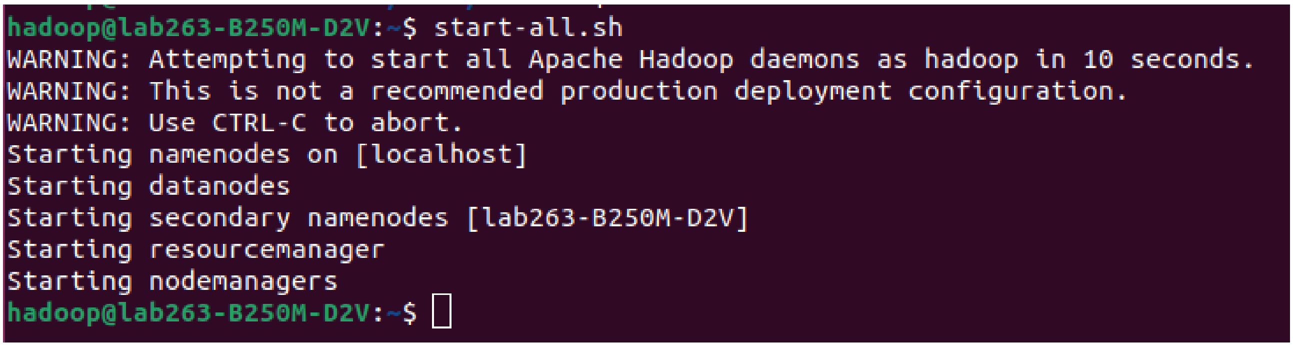
Practical No - 1

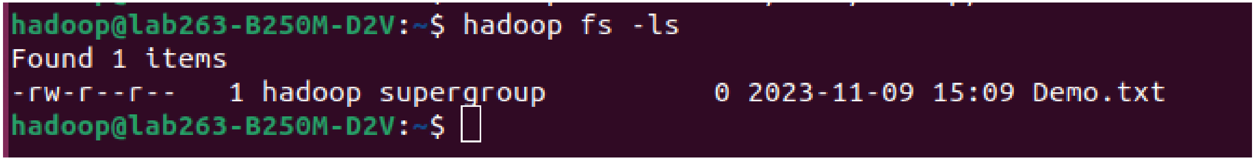
Aim:List of Commands(mkdir, touchz, copy from local/put,copy to local/get move from local,cp,rmr,du,dus,stat)

start-all.sh



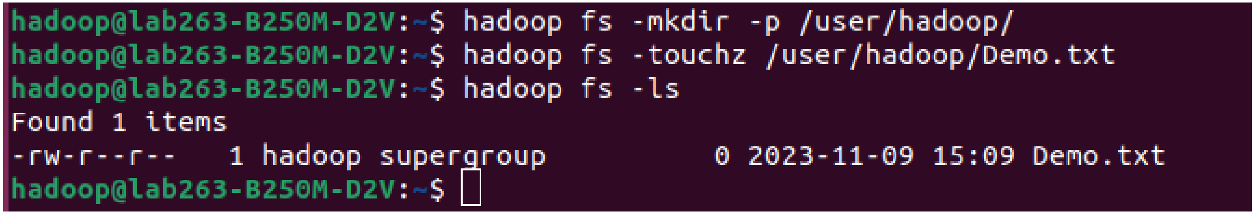
```
hadoop@lab263-B250M-D2V:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [lab263-B250M-D2V]
Starting resourcemanager
Starting nodemanagers
hadoop@lab263-B250M-D2V:~$ 
```

1)hadoop fs
hadoop fs -ls



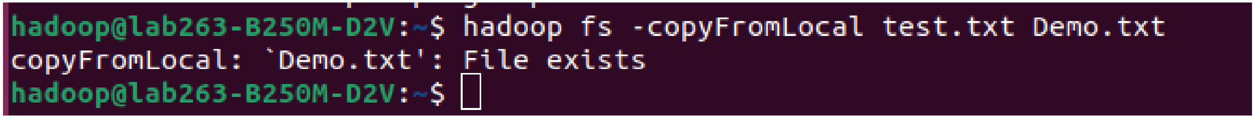
```
hadoop@lab263-B250M-D2V:~$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2023-11-09 15:09 Demo.txt
hadoop@lab263-B250M-D2V:~$ 
```

2)touchz: It creates an empty file.
hadoop fs -mkdir -p /user/hadoop/
hadoop fs -touchz /user/hadoop/Demo.txt



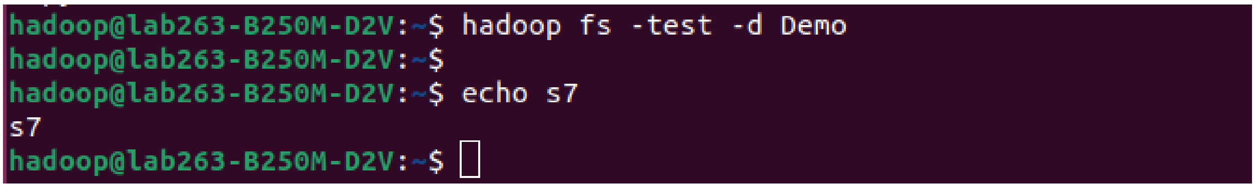
```
hadoop@lab263-B250M-D2V:~$ hadoop fs -mkdir -p /user/hadoop/
hadoop@lab263-B250M-D2V:~$ hadoop fs -touchz /user/hadoop/Demo.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop supergroup          0 2023-11-09 15:09 Demo.txt
hadoop@lab263-B250M-D2V:~$ 
```

3)copyFromLocal (or) put:
hadoop fs -copyFromLocal test.txt Demo.txt

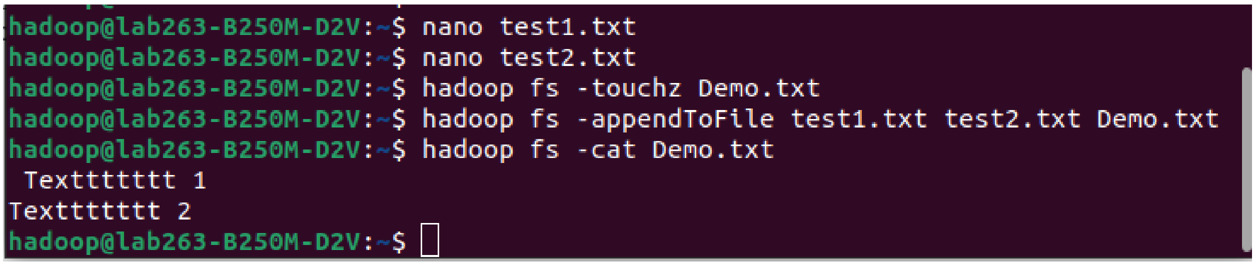


```
hadoop@lab263-B250M-D2V:~$ hadoop fs -copyFromLocal test.txt Demo.txt
copyFromLocal: `Demo.txt': File exists
hadoop@lab263-B250M-D2V:~$ 
```

4) test
hadoop fs -test -d Demo
echo s7



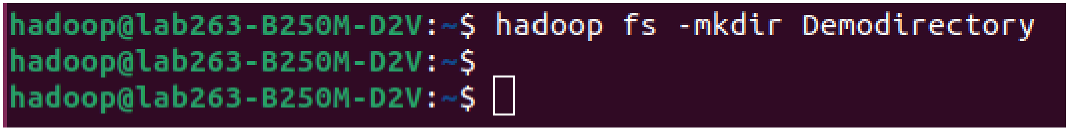
```
hadoop@lab263-B250M-D2V:~$ hadoop fs -test -d Demo
hadoop@lab263-B250M-D2V:~$
hadoop@lab263-B250M-D2V:~$ echo s7
s7
hadoop@lab263-B250M-D2V:~$ 
```



```
hadoop@lab263-B250M-D2V:~$ nano test1.txt
hadoop@lab263-B250M-D2V:~$ nano test2.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -touchz Demo.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -appendToFile test1.txt test2.txt Demo.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -cat Demo.txt
 Texttttttt 1
Texttttttt 2
hadoop@lab263-B250M-D2V:~$ 
```

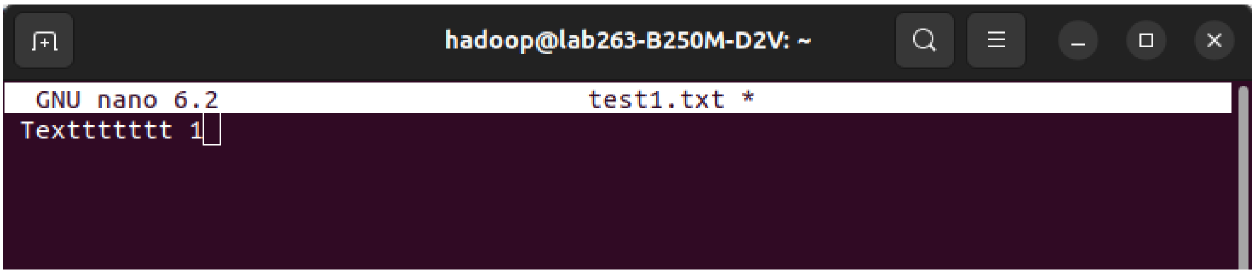
5)mkdir

hadoop fs -mkdir Demodirectory



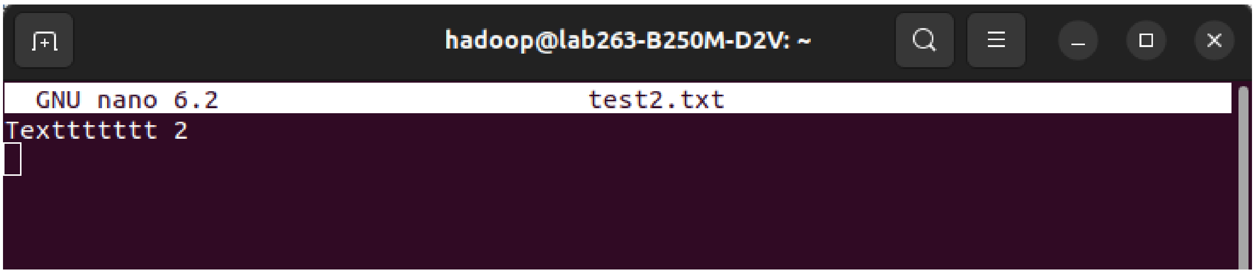
```
hadoop@lab263-B250M-D2V:~$ hadoop fs -mkdir Demodirectory
hadoop@lab263-B250M-D2V:~$
hadoop@lab263-B250M-D2V:~$ 
```

nano test1.txt



```
hadoop@lab263-B250M-D2V: ~

  GNU nano 6.2                      test1.txt *
 Texttttttt 1
```

nano test2.txt



```
hadoop@lab263-B250M-D2V: ~

  GNU nano 6.2                      test2.txt
Texttttttt 2
```

6)appendToFile

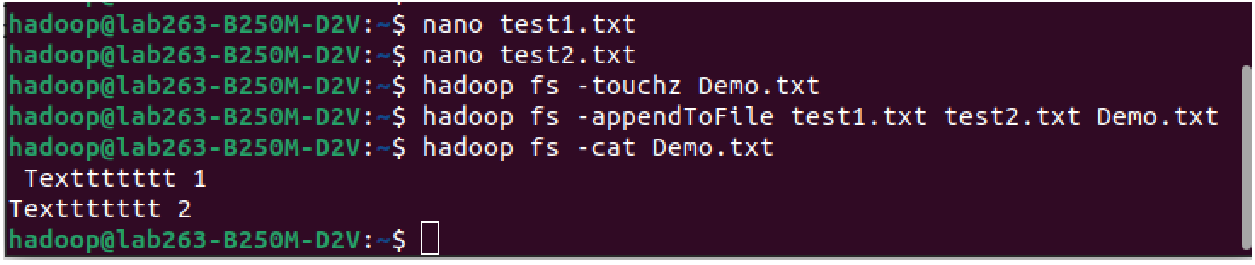
nano test1.txt

nano test2.txt

hadoop fs -touchz Demo.txt

hadoop fs -appendToFile test1.txt test2.txt Demo.txt

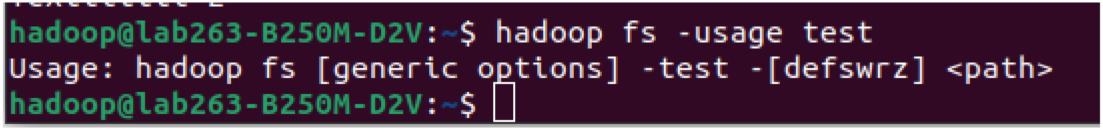
hadoop fs -cat Demo.txt



```
hadoop@lab263-B250M-D2V:~$ nano test1.txt
hadoop@lab263-B250M-D2V:~$ nano test2.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -touchz Demo.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -appendToFile test1.txt test2.txt Demo.txt
hadoop@lab263-B250M-D2V:~$ hadoop fs -cat Demo.txt
 Texttttttt 1
Texttttttt 2
hadoop@lab263-B250M-D2V:~$ 
```

7)usage

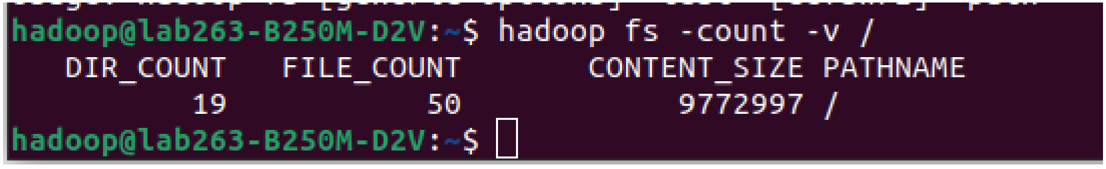
hadoop fs -usage test



```
hadoop@lab263-B250M-D2V:~$ hadoop fs -usage test
Usage: hadoop fs [generic options] -test -[defswrz] <path>
hadoop@lab263-B250M-D2V:~$ 
```

8)Count

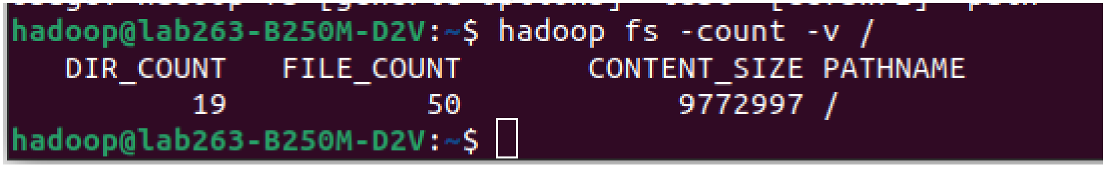
hadoop fs -count -v /



```
hadoop@lab263-B250M-D2V:~$ hadoop fs -count -v /
   DIR_COUNT   FILE_COUNT       CONTENT_SIZE PATHNAME
          19           50            9772997 /
hadoop@lab263-B250M-D2V:~$ 
```

9)find

hadoop fs -find / -name Demodirectory

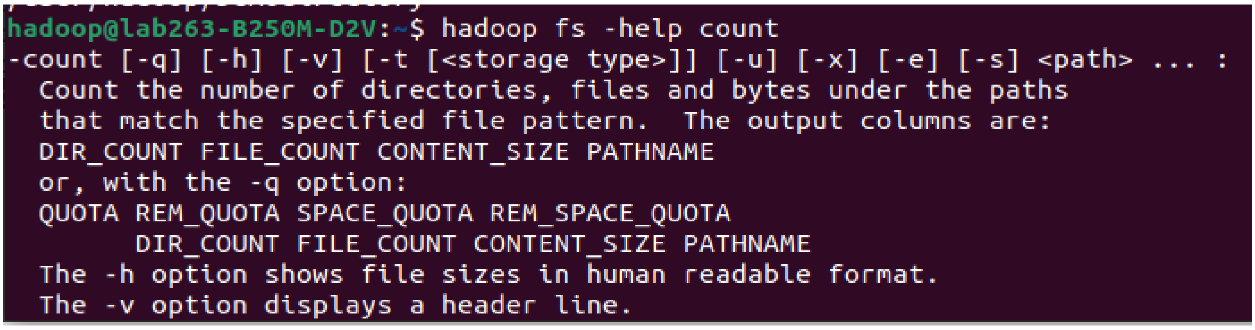


```
hadoop@lab263-B250M-D2V:~$ hadoop fs -count -v /
   DIR_COUNT   FILE_COUNT       CONTENT_SIZE PATHNAME
          19           50            9772997 /
hadoop@lab263-B250M-D2V:~$ 
```

10)help

hadoop fs -help

count



```
hadoop@lab263-B250M-D2V:~$ hadoop fs -help count
-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] [-s] <path> ... :
  Count the number of directories, files and bytes under the paths
  that match the specified file pattern.  The output columns are:
  DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  or, with the -q option:
  QUOTA REM_QUOTA SPACE_QUOTA REM_SPACE_QUOTA
        DIR_COUNT FILE_COUNT CONTENT_SIZE PATHNAME
  The -h option shows file sizes in human readable format.
  The -v option displays a header line.
```

Practical No - 2

Aim: Write a Program in Map Reduce for WordCount operation.

WordCountDriver.java

```
import org.apache.hadoop.fs.Path; import
org.apache.hadoop.io.Text; import
org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.mapreduce.Job;

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

import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.conf.Configuration;
public class WordCountDriver {

        public static void main(String[] args) throws Exception{Job
                j1=Job.getInstance(new Configuration());
                j1.setJarByClass(WordCountDriver.class);
                j1.setJobName("Average Word Count");


                FileInputFormat.addInputPath(j1,new Path(args[0]));
                FileOutputFormat.setOutputPath(j1, new Path(args[1]));


                j1.setMapperClass(WordCountMapper.class);
                j1.setReducerClass(WordCountReducer.class);
                j1.setOutputKeyClass(Text.class); j1.setOutputValueClass(IntWritable.class);
                System.exit(j1.waitForCompletion(true)? 0:1);

        }
}
```

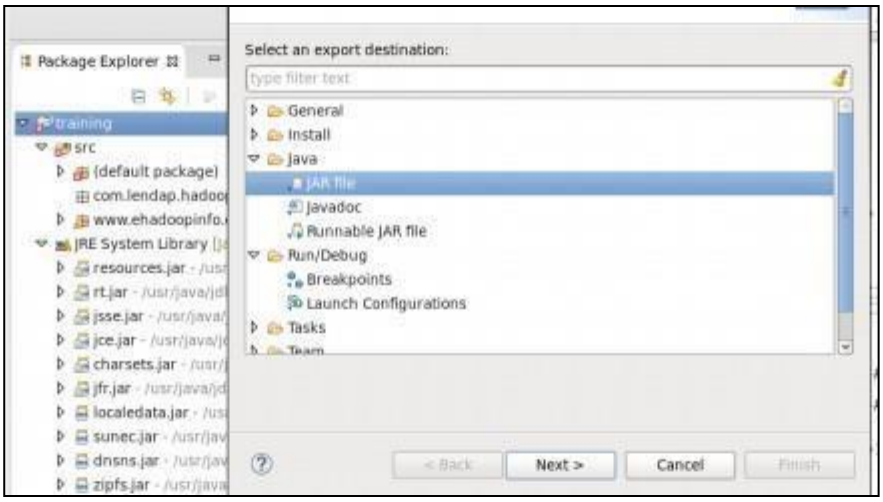
WordCountMapper.java

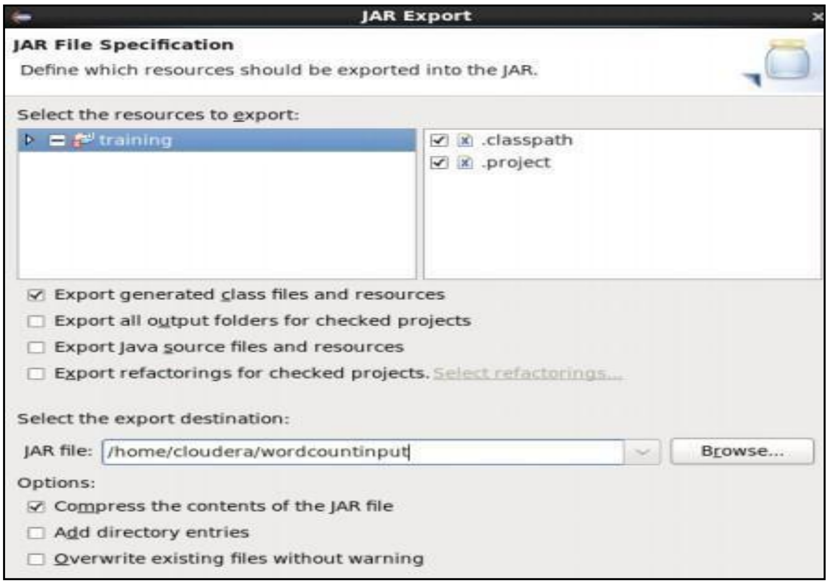
```
import java.io.IOException; import
java.util.StringTokenizer;0import
org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.LongWritable; import
org.apache.hadoop.mapreduce.Mapper;import
org.apache.hadoop.mapreduce.Reducer;

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


public class WordCountMapper extends
Mapper<LongWritable,Text,Text,IntWritable> {


    private final static IntWritable one=new IntWritable(1)


private Text word=new Text();
    public void map(LongWritable key, Text value, Context context) throws
IOException,InterruptedException {
        String line=value.toString();
        StringTokenizer ltr=new StringTokenizer(line);
        while(ltr.hasMoreTokens()){
            word.set(ltr.nextToken());
            context.write(word ,one);
        }
    }
}
```

WordCountReducer.java

```
import java.io.IOException; import
org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.mapreduce.Reducer;public class
WordCountReducer extends
Reducer<Text,IntWritable,Text,IntWritable> {


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

                int sum=0;

                for(IntWritable value:values)

                {

                        sum+=value.get();

                }

                context.write(key,  new IntWritable(sum));

        }

}
```

Export .jar file. Right click on training and select export.





JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

training

☑ .classpath

☑ .project

☑ Export generated class files and resources

□ Export all output folders for checked projects

□ Export Java source files and resources

□ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: /home/cloudera/wordcountinput

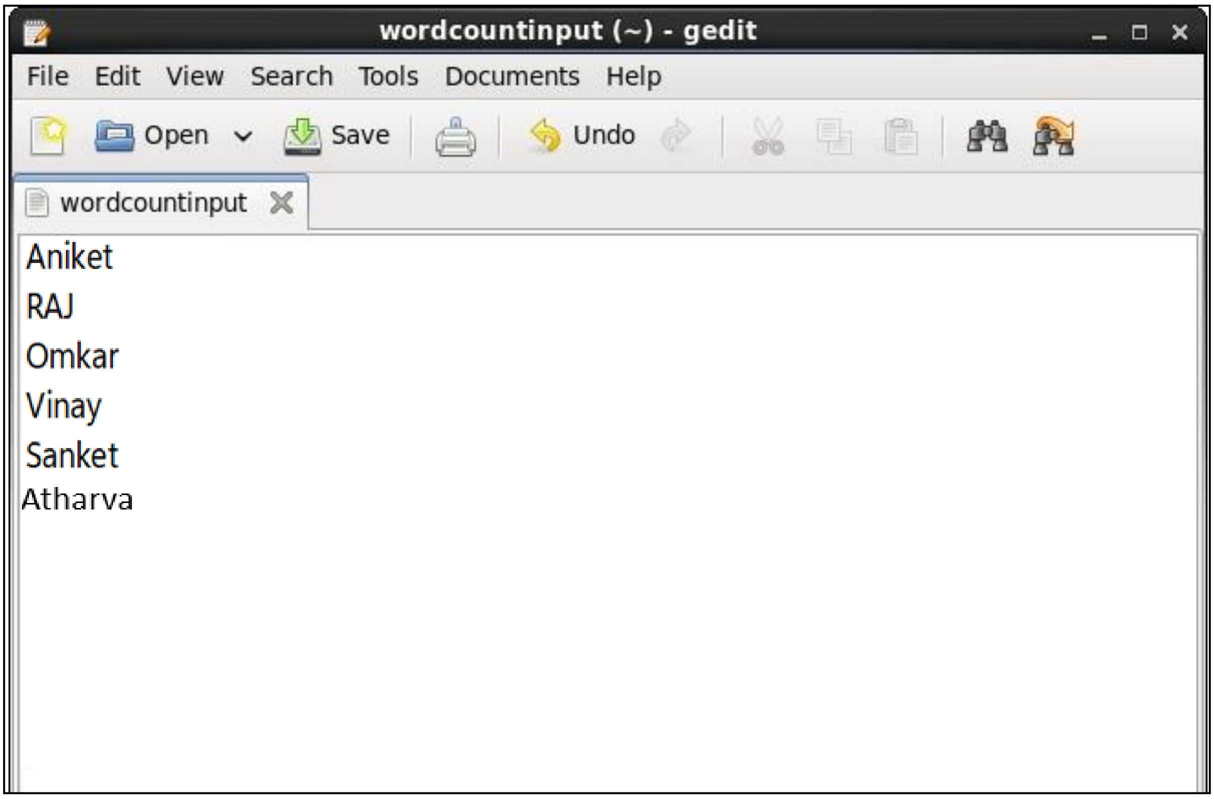
Options:

☑ Compress the contents of the JAR file

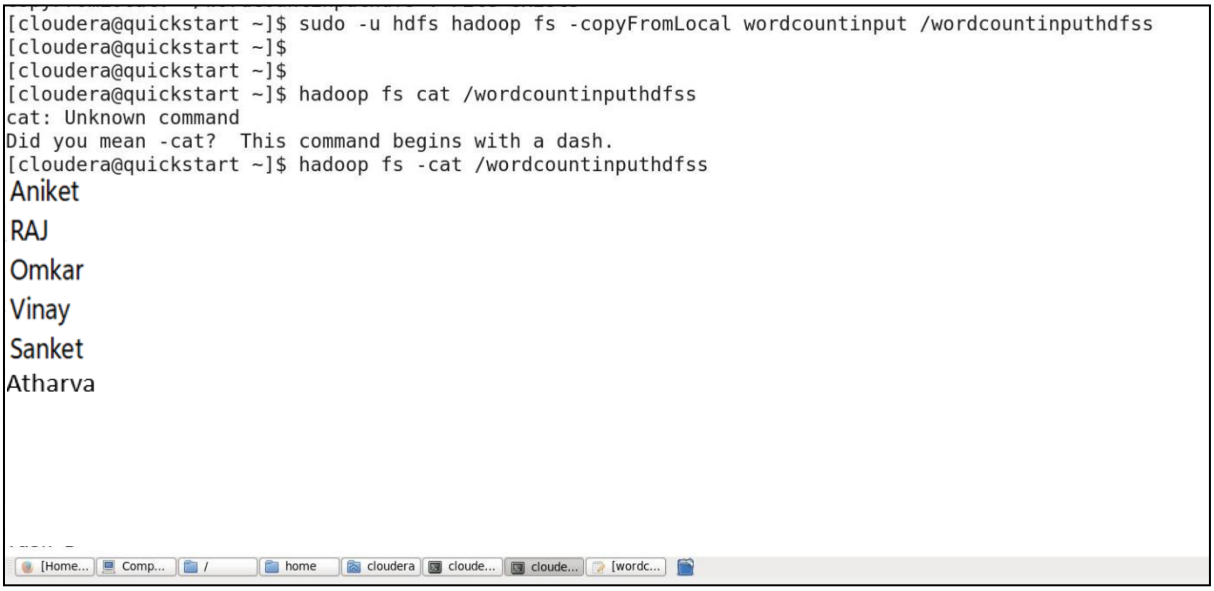
□ Add directory entries

□ Overwrite existing files without warning

Follow below HDFS commands

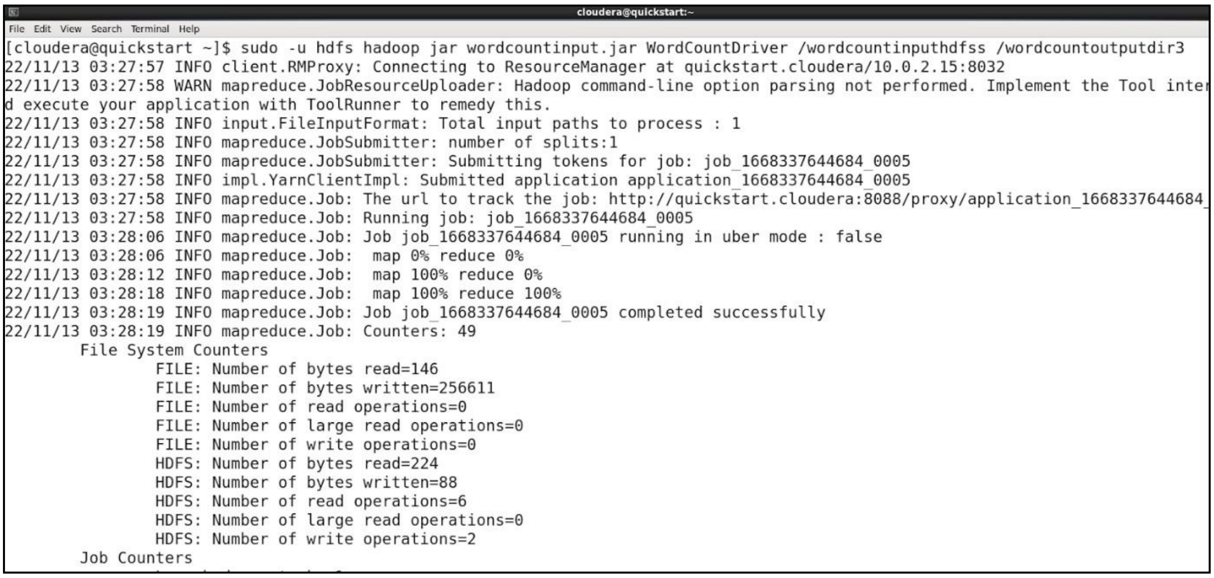


```
Aniket
RAJ
Omkar
Vinay
Sanket
Atharva
```



```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -copyFromLocal wordcountinput /wordcountinputhdfss
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ hadoop fs cat /wordcountinputhdfss
cat: Unknown command
Did you mean -cat?  This command begins with a dash.
[cloudera@quickstart ~]$ hadoop fs -cat /wordcountinputhdfss
Aniket
RAJ
Omkar
Vinay
Sanket
Atharva
```

Executing the jar file using hadoop command:



```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop jar wordcountinput.jar WordCountDriver /wordcountinputhdfss /wordcountoutputdir3
22/11/13 03:27:57 INFO client.RMProxy: Connecting to ResourceManager at quickstart.cloudera/10.0.2.15:8032
22/11/13 03:27:58 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
d execute your application with ToolRunner to remedy this.
22/11/13 03:27:58 INFO input.FileInputFormat: Total input paths to process : 1
22/11/13 03:27:58 INFO mapreduce.JobSubmitter: number of splits:1
22/11/13 03:27:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668337644684_0005
22/11/13 03:27:58 INFO impl.YarnClientImpl: Submitted application application_1668337644684_0005
22/11/13 03:27:58 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1668337644684_
22/11/13 03:27:58 INFO mapreduce.Job: Running job: job_1668337644684_0005
22/11/13 03:28:06 INFO mapreduce.Job: Job job_1668337644684_0005 running in uber mode : false
22/11/13 03:28:06 INFO mapreduce.Job:  map 0% reduce 0%
22/11/13 03:28:12 INFO mapreduce.Job:  map 100% reduce 0%
22/11/13 03:28:18 INFO mapreduce.Job:  map 100% reduce 100%
22/11/13 03:28:19 INFO mapreduce.Job: Job job_1668337644684_0005 completed successfully
22/11/13 03:28:19 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=146
                FILE: Number of bytes written=256611
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=224
                HDFS: Number of bytes written=88
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
```

Aim: write a Program in Map Reduce for Matrix Multiplication.

MatrixMultiplication.java

```
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import java.util.ArrayList;


import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.Writable;

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


import  org.apache.hadoop.mapreduce.Job; import
org.apache.hadoop.mapreduce.Mapper; import
org.apache.hadoop.mapreduce.Reducer; import
org.apache.hadoop.mapreduce.lib.input.*;

import org.apache.hadoop.mapreduce.lib.output.*;
import org.apache.hadoop.util.ReflectionUtils;


class Element implements Writable {int
        tag;

        int index;
        double value;
        Element() {

                tag = 0;

                index = 0;

                value = 0.0;

        }

        Element(int tag, int index, double value) {
                this.tag = tag;

                this.index = index;
                this.value = value;

        }
```

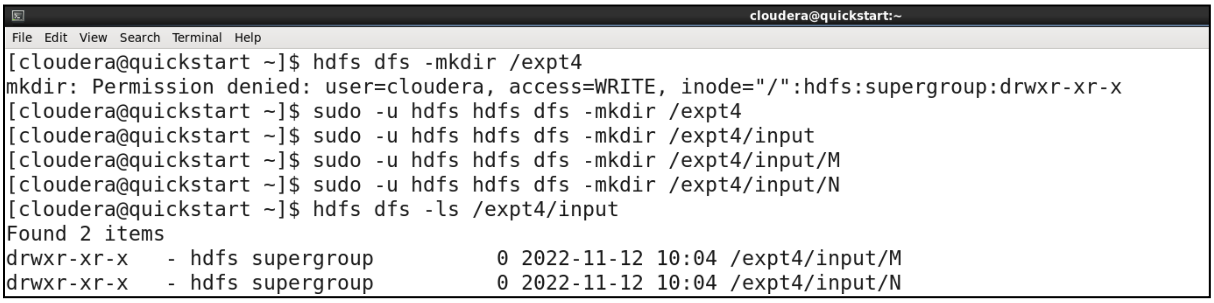
```
          @Override
          public void readFields(DataInput input) throws IOException {tag =
                input.readInt();
                index = input.readInt(); value
                = input.readDouble();
          }
          @Override
          public void write(DataOutput output) throws IOException {
                output.writeInt(tag);
output.writeInt(index);
output.writeDouble(value);
          }
  }
  class Pair implements WritableComparable<Pair> {int i;
          int j;
          Pair() {
                i = 0;
                j = 0;
          }
          Pair(int i, int j) {
                this.i = i;
                this.j = j;
          }
          @Override
          public void readFields(DataInput input) throws IOException {i =
                input.readInt();
                j = input.readInt();
          }
          @Override
          public void write(DataOutput output) throws IOException {

               output.writeInt(i);
                output.writeInt(j);
          }
```

```
        @Override
        public int compareTo(Pair compare) {if (i
                > compare.i) {

                        return 1;

                }

                else if (i < compare.i) {
                        return -1;

                }

                else {

                        if (j > compare.j) {
                                return 1;

                        }

                        else if (j < compare.j) {
                                return -1;

                        }

                }

                return 0;

        }

        public String toString() { return
                i + " " + j + " ";


        }

}

public class MatrixMultiply {

        public static class MatrixMapperM extends Mapper<Object,
                        Text, IntWritable, Element> {

                @Override

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

                        String readLine = value.toString();
                        String[] tokens = readLine.split(","); int
                        index = Integer.parseInt(tokens[0]);

                        double elementVal = Double.parseDouble(tokens[2]);
                        Element e = new Element(0, index, elementVal);

                        IntWritable keyval = new IntWritable(Integer.parseInt(tokens[1]));
                        context.write(keyval, e);

                }

        }
```

```
public static class MatrixMapperN extends Mapper<Object,
        Text, IntWritable, Element> {


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

         String readLine = value.toString();
         String[] tokens = readLine.split(","); int
         index = Integer.parseInt(tokens[1]);

         double elementVal = Double.parseDouble(tokens[2]);
         Element e = new Element(1, index, elementVal);

         IntWritable keyval = new IntWritable(Integer.parseInt(tokens[0]));
         context.write(keyval, e);

     }
}


public static class ReducerMN extends

        Reducer<IntWritable, Element, Pair, DoubleWritable> {


    @Override
    public void reduce(IntWritable key, Iterable<Element> values,

             Context context) throws IOException, InterruptedException {
         ArrayList<Element> M = new ArrayList<Element>();
         ArrayList<Element> N = new ArrayList<Element>(); Configuration
         conf = context.getConfiguration();

         for (Element element : values) {

             Element temp = ReflectionUtils.newInstance(Element.class, conf);
             ReflectionUtils.copy(conf, element, temp);

             if (temp.tag == 0)


             {

                 M.add(temp);

             }

             else if (temp.tag == 1)

             {

                 N.add(temp);

             }

         }
         for (int i = 0; i < M.size(); i++) {
             for (int j = 0; j < N.size(); j++) {
                 Pair p = new Pair(M.get(i).index, N.get(j).index);
                 double mul = M.get(i).value * N.get(j).value;


                 context.write(p, new DoubleWritable(mul));
             }
         }
     }


}
public static class MapMN extends
         Mapper<Object, Text, Pair, DoubleWritable> {
     @Override
    public void map(Object key, Text value, Context context)
             throws IOException, InterruptedException {
         String readLine = value.toString(); String[]
        pairValue = readLine.split(" ");
         Pair p = new Pair(Integer.parseInt(pairValue[0]),
                 Integer.parseInt(pairValue[1]));
        DoubleWritable val = new DoubleWritable(
                 Double.parseDouble(pairValue[2]));
         context.write(p, val);
     }
}
public static class ReduceMN extends
         Reducer<Pair, DoubleWritable, Pair, DoubleWritable> {
     @Override
     public void reduce(Pair key, Iterable<DoubleWritable> values, Context
             context) throws IOException, InterruptedException {
         double sum = 0.0;
         for (DoubleWritable value : values) {
             sum += value.get();
         }
         context.write(key, new DoubleWritable(sum));
     }
}
```

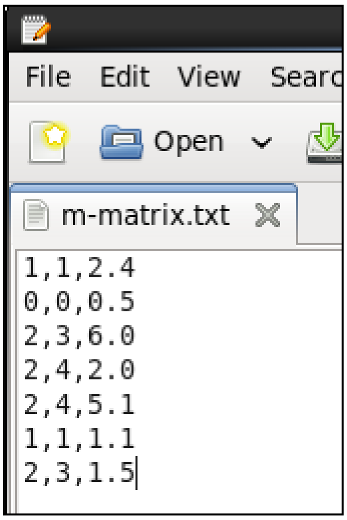
```
public static void main(String[] args) throws Exception {
        Path MPath = new Path("/expt4/input/M");Path NPath = new Path("/expt4/input/N");
              Path intermediatePath = new Path("/expt4/interim");Path
              outputPath = new Path("/expt4/output");
              Job job1 = Job.getInstance();
              job1.setJobName("Map Intermediate");
              job1.setJarByClass(MatrixMultiply.class);
              MultipleInputs.addInputPath(job1, MPath, TextInputFormat.class,
                          MatrixMapperM.class);


             MultipleInputs.addInputPath(job1, NPath, TextInputFormat.class,
                          MatrixMapperN.class);


              job1.setReducerClass(ReducerMN.class);
             job1.setMapOutputKeyClass(IntWritable.class);
             job1.setMapOutputValueClass(Element.class); job1.setOutputKeyClass(Pair.class);
             job1.setOutputValueClass(DoubleWritable.class);
             job1.setOutputFormatClass(TextOutputFormat.class);
             FileOutputFormat.setOutputPath(job1, intermediatePath); job1.waitForCompletion(true);
              Job job2 = Job.getInstance(); job2.setJobName("Map
              Final Output");
              job2.setJarByClass(MatrixMultiply.class);
              job2.setMapperClass(MapMN.class);
              job2.setReducerClass(ReduceMN.class);
              job2.setOutputKeyClass(Pair.class);
              job2.setOutputValueClass(DoubleWritable.class);
              job2.setInputFormatClass(TextInputFormat.class);
              job2.setOutputFormatClass(TextOutputFormat.class);
              FileInputFormat.addInputPath(job2, intermediatePath);
              FileOutputFormat.setOutputPath(job2, outputPath);
              job2.waitForCompletion(true);
        }


 }
```

Prerequisites create the input directories to store the input matrices M and N

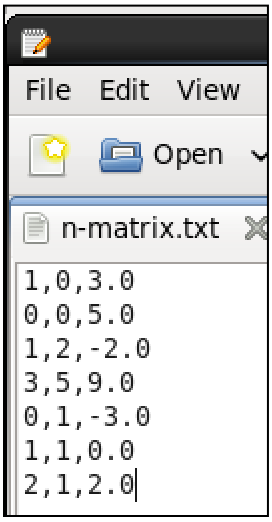


```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /expt4
mkdir: Permission denied: user=cloudera, access=WRITE, inode="/":hdfs:supergroup:drwxr-xr-x
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /expt4
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /expt4/input
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /expt4/input/M
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /expt4/input/N
[cloudera@quickstart ~]$ hdfs dfs -ls /expt4/input
Found 2 items
drwxr-xr-x   - hdfs supergroup          0 2022-11-12 10:04 /expt4/input/M
drwxr-xr-x   - hdfs supergroup          0 2022-11-12 10:04 /expt4/input/N
```

The figure below shows the matrix data used for this implementation:

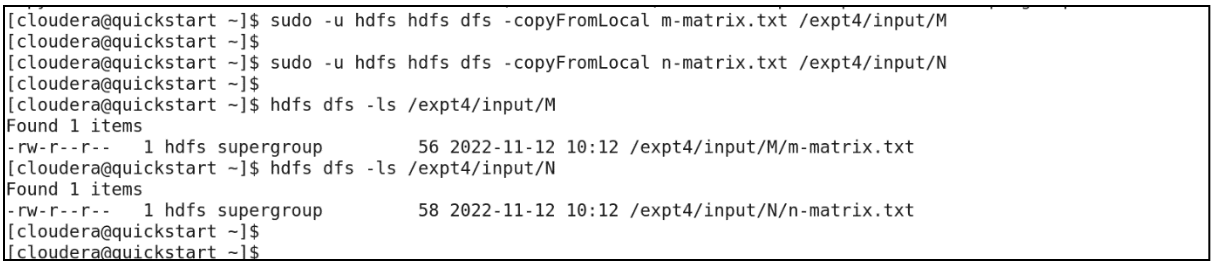


```
1,1,2.4
0,0,0.5
2,3,6.0
2,4,2.0
2,4,5.1
1,1,1.1
2,3,1.5
```



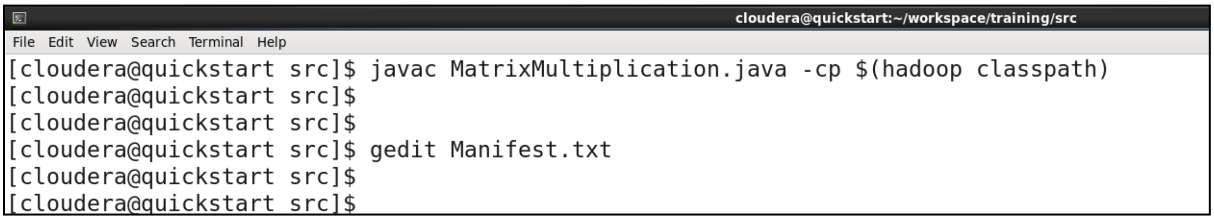
```
1,0,3.0
0,0,5.0
1,2,-2.0
3,5,9.0
0,1,-3.0
1,1,0.0
2,1,2.0
```

Copy the matrix data from the local system to HDFS



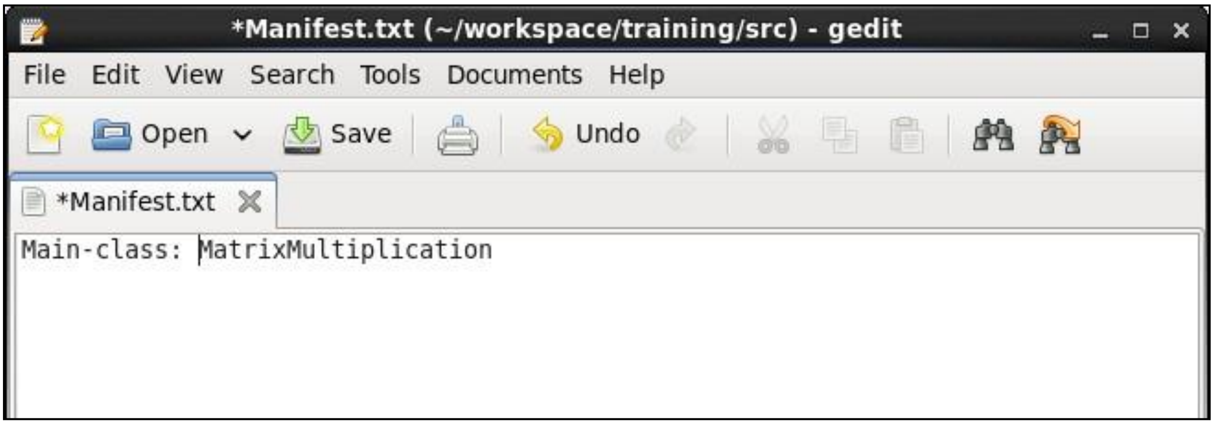
```
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -copyFromLocal m-matrix.txt /expt4/input/M
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -copyFromLocal n-matrix.txt /expt4/input/N
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ hdfs dfs -ls /expt4/input/M
Found 1 items
-rw-r--r--   1 hdfs supergroup         56 2022-11-12 10:12 /expt4/input/M/m-matrix.txt
[cloudera@quickstart ~]$ hdfs dfs -ls /expt4/input/N
Found 1 items
-rw-r--r--   1 hdfs supergroup         58 2022-11-12 10:12 /expt4/input/N/n-matrix.txt
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$
```

Compile the code to create the classes


cloudera@quickstart:~/workspace/training/src
File Edit View Search Terminal Help

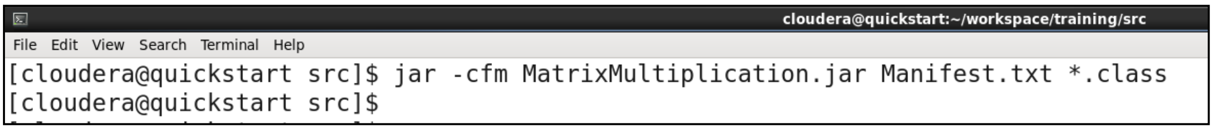
```
[cloudera@quickstart src]$ javac MatrixMultiplication.java -cp $(hadoop classpath)
[cloudera@quickstart src]$
[cloudera@quickstart src]$
[cloudera@quickstart src]$ gedit Manifest.txt
[cloudera@quickstart src]$
[cloudera@quickstart src]$
```

To indicate the main class file, create a Manifest file to point out to the main driver class.


*Manifest.txt (~/workspace/training/src) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
*Manifest.txt

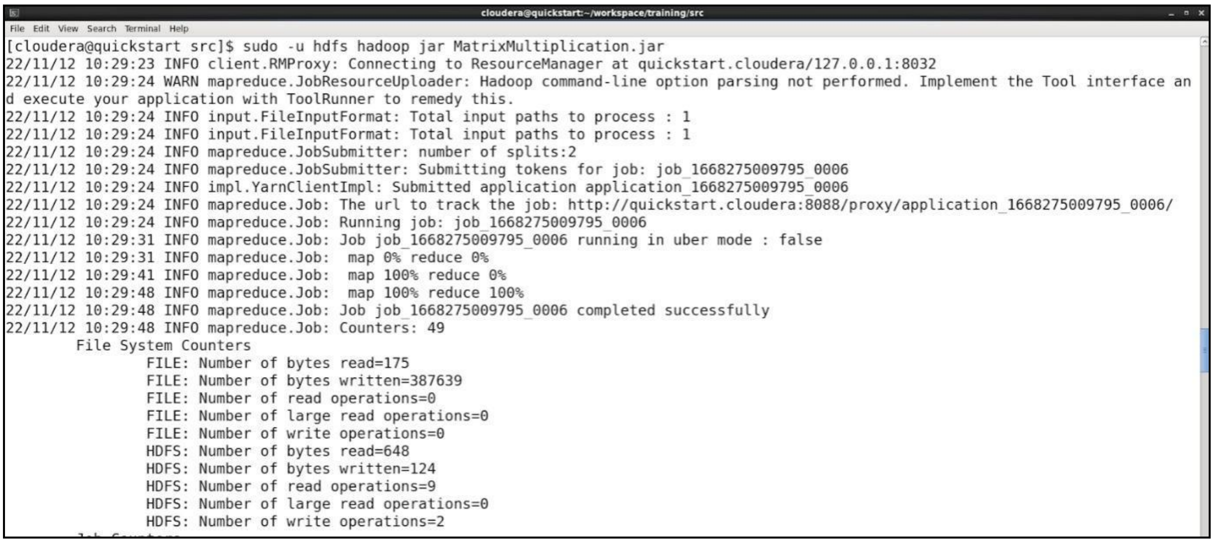
```
Main-class: MatrixMultiplication
```

Compile and create the Jar file required to run the MapReduce Task


cloudera@quickstart:~/workspace/training/src
File Edit View Search Terminal Help

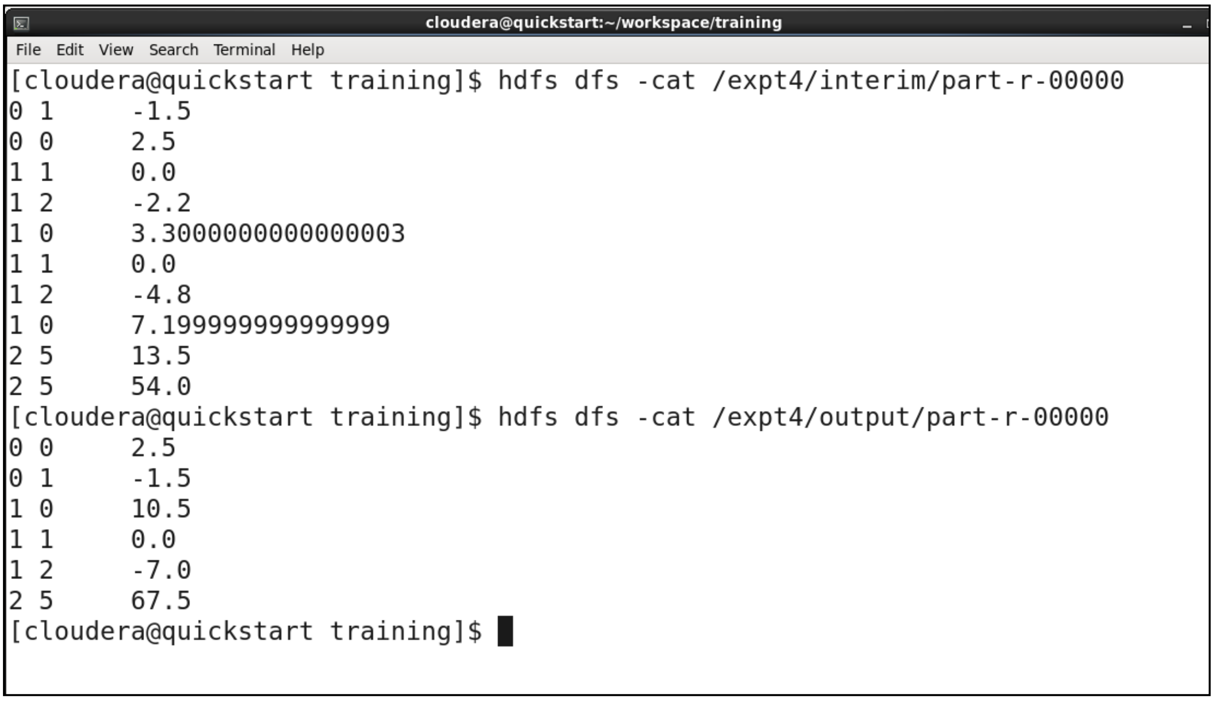
```
[cloudera@quickstart src]$ jar -cfm MatrixMultiplication.jar Manifest.txt *.class
[cloudera@quickstart src]$
```

Run the jar file on the Hadoop ecosystem to trigger all the MapReduce classes.


cloudera@quickstart:~/workspace/training/src
File Edit View Search Terminal Help

```
[cloudera@quickstart src]$ sudo -u hdfs hadoop jar MatrixMultiplication.jar
22/11/12 10:29:23 INFO client.RMProxy: Connecting to ResourceManager at quickstart.cloudera/127.0.0.1:8032
22/11/12 10:29:24 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface an
d execute your application with ToolRunner to remedy this.
22/11/12 10:29:24 INFO input.FileInputFormat: Total input paths to process : 1
22/11/12 10:29:24 INFO input.FileInputFormat: Total input paths to process : 1
22/11/12 10:29:24 INFO mapreduce.JobSubmitter: number of splits:2
22/11/12 10:29:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1668275009795_0006
22/11/12 10:29:24 INFO impl.YarnClientImpl: Submitted application application_1668275009795_0006
22/11/12 10:29:24 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1668275009795_0006/
22/11/12 10:29:24 INFO mapreduce.Job: Running job: job_1668275009795_0006
22/11/12 10:29:31 INFO mapreduce.Job: Job job_1668275009795_0006 running in uber mode : false
22/11/12 10:29:31 INFO mapreduce.Job:  map 0% reduce 0%
22/11/12 10:29:41 INFO mapreduce.Job:  map 100% reduce 0%
22/11/12 10:29:48 INFO mapreduce.Job:  map 100% reduce 100%
22/11/12 10:29:48 INFO mapreduce.Job: Job job_1668275009795_0006 completed successfully
22/11/12 10:29:48 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=175
                FILE: Number of bytes written=387639
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=648
                HDFS: Number of bytes written=124
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

Output of the file


cloudera@quickstart:~/workspace/training
File Edit View Search Terminal Help

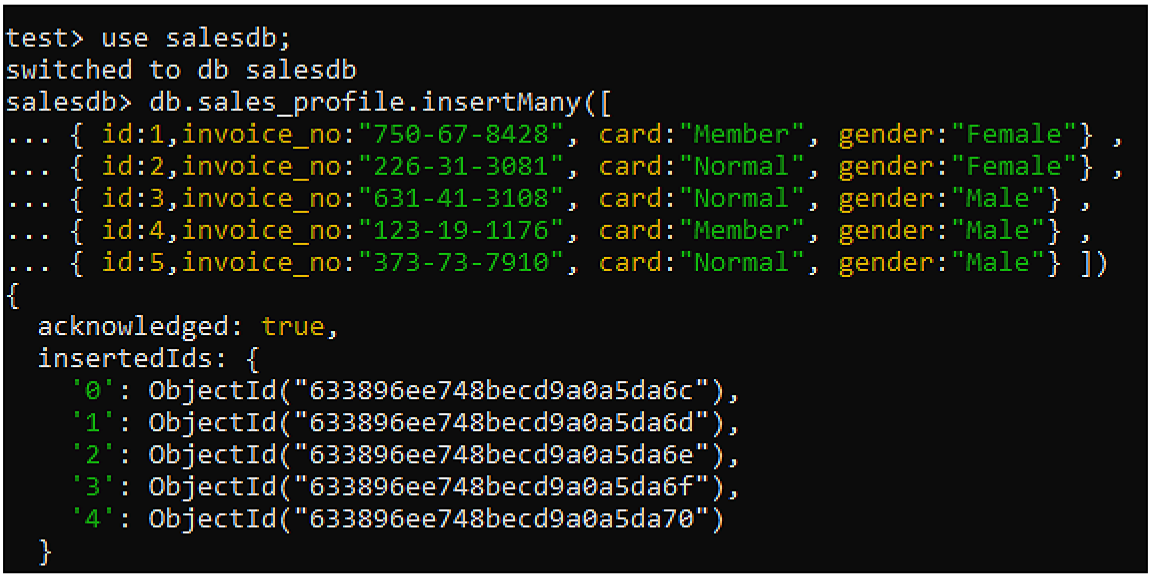
```
[cloudera@quickstart training]$ hdfs dfs -cat /expt4/interim/part-r-00000
0 1     -1.5
0 0     2.5
1 1     0.0
1 2     -2.2
1 0     3.3000000000000003
1 1     0.0
1 2     -4.8
1 0     7.199999999999999
2 5     13.5
2 5     54.0
[cloudera@quickstart training]$ hdfs dfs -cat /expt4/output/part-r-00000
0 0     2.5
0 1     -1.5
1 0     10.5
1 1     0.0
1 2     -7.0
2 5     67.5
[cloudera@quickstart training]$
```

Practical No - 3

Aim: Query the Sample Database using MongoDB querring commands.

1. create a database named salesdb and switch to salesdb.

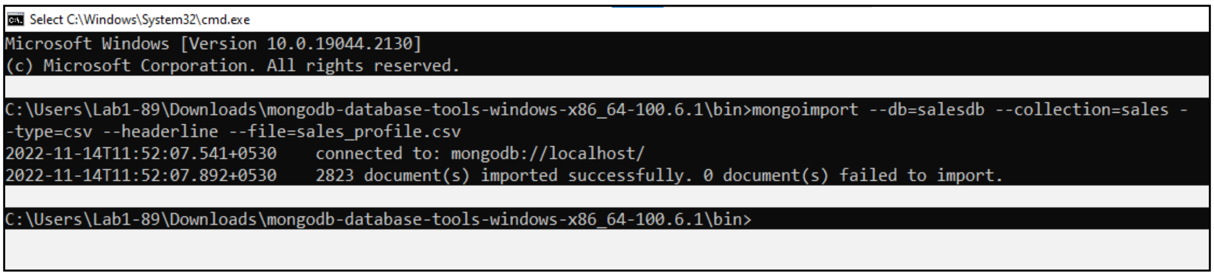
2. **We have two collections namely sales and sales_profile. Let us populate them. First run a query to insert data in sales_profile.**



```
test> use salesdb;
switched to db salesdb
salesdb> db.sales_profile.insertMany([
... { id:1,invoice_no:"750-67-8428", card:"Member", gender:"Female"} ,
... { id:2,invoice_no:"226-31-3081", card:"Normal", gender:"Female"} ,
... { id:3,invoice_no:"631-41-3108", card:"Normal", gender:"Male"} ,
... { id:4,invoice_no:"123-19-1176", card:"Member", gender:"Male"} ,
... { id:5,invoice_no:"373-73-7910", card:"Normal", gender:"Male"} ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("633896ee748becd9a0a5da6c"),
    '1': ObjectId("633896ee748becd9a0a5da6d"),
    '2': ObjectId("633896ee748becd9a0a5da6e"),
    '3': ObjectId("633896ee748becd9a0a5da6f"),
    '4': ObjectId("633896ee748becd9a0a5da70")
  }
```

Then we run a query to import sales_profile.csv in sales_profile collection.

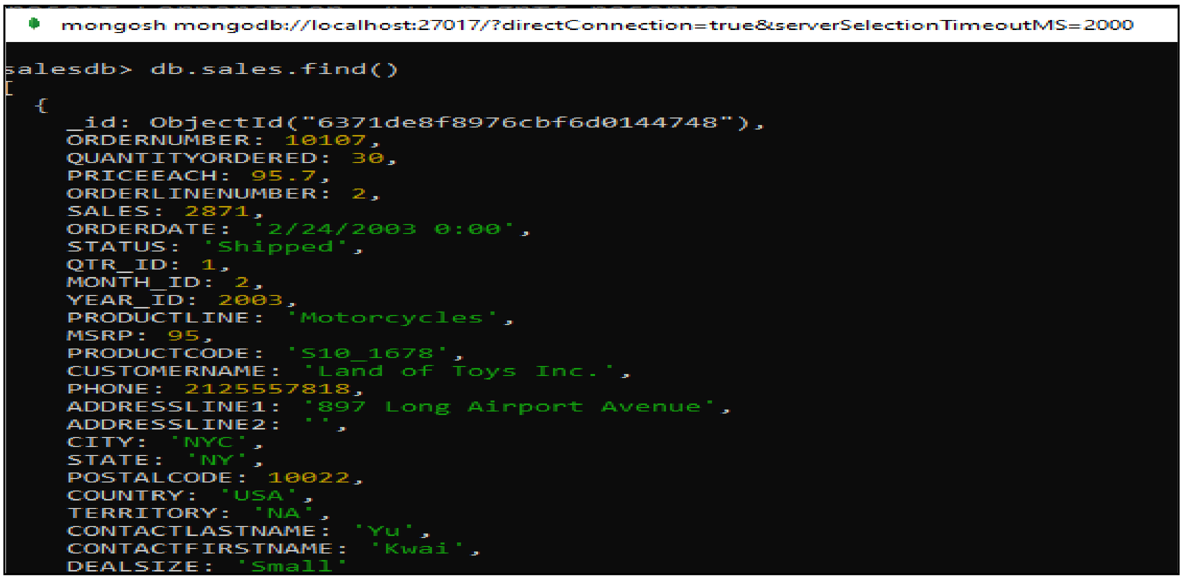
Query :mongoimport --db=salesdb --collection=sales --type=csv --headerline --file=sales_profile.csv



```
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lab1-89\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\bin>mongoimport --db=salesdb --collection=sales -
-type=csv --headerline --file=sales_profile.csv
2022-11-14T11:52:07.541+0530    connected to: mongodb://localhost/
2022-11-14T11:52:07.892+0530    2823 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\Lab1-89\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\bin>
```

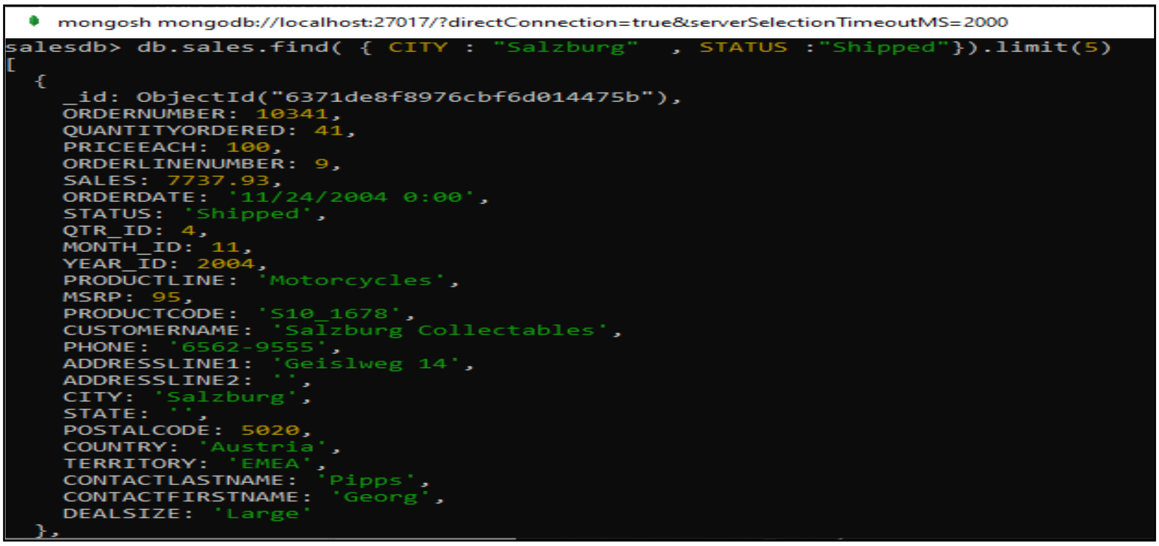
3. Find data query

Query : db.sales.find()



```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000

salesdb> db.sales.find()
[
  {
    _id: ObjectId("6371de8f8976cbf6d0144748"),
    ORDERNUMBER: 10107,
    QUANTITYORDERED: 30,
    PRICEEACH: 95.7,
    ORDERLINENUMBER: 2,
    SALES: 2871,
    ORDERDATE: '2/24/2003 0:00',
    STATUS: 'Shipped',
    QTR_ID: 1,
    MONTH_ID: 2,
    YEAR_ID: 2003,
    PRODUCTLINE: 'Motorcycles',
    MSRP: 95,
    PRODUCTCODE: 'S10_1678',
    CUSTOMERNAME: 'Land of Toys Inc.',
    PHONE: 2125557818,
    ADDRESSLINE1: '897 Long Airport Avenue',
    ADDRESSLINE2: '',
    CITY: 'NYC',
    STATE: 'NY',
    POSTALCODE: 10022,
    COUNTRY: 'USA',
    TERRITORY: 'NA',
    CONTACTLASTNAME: 'Yu',
    CONTACTFIRSTNAME: 'Kwai',
    DEALSIZE: 'Small'
```

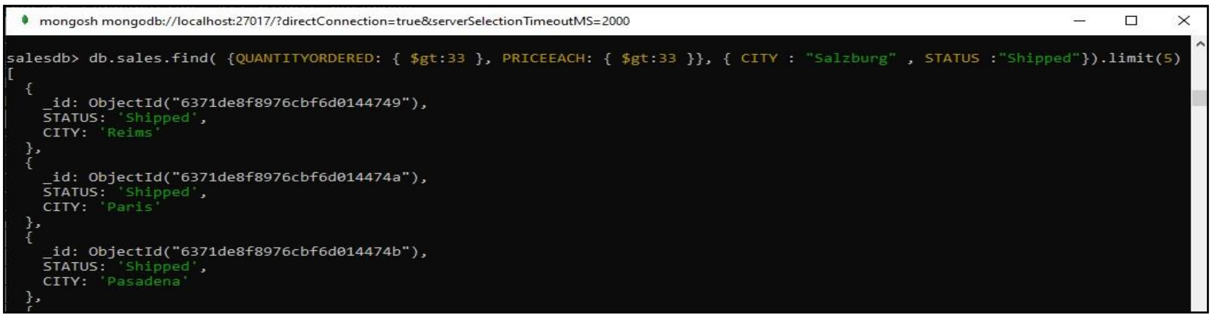
db.sales.find({ CITY : "Salzburg" , STATUS :"Shipped"}).limit(5)



```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
salesdb> db.sales.find( { CITY : "Salzburg"  , STATUS :"Shipped"}).limit(5)
[
  {
    _id: ObjectId("6371de8f8976cbf6d014475b"),
    ORDERNUMBER: 10341,
    QUANTITYORDERED: 41,
    PRICEEACH: 100,
    ORDERLINENUMBER: 9,
    SALES: 7737.93,
    ORDERDATE: '11/24/2004 0:00',
    STATUS: 'Shipped',
    QTR_ID: 4,
    MONTH_ID: 11,
    YEAR_ID: 2004,
    PRODUCTLINE: 'Motorcycles',
    MSRP: 95,
    PRODUCTCODE: 'S10_1678',
    CUSTOMERNAME: 'Salzburg Collectables',
    PHONE: '6562-9555',
    ADDRESSLINE1: 'Geislweg 14',
    ADDRESSLINE2: '',
    CITY: 'Salzburg',
    STATE: '',
    POSTALCODE: 5020,
    COUNTRY: 'Austria',
    TERRITORY: 'EMEA',
    CONTACTLASTNAME: 'Pipps',
    CONTACTFIRSTNAME: 'Georg',
    DEALSIZE: 'Large'
  },
```

4. Find data using greater than condition

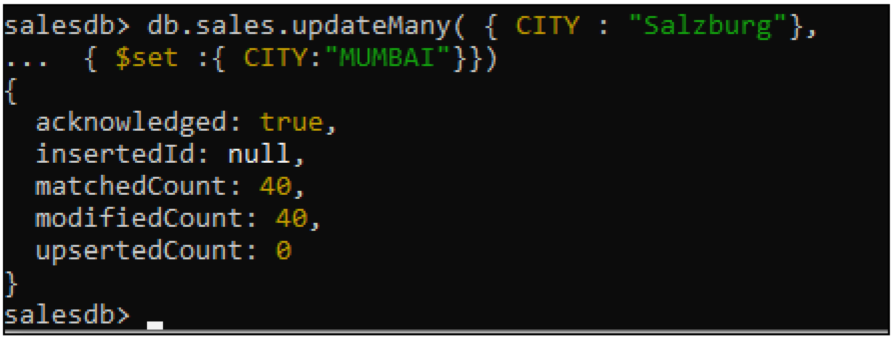
db.sales.find({QUANTITYORDERED: { $gt:33 }, PRICEEACH: { $gt:33 }}, { CITY :"Salzburg" , STATUS :"Shipped"}).limit(5)



```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000

salesdb> db.sales.find( {QUANTITYORDERED: { $gt:33 }, PRICEEACH: { $gt:33 }}, { CITY : "Salzburg" , STATUS :"Shipped"}).limit(5)
[
  {
    _id: ObjectId("6371de8f8976cbf6d0144749"),
    STATUS: 'Shipped',
    CITY: 'Reims'
  },
  {
    _id: ObjectId("6371de8f8976cbf6d014474a"),
    STATUS: 'Shipped',
    CITY: 'Paris'
  },
  {
    _id: ObjectId("6371de8f8976cbf6d014474b"),
    STATUS: 'Shipped',
    CITY: 'Pasadena'
  },
```

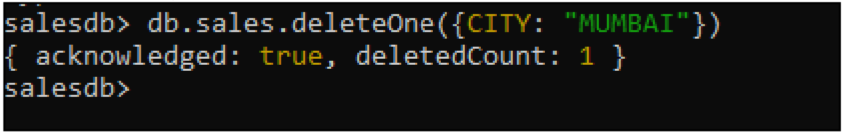
5. Update document

db.sales.updateMany({ CITY : “Salzburg”}, { $set :{ CITY:”MUMBAI” }})



```
salesdb> db.sales.updateMany( { CITY : "Salzburg"},
...   { $set :{ CITY:"MUMBAI"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 40,
  modifiedCount: 40,
  upsertedCount: 0
}
salesdb> _
```

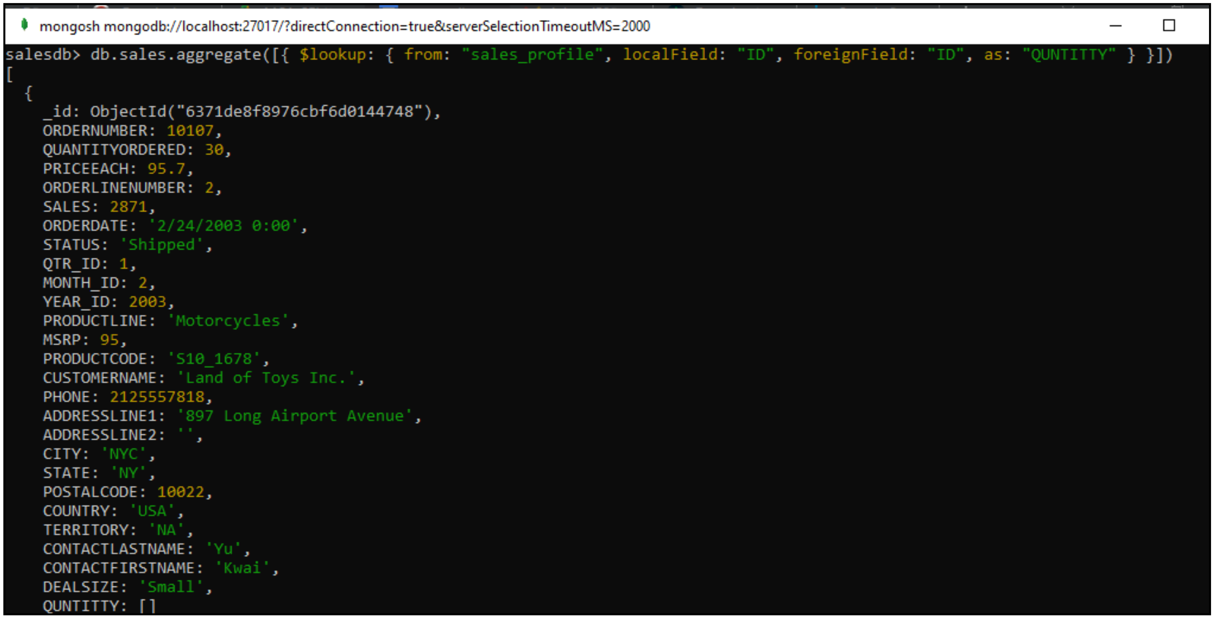
6. Delete document db.sales.deleteOne({CITY : “MUMBAI” })



```
salesdb> db.sales.deleteOne({CITY: "MUMBAI"})
{ acknowledged: true, deletedCount: 1 }
salesdb>
```

7. Aggregate function

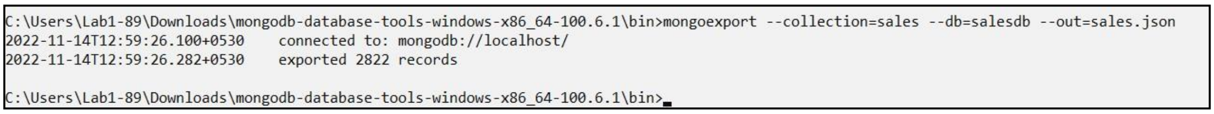
db.sales.aggregate([{ $lookup: {from: "sales_profile",localField: "ID", foreignField: "ID", as: "QUNTITTY" }}])


mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
salesdb> db.sales.aggregate([{ $lookup: { from: "sales_profile", localField: "ID", foreignField: "ID", as: "QUNTITTY" } }])
[
  {
    _id: ObjectId("6371de8f8976cbf6d0144748"),
    ORDERNUMBER: 10107,
    QUANTITYORDERED: 30,
    PRICEEACH: 95.7,
    ORDERLINENUMBER: 2,
    SALES: 2871,
    ORDERDATE: '2/24/2003 0:00',
    STATUS: 'Shipped',
    QTR_ID: 1,
    MONTH_ID: 2,
    YEAR_ID: 2003,
    PRODUCTLINE: 'Motorcycles',
    MSRP: 95,
    PRODUCTCODE: 'S10_1678',
    CUSTOMERNAME: 'Land of Toys Inc.',
    PHONE: 2125557818,
    ADDRESSLINE1: '897 Long Airport Avenue',
    ADDRESSLINE2: '',
    CITY: 'NYC',
    STATE: 'NY',
    POSTALCODE: 10022,
    COUNTRY: 'USA',
    TERRITORY: 'NA',
    CONTACTLASTNAME: 'Yu',
    CONTACTFIRSTNAME: 'Kwai',
    DEALSIZE: 'Small',
    QUNTITTY: []
```

8. Extract the salesdb database in json

mongoexport --collection=sales --db=salesdb --out=sales.json

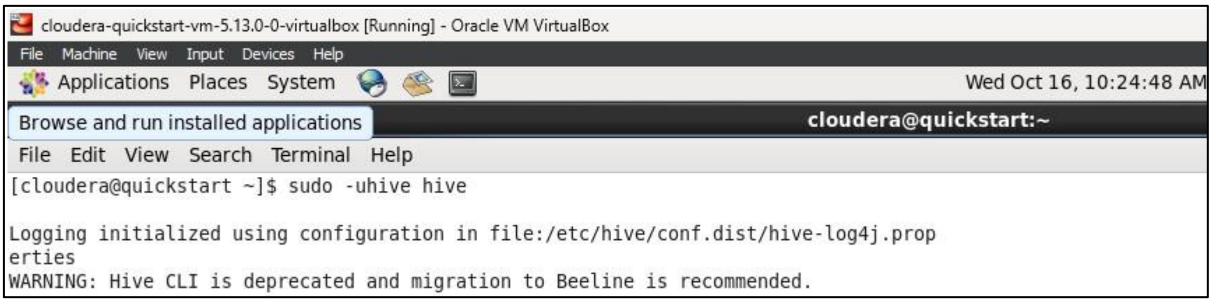


```
C:\Users\Lab1-89\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\bin>mongoexport --collection=sales --db=salesdb --out=sales.json
2022-11-14T12:59:26.100+0530    connected to: mongodb://localhost/
2022-11-14T12:59:26.282+0530    exported 2822 records

C:\Users\Lab1-89\Downloads\mongodb-database-tools-windows-x86_64-100.6.1\bin>_
```

Practical No : 4

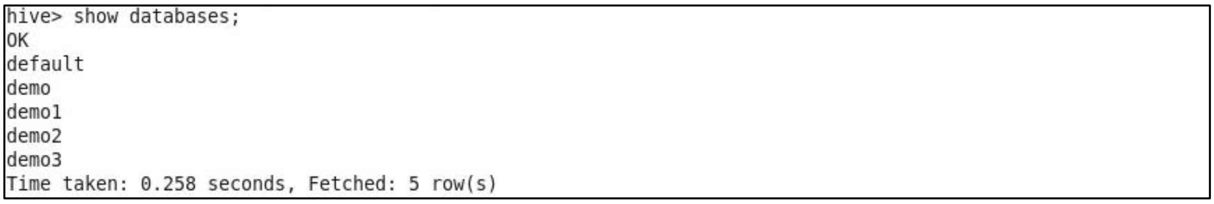
Aim: Basic HIVE Commands

1. Log in into Hive.


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Wed Oct 16, 10:24:48 AM
Browse and run installed applications
cloudera@quickstart:~
File Edit View Search Terminal Help

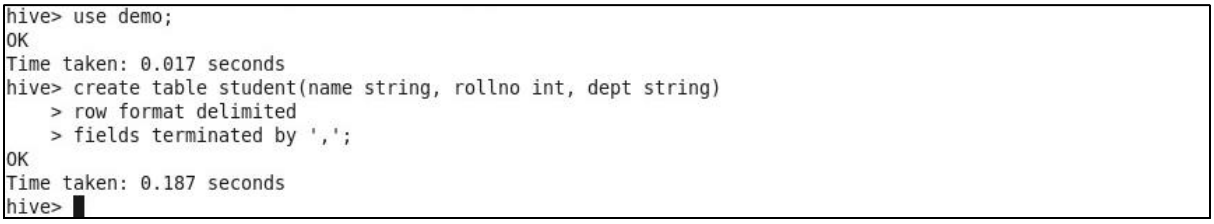
```
[cloudera@quickstart ~]$ sudo -uhive hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.prop
erties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
```

2. Show available databases.



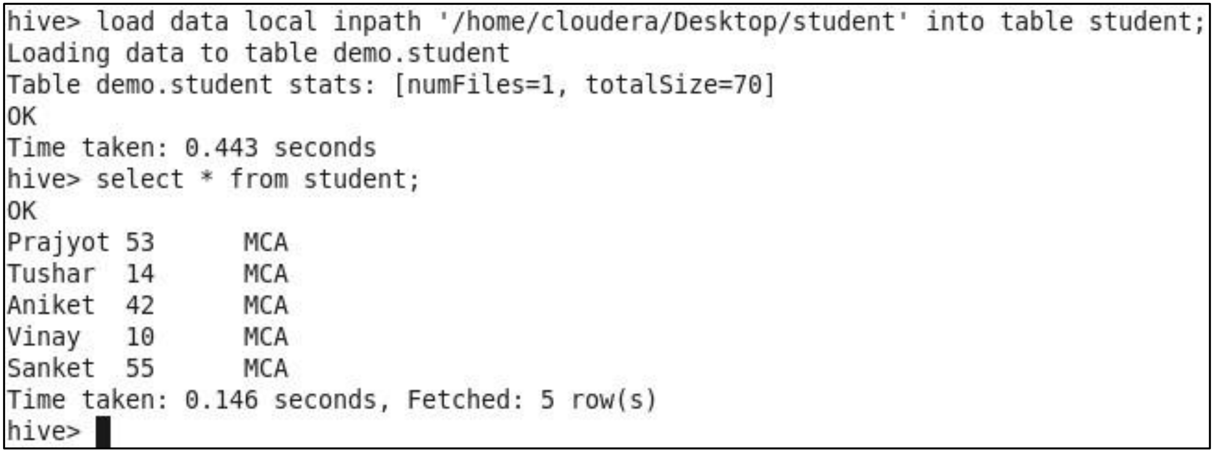
```
hive> show databases;
OK
default
demo
demo1
demo2
demo3
Time taken: 0.258 seconds, Fetched: 5 row(s)
```

3. Select a database and create a table name student.



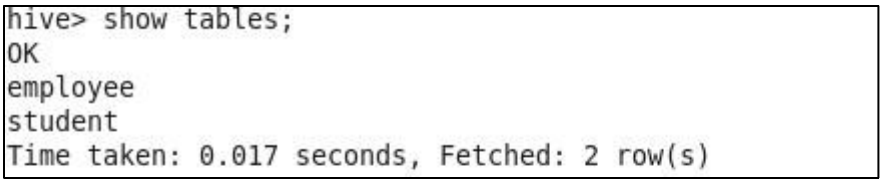
```
hive> use demo;
OK
Time taken: 0.017 seconds
hive> create table student(name string, rollno int, dept string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.187 seconds
hive> 
```

4. Import data from a file and display the table with the data.



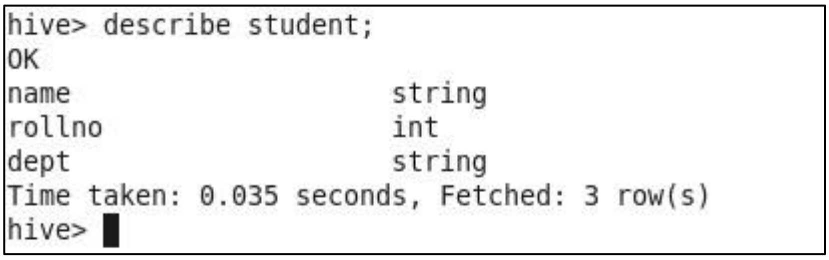
```
hive> load data local inpath '/home/cloudera/Desktop/student' into table student;
Loading data to table demo.student
Table demo.student stats: [numFiles=1, totalSize=70]
OK
Time taken: 0.443 seconds
hive> select * from student;
OK
Prajyot 53      MCA
Tushar  14      MCA
Aniket  42      MCA
Vinay   10      MCA
Sanket  55      MCA
Time taken: 0.146 seconds, Fetched: 5 row(s)
hive> 
```

5. Show available tables.



```
hive> show tables;
OK
employee
student
Time taken: 0.017 seconds, Fetched: 2 row(s)
```

6. Describe the table.



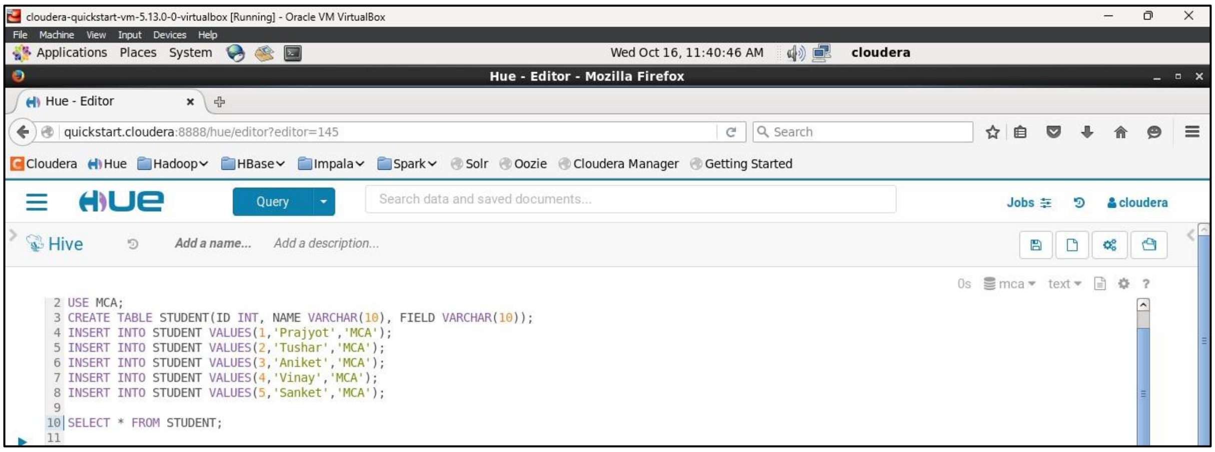
```
hive> describe student;
OK
name                    string
rollno                  int
dept                    string
Time taken: 0.035 seconds, Fetched: 3 row(s)
hive> █
```

USING HUE EDITOR

7. Create a database and a table.

Input:

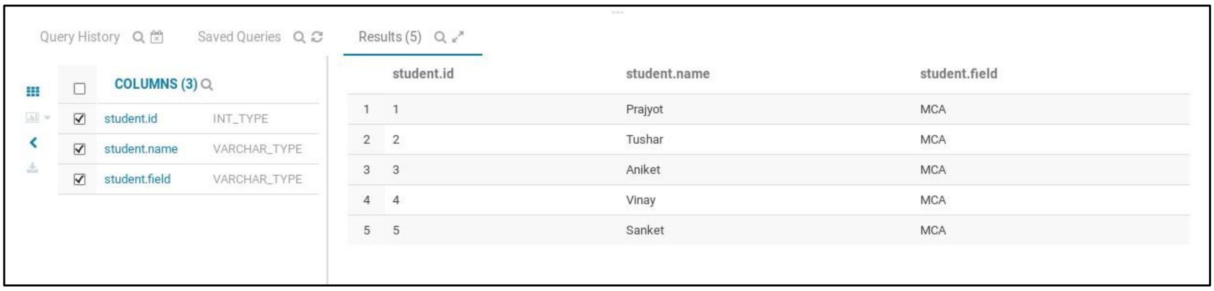
CREATE DATABASE MCA;
USE MCA;
CREATE TABLE STUDENT(ID INT, NAME VARCHAR(10), FIELD VARCHAR(10));
INSERT INTO STUDENT VALUES(1, ’Prajyot’, ‘MCA’);
INSERT INTO STUDENT VALUES(2, ’Tushar’, ‘MCA’);
INSERT INTO STUDENT VALUES(3, ’Aniket’, ‘MCA’);
INSERT INTO STUDENT VALUES(4, ’Atharva’, ‘MCA’);
INSERT INTO STUDENT VALUES(5, ’Sanket’, ‘MCA’);
SELECT * FROM STUDENT;



cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Wed Oct 16, 11:40:46 AM
cloudera
Hue - Editor - Mozilla Firefox
Hue - Editor
quickstart.cloudera:8888/hue/editor?editor=145
Search
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
HUE
Query
Search data and saved documents...
Jobs
cloudera
Hive
Add a name...
Add a description...
0s mca text ?

```
USE MCA;
CREATE TABLE STUDENT(ID INT, NAME VARCHAR(10), FIELD VARCHAR(10));
INSERT INTO STUDENT VALUES(1,'Prajyot','MCA');
INSERT INTO STUDENT VALUES(2,'Tushar','MCA');
INSERT INTO STUDENT VALUES(3,'Aniket','MCA');
INSERT INTO STUDENT VALUES(4,'Vinay','MCA');
INSERT INTO STUDENT VALUES(5,'Sanket','MCA');

SELECT * FROM STUDENT;
```

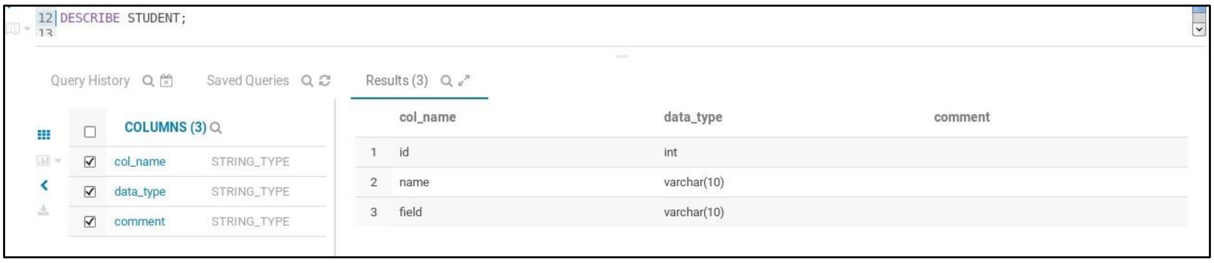
Output:



☐	
☑ student.id	INT_TYPE
☑ student.name	VARCHAR_TYPE
☑ student.field	VARCHAR_TYPE

	student.id	student.name	student.field
1	1	Prajyot	MCA
2	2	Tushar	MCA
3	3	Aniket	MCA
4	4	Vinay	MCA
5	5	Sanket	MCA

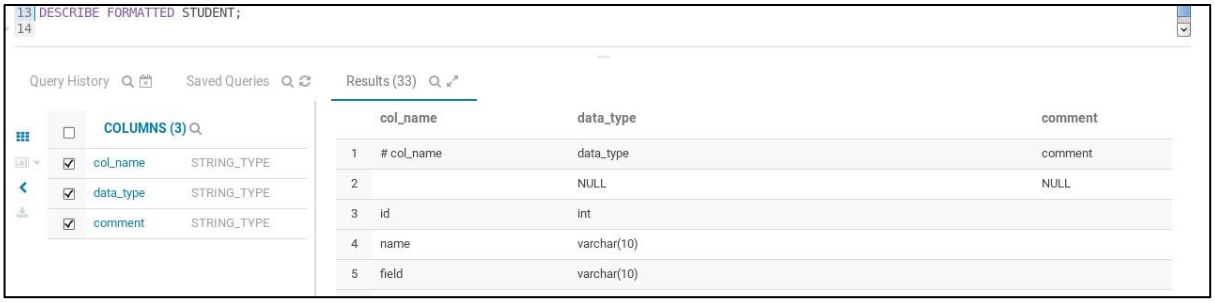
8. Describe the table.


```
DESCRIBE STUDENT;
```

☐	
☑ col_name	STRING_TYPE
☑ data_type	STRING_TYPE
☑ comment	STRING_TYPE

	col_name	data_type	comment
1	id	int	
2	name	varchar(10)	
3	field	varchar(10)	

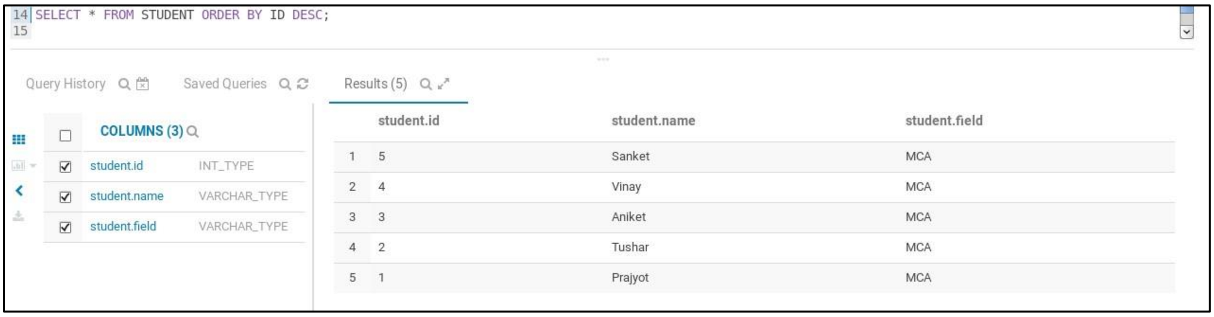
9. Describe formatted student.


```
DESCRIBE FORMATTED STUDENT;
```

☐	
☑ col_name	STRING_TYPE
☑ data_type	STRING_TYPE
☑ comment	STRING_TYPE

	col_name	data_type	comment
1	# col_name	data_type	comment
2		NULL	NULL
3	id	int	
4	name	varchar(10)	
5	field	varchar(10)	

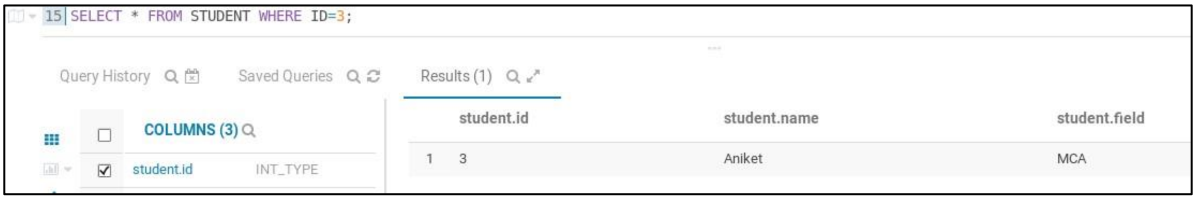
10. Student order by id desc;


```
SELECT * FROM STUDENT ORDER BY ID DESC;
```

☐	
☑ student.id	INT_TYPE
☑ student.name	VARCHAR_TYPE
☑ student.field	VARCHAR_TYPE

	student.id	student.name	student.field
1	5	Sanket	MCA
2	4	Vinay	MCA
3	3	Aniket	MCA
4	2	Tushar	MCA
5	1	Prajyot	MCA

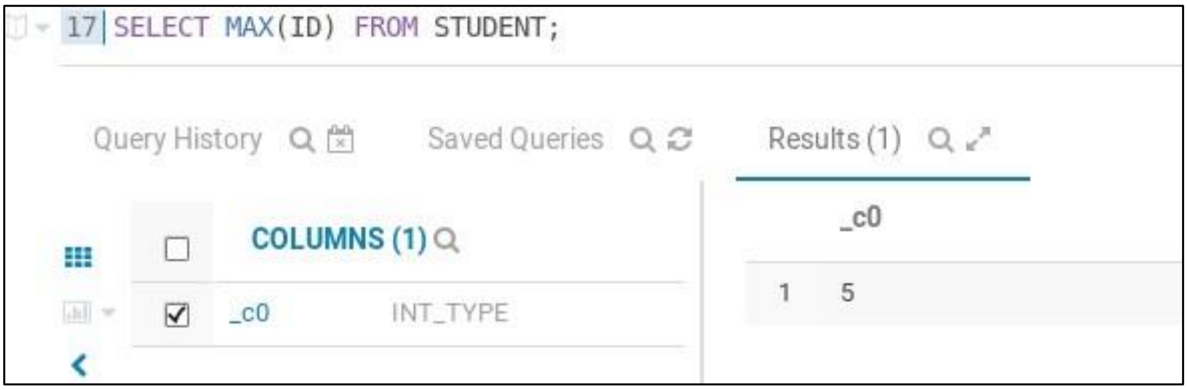
11. Select * From student where id=3;



```
SELECT * FROM STUDENT WHERE ID=3;
```

	student.id	student.name	student.field
1	3	Aniket	MCA

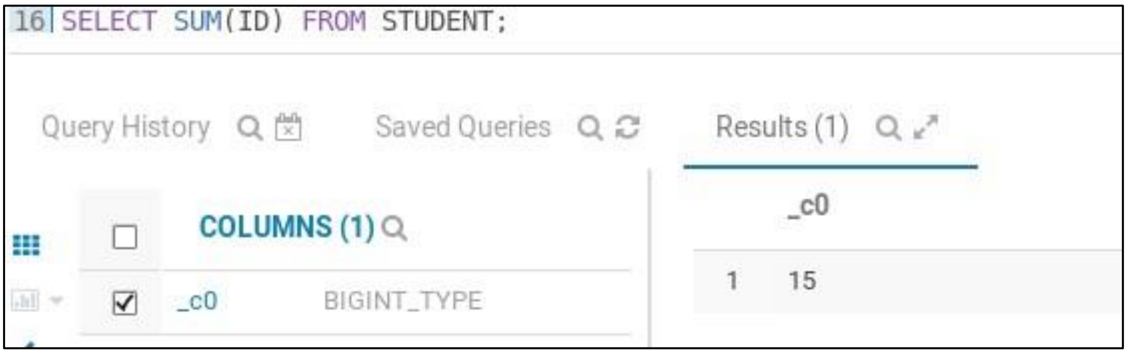
12.Select sum(id) from student;



```
SELECT MAX(ID) FROM STUDENT;
```

	_c0
1	5

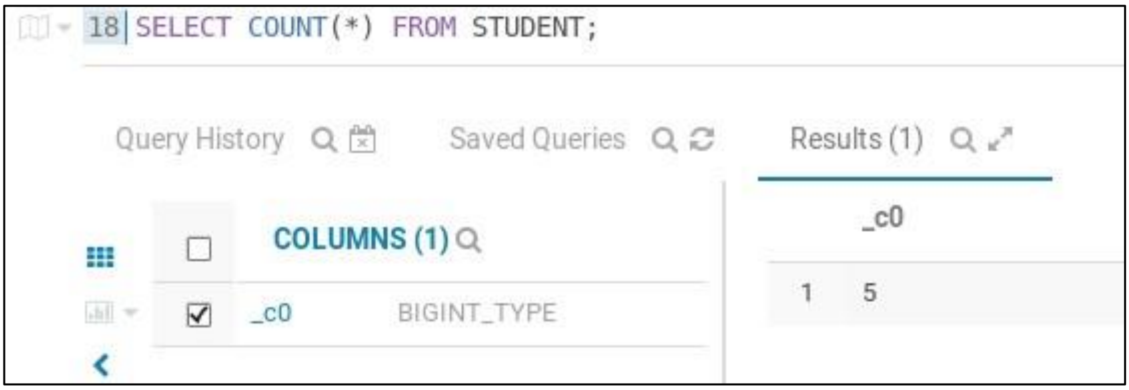
13. Select Max(id) from student;



```
SELECT SUM(ID) FROM STUDENT;
```

	_c0
1	15

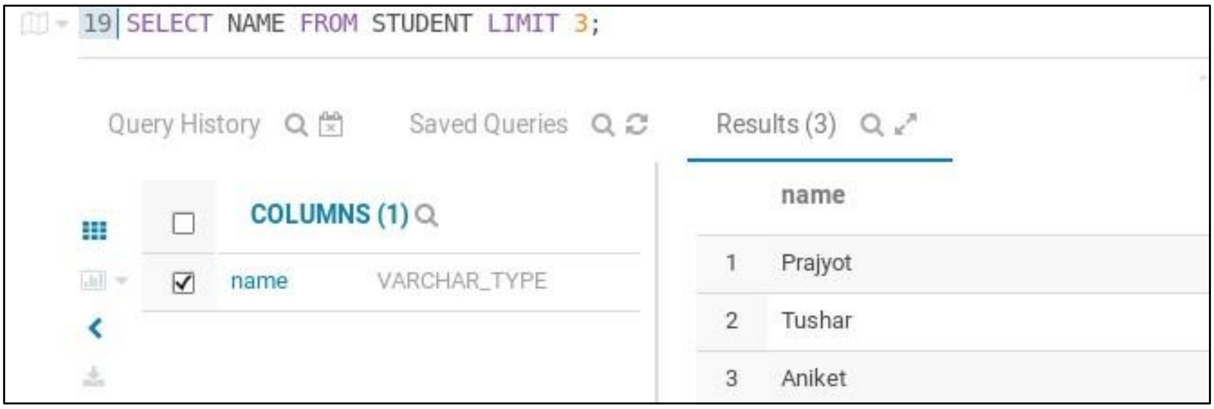
14. Select count(*) from student;



```
SELECT COUNT(*) FROM STUDENT;
```

	_c0
1	5

15.Select name from student limit 3;



```
SELECT NAME FROM STUDENT LIMIT 3;
```

☐	COLUMNS (1)
☑	name VARCHAR_TYPE

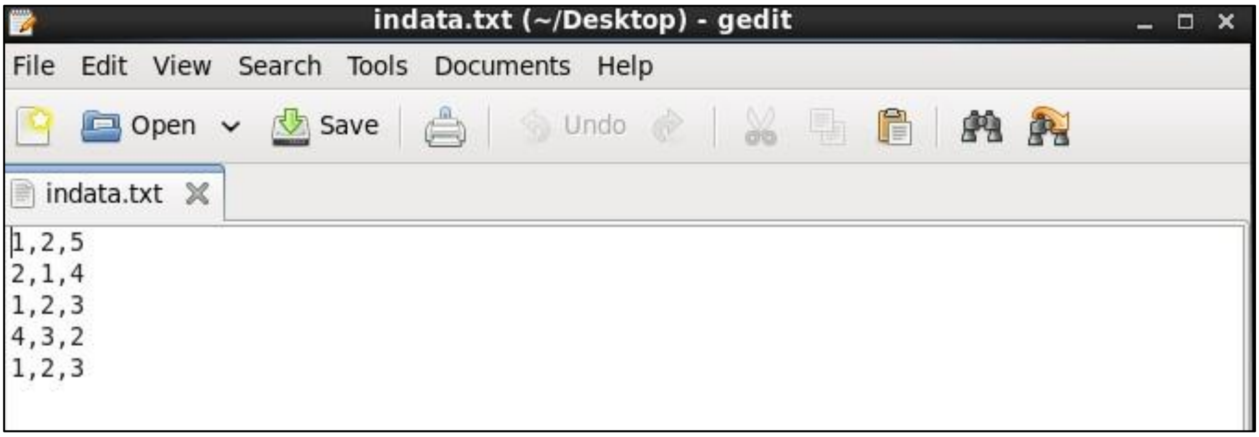
	name
1	Prajyot
2	Tushar
3	Aniket

Practical No : 5

Aim: - PIG List of Commands.

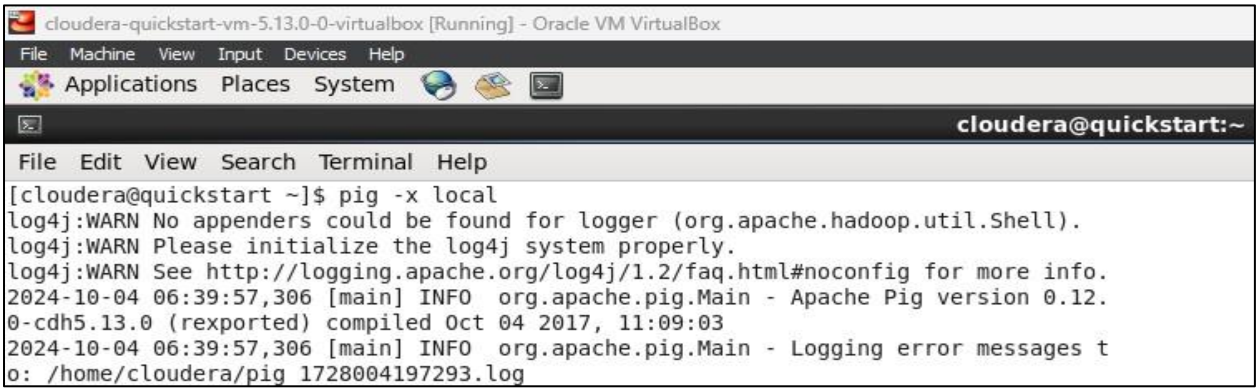
A) Student1.txt

1. Indata.txt


indata.txt (~/Desktop) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
indata.txt

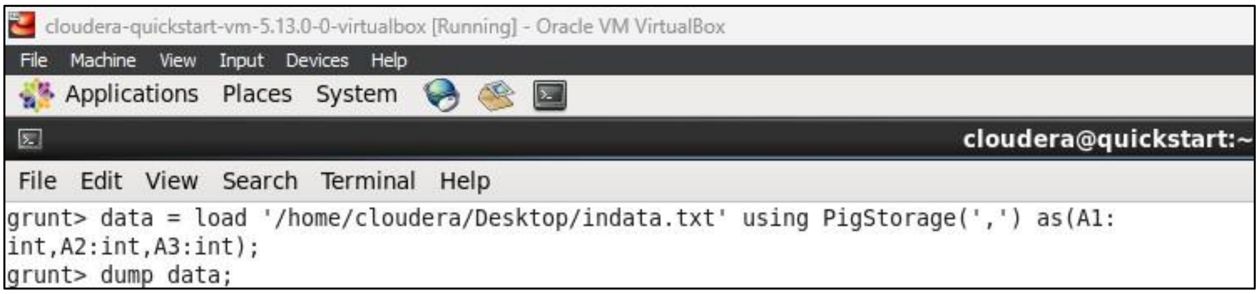
```
1,2,5
2,1,4
1,2,3
4,3,2
1,2,3
```

2. Pig –x local

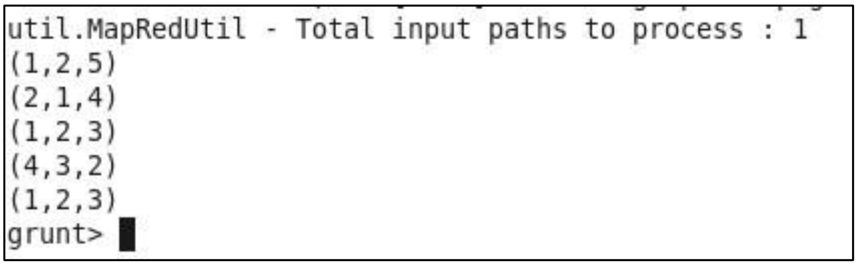

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ pig -x local
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2024-10-04 06:39:57,306 [main] INFO  org.apache.pig.Main - Apache Pig version 0.12.
0-cdh5.13.0 (rexported) compiled Oct 04 2017, 11:09:03
2024-10-04 06:39:57,306 [main] INFO  org.apache.pig.Main - Logging error messages t
o: /home/cloudera/pig_1728004197293.log
```

3. Local data and dump data.

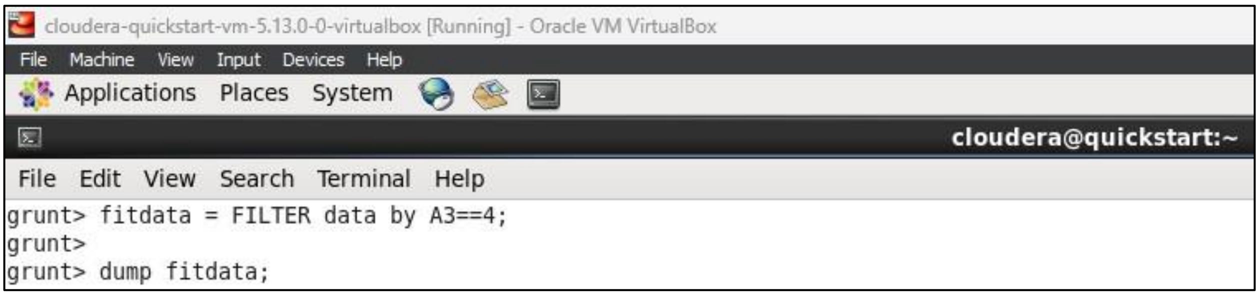

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help

```
grunt> data = load '/home/cloudera/Desktop/indata.txt' using PigStorage(',') as(A1:
int,A2:int,A3:int);
grunt> dump data;
```

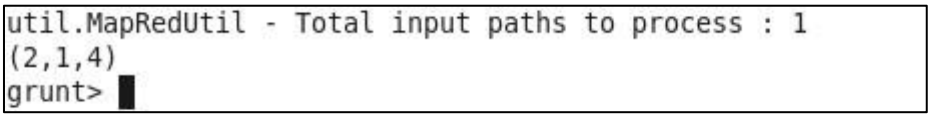


```
util.MapRedUtil - Total input paths to process : 1
(1,2,5)
(2,1,4)
(1,2,3)
(4,3,2)
(1,2,3)
grunt> █
```

4. Filter

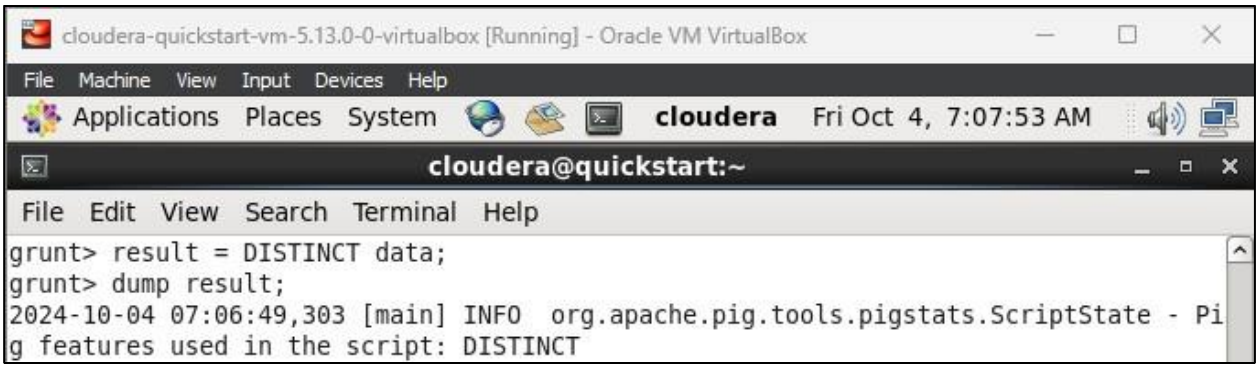

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help

```
grunt> fitdata = FILTER data by A3==4;
grunt>
grunt> dump fitdata;
```

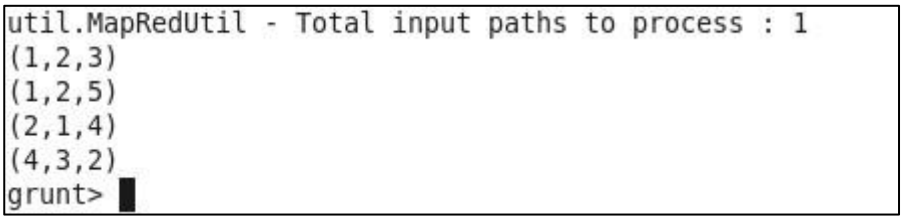


```
util.MapRedUtil - Total input paths to process : 1
(2,1,4)
grunt> █
```

5. Distinct

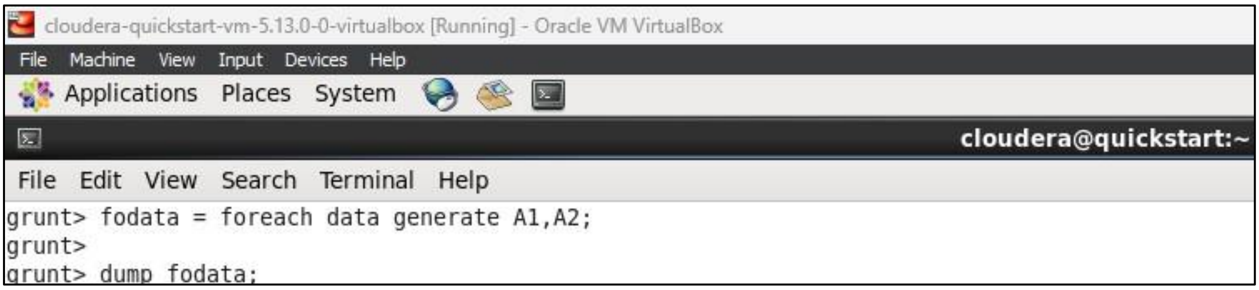

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera
Fri Oct 4, 7:07:53 AM
cloudera@quickstart:~
File Edit View Search Terminal Help

```
grunt> result = DISTINCT data;
grunt> dump result;
2024-10-04 07:06:49,303 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pi
g features used in the script: DISTINCT
```

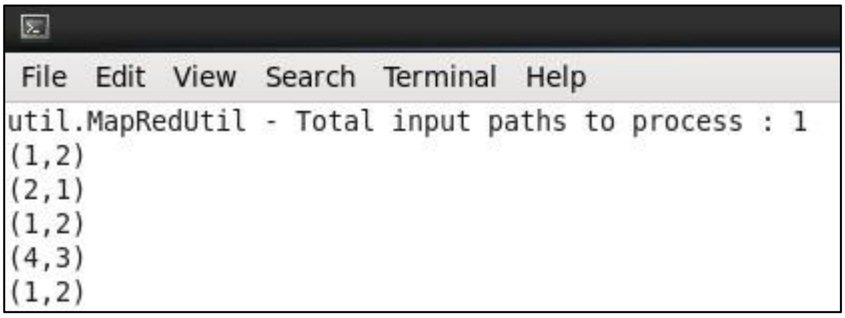


```
util.MapRedUtil - Total input paths to process : 1
(1,2,3)
(1,2,5)
(2,1,4)
(4,3,2)
grunt> █
```

6. Foreach

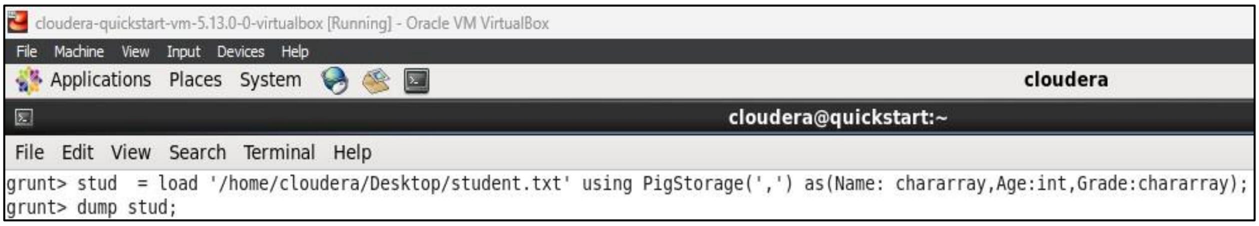

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help

```
grunt> fodata = foreach data generate A1,A2;
grunt>
grunt> dump fodata;
```

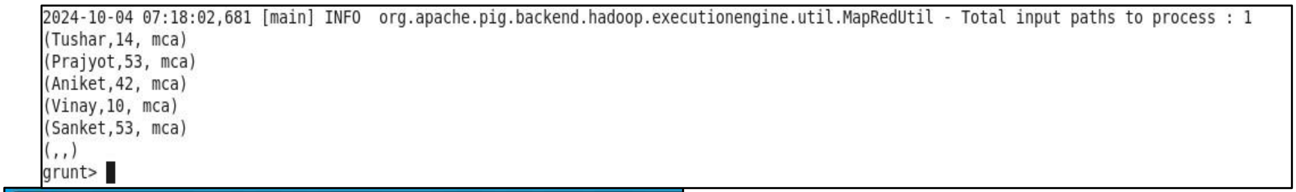


```
util.MapRedUtil - Total input paths to process : 1
(1,2)
(2,1)
(1,2)
(4,3)
(1,2)
```

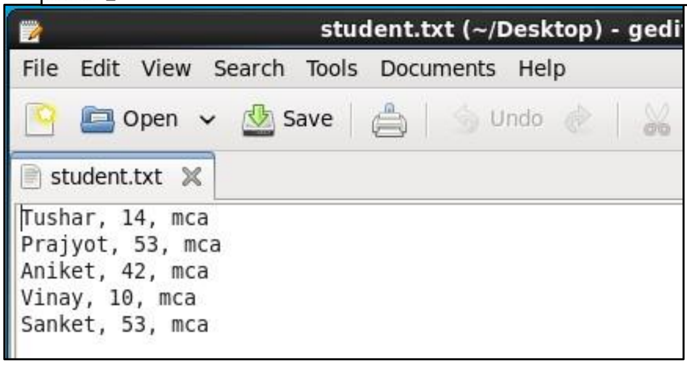
7. Load


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help

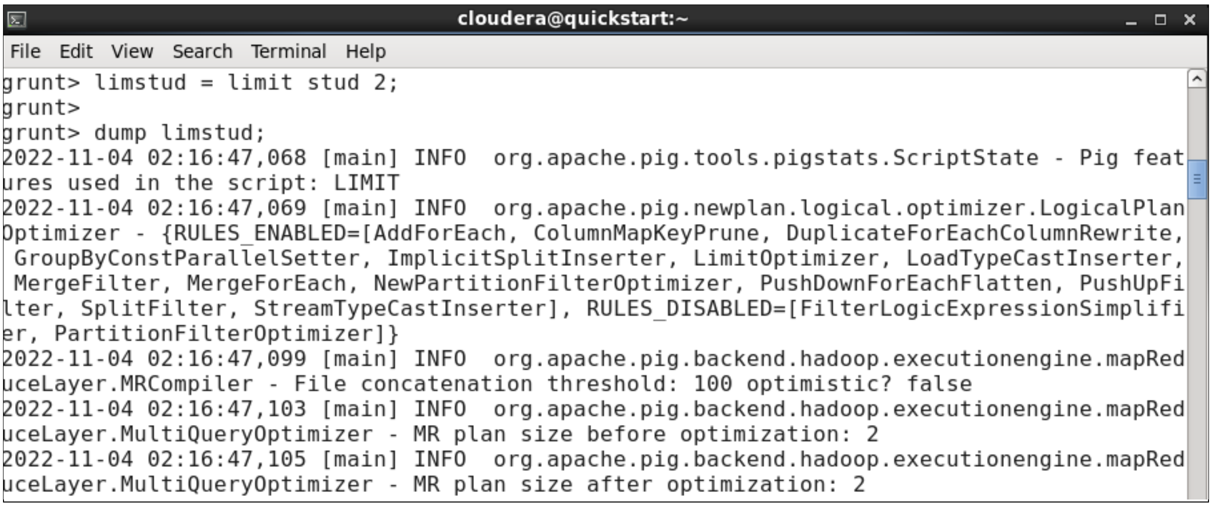
```
grunt> stud  = load '/home/cloudera/Desktop/student.txt' using PigStorage(',') as(Name: chararray,Age:int,Grade:chararray);
grunt> dump stud;
```



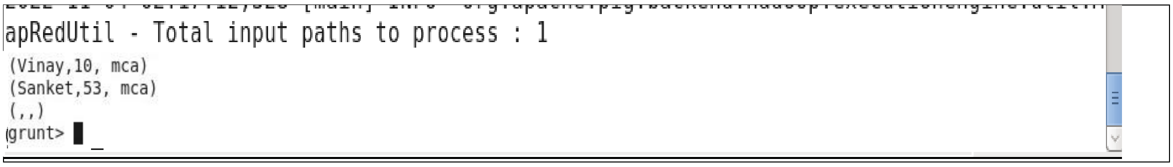
```
2024-10-04 07:18:02,681 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(Tushar,14, mca)
(Prajyot,53, mca)
(Aniket,42, mca)
(Vinay,10, mca)
(Sanket,53, mca)
(,,)
grunt> 
```


student.txt (~/Desktop) - gedi
File Edit View Search Tools Documents Help
Open
Save
Undo
student.txt

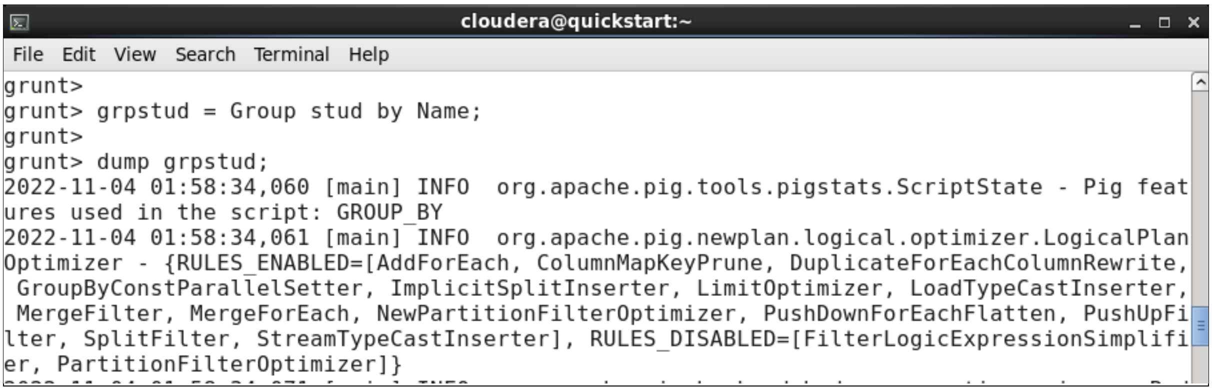
```
Tushar, 14, mca
Prajyot, 53, mca
Aniket, 42, mca
Vinay, 10, mca
Sanket, 53, mca
```



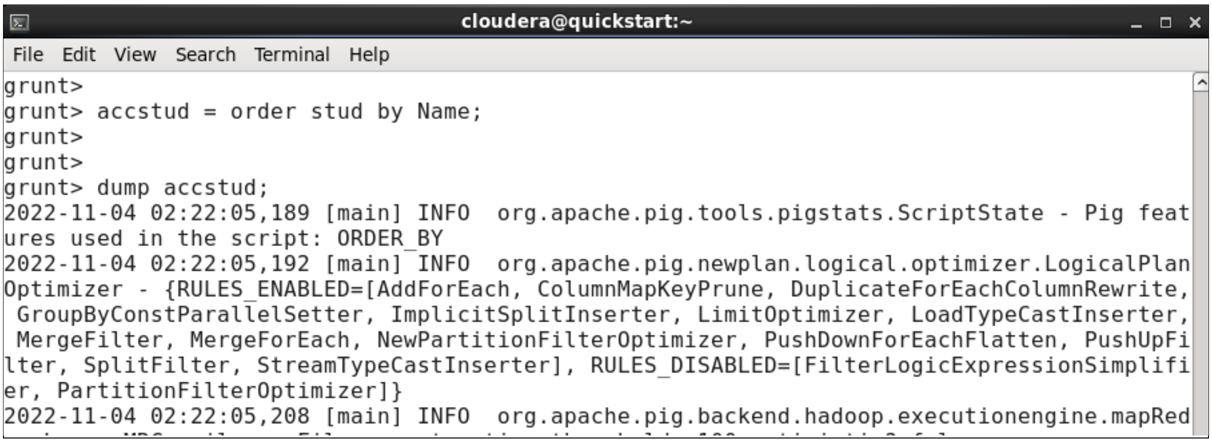
```
grunt> limstud = limit stud 2;
grunt>
grunt> dump limstud;
2022-11-04 02:16:47,068 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig feat
ures used in the script: LIMIT
2022-11-04 02:16:47,069 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlan
Optimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite,
 GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter,
 MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFi
lter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifi
er, PartitionFilterOptimizer]}
2022-11-04 02:16:47,099 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
uceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2022-11-04 02:16:47,103 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
uceLayer.MultiQueryOptimizer - MR plan size before optimization: 2
2022-11-04 02:16:47,105 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
uceLayer.MultiQueryOptimizer - MR plan size after optimization: 2
```



```
apRedUtil - Total input paths to process : 1
(Vinay,10, mca)
(Sanket,53, mca)
(,,)
grunt> 
```



```
grunt>
grunt> grpstud = Group stud by Name;
grunt>
grunt> dump grpstud;
2022-11-04 01:58:34,060 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig feat
ures used in the script: GROUP_BY
2022-11-04 01:58:34,061 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlan
Optimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite,
 GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter,
 MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFi
lter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifi
er, PartitionFilterOptimizer]}
```

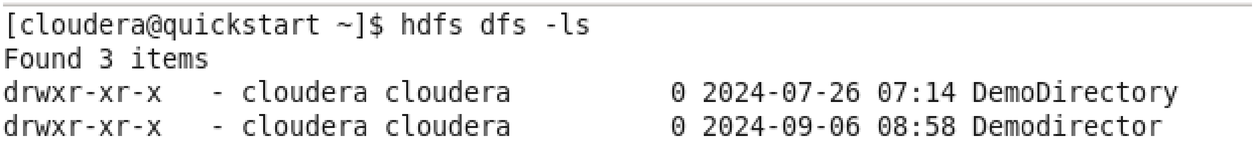


```
grunt>
grunt> accstud = order stud by Name;
grunt>
grunt>
grunt> dump accstud;
2022-11-04 02:22:05,189 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig feat
ures used in the script: ORDER_BY
2022-11-04 02:22:05,192 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlan
Optimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite,
 GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter,
 MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFi
lter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifi
er, PartitionFilterOptimizer]}
2022-11-04 02:22:05,208 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
```

Practical No : 6

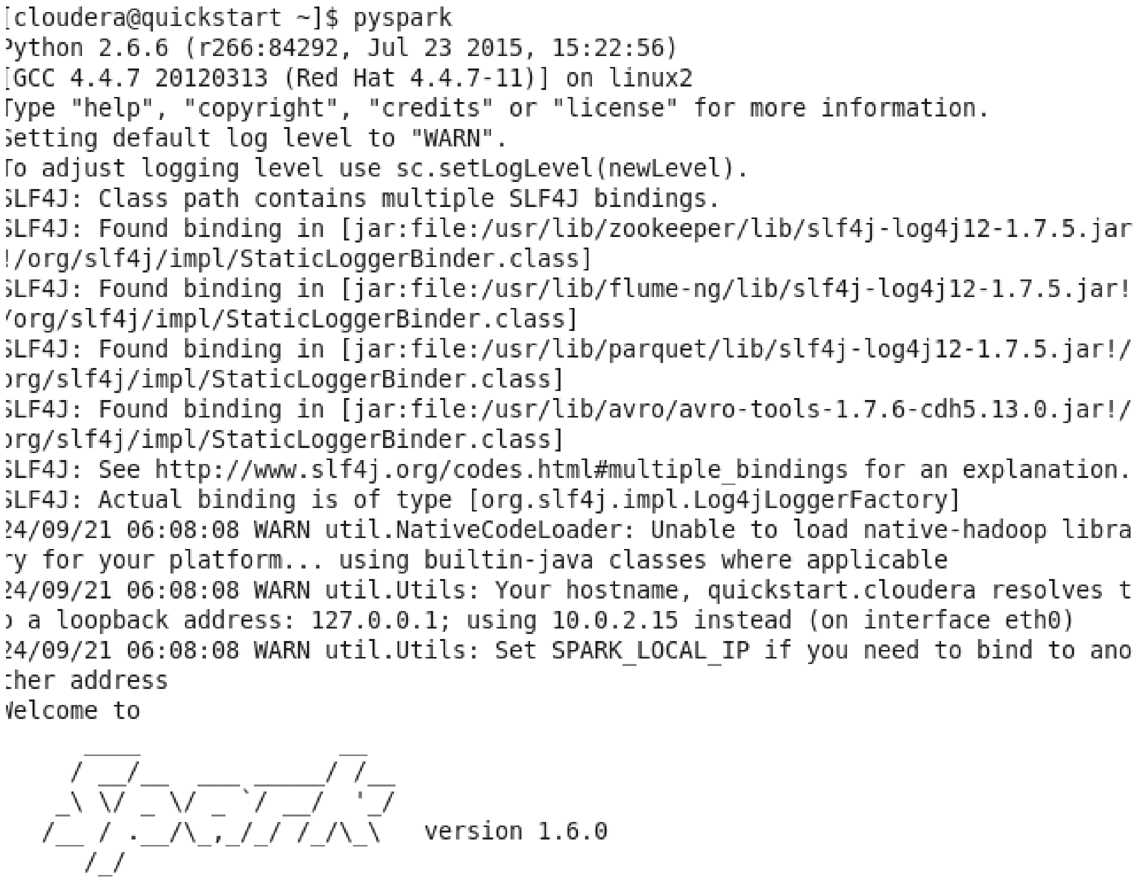
AIM : BASIC SPARK COMMANDS

1. Move the data into Hadoop file system



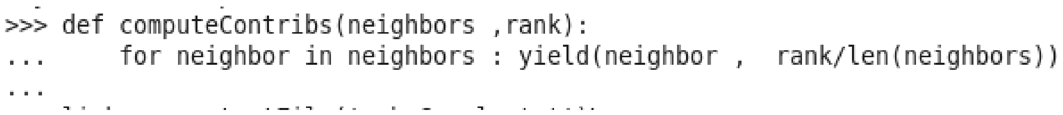
```
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 3 items
drwxr-xr-x   - cloudera cloudera          0 2024-07-26 07:14 DemoDirectory
drwxr-xr-x   - cloudera cloudera          0 2024-09-06 08:58 Demodirector
```

2. Start pyspark in terminal



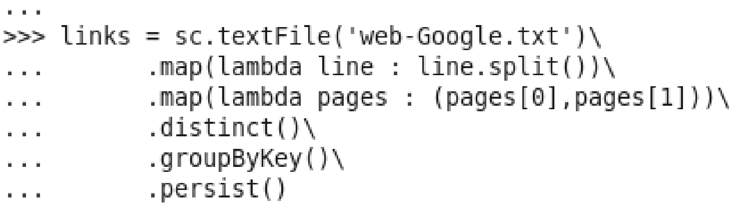
```
[cloudera@quickstart ~]$ pyspark
Python 2.6.6 (r266:84292, Jul 23 2015, 15:22:56)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/parquet/lib/slf4j-log4j12-1.7.5.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/avro/avro-tools-1.7.6-cdh5.13.0.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
24/09/21 06:08:08 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
24/09/21 06:08:08 WARN util.Utils: Your hostname, quickstart.cloudera resolves t
o a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth0)
24/09/21 06:08:08 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to ano
ther address
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/
```

3. Writing compute contrib function



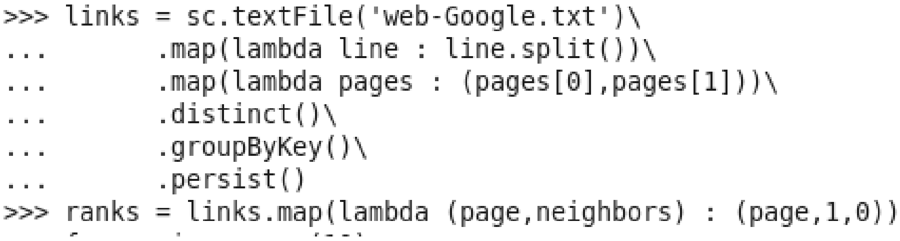
```
>>> def computeContribs(neighbors ,rank):
...     for neighbor in neighbors : yield(neighbor ,  rank/len(neighbors))
...
```

4. Create RDD named links with following command



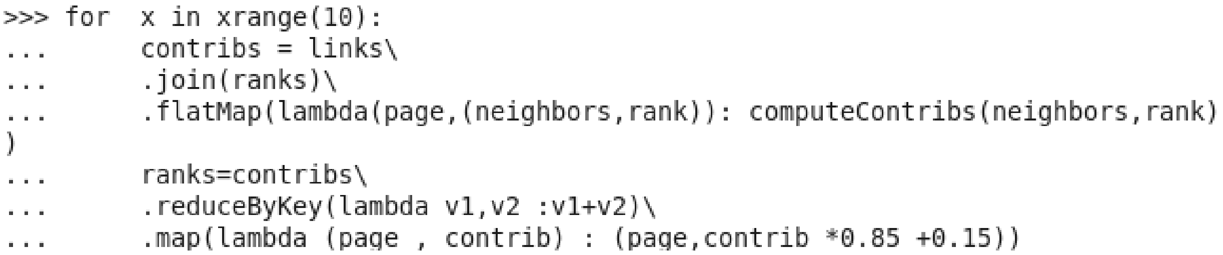
```
...
>>> links = sc.textFile('web-Google.txt')\
...       .map(lambda line : line.split())\
...       .map(lambda pages : (pages[0],pages[1]))\
...       .distinct()\
...       .groupByKey()\
...       .persist()
```

5. Create ranks rdd storing the ranks data



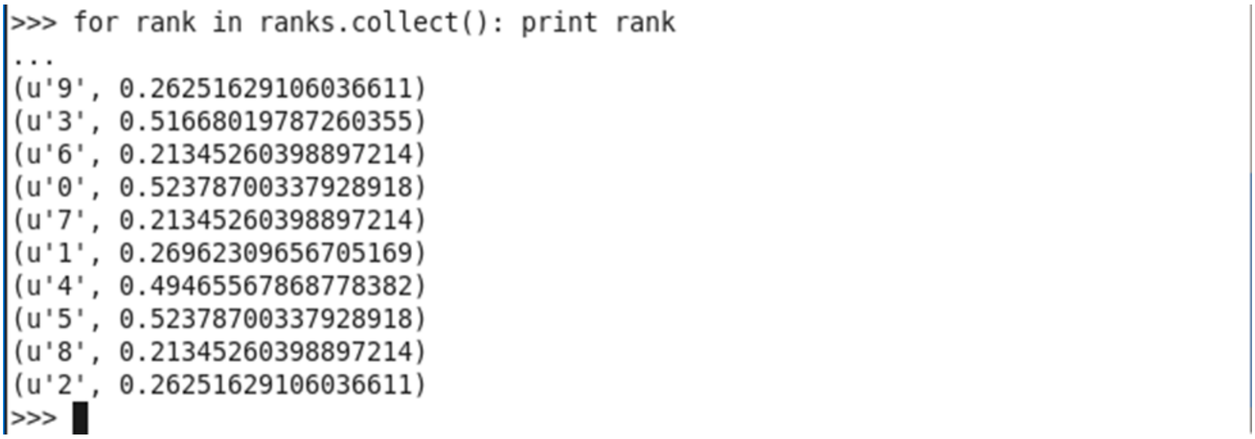
```
>>> links = sc.textFile('web-Google.txt')\
...       .map(lambda line : line.split())\
...       .map(lambda pages : (pages[0],pages[1]))\
...       .distinct()\
...       .groupByKey()\
...       .persist()
>>> ranks = links.map(lambda (page,neighbors) : (page,1,0))
```

6. Create a loop in order to calculate contribs and ranks



```
>>> for  x in xrange(10):
...      contribs = links\
...      .join(ranks)\
...      .flatMap(lambda(page,(neighbors,rank)): computeContribs(neighbors,rank)
)
...      ranks=contribs\
...      .reduceByKey(lambda v1,v2 :v1+v2)\
...      .map(lambda (page , contrib) : (page,contrib *0.85 +0.15))
```

7. Code to collect all ranks



```
>>> for rank in ranks.collect(): print rank
...
(u'9', 0.26251629106036611)
(u'3', 0.51668019787260355)
(u'6', 0.21345260398897214)
(u'0', 0.52378700337928918)
(u'7', 0.21345260398897214)
(u'1', 0.26962309656705169)
(u'4', 0.49465567868778382)
(u'5', 0.52378700337928918)
(u'8', 0.21345260398897214)
(u'2', 0.26251629106036611)
>>> █
```

Practical No : 7

AIM : To Demonstrate Visualization Using Tableau

1. Create an Account On Tableau Public

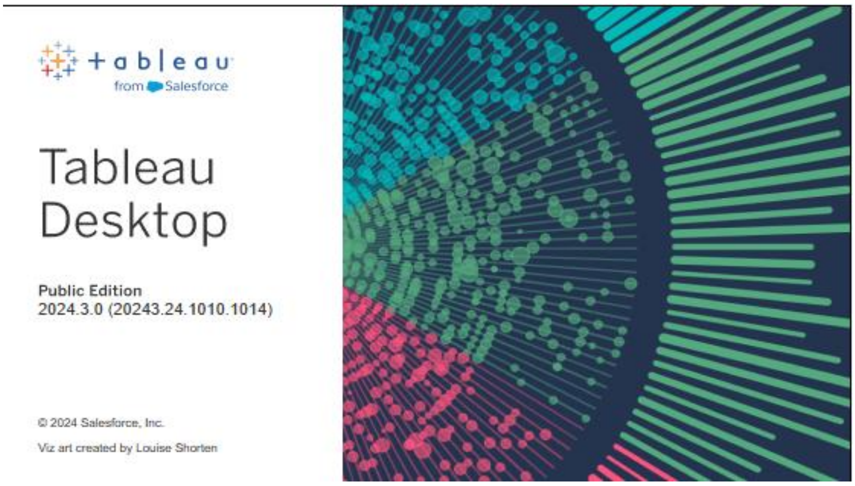


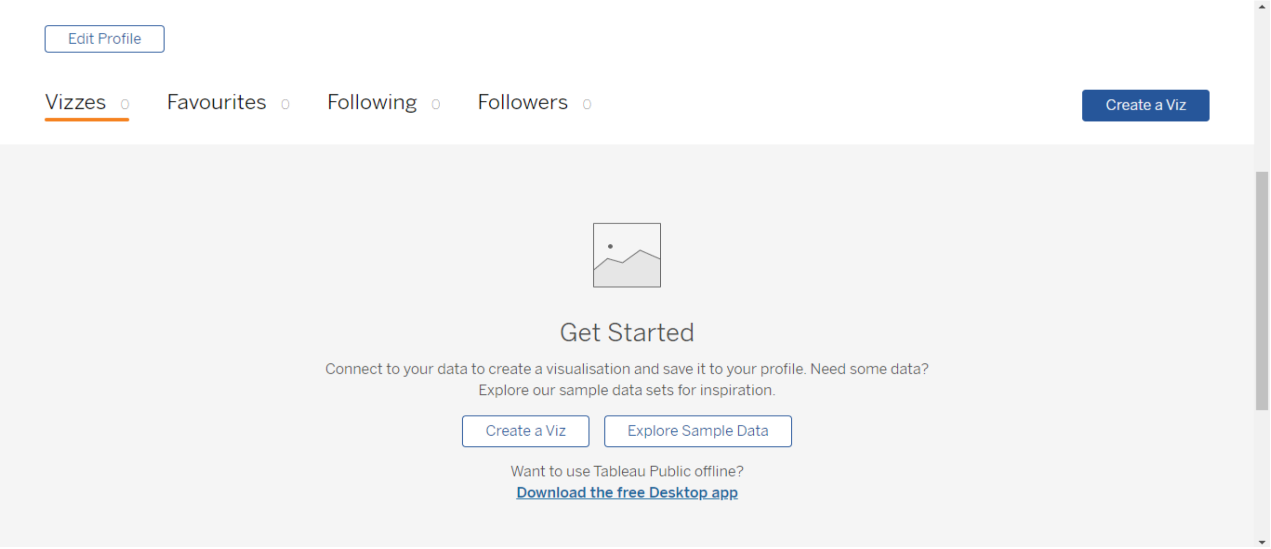
Tableau Desktop

Public Edition
2024.3.0 (20243.24.1010.1014)

© 2024 Salesforce, Inc.

Viz art created by Louise Shorten

2. Create VIZ

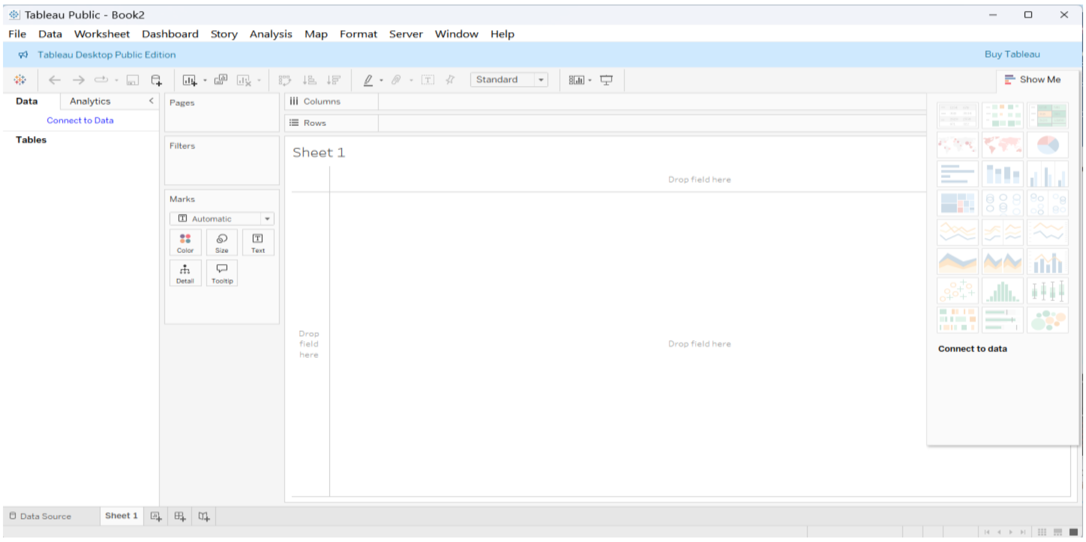


Get Started

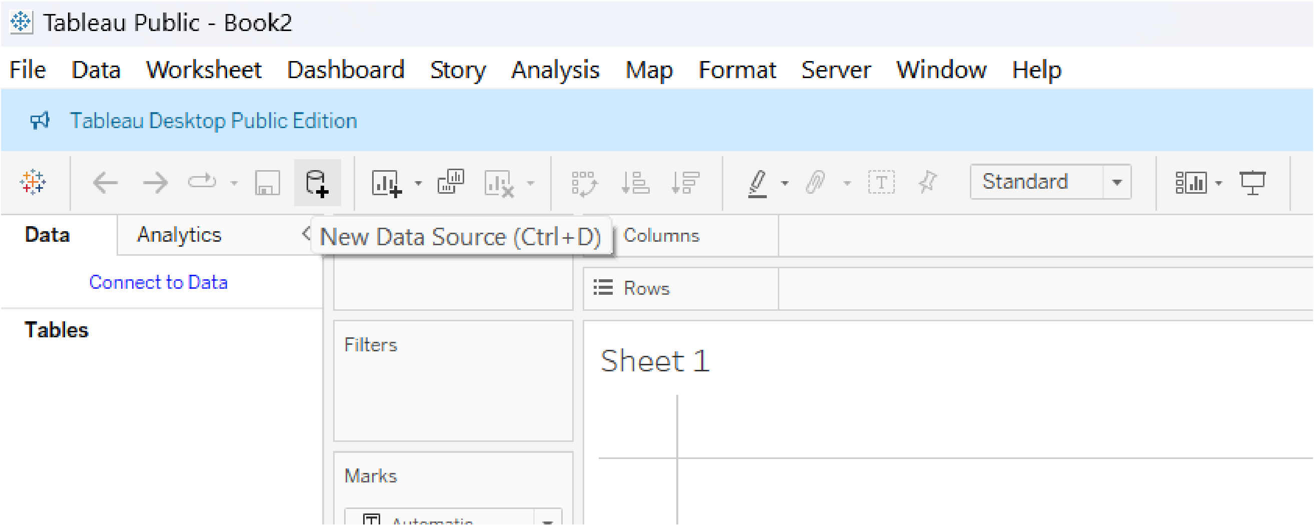
Connect to your data to create a visualisation and save it to your profile. Need some data? Explore our sample data sets for inspiration.

Want to use Tableau Public offline?

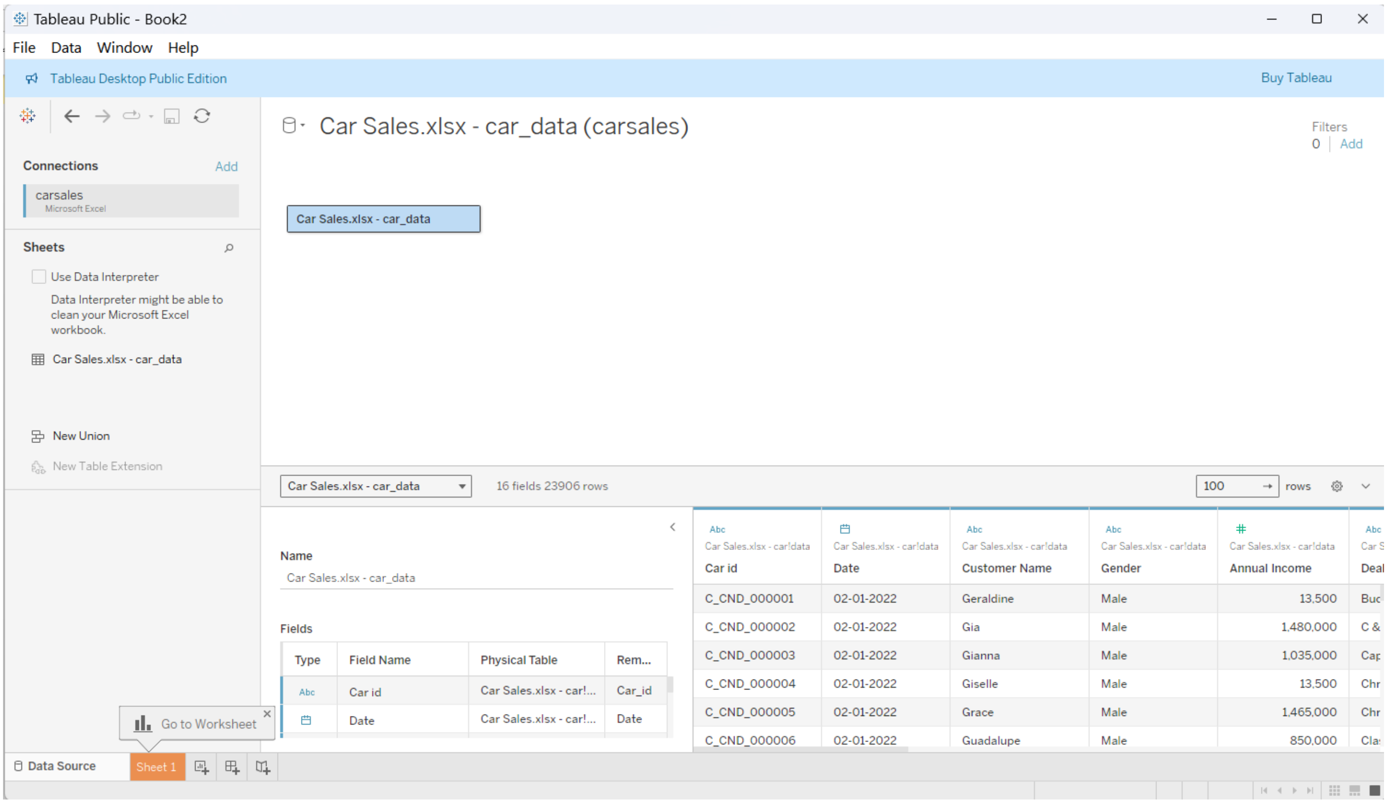
Download the free Desktop app



3. Click on the icon Shown below



4. Now Drag and Drop the file you want to use



Name

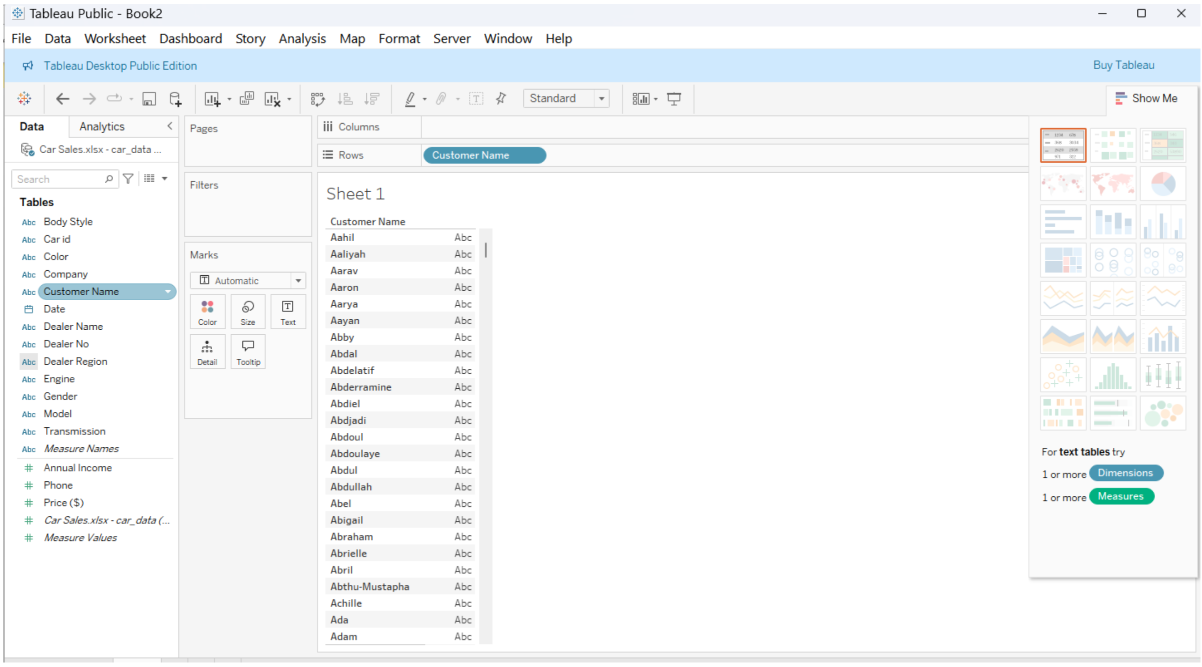
Car Sales.xlsx - car_data

Fields

Type	Field Name	Physical Table	Rem...
Abc	Car id	Car Sales.xlsx - car!...	Car_id
▦	Date	Car Sales.xlsx - car!...	Date

Car id	Date	Customer Name	Gender	Annual Income	Deal
C_CND_000001	02-01-2022	Geraldine	Male	13,500	Buc
C_CND_000002	02-01-2022	Gia	Male	1,480,000	C &
C_CND_000003	02-01-2022	Gianna	Male	1,035,000	Cap
C_CND_000004	02-01-2022	Giselle	Male	13,500	Chr
C_CND_000005	02-01-2022	Grace	Male	1,465,000	Chr
C_CND_000006	02-01-2022	Guadalupe	Male	850,000	Cla

5. Drag and Drop any field you want to use on row



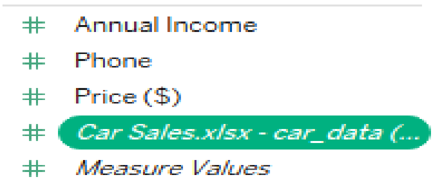
Columns

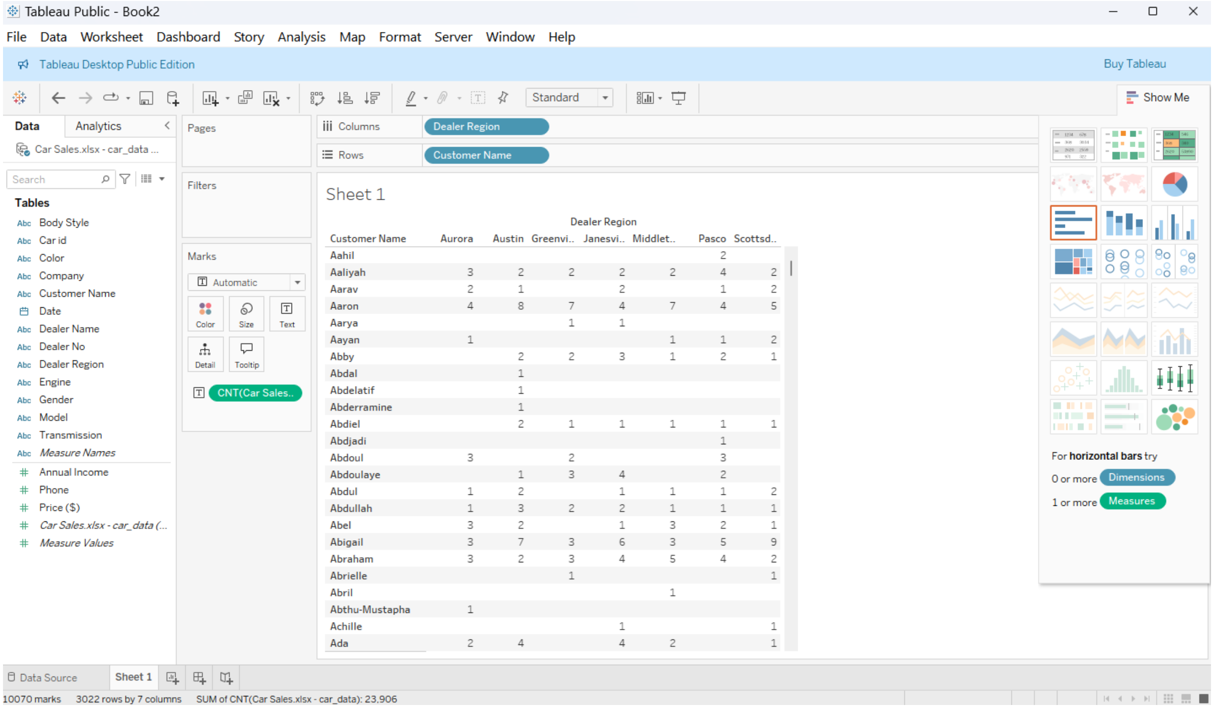
Rows: Customer Name

Sheet 1

Customer Name	
Aahil	Abc
Aaliyah	Abc
Aarav	Abc
Aaron	Abc
Aarya	Abc
Aayan	Abc
Abby	Abc
Abdal	Abc
Abdelatif	Abc
Abderramine	Abc
Abdiel	Abc
Abdjadi	Abc
Abdoul	Abc
Abdoulaye	Abc
Abdul	Abc
Abdullah	Abc
Abel	Abc
Abigail	Abc
Abraham	Abc
Abrielle	Abc
Abril	Abc
Abthu-Mustapha	Abc
Achille	Abc
Ada	Abc
Adam	Abc

6. Drag and Drop csv file onto data part





Columns: Dealer Region

Rows: Customer Name

Sheet 1

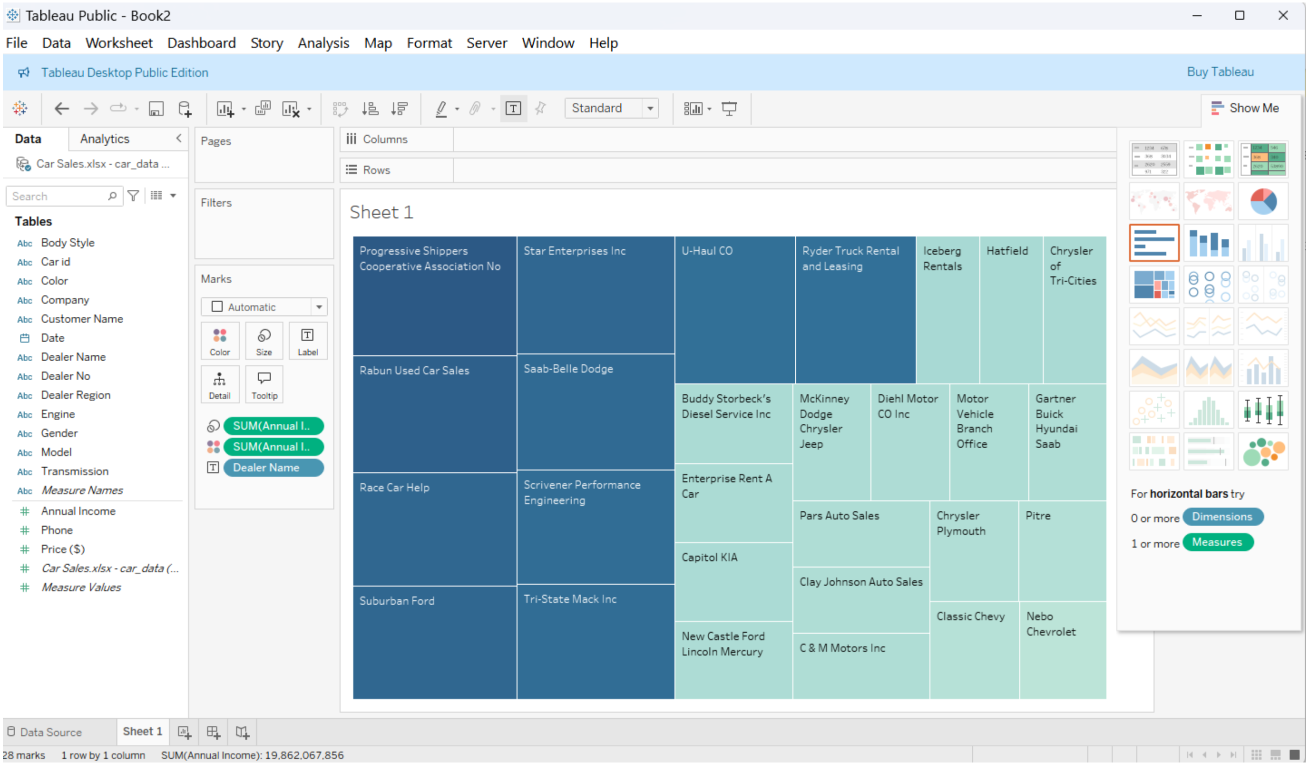
	Dealer Region						
Customer Name	Aurora	Austin	Greenvi..	Janesvi..	Middlet..	Pasco	Scottsd..
Aahil						2	
Aaliyah	3	2	2	2	2	4	2
Aarav	2	1		2		1	2
Aaron	4	8	7	4	7	4	5
Aarya			1	1			
Aayan	1				1	1	2
Abby		2	2	3	1	2	1
Abdal		1					
Abdelatif		1					
Abderramine		1					
Abdiel		2	1	1	1	1	1
Abdjadi						1	
Abdoul	3		2			3	
Abdoulaye		1	3	4		2	
Abdul	1	2		1	1	1	2
Abdullah	1	3	2	2	1	1	1
Abel	3	2		1	3	2	1
Abigail	3	7	3	6	3	5	9
Abraham	3	2	3	4	5	4	2
Abrielle			1				1
Abril					1		
Abthu-Mustapha	1						
Achille				1			1
Ada	2	4		4	2		1

10070 marks 3022 rows by 7 columns SUM of CNT(Car Sales.xlsx - car_data): 23,906

Atharva Nalawadee ROLL NO 30

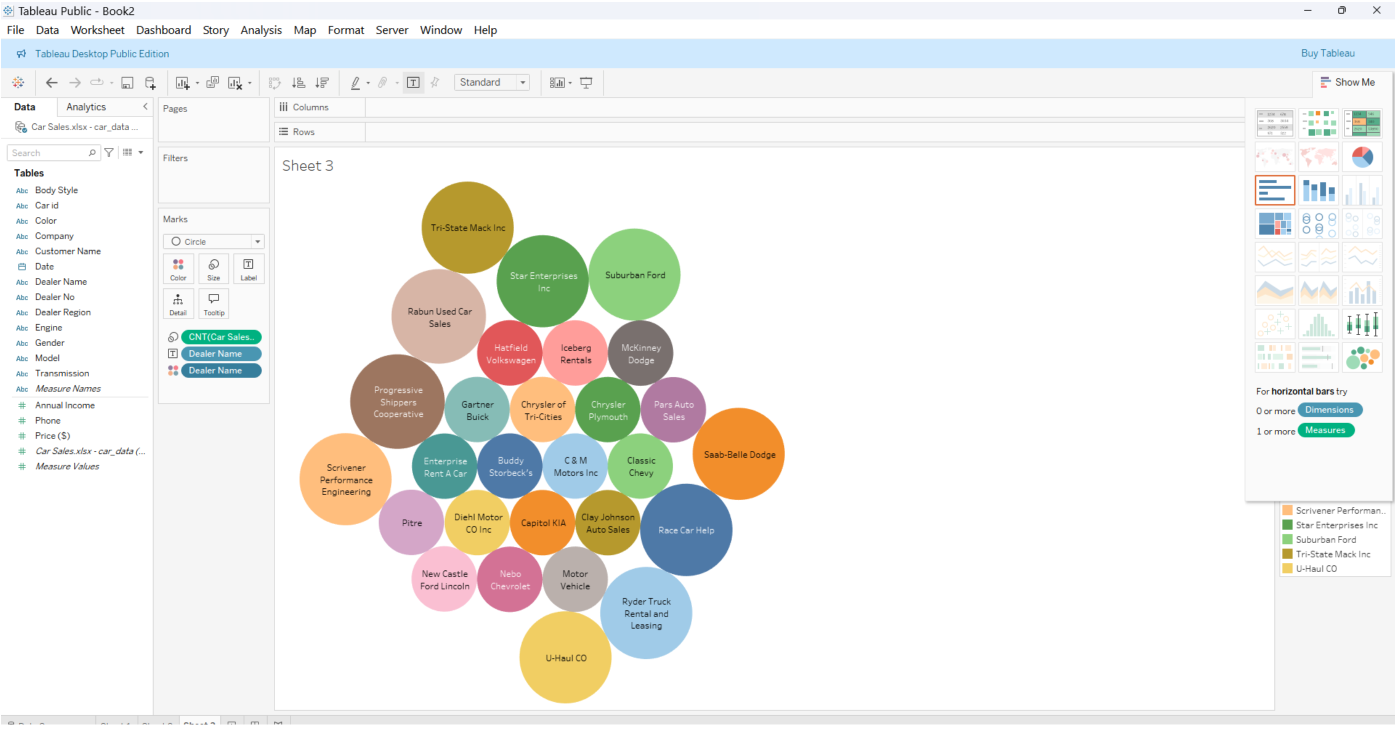
7. Analysing Using Charts

◆ TREEMAP



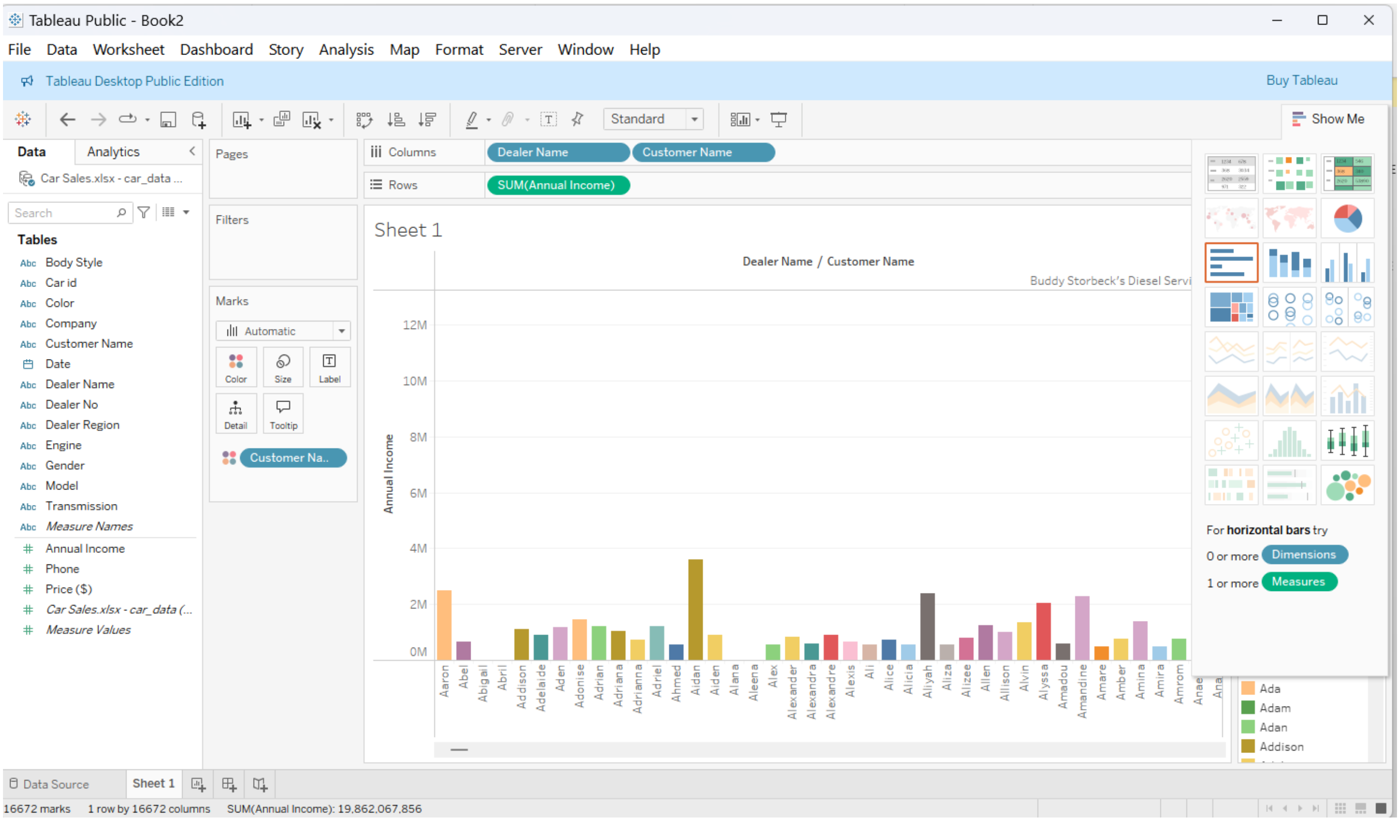
28 marks 1 row by 1 column SUM(Annual Income): 19,862,067,856

◆ PACKED BUBBLES



Atharva Nalawadee ROLL NO 30

◆ SIDE-BY-SIDE BARS



Columns: Dealer Name, Customer Name

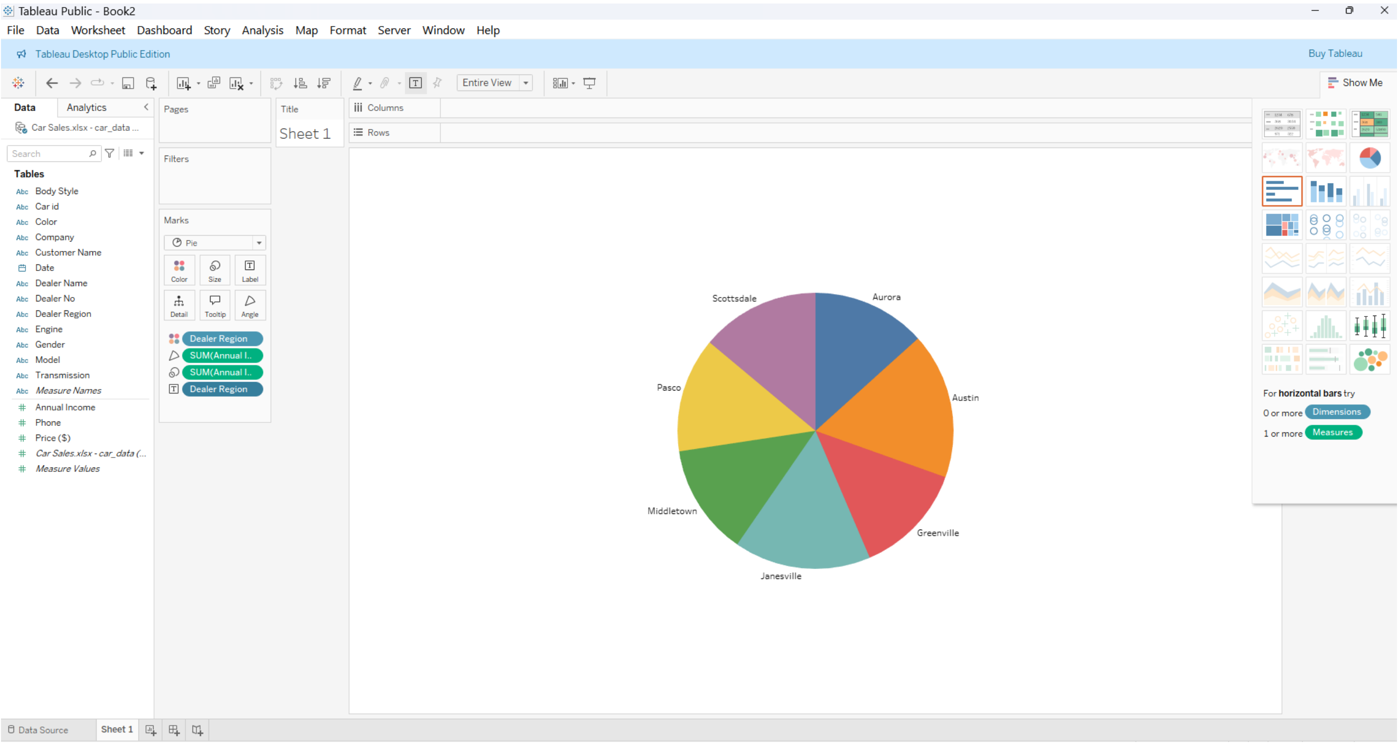
Rows: SUM(Annual Income)

Marks: Automatic; Color: Customer Na..

16672 marks 1 row by 16672 columns SUM(Annual Income): 19,862,067,856

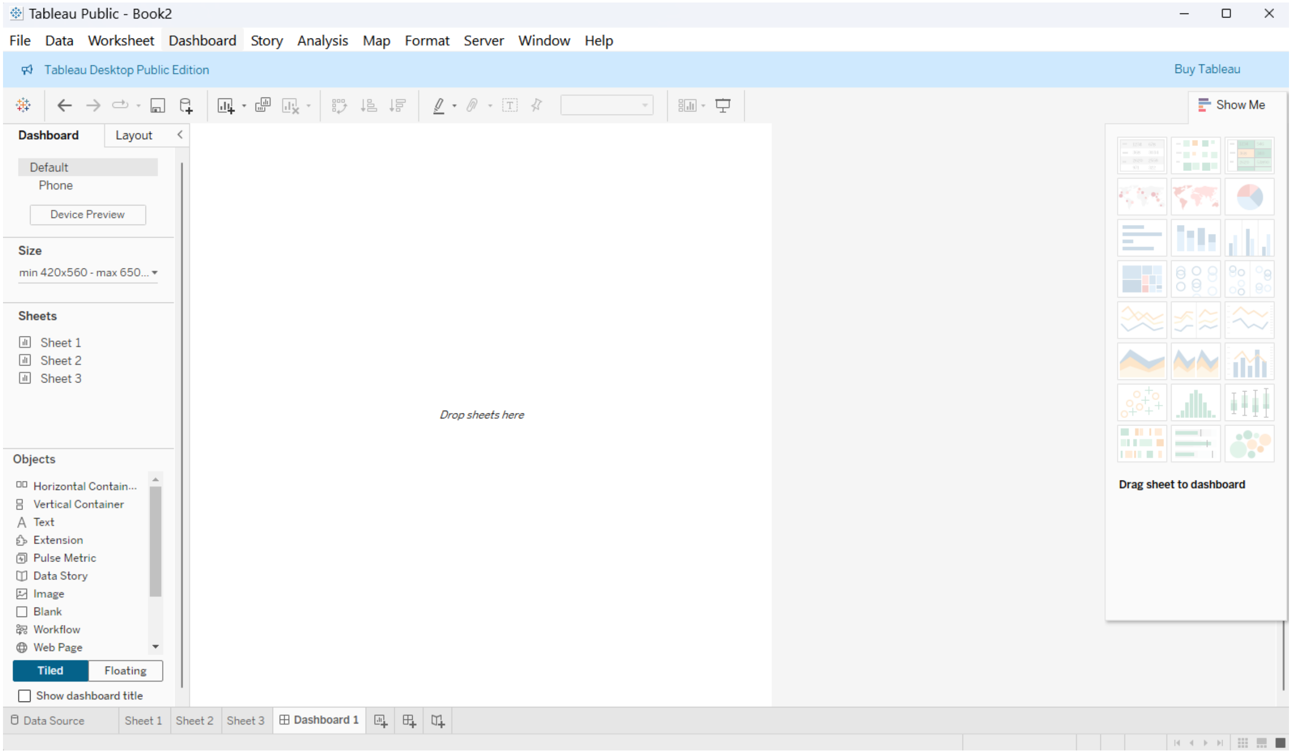
Tableau Public - Book2
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Tableau Desktop Public Edition
Buy Tableau
Standard
Show Me
Data Analytics
Car Sales.xlsx - car_data ...
Pages
Filters
Marks
Tables
Body Style
Car id
Color
Company
Customer Name
Date
Dealer Name
Dealer No
Dealer Region
Engine
Gender
Model
Transmission
Measure Names
Annual Income
Phone
Price ($)
Car Sales.xlsx - car_data (...
Measure Values
Color Size Label Detail Tooltip
Sheet 1
Dealer Name / Customer Name
Buddy Storbeck's Diesel Servi
Annual Income
12M
10M
8M
6M
4M
2M
0M
Aaron Abel Abigail Abril Addison Adelaide Aden Adonise Adrian Adriana Adrianna Adriel Ahmed Aidan Aiden Alana Aleena Alex Alexander Alexandra Alexandre Alexis Ali Alice Alicia Aliyah Aliza Alizee Allen Allison Alvin Alyssa Amadou Amandine Amare Amber Amina Amira Amrom Anae Ana
For horizontal bars try
0 or more Dimensions
1 or more Measures
Ada
Adam
Adan
Addison
Data Source
Sheet 1

◆ PIE CHARTS

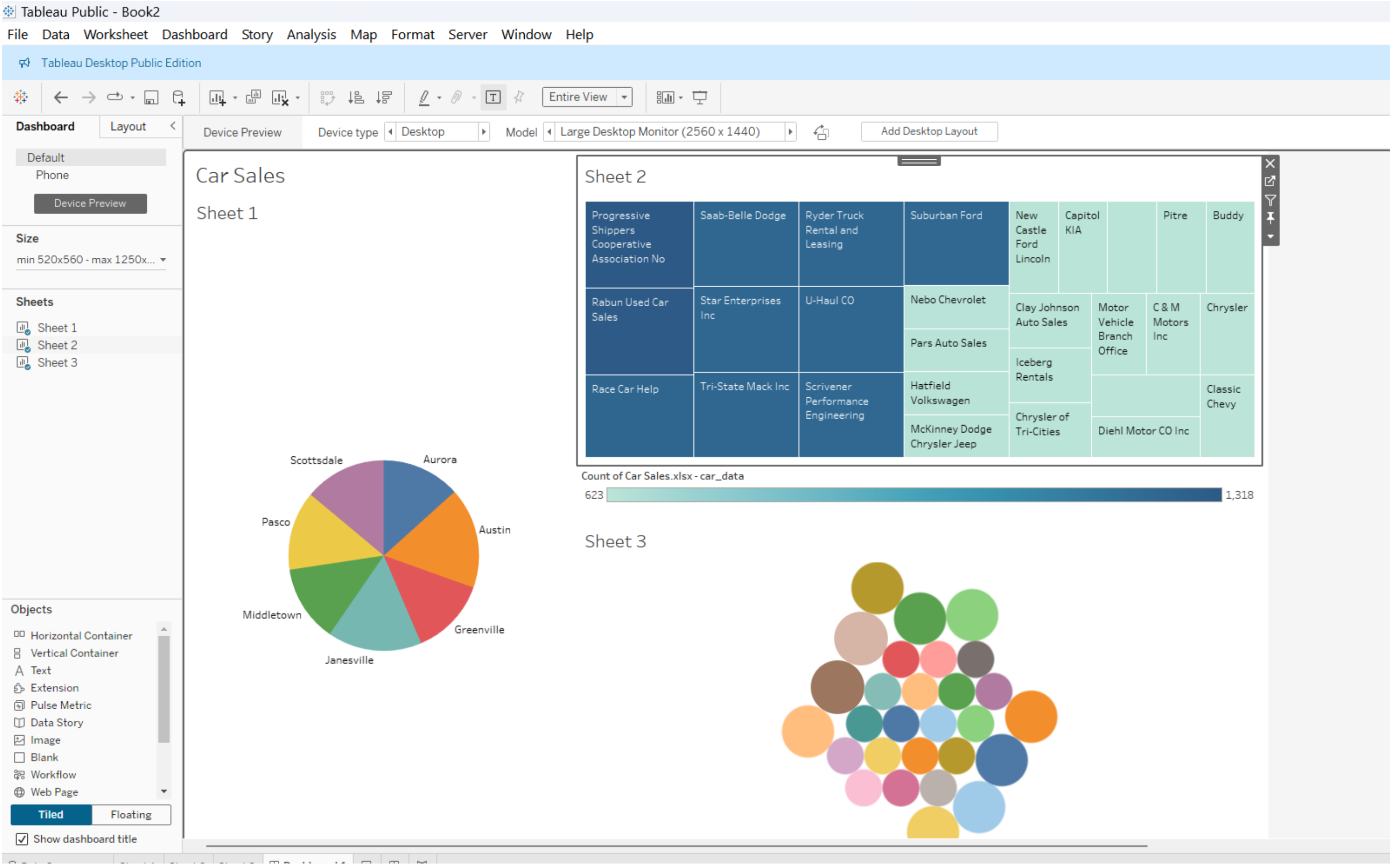


Marks: Pie; Color: Dealer Region; Angle: SUM(Annual I..; Size: SUM(Annual I..; Label: Dealer Region

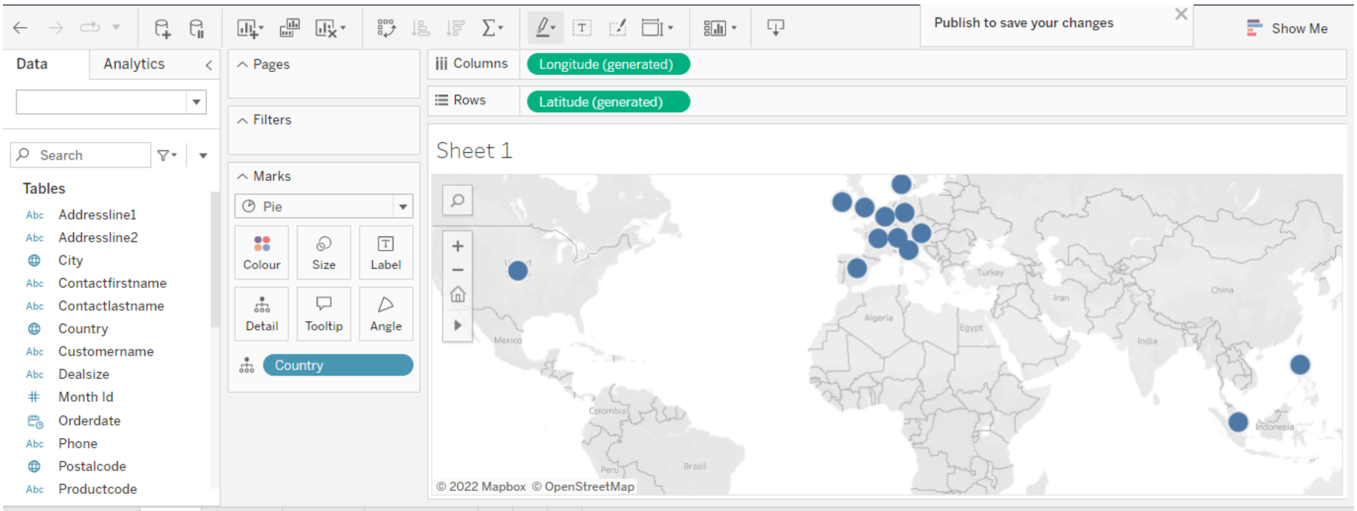
8. Creating Dashboard



- **Drag and drop your sheet at your left side at main working space**



9. Working with Maps



Columns: Longitude (generated)

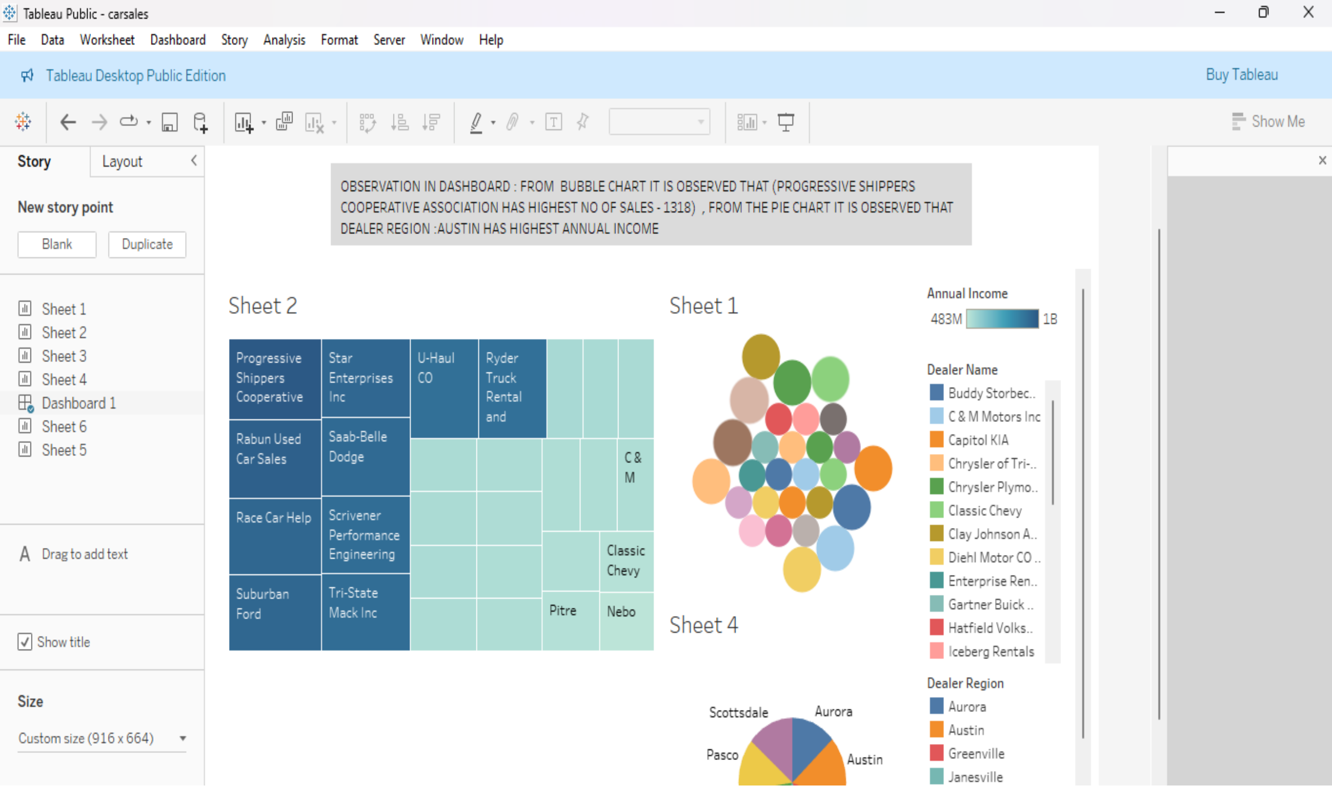
Rows: Latitude (generated)

Marks: Pie

Detail: Country

10. Telling Stories With Tableau

•ANALYSIS OF DASHBOARD



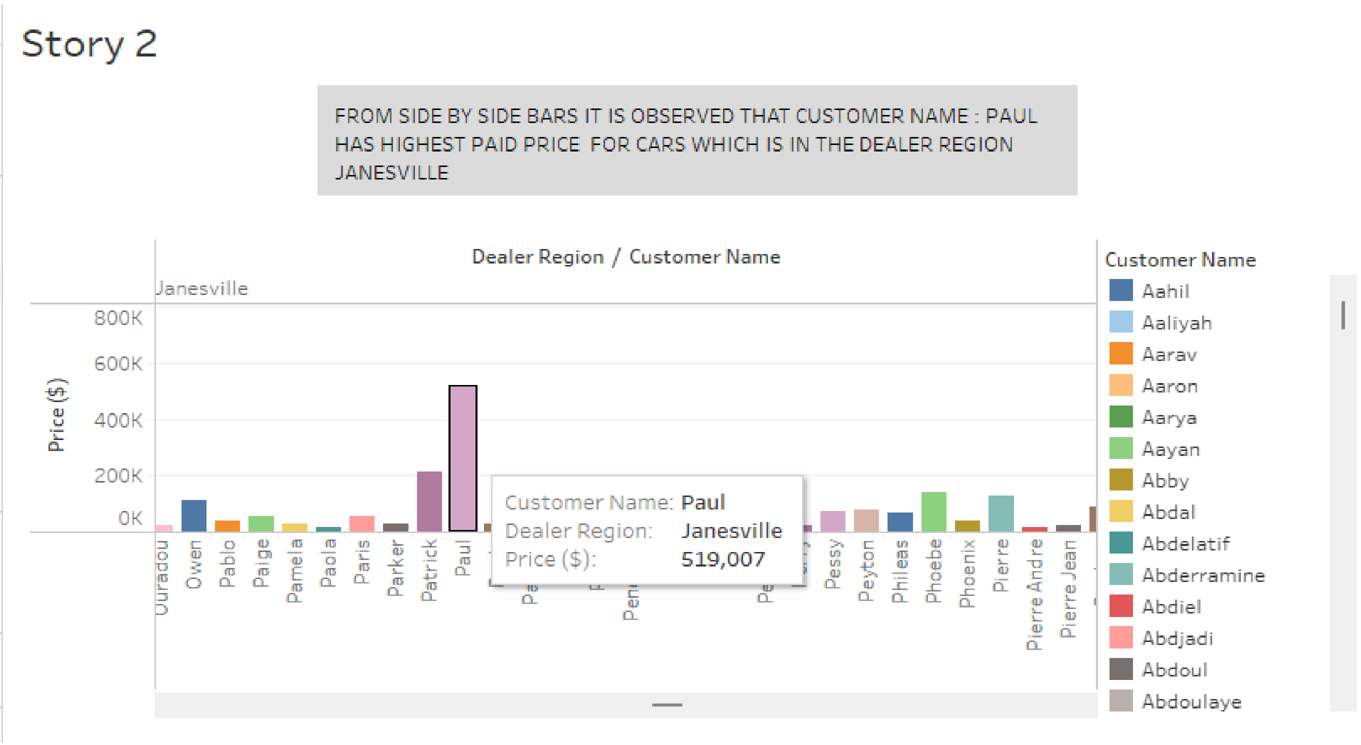
OBSERVATION IN DASHBOARD : FROM BUBBLE CHART IT IS OBSERVED THAT (PROGRESSIVE SHIPPERS COOPERATIVE ASSOCIATION HAS HIGHEST NO OF SALES - 1318) , FROM THE PIE CHART IT IS OBSERVED THAT DEALER REGION :AUSTIN HAS HIGHEST ANNUAL INCOME

Size: Custom size (916 x 664)

☑ Show title

Atharva Nalawadee ROLL NO 30

- **CUSTOMER NAME PAID HIGHEST FOR CARS**

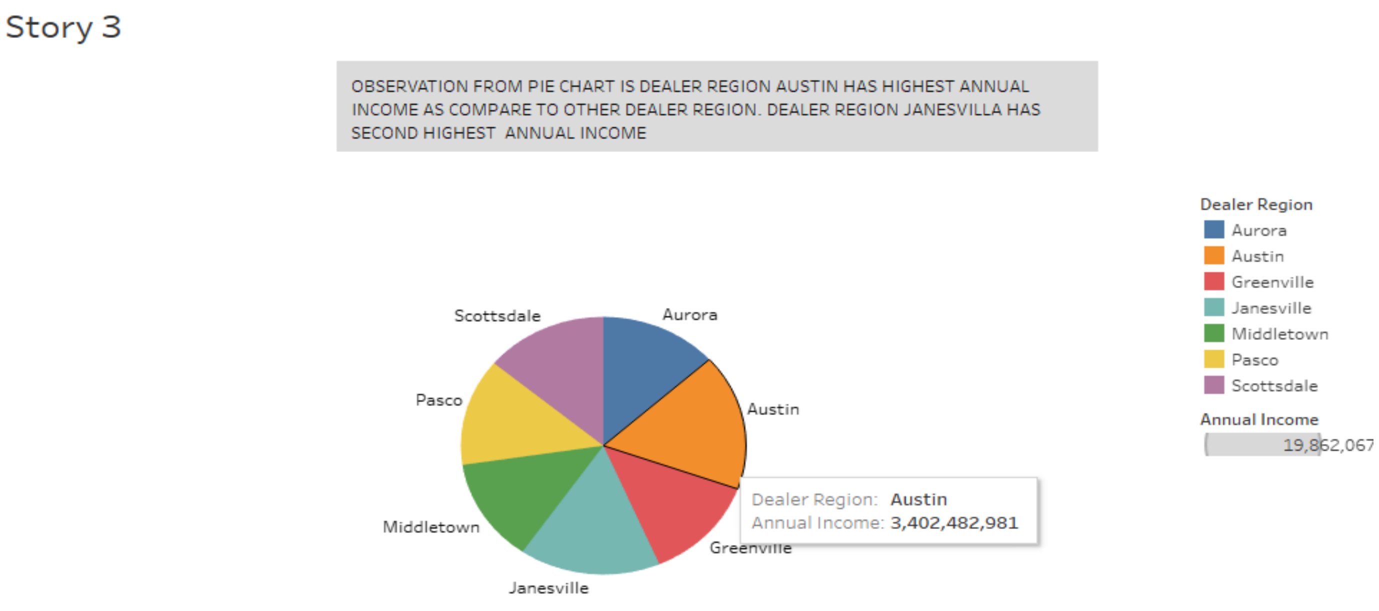


Story 2

FROM SIDE BY SIDE BARS IT IS OBSERVED THAT CUSTOMER NAME : PAUL HAS HIGHEST PAID PRICE FOR CARS WHICH IS IN THE DEALER REGION JANESVILLE

Customer Name: Paul
Dealer Region: Janesville
Price ($): 519,007

- **DEALER REGION HAVING HIGHEST ANNUAL INCOME**

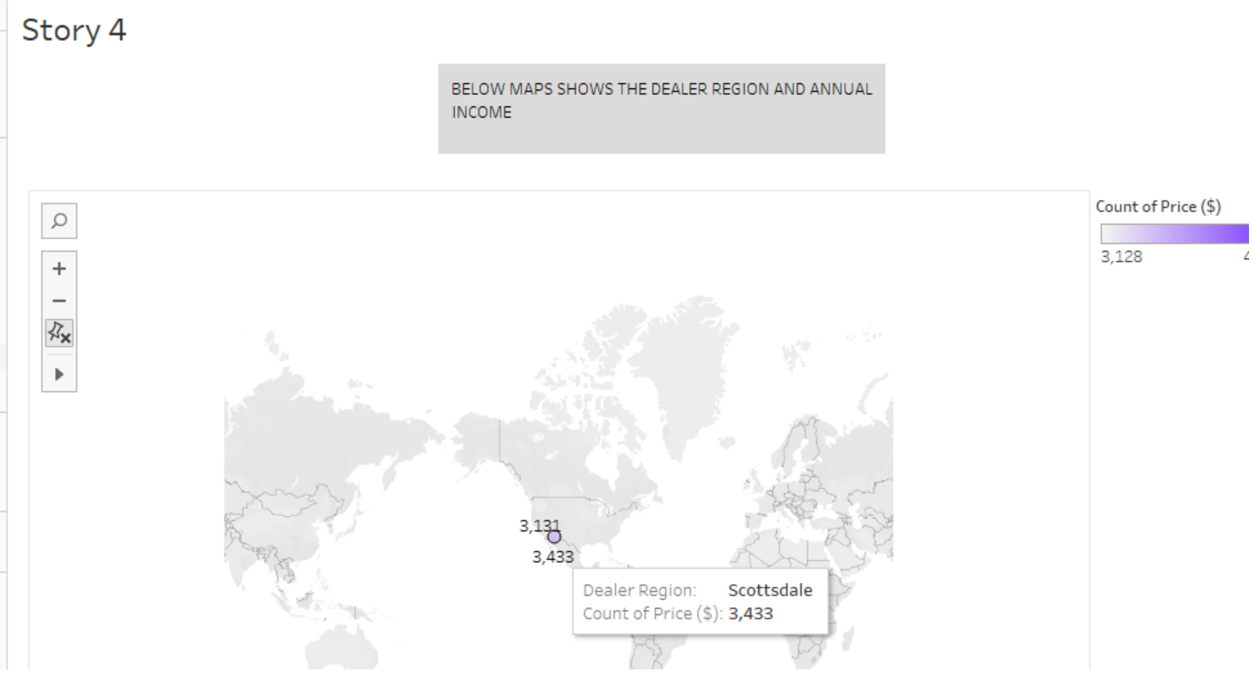


Story 3

OBSERVATION FROM PIE CHART IS DEALER REGION AUSTIN HAS HIGHEST ANNUAL INCOME AS COMPARE TO OTHER DEALER REGION. DEALER REGION JANESVILLA HAS SECOND HIGHEST ANNUAL INCOME

Dealer Region: Austin
Annual Income: 3,402,482,981

- **MAP OBSERVATION HAVING DEALER REGION AND ANNUAL INCOME**



Story 4

BELOW MAPS SHOWS THE DEALER REGION AND ANNUAL INCOME