**BIG DATA LAB-06**

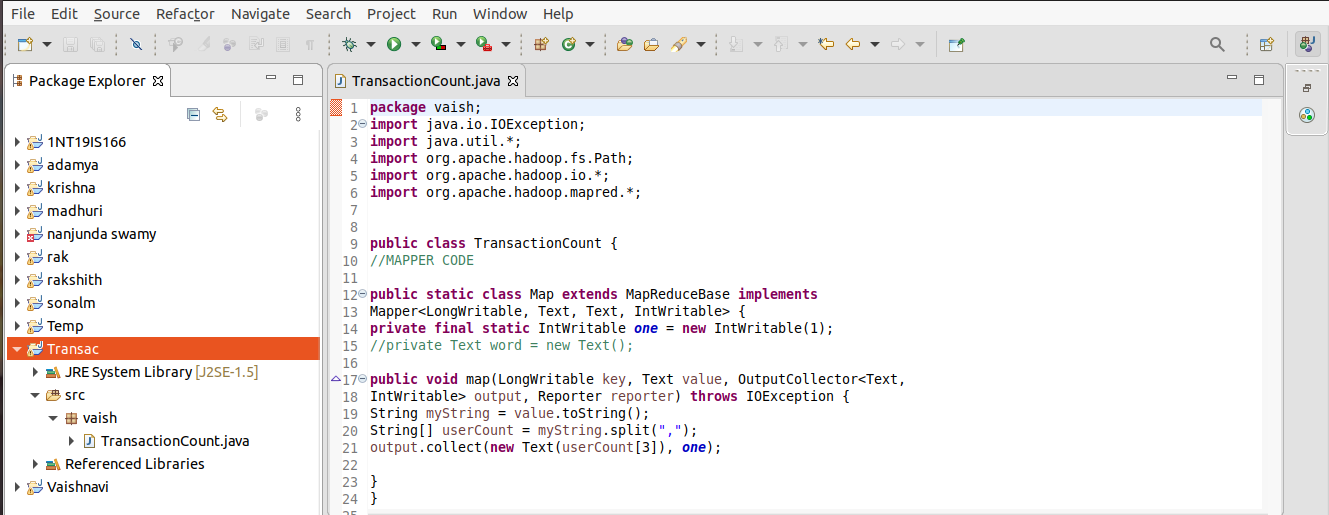
**PROGRAM-04**

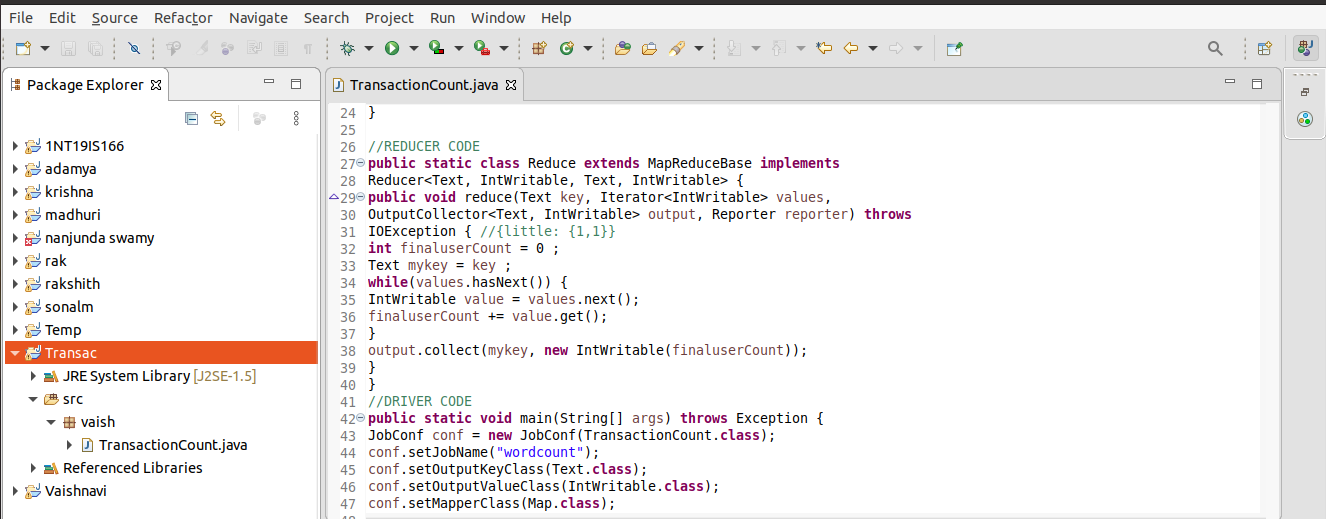
**Use the Hadoop framework to write a MapReduce program to read a .csv file into a single node Hadoop containing Sl. CARD UserName Amount Implement**

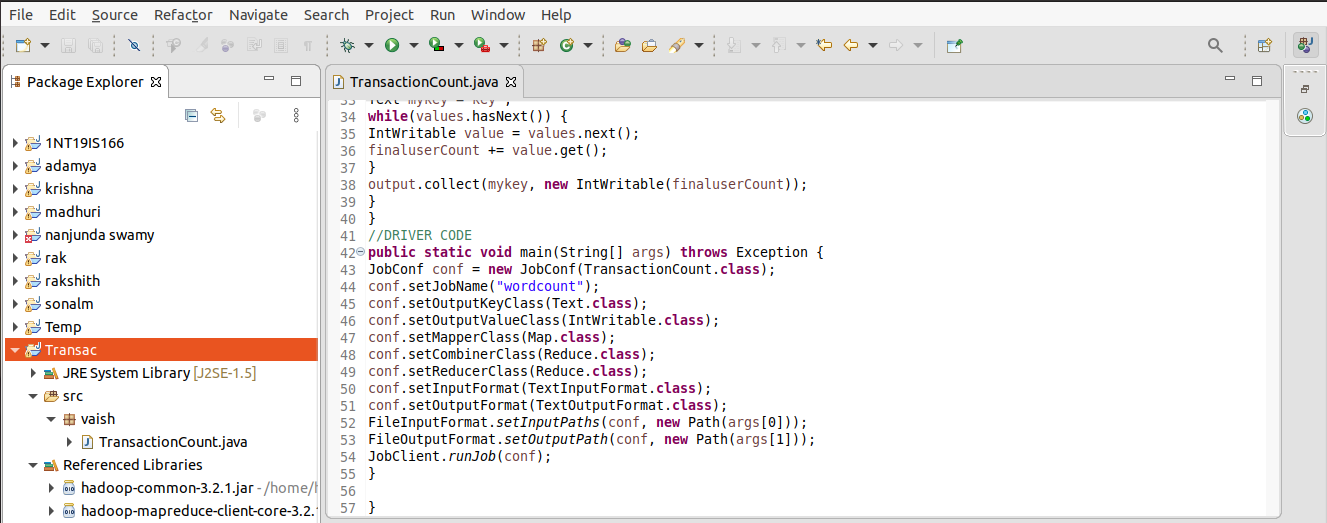
**1. Count the Number of transactions done by each user**

**2. Find the total amount of money transacted by each user**

package vaish;  
import java.io.IOException;  
import java.util.\*;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.\*;  
import org.apache.hadoop.mapred.\*;  
  
  
public class TransactionCount {  
//MAPPER CODE  
  
public static class Map extends MapReduceBase implements  
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, OutputCollector<Text,  
IntWritable> output, Reporter reporter) throws IOException {  
String myString = value.toString();  
String[] userCount = myString.split(",");  
output.collect(new Text(userCount[3]), one);  
  
}  
}  
  
//REDUCER CODE  
public static class Reduce extends MapReduceBase implements  
Reducer<Text, IntWritable, Text, IntWritable> {  
public void reduce(Text key, Iterator<IntWritable> values,  
OutputCollector<Text, IntWritable> output, Reporter reporter) throws  
IOException { //{little: {1,1}}  
int finaluserCount = 0 ;  
Text mykey = key ;  
while(values.hasNext()) {  
IntWritable value = values.next();  
finaluserCount += value.get();  
}  
output.collect(mykey, new IntWritable(finaluserCount));  
}  
}  
//DRIVER CODE  
public static void main(String[] args) throws Exception {  
JobConf conf = new JobConf(TransactionCount.class);  
conf.setJobName("wordcount");  
conf.setOutputKeyClass(Text.class);  
conf.setOutputValueClass(IntWritable.class);  
conf.setMapperClass(Map.class);  
conf.setCombinerClass(Reduce.class);  
conf.setReducerClass(Reduce.class);  
conf.setInputFormat(TextInputFormat.class);  
conf.setOutputFormat(TextOutputFormat.class);  
FileInputFormat.setInputPaths(conf, new Path(args[0]));  
FileOutputFormat.setOutputPath(conf, new Path(args[1]));  
JobClient.runJob(conf);  
}  
  
}







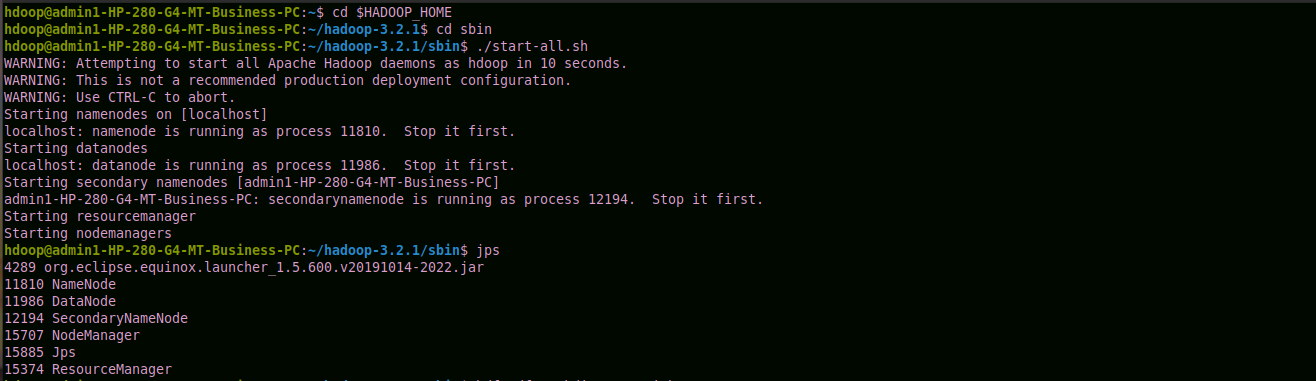
CREATED A JAR FILE WITH NAME

CREATING A CSV FILE

COMMAND PERFORMED ON TEMINAL

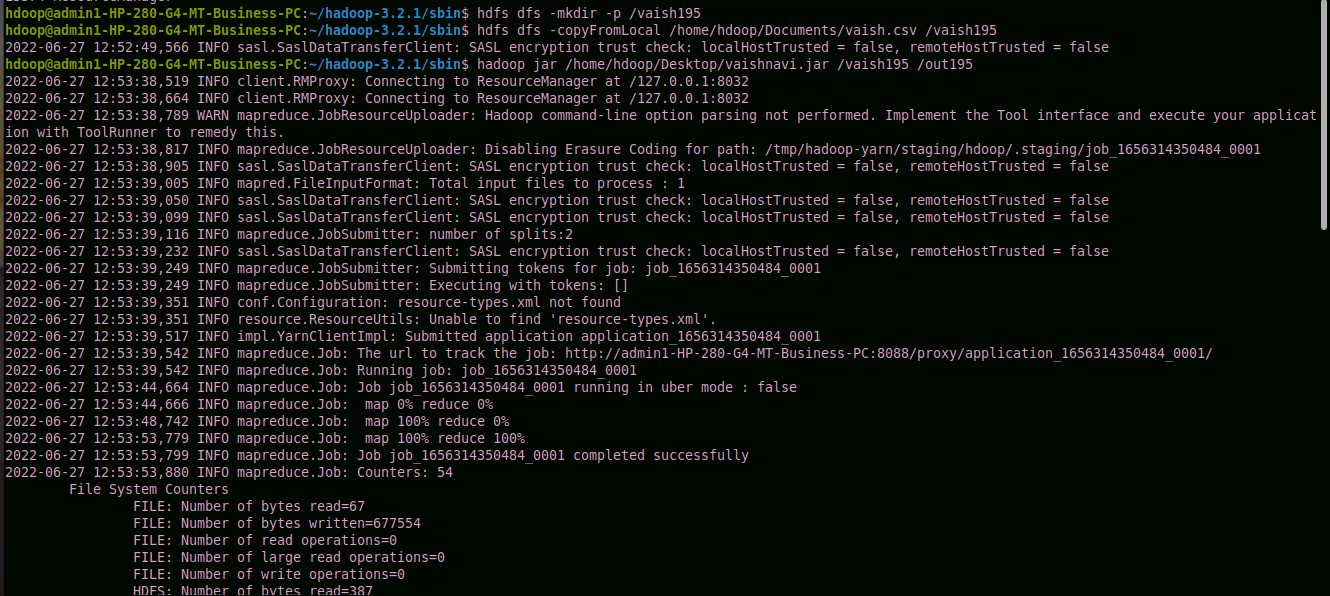
hdoop@admin1-HP-280-G4-MT-Business-PC:~$ cd $HADOOP\_HOME  
hdoop@admin1-HP-280-G4-MT-Business-PC:~/hadoop-3.2.1$ cd sbin  
hdoop@admin1-HP-280-G4-MT-Business-PC:~/hadoop-3.2.1/sbin$ jps

hdoop@admin1-HP-280-G4-MT-Business-PC:~/hadoop-3.2.1/sbin$ start-all.sh

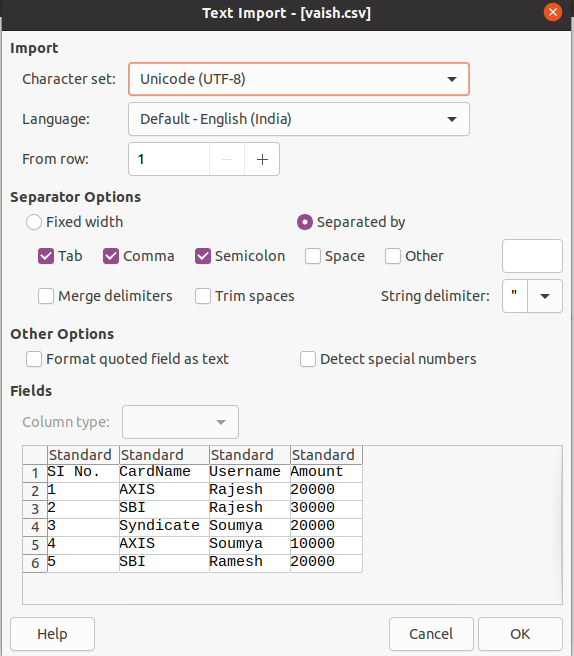


CREATING AN INPUT DIRECTORY

hdfs dfs -mkdir -p ~/vaish195

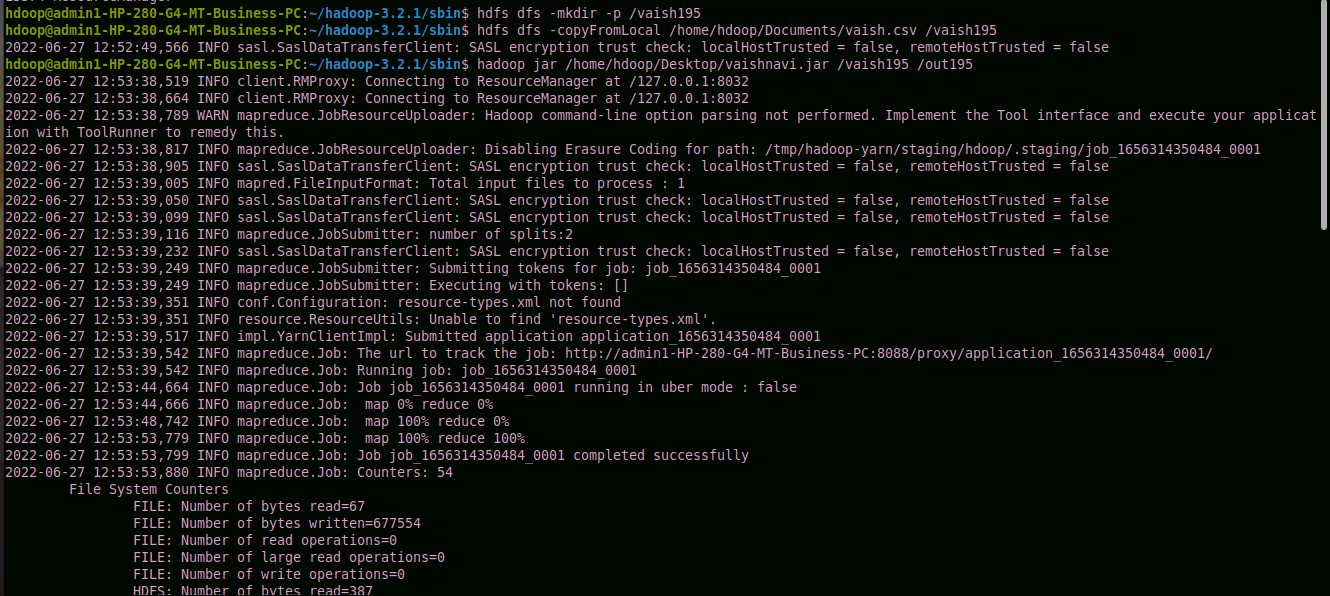


CSV file



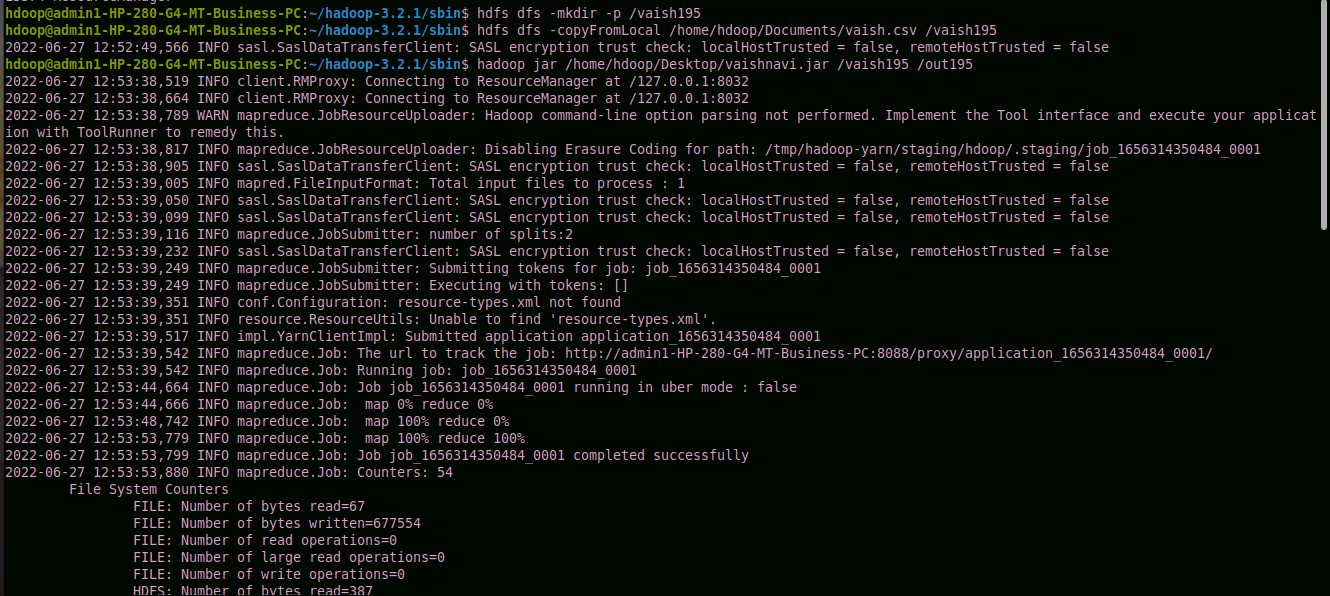
COPYING THE CSV FILE FROM LOCAL

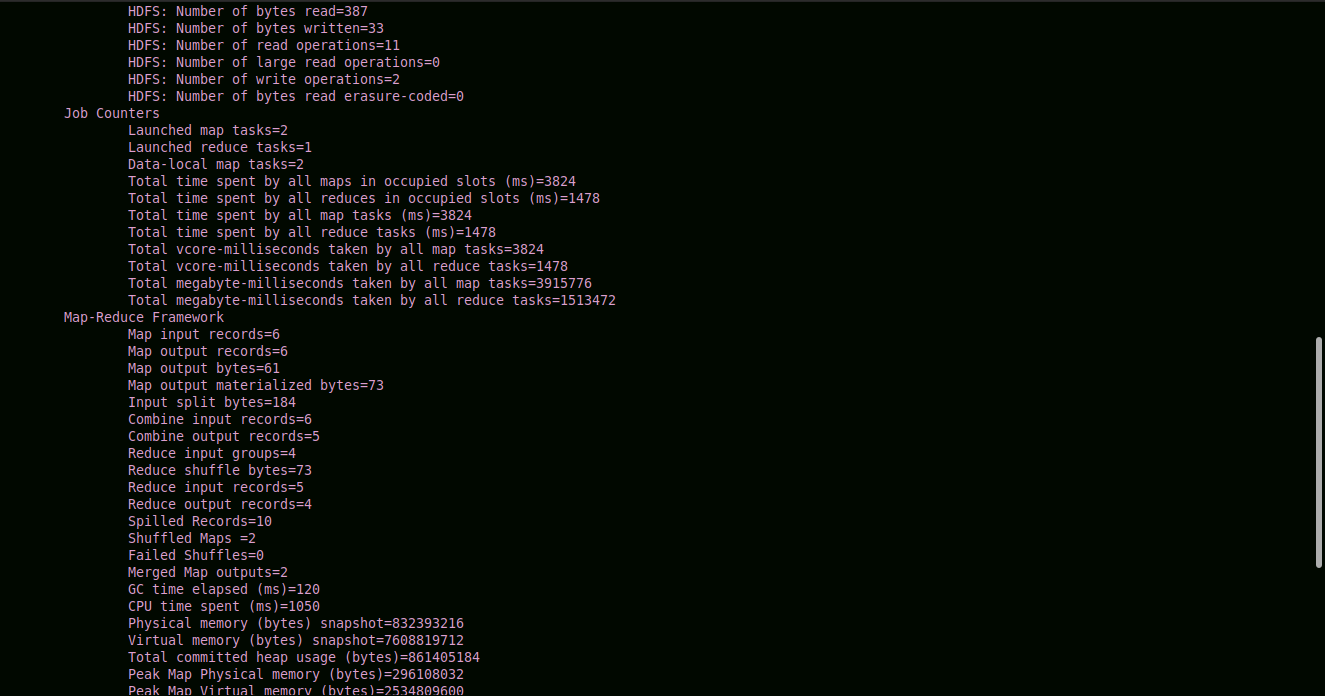
 dfs -copyFromLocal /home/hdoop/Documents/vaish.csv ~/vaish195



ADDING EXTERNAL JARS TO THE PATH

 hadoop jar /home/hdoop/ Documents /vaishnavi.jar ~/vaish195 ~/out195







CAT THE FILE TO GET THE OUTPUT

 hdfs dfs -cat /out195/part\*

OUTPUT

