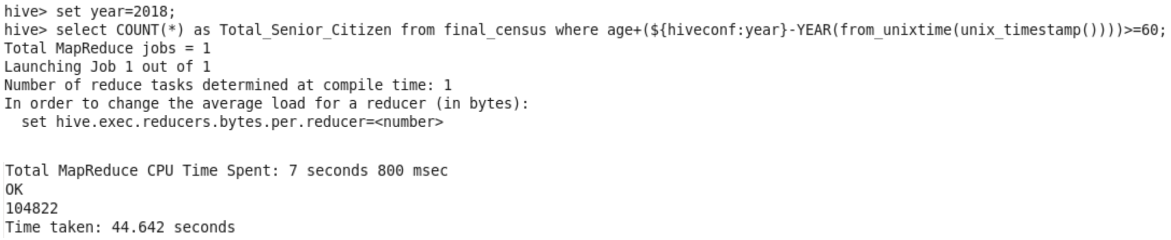
2)



```
[cloudera@localhost Desktop]$ hadoop fs -cat /user/cloudera/ptask1/part-m-00000
446198
```

TASK 10 Senior Citizen Count in x year

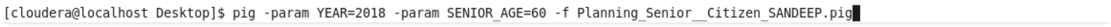
Using HIVE



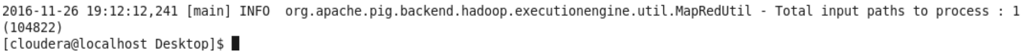
```
hive> set year=2018;
hive> select COUNT(*) as Total_Senior_Citizen from final_census where age+(${hiveconf:year}-YEAR(from_unixtime(unix_timestamp())))>=60;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>

Total MapReduce CPU Time Spent: 7 seconds 800 msec
OK
104822
Time taken: 44.642 seconds
```

Using PIG

```
step1 = LOAD '/user/cloudera/CensusData' using PigStorage(',') as (age : int , education , marital_status , gender , tax_fil_status , income: double ,
 parents , country_birth , citizenship , weeks_worked );
step2 = FILTER step1 by age + ($YEAR-GetYear(CurrentTime()))>=$SENIOR_AGE;
step3 = FOREACH step2 GENERATE 1 as one, age;
step4 = GROUP step3 by one;
step5 = FOREACH step4 GENERATE COUNT(step3.age) as TOTAL_SENIOR_CITIZEN;
DUMP step5;
```

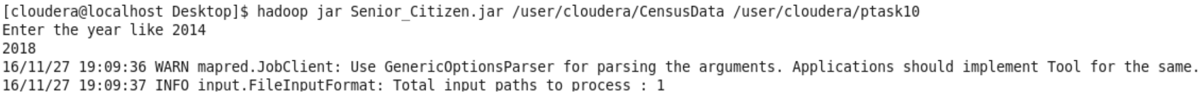


```
[cloudera@localhost Desktop]$ pig -param YEAR=2018 -param SENIOR_AGE=60 -f Planning_Senior__Citizen_SANDEEP.pig
```



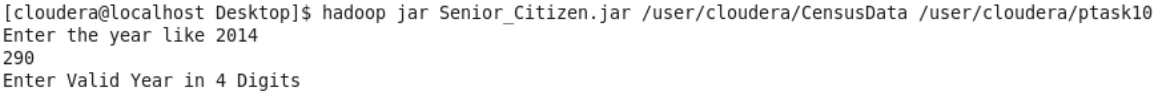
```
2016-11-26 19:12:12,241 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(104822)
[cloudera@localhost Desktop]$
```

Using MapReduce

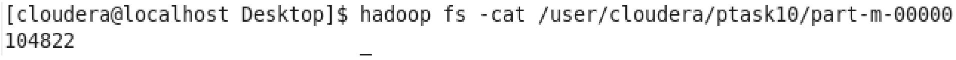


```
[cloudera@localhost Desktop]$ hadoop jar Senior_Citizen.jar /user/cloudera/CensusData /user/cloudera/ptask10
Enter the year like 2014
2018
16/11/27 19:09:36 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
16/11/27 19:09:37 INFO input.FileInputFormat: Total input paths to process : 1
```

Validation



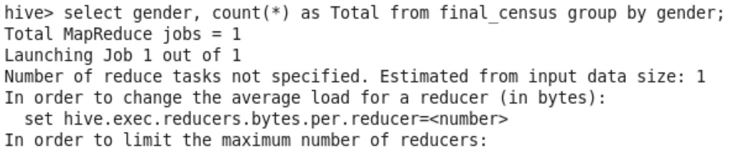
```
[cloudera@localhost Desktop]$ hadoop jar Senior_Citizen.jar /user/cloudera/CensusData /user/cloudera/ptask10
Enter the year like 2014
290
Enter Valid Year in 4 Digits
```



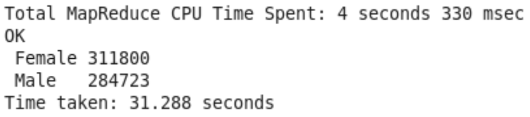
```
[cloudera@localhost Desktop]$ hadoop fs -cat /user/cloudera/ptask10/part-m-00000
104822
```

TASK 11Total Male/Female

Using HIVE



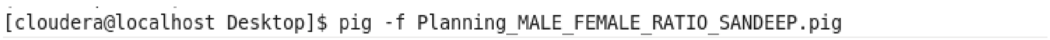
```
hive> select gender, count(*) as Total from final_census group by gender;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
```



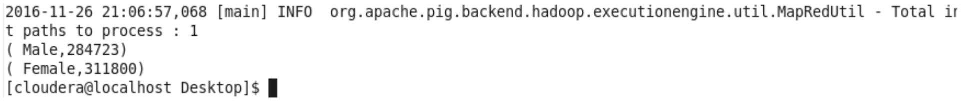
```
Total MapReduce CPU Time Spent: 4 seconds 330 msec
OK
 Female 311800
 Male   284723
Time taken: 31.288 seconds
```

Using PIG

step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:chararray');
step2 = GROUP step1 by Gender;
step3 = FOREACH step2 GENERATE group, COUNT(step1.Gender) as Total_Num;
DUMP step3;

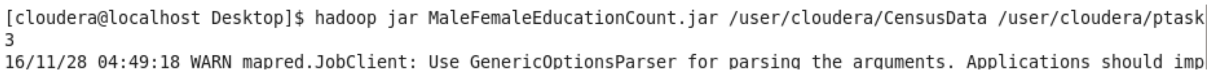


```
[cloudera@localhost Desktop]$ pig -f Planning_MALE_FEMALE_RATIO_SANDEEP.pig
```

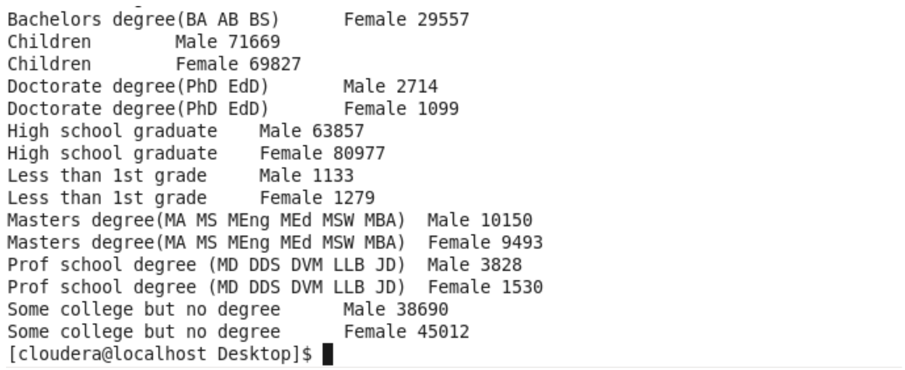


```
2016-11-26 21:06:57,068 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total i
t paths to process : 1
( Male,284723)
( Female,311800)
[cloudera@localhost Desktop]$ 
```

Using Simple MapReduce



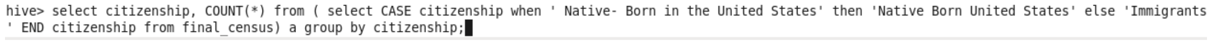
```
[cloudera@localhost Desktop]$ hadoop jar MaleFemaleEducationCount.jar /user/cloudera/CensusData /user/cloudera/ptask
3
16/11/28 04:49:18 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should imp
```



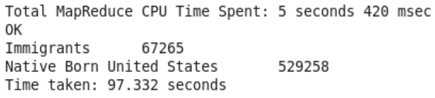
```
Bachelors degree(BA AB BS)      Female 29557
Children        Male 71669
Children        Female 69827
Doctorate degree(PhD EdD)       Male 2714
Doctorate degree(PhD EdD)       Female 1099
High school graduate    Male 63857
High school graduate    Female 80977
Less than 1st grade     Male 1133
Less than 1st grade     Female 1279
Masters degree(MA MS MEng MEd MSW MBA)  Male 10150
Masters degree(MA MS MEng MEd MSW MBA)  Female 9493
Prof school degree (MD DDS DVM LLB JD)  Male 3828
Prof school degree (MD DDS DVM LLB JD)  Female 1530
Some college but no degree      Male 38690
Some college but no degree      Female 45012
[cloudera@localhost Desktop]$ 
```

TASK 12 Citizens and Immigrants ratio

Using HIVE



```
hive> select citizenship, COUNT(*) from ( select CASE citizenship when ' Native- Born in the United States' then 'Native Born United States' else 'Immigrants
' END citizenship from final_census) a group by citizenship;
```



```
Total MapReduce CPU Time Spent: 5 seconds 420 msec
OK
Immigrants      67265
Native Born United States       529258
Time taken: 97.332 seconds
```

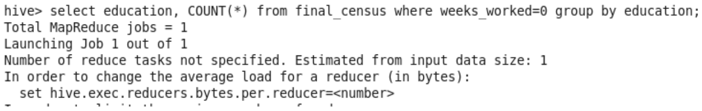
Miscellaneous

TASK 13 Degree Wise count for employability

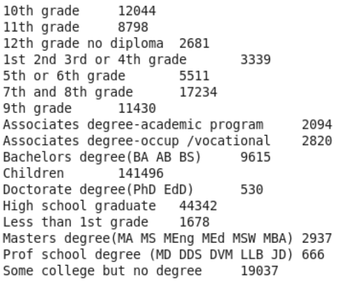
USE CASE:

RURAL development is going on

Using HIVE



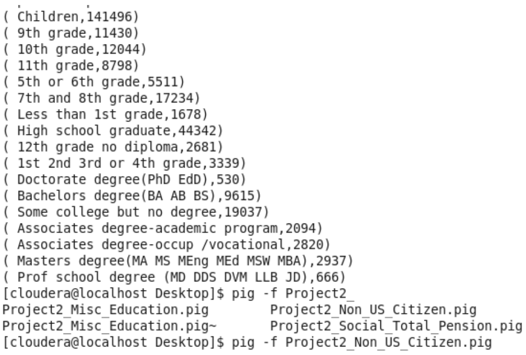
```
hive> select education, COUNT(*) from final_census where weeks_worked=0 group by education;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```



```
10th grade      12044
11th grade      8798
12th grade no diploma  2681
1st 2nd 3rd or 4th grade       3339
5th or 6th grade       5511
7th and 8th grade      17234
9th grade      11430
Associates degree-academic program     2094
Associates degree-occup /vocational    2820
Bachelors degree(BA AB BS)     9615
Children       141496
Doctorate degree(PhD EdD)      530
High school graduate   44342
Less than 1st grade    1678
Masters degree(MA MS MEng MEd MSW MBA) 2937
Prof school degree (MD DDS DVM LLB JD) 666
Some college but no degree     19037
```

Using PIG

step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:double');
step2 = FILTER step1 by WeeksWorked==0;
step3 = GROUP step2 by Education;
step4 = FOREACH step3 GENERATE group, COUNT(step2.Age) as Total;
DUMP step4;



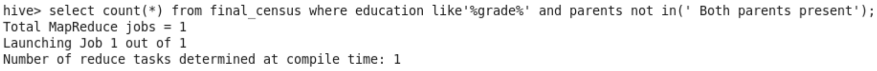
```
( Children,141496)
( 9th grade,11430)
( 10th grade,12044)
( 11th grade,8798)
( 5th or 6th grade,5511)
( 7th and 8th grade,17234)
( Less than 1st grade,1678)
( High school graduate,44342)
( 12th grade no diploma,2681)
( 1st 2nd 3rd or 4th grade,3339)
( Doctorate degree(PhD EdD),530)
( Bachelors degree(BA AB BS),9615)
( Some college but no degree,19037)
( Associates degree-academic program,2094)
( Associates degree-occup /vocational,2820)
( Masters degree(MA MS MEng MEd MSW MBA),2937)
( Prof school degree (MD DDS DVM LLB JD),666)
[cloudera@localhost Desktop]$ pig -f Project2_
Project2_Misc_Education.pig        Project2_Non_US_Citizen.pig
Project2_Misc_Education.pig~       Project2_Social_Total_Pension.pig
[cloudera@localhost Desktop]$ pig -f Project2_Non_US_Citizen.pig
```

TASK 14 Customer base analyses

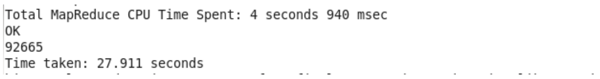
People doing graduation and either have one parent or none

USE CASE:

Using HIVE



```
hive> select count(*) from final_census where education like'%grade%' and parents not in(' Both parents present');
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
```



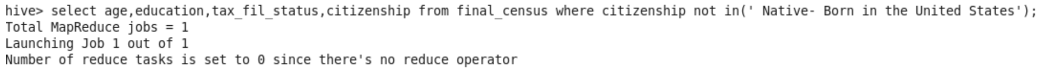
```
Total MapReduce CPU Time Spent: 4 seconds 940 msec
OK
92665
Time taken: 27.911 seconds
```

Using PIG

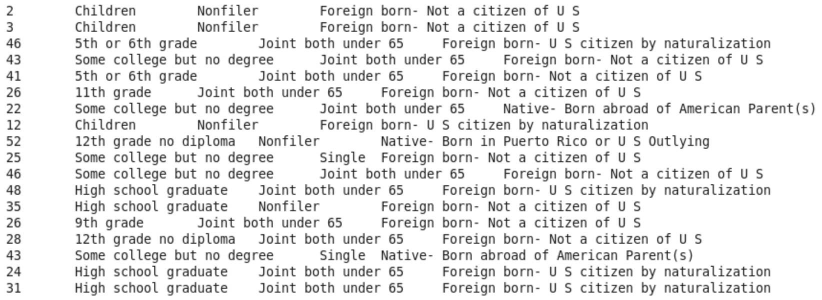
```
step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray,
Gender:chararray, TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray,
Citizenship:chararray, WeeksWorked:double');
step2 = FILTER step1 by Education matches '.*grade.*';
step3 = FILTER step2 by not(Parents matches ' Both parents present');
step4 = FOREACH step3 GENERATE 1 as one, Education;
step5 = GROUP step4 by one;
step6 = FOREACH step5 GENERATE COUNT(step4.Education) as Edcnt;
DUMP step6;
```

TASK 15 Non-US citizen tax filer statuses

Using HIVE



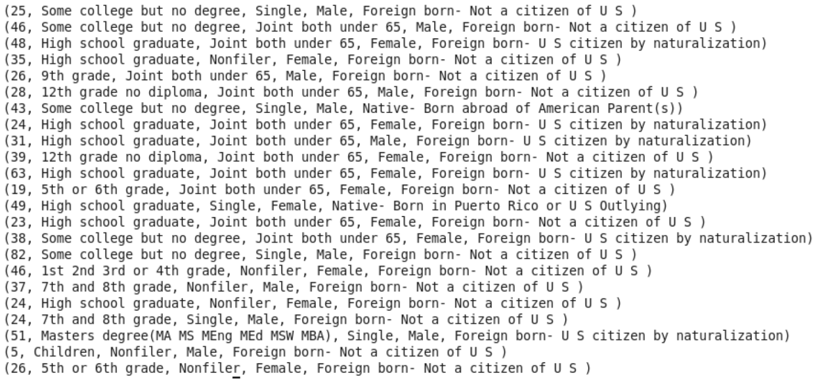
```
hive> select age,education,tax_fil_status,citizenship from final_census where citizenship not in(' Native- Born in the United States');
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
```



```
2       Children        Nonfiler        Foreign born- Not a citizen of U S
3       Children        Nonfiler        Foreign born- Not a citizen of U S
46      5th or 6th grade        Joint both under 65     Foreign born- U S citizen by naturalization
43      Some college but no degree      Joint both under 65     Foreign born- Not a citizen of U S
41      5th or 6th grade        Joint both under 65     Foreign born- Not a citizen of U S
26      11th grade      Joint both under 65     Foreign born- Not a citizen of U S
22      Some college but no degree      Joint both under 65     Native- Born abroad of American Parent(s)
12      Children        Nonfiler        Foreign born- U S citizen by naturalization
52      12th grade no diploma   Nonfiler        Native- Born in Puerto Rico or U S Outlying
25      Some college but no degree      Single  Foreign born- Not a citizen of U S
46      Some college but no degree      Joint both under 65     Foreign born- Not a citizen of U S
48      High school graduate    Joint both under 65     Foreign born- U S citizen by naturalization
35      High school graduate    Nonfiler        Foreign born- Not a citizen of U S
26      9th grade       Joint both under 65     Foreign born- Not a citizen of U S
28      12th grade no diploma   Joint both under 65     Foreign born- Not a citizen of U S
43      Some college but no degree      Single  Native- Born abroad of American Parent(s)
24      High school graduate    Joint both under 65     Foreign born- U S citizen by naturalization
31      High school graduate    Joint both under 65     Foreign born- U S citizen by naturalization
```

Using PIG

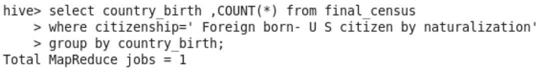
```
step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray,
Gender:chararray, TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray,
WeeksWorked:double');
step2 = FILTER step1 by not (Citizenship matches ' Native- Born in the United States');
step3 =OREACH step2 GENERATE Age, Education, TaxFilerStatus,Gender, Citizenship;
DUMP step3;
```



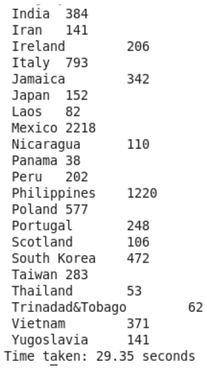
```
(25, Some college but no degree, Single, Male, Foreign born- Not a citizen of U S )
(46, Some college but no degree, Joint both under 65, Male, Foreign born- Not a citizen of U S )
(48, High school graduate, Joint both under 65, Female, Foreign born- U S citizen by naturalization)
(35, High school graduate, Nonfiler, Female, Foreign born- Not a citizen of U S )
(26, 9th grade, Joint both under 65, Male, Foreign born- Not a citizen of U S )
(28, 12th grade no diploma, Joint both under 65, Male, Foreign born- Not a citizen of U S )
(43, Some college but no degree, Single, Male, Native- Born abroad of American Parent(s))
(24, High school graduate, Joint both under 65, Female, Foreign born- U S citizen by naturalization)
(31, High school graduate, Joint both under 65, Male, Foreign born- U S citizen by naturalization)
(39, 12th grade no diploma, Joint both under 65, Female, Foreign born- Not a citizen of U S )
(63, High school graduate, Joint both under 65, Female, Foreign born- U S citizen by naturalization)
(19, 5th or 6th grade, Joint both under 65, Female, Foreign born- Not a citizen of U S )
(49, High school graduate, Single, Female, Native- Born in Puerto Rico or U S Outlying)
(23, High school graduate, Joint both under 65, Female, Foreign born- Not a citizen of U S )
(38, Some college but no degree, Joint both under 65, Female, Foreign born- U S citizen by naturalization)
(82, Some college but no degree, Single, Male, Foreign born- Not a citizen of U S )
(46, 1st 2nd 3rd or 4th grade, Nonfiler, Female, Foreign born- Not a citizen of U S )
(37, 7th and 8th grade, Nonfiler, Male, Foreign born- Not a citizen of U S )
(24, High school graduate, Nonfiler, Female, Foreign born- Not a citizen of U S )
(24, 7th and 8th grade, Single, Male, Foreign born- Not a citizen of U S )
(51, Masters degree(MA MS MEng MEd MSW MBA), Single, Male, Foreign born- U S citizen by naturalization)
(5, Children, Nonfiler, Male, Foreign born- Not a citizen of U S )
(26, 5th or 6th grade, Nonfiler, Female, Foreign born- Not a citizen of U S )
```

TASK 16 Country of Birth wise counts for US citizenship by naturalisation

Using HIVE



```
hive> select country_birth ,COUNT(*) from final_census
    > where citizenship=' Foreign born- U S citizen by naturalization'
    > group by country_birth;
Total MapReduce jobs = 1
```



```
 India  384
 Iran   141
 Ireland        206
 Italy  793
 Jamaica        342
 Japan  152
 Laos   82
 Mexico 2218
 Nicaragua      110
 Panama 38
 Peru   202
 Philippines    1220
 Poland 577
 Portugal       248
 Scotland       106
 South Korea    472
 Taiwan 283
 Thailand       53
 Trinadad&Tobago        62
 Vietnam        371
 Yugoslavia     141
Time taken: 29.35 seconds
```

Using PIG

step1 = LOAD '/user/cloudera/Census_Records.json' using JsonLoader('Age:chararray,   Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income: double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:double');
step2 = FILTER step1 by Citizenship matches ' Foreign born- U S citizen by naturalization';
step3 = GROUP step2 by CountryOfBirth;
step4 = FOREACH step3 GENERATE group, COUNT(step2.Age);
DUMP step4;

```
( Taiwan,283)
( Ecuador,192)
( England,496)
( Germany,1054)
( Hungary,187)
( Ireland,206)
( Jamaica,342)
( Vietnam,371)
( Cambodia,75)
( Columbia,397)
( Honduras,87)
( Portugal,248)
( Scotland,106)
( Thailand,53)
( Guatemala,98)
( Hong Kong,99)
( Nicaragua,110)
( Yugoslavia,141)
( El-Salvador,227)
( Philippines,1220)
( South Korea,472)
( Trinadad&Tobago,62)
( Dominican-Republic,379)
( Holand-Netherlands,28)
```