Internal Assignment 2

Student Id: 2048494

Student Name: Dhaniswar B.K.

Module Leader: Jnaneshwar Bohara

Tutor: Sujan Shrestha

Section: L6CG5

Submitted on : 13/04/2022

Table of Content

1. Dividing my university id by 3.

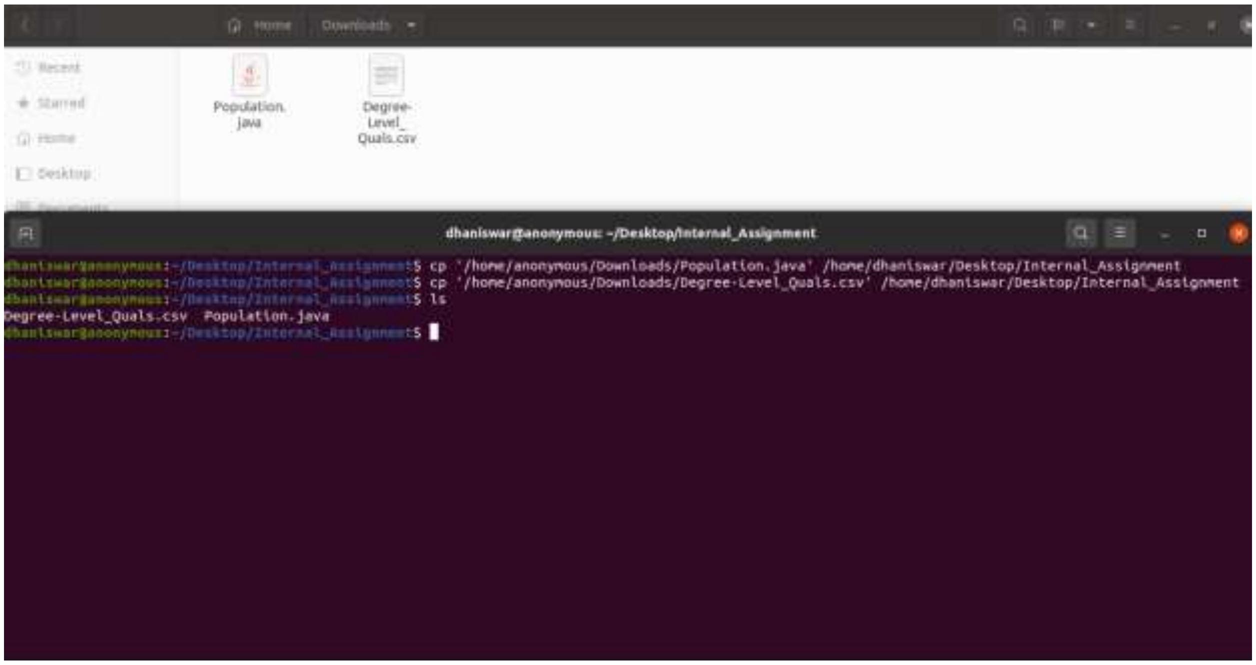
Answer: calculating the remainder of the university id using short trick

Step I => 2048494 = 2+0+4+8+4+9+4 => 31 => 3+1 = 4

Now, 4%3 = 1 So, I have to use Degree-Level_Quals.csv data set for this assessment.

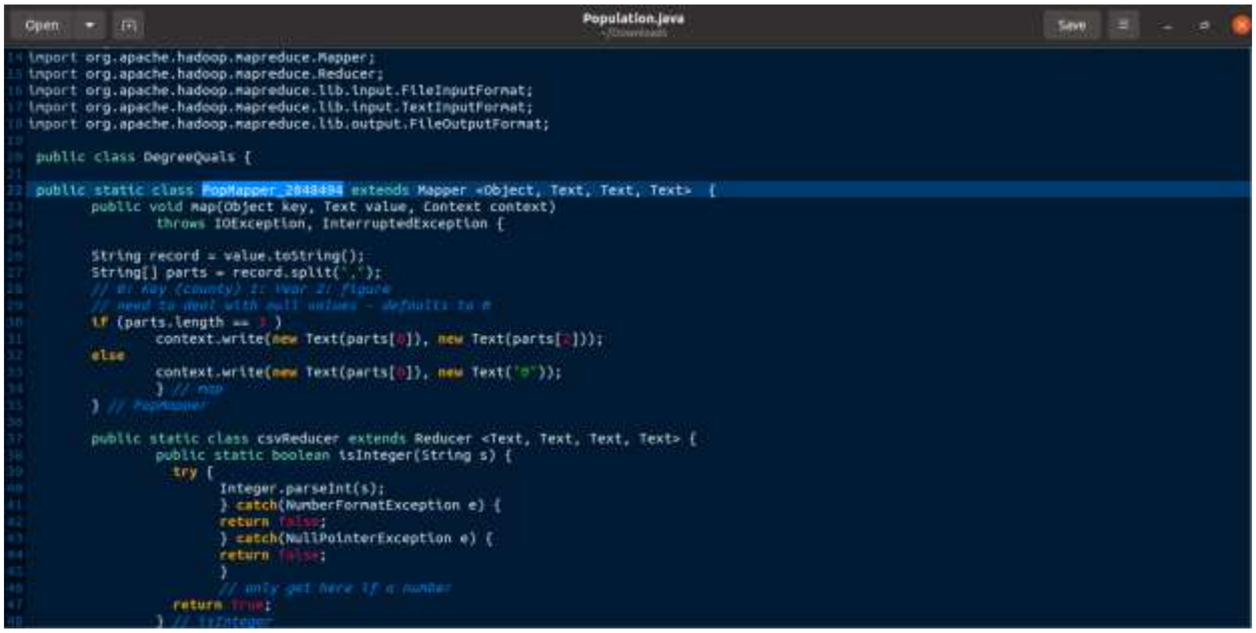
2. Java and Hadoop

2.1. Copying the required files to the destination



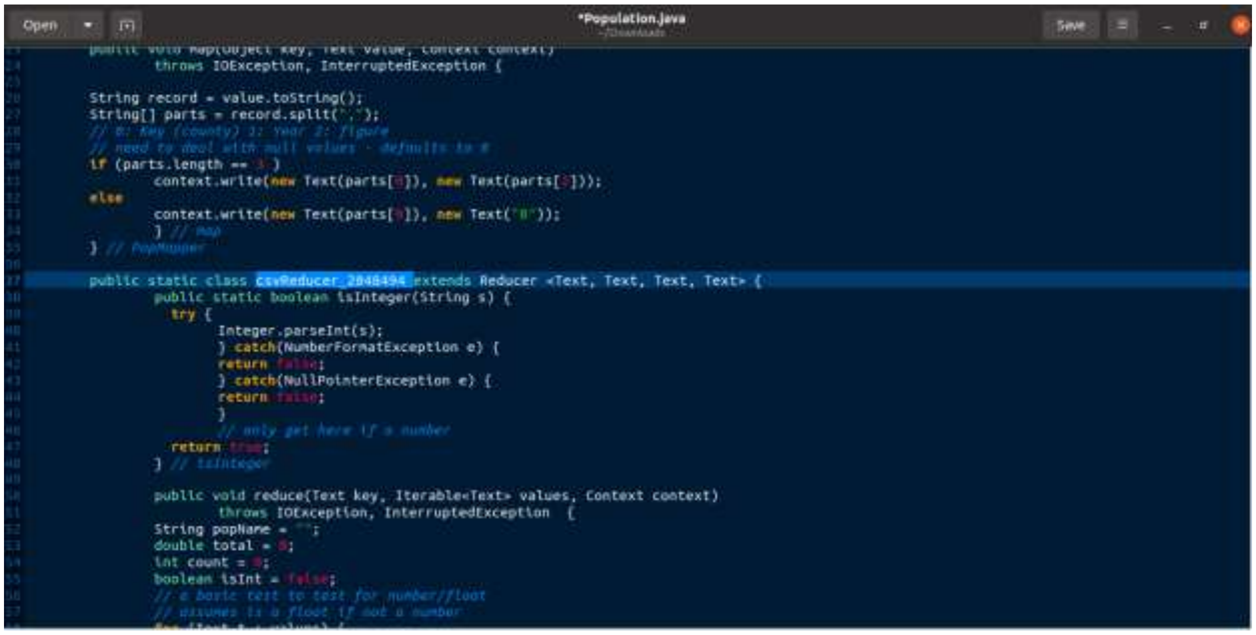
```
dhaniswar@anonymous: ~/Desktop/Internal_Assignment
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ cp '/home/anonymous/Downloads/Population.java' /home/dhaniswar/Desktop/Internal_Assignment
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ cp '/home/anonymous/Downloads/Degree-Level_Quals.csv' /home/dhaniswar/Desktop/Internal_Assignment
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
Degree-Level_Quals.csv  Population.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ 
```

2.2. Amending the Population class name to the DegreeQuals class



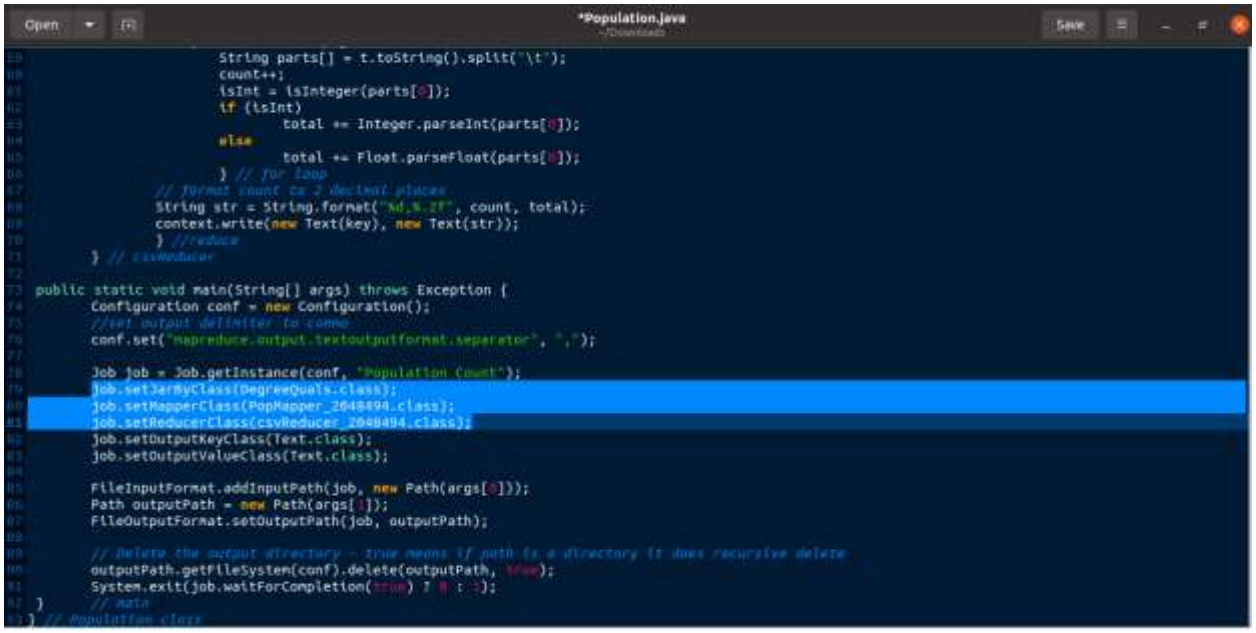
```
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

 public class DegreeQuals {

 public static class PopMapper_2048494 extends Mapper <Object, Text, Text, Text>  {
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {

        String record = value.toString();
        String[] parts = record.split(",");
        // 0: key (county) 1: Year 2: figure
        // need to deal with null values - defaults to 0
        if (parts.length == 3 )
                context.write(new Text(parts[0]), new Text(parts[2]));
        else
                context.write(new Text(parts[0]), new Text("0"));
                } // map
        } // PopMapper

        public static class csvReducer extends Reducer <Text, Text, Text, Text> {
                public static boolean isInteger(String s) {
                  try {
                        Integer.parseInt(s);
                        } catch(NumberFormatException e) {
                        return false;
                        } catch(NullPointerException e) {
                        return false;
                        }
                        // only get here if a number
                  return true;
                } // isInteger
```

2.3. Amend the Mapper and Reducer class names to Maper_2048494 and Reducer_2048494 respectively



```
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {

        String record = value.toString();
        String[] parts = record.split(",");
        // 0: key (county) 1: Year 2: figure
        // need to deal with null values - defaults to 0
        if (parts.length == 3 )
                context.write(new Text(parts[0]), new Text(parts[2]));
        else
                context.write(new Text(parts[0]), new Text("0"));
                } // map
        } // PopMapper

        public static class csvReducer_2048494 extends Reducer <Text, Text, Text, Text> {
                public static boolean isInteger(String s) {
                  try {
                        Integer.parseInt(s);
                        } catch(NumberFormatException e) {
                        return false;
                        } catch(NullPointerException e) {
                        return false;
                        }
                        // only get here if a number
                  return true;
                } // isInteger

                public void reduce(Text key, Iterable<Text> values, Context context)
                        throws IOException, InterruptedException  {
                String popName = "";
                double total = 0;
                int count = 0;
                boolean isInt = false;
                // a basic test to test for number/float
                // assumes is a float if not a number
```

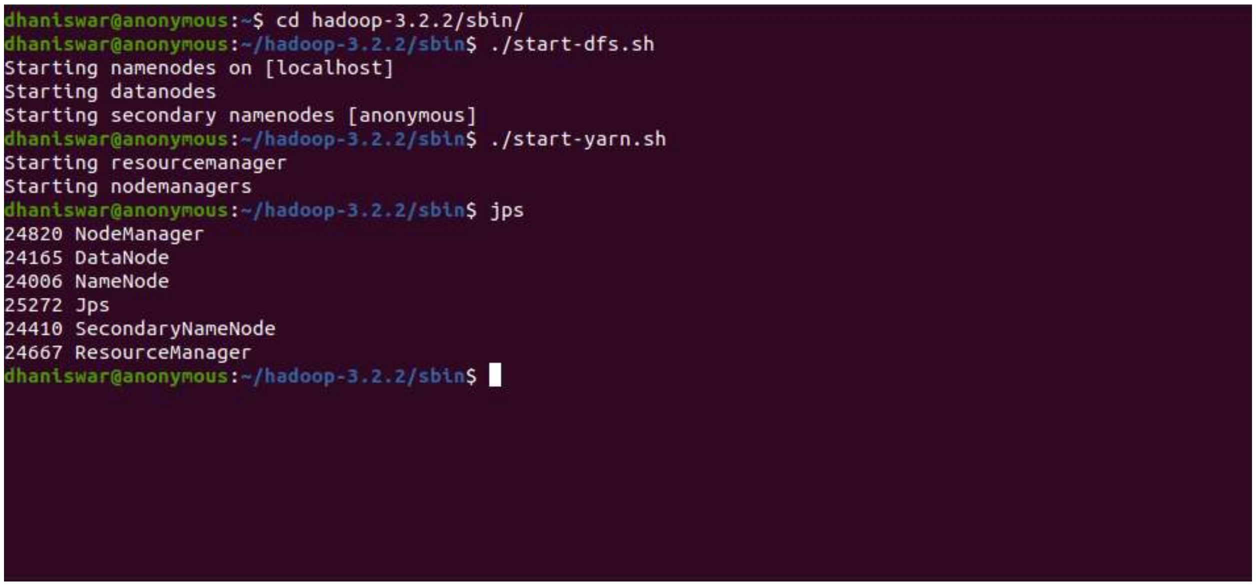
Reflecting these changes in the main method



```
                        String parts[] = t.toString().split('\t');
                        count++;
                        isInt = isInteger(parts[0]);
                        if (isInt)
                                total += Integer.parseInt(parts[0]);
                        else
                                total += Float.parseFloat(parts[0]);
                        } // for loop
                // format count to 2 decimal places
                String str = String.format("%d,%.2f", count, total);
                context.write(new Text(key), new Text(str));
                } //reduce
        } // csvReducer

public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        //set output delimiter to comma
        conf.set("mapreduce.output.textoutputformat.seperator", ",");

        Job job = Job.getInstance(conf, "Population Count");
        job.setJarByClass(DegreeQuals.class);
        job.setMapperClass(PopMapper_2048494.class);
        job.setReducerClass(csvReducer_2048494.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        Path outputPath = new Path(args[1]);
        FileOutputFormat.setOutputPath(job, outputPath);

        // Delete the output directory - true means if path is a directory it does recursive delete
        outputPath.getFileSystem(conf).delete(outputPath, true);
        System.exit(job.waitForCompletion(true) ? 0 : 1);
}       // main
} // Population class
```

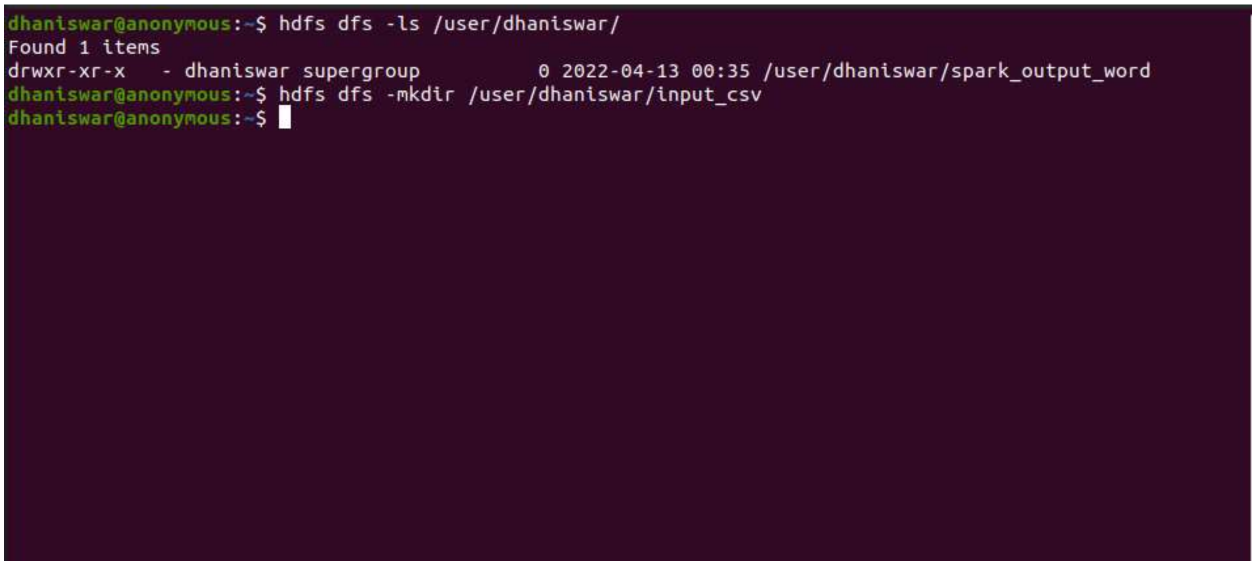
3. Run the code

3.1. Starting the jps



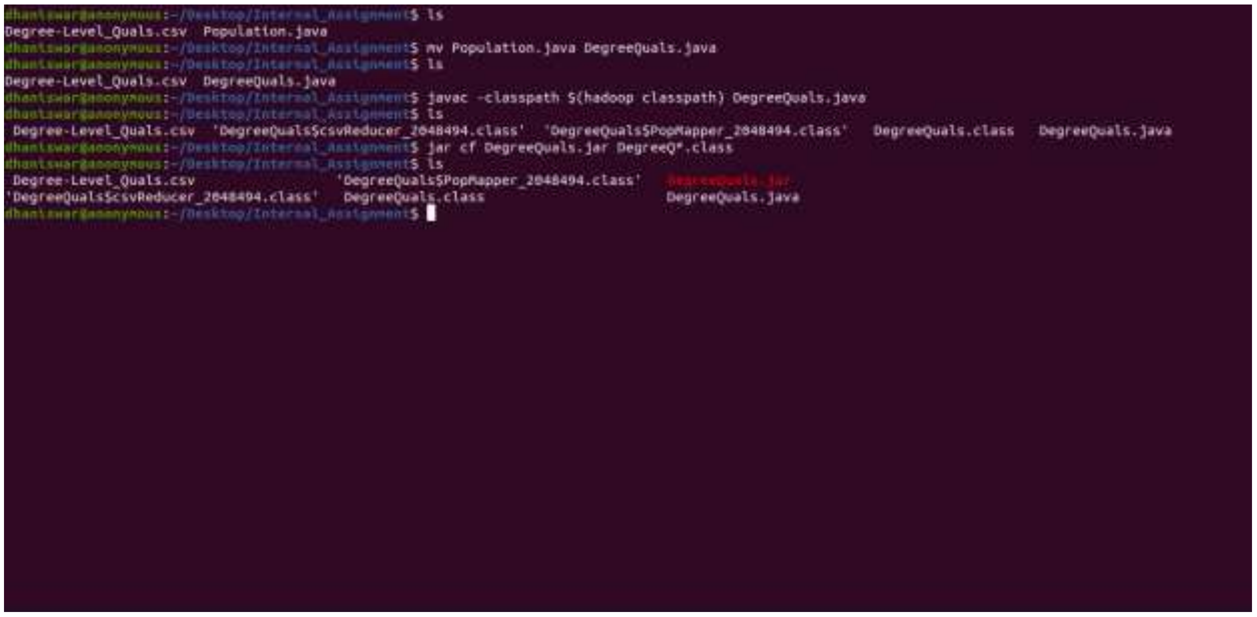
```
dhaniswar@anonymous:~$ cd hadoop-3.2.2/sbin/
dhaniswar@anonymous:~/hadoop-3.2.2/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [anonymous]
dhaniswar@anonymous:~/hadoop-3.2.2/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
dhaniswar@anonymous:~/hadoop-3.2.2/sbin$ jps
24820 NodeManager
24165 DataNode
24006 NameNode
25272 Jps
24410 SecondaryNameNode
24667 ResourceManager
dhaniswar@anonymous:~/hadoop-3.2.2/sbin$ 
```

3.2. Checking the directory in the Hadoop file system



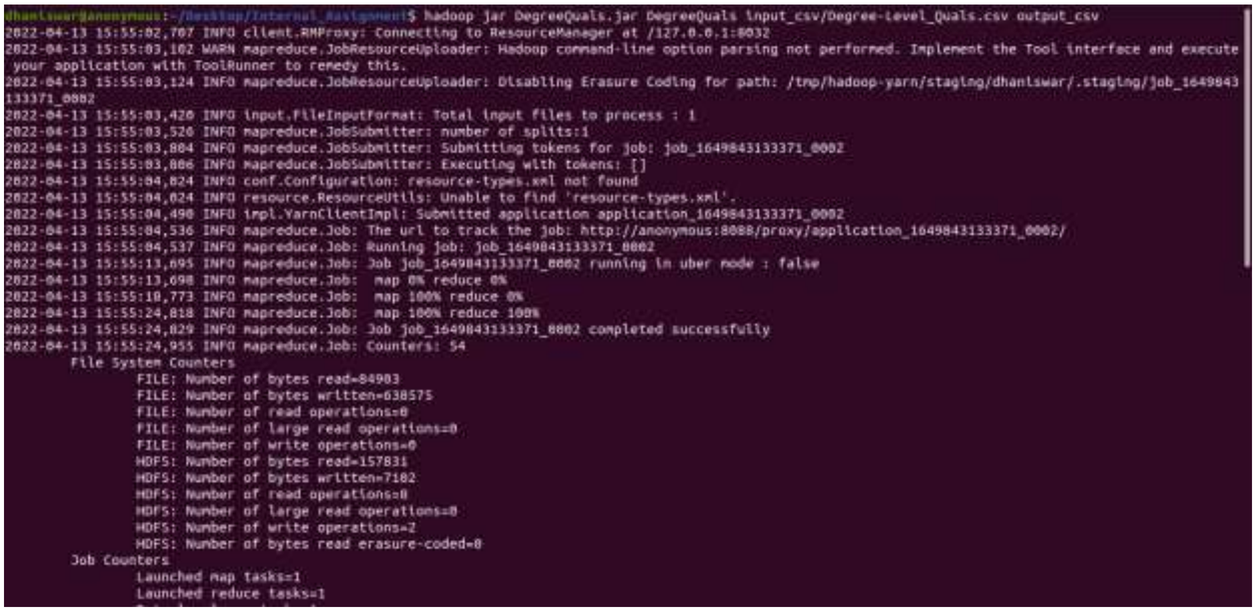
```
dhaniswar@anonymous:~$ hdfs dfs -ls /user/dhaniswar/
Found 1 items
drwxr-xr-x   - dhaniswar supergroup          0 2022-04-13 00:35 /user/dhaniswar/spark_output_word
dhaniswar@anonymous:~$ hdfs dfs -mkdir /user/dhaniswar/input_csv
dhaniswar@anonymous:~$
```

3.3. Compiling the java file and creating the jar file

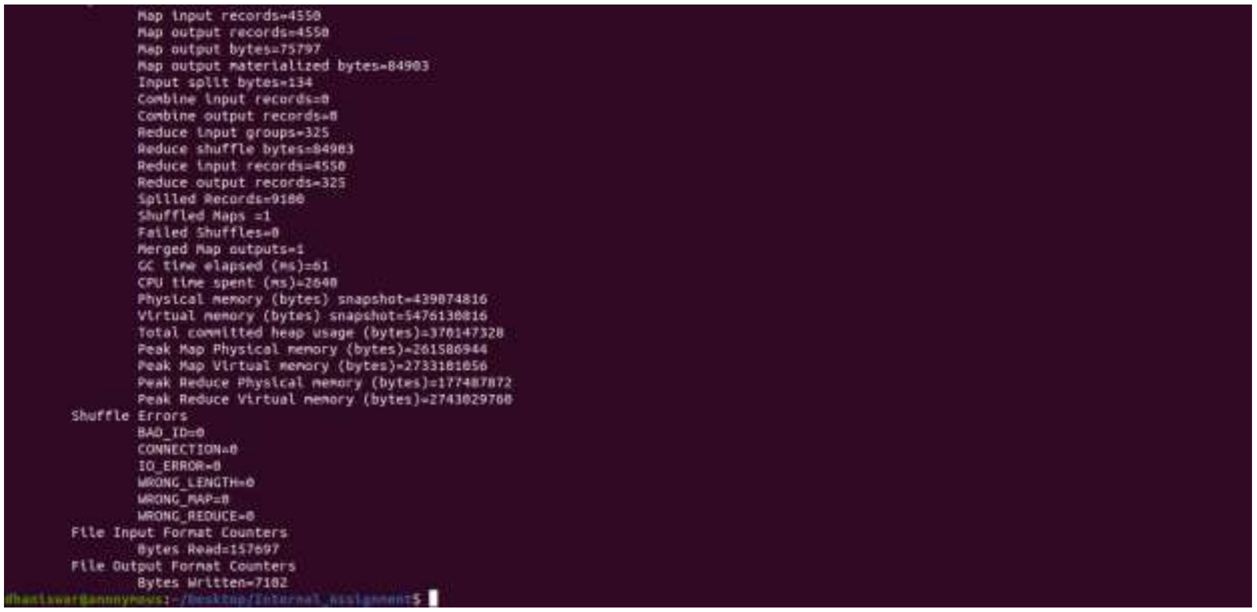


```
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
Degree-Level_Quals.csv  Population.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ mv Population.java DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
Degree-Level_Quals.csv  DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ javac -classpath $(hadoop classpath) DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
 Degree-Level_Quals.csv  'DegreeQuals$csvReducer_2048494.class'  'DegreeQuals$PopMapper_2048494.class'   DegreeQuals.class   DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ jar cf DegreeQuals.jar DegreeQ*.class
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
 Degree-Level_Quals.csv                   'DegreeQuals$PopMapper_2048494.class'   DegreeQuals.jar
'DegreeQuals$csvReducer_2048494.class'   DegreeQuals.class                       DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$
```

3.4. Executing the java programme

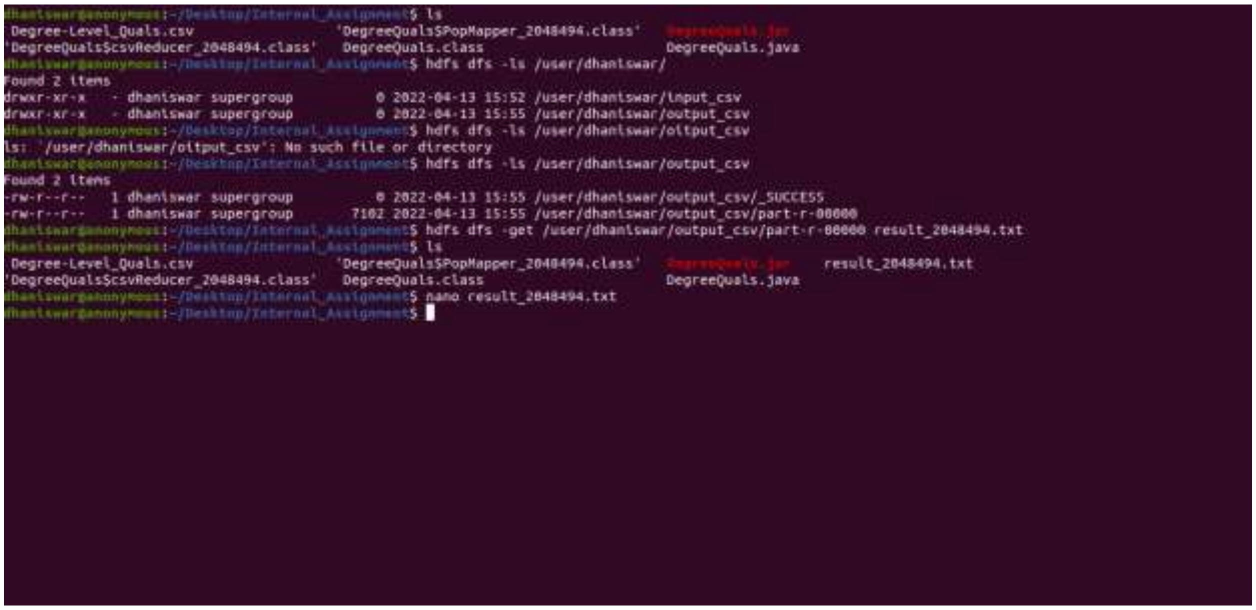


```
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ hadoop jar DegreeQuals.jar DegreeQuals input_csv/Degree-Level_Quals.csv output_csv
2022-04-13 15:55:02,707 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-13 15:55:03,102 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute
 your application with ToolRunner to remedy this.
2022-04-13 15:55:03,124 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/dhaniswar/.staging/job_1649843
133371_0002
2022-04-13 15:55:03,420 INFO input.FileInputFormat: Total input files to process : 1
2022-04-13 15:55:03,520 INFO mapreduce.JobSubmitter: number of splits:1
2022-04-13 15:55:03,804 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1649843133371_0002
2022-04-13 15:55:03,806 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-13 15:55:04,024 INFO conf.Configuration: resource-types.xml not found
2022-04-13 15:55:04,024 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-13 15:55:04,490 INFO impl.YarnClientImpl: Submitted application application_1649843133371_0002
2022-04-13 15:55:04,536 INFO mapreduce.Job: The url to track the job: http://anonymous:8088/proxy/application_1649843133371_0002/
2022-04-13 15:55:04,537 INFO mapreduce.Job: Running job: job_1649843133371_0002
2022-04-13 15:55:13,695 INFO mapreduce.Job: Job job_1649843133371_0002 running in uber mode : false
2022-04-13 15:55:13,698 INFO mapreduce.Job:  map 0% reduce 0%
2022-04-13 15:55:18,773 INFO mapreduce.Job:  map 100% reduce 0%
2022-04-13 15:55:24,818 INFO mapreduce.Job:  map 100% reduce 100%
2022-04-13 15:55:24,829 INFO mapreduce.Job: Job job_1649843133371_0002 completed successfully
2022-04-13 15:55:24,955 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=84903
                FILE: Number of bytes written=638575
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=157831
                HDFS: Number of bytes written=7182
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
```



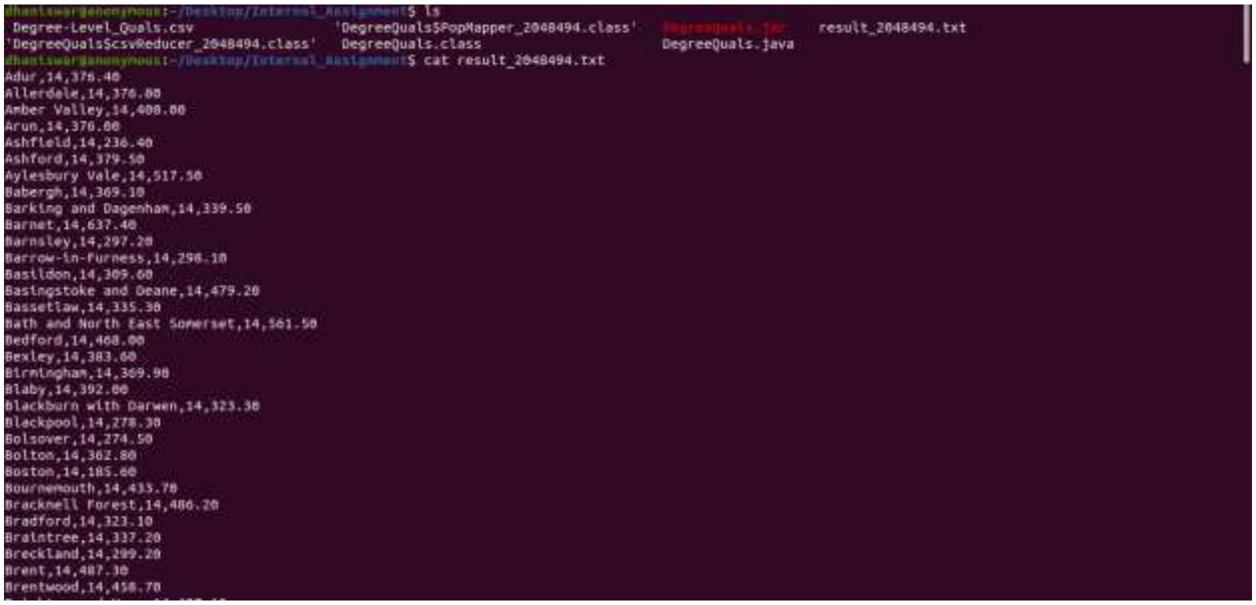
```
                Map input records=4550
                Map output records=4550
                Map output bytes=75797
                Map output materialized bytes=84903
                Input split bytes=134
                Combine input records=0
                Combine output records=0
                Reduce input groups=325
                Reduce shuffle bytes=84903
                Reduce input records=4550
                Reduce output records=325
                Spilled Records=9100
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=61
                CPU time spent (ms)=2640
                Physical memory (bytes) snapshot=439074816
                Virtual memory (bytes) snapshot=5476130816
                Total committed heap usage (bytes)=370147328
                Peak Map Physical memory (bytes)=261586944
                Peak Map Virtual memory (bytes)=2733101056
                Peak Reduce Physical memory (bytes)=177487872
                Peak Reduce Virtual memory (bytes)=2743029760
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=157697
        File Output Format Counters
                Bytes Written=7182
dhaniswar@anonymous:~/Desktop/Internal_Assignment$
```

3.5. Navigating the input and output directories of Hadoop file system and found success message in output_csv directory and storing the successfully created file from Hadoop file system to local directory.



```
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
 Degree-Level_Quals.csv                     'DegreeQuals$PopMapper_2048494.class'   DegreeQuals.jar
'DegreeQuals$csvReducer_2048494.class'   DegreeQuals.class                       DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ hdfs dfs -ls /user/dhaniswar/
Found 2 items
drwxr-xr-x   - dhaniswar supergroup          0 2022-04-13 15:52 /user/dhaniswar/input_csv
drwxr-xr-x   - dhaniswar supergroup          0 2022-04-13 15:55 /user/dhaniswar/output_csv
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ hdfs dfs -ls /user/dhaniswar/oitput_csv
ls: `/user/dhaniswar/oitput_csv': No such file or directory
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ hdfs dfs -ls /user/dhaniswar/output_csv
Found 2 items
-rw-r--r--   1 dhaniswar supergroup          0 2022-04-13 15:55 /user/dhaniswar/output_csv/_SUCCESS
-rw-r--r--   1 dhaniswar supergroup       7102 2022-04-13 15:55 /user/dhaniswar/output_csv/part-r-00000
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ hdfs dfs -get /user/dhaniswar/output_csv/part-r-00000 result_2048494.txt
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
 Degree-Level_Quals.csv                     'DegreeQuals$PopMapper_2048494.class'   DegreeQuals.jar    result_2048494.txt
'DegreeQuals$csvReducer_2048494.class'   DegreeQuals.class                       DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ nano result_2048494.txt
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ █
```

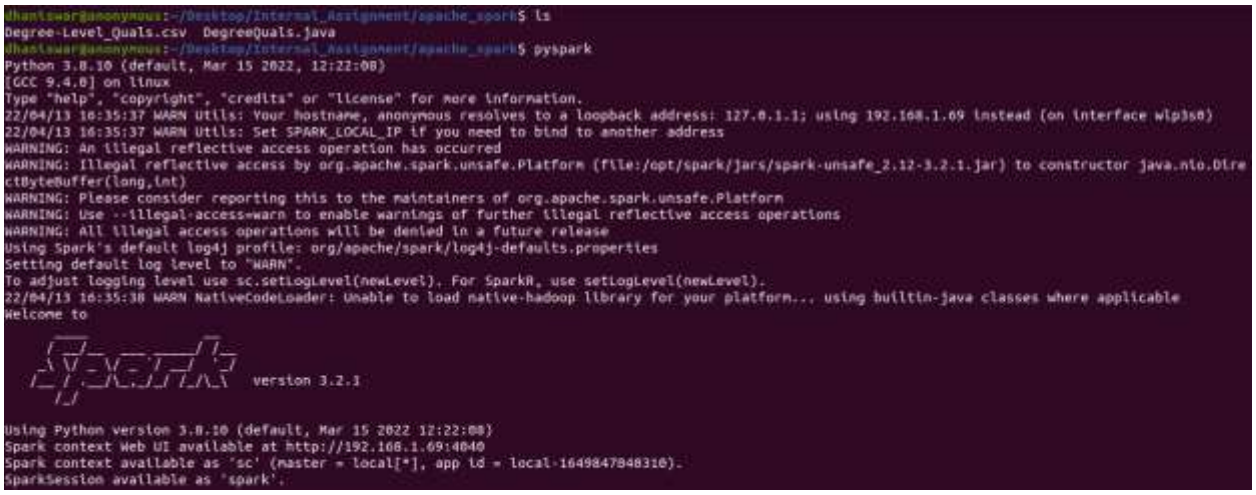
3.6. Content of the output directory



```
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ ls
 Degree-Level_Quals.csv                     'DegreeQuals$PopMapper_2048494.class'   DegreeQuals.jar    result_2048494.txt
'DegreeQuals$csvReducer_2048494.class'   DegreeQuals.class                       DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment$ cat result_2048494.txt
Adur,14,376.40
Allerdale,14,376.00
Amber Valley,14,400.00
Arun,14,376.00
Ashfield,14,236.40
Ashford,14,379.50
Aylesbury Vale,14,517.50
Babergh,14,369.10
Barking and Dagenham,14,339.50
Barnet,14,637.40
Barnsley,14,297.20
Barrow-in-Furness,14,298.10
Basildon,14,309.60
Basingstoke and Deane,14,479.20
Bassetlaw,14,335.30
Bath and North East Somerset,14,561.50
Bedford,14,460.00
Bexley,14,383.60
Birmingham,14,369.90
Blaby,14,392.00
Blackburn with Darwen,14,323.30
Blackpool,14,278.30
Bolsover,14,274.50
Bolton,14,362.80
Boston,14,185.60
Bournemouth,14,435.70
Bracknell Forest,14,486.20
Bradford,14,323.10
Braintree,14,337.20
Breckland,14,299.20
Brent,14,487.30
Brentwood,14,458.70
```

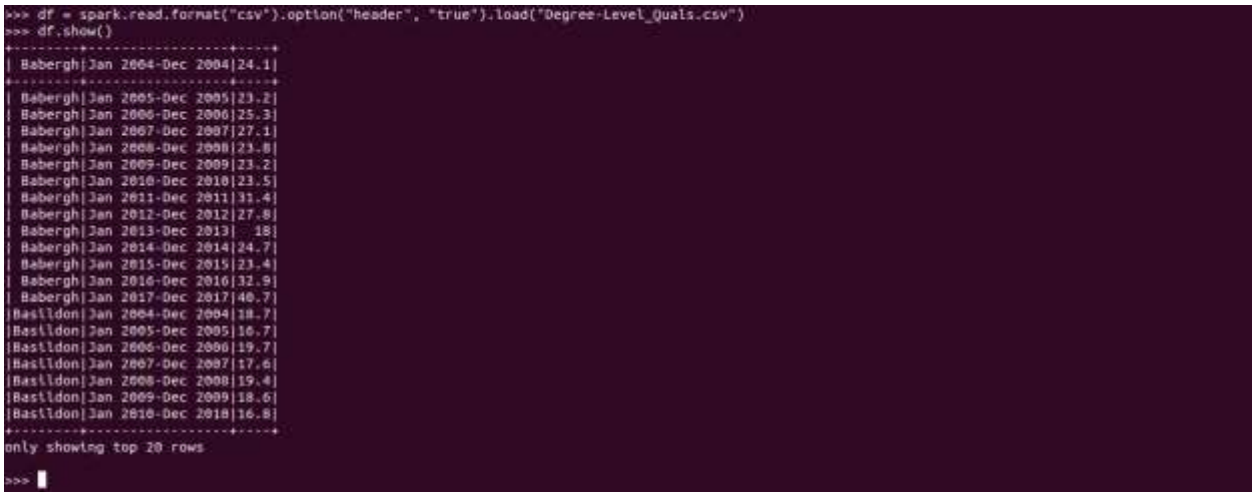
4. Apache Spark

4.1. Starting the apace spark using command pyspark



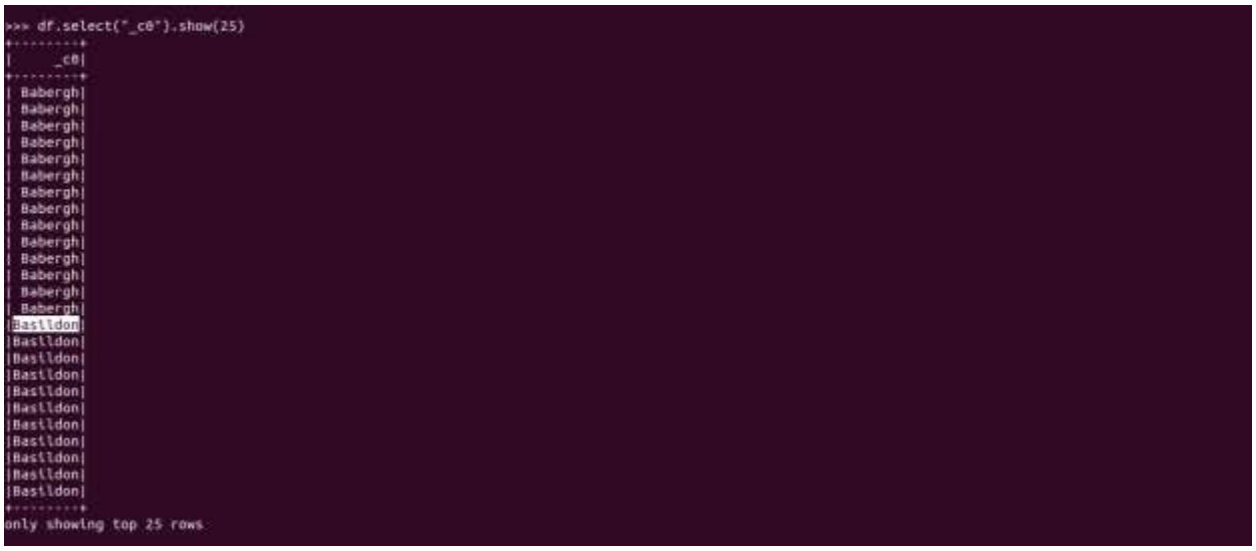
```
dhaniswar@anonymous:~/Desktop/Internal_Assignment/apache_spark$ ls
Degree-Level_Quals.csv  DegreeQuals.java
dhaniswar@anonymous:~/Desktop/Internal_Assignment/apache_spark$ pyspark
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
22/04/13 16:35:37 WARN Utils: Your hostname, anonymous resolves to a loopback address: 127.0.1.1; using 192.168.1.69 instead (on interface wlp3s0)
22/04/13 16:35:37 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.2.1.jar) to constructor java.nio.Dire
ctByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/13 16:35:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Python version 3.8.10 (default, Mar 15 2022 12:22:08)
Spark context Web UI available at http://192.168.1.69:4040
Spark context available as 'sc' (master = local[*], app id = local-1649847848310).
SparkSession available as 'spark'.
```

4.2. Loading the same CSV file into Apache Spark



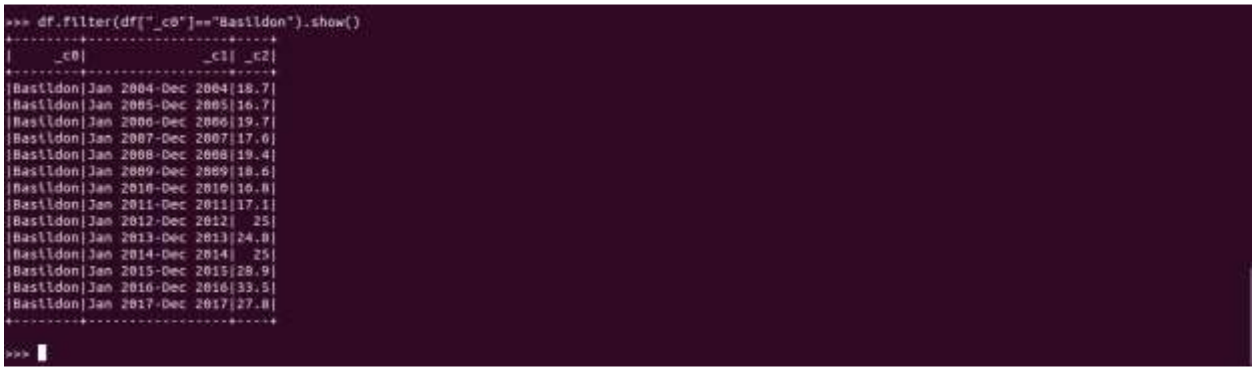
```
>>> df = spark.read.format("csv").option("header", "true").load("Degree-Level_Quals.csv")
>>> df.show()
+--------+-------------------+----+
| Babergh|Jan 2004-Dec 2004|24.1|
+--------+-------------------+----+
| Babergh|Jan 2005-Dec 2005|23.2|
| Babergh|Jan 2006-Dec 2006|25.3|
| Babergh|Jan 2007-Dec 2007|27.1|
| Babergh|Jan 2008-Dec 2008|23.8|
| Babergh|Jan 2009-Dec 2009|23.2|
| Babergh|Jan 2010-Dec 2010|23.5|
| Babergh|Jan 2011-Dec 2011|31.4|
| Babergh|Jan 2012-Dec 2012|27.8|
| Babergh|Jan 2013-Dec 2013|  18|
| Babergh|Jan 2014-Dec 2014|24.7|
| Babergh|Jan 2015-Dec 2015|23.4|
| Babergh|Jan 2016-Dec 2016|32.9|
| Babergh|Jan 2017-Dec 2017|40.7|
|Basildon|Jan 2004-Dec 2004|18.7|
|Basildon|Jan 2005-Dec 2005|16.7|
|Basildon|Jan 2006-Dec 2006|19.7|
|Basildon|Jan 2007-Dec 2007|17.6|
|Basildon|Jan 2008-Dec 2008|19.4|
|Basildon|Jan 2009-Dec 2009|18.6|
|Basildon|Jan 2010-Dec 2010|16.8|
+--------+-------------------+----+
only showing top 20 rows

>>> 
```

4.3. Selecting the first columns of the dataset with 25 rows using data frame



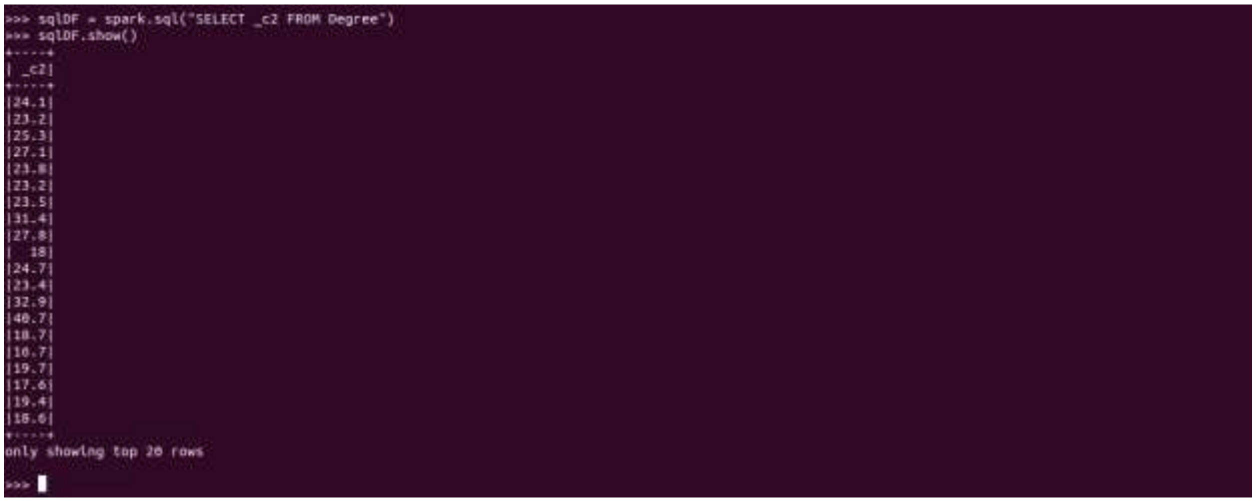
```
>>> df.select("_c0").show(25)
+--------+
|     _c0|
+--------+
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
| Babergh|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
|Basildon|
+--------+
only showing top 25 rows
```

4.4. Filtering specific records using data frame



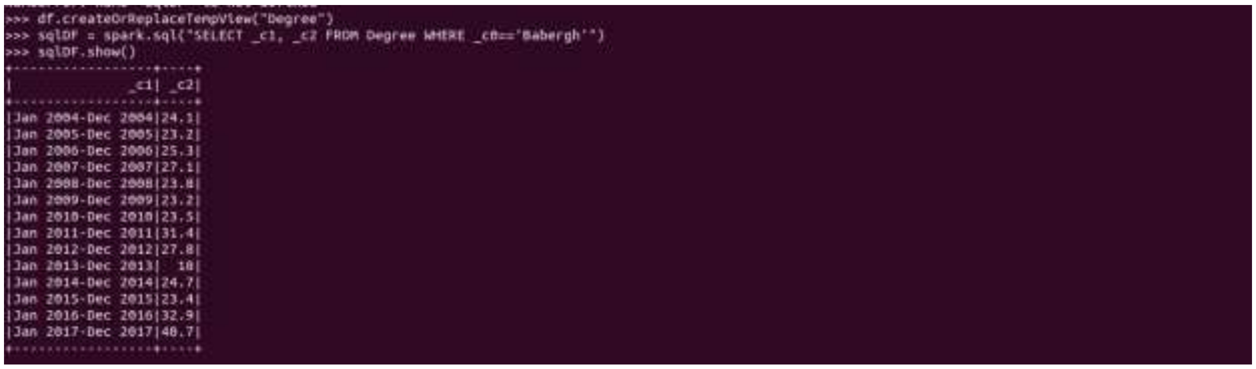
```
>>> df.filter(df["_c0"]=="Basildon").show()
+--------+-----------------+----+
|     _c0|              _c1| _c2|
+--------+-----------------+----+
|Basildon|Jan 2004-Dec 2004|18.7|
|Basildon|Jan 2005-Dec 2005|16.7|
|Basildon|Jan 2006-Dec 2006|19.7|
|Basildon|Jan 2007-Dec 2007|17.6|
|Basildon|Jan 2008-Dec 2008|19.4|
|Basildon|Jan 2009-Dec 2009|18.6|
|Basildon|Jan 2010-Dec 2010|16.8|
|Basildon|Jan 2011-Dec 2011|17.1|
|Basildon|Jan 2012-Dec 2012|  25|
|Basildon|Jan 2013-Dec 2013|24.8|
|Basildon|Jan 2014-Dec 2014|  25|
|Basildon|Jan 2015-Dec 2015|28.9|
|Basildon|Jan 2016-Dec 2016|33.5|
|Basildon|Jan 2017-Dec 2017|27.8|
+--------+-----------------+----+

>>> 
```

4.5. Selecting one columns using SQL query in spark



```
>>> sqlDF = spark.sql("SELECT _c2 FROM Degree")
>>> sqlDF.show()
+----+
| _c2|
+----+
|24.1|
|23.2|
|25.3|
|27.1|
|23.8|
|23.2|
|23.5|
|31.4|
|27.8|
|  18|
|24.7|
|23.4|
|32.9|
|40.7|
|18.7|
|16.7|
|19.7|
|17.6|
|19.4|
|18.6|
+----+
only showing top 20 rows

>>> 
```

4.6. Selecting two columns on the basis of first column value equals to Babergh using SQL query in spark



```
>>> df.createOrReplaceTempView("Degree")
>>> sqlDF = spark.sql("SELECT _c1, _c2 FROM Degree WHERE _c0=='Babergh'")
>>> sqlDF.show()
+-----------------+----+
|              _c1| _c2|
+-----------------+----+
|Jan 2004-Dec 2004|24.1|
|Jan 2005-Dec 2005|23.2|
|Jan 2006-Dec 2006|25.3|
|Jan 2007-Dec 2007|27.1|
|Jan 2008-Dec 2008|23.8|
|Jan 2009-Dec 2009|23.2|
|Jan 2010-Dec 2010|23.5|
|Jan 2011-Dec 2011|31.4|
|Jan 2012-Dec 2012|27.8|
|Jan 2013-Dec 2013|  18|
|Jan 2014-Dec 2014|24.7|
|Jan 2015-Dec 2015|23.4|
|Jan 2016-Dec 2016|32.9|
|Jan 2017-Dec 2017|40.7|
+-----------------+----+
```

5. Hadoop is fast (advantage)

Data processing tools are frequently housed on the same servers as the data, resulting in the substantially quicker data processing. Hadoop can easily handle terabytes of data in minutes or petabytes in hours if you're working with enormous amounts of unstructured data. Hadoop's one-of-a-kind storage system is built on a distributed file system that essentially "maps" data to any location on a cluster. Data processing tools are frequently housed on the same servers as the data, resulting in the substantially quicker data processing. Hadoop can easily handle terabytes of data in minutes or petabytes in hours if you're working with enormous amounts of unstructured data.

6. Issue with Small Files (disadvantage)

Hadoop is not suitable for handling modest amounts of data. (HDFS) Because of its high-capacity architecture, the Hadoop distributed file system is unable to efficiently support the random reading of tiny files. In HDFS, small files are the most common issue. The size of a tiny file is much smaller than the HDFS block size (default 128MB). HDFS can't manage this many little files since it's designed to deal with a limited number of large files for storing massive data sets rather than a large number of small files. Because it holds the HDFS namespace, the Name Node will get overloaded if there are too many tiny files.