Script to run the code.

Before starting anything, we will start our hadoop by first navigate to the directory of the hadoop. By using the following prompt in command prompt.

C:\Program_files\hadoop-3.2.1\sbin

Now, we will run the following command:

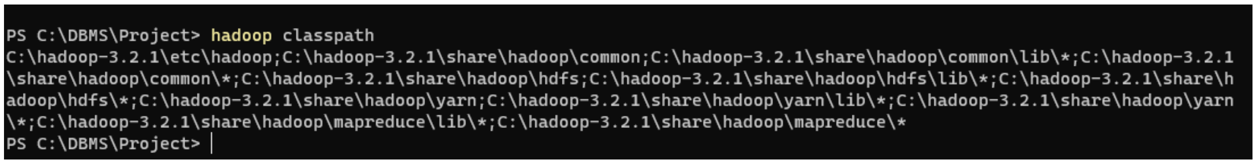
.\start-dfs.cmd

The above command prompt will pop up, one for the name node and the other for the data node.
Next, we will start the yarn service using the following command:

.\start-yarn.cmd

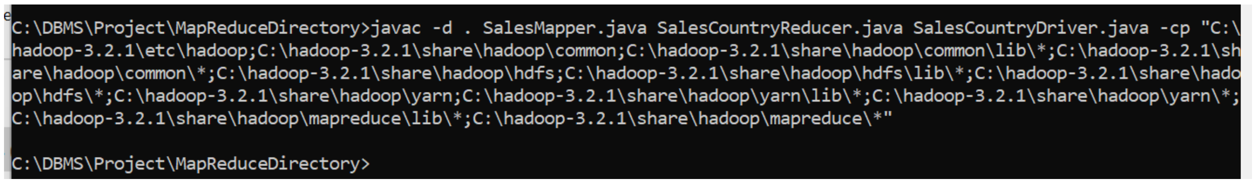
The main problem is divided into three different parts, as mentioned in our report.

We will run the following command to get the hadoop classpath



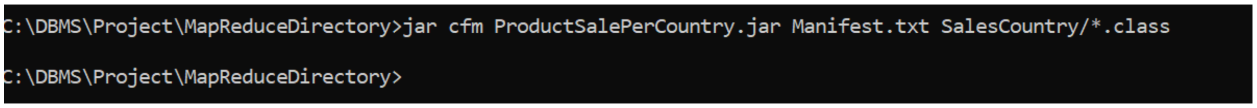
```
PS C:\DBMS\Project> hadoop classpath
C:\hadoop-3.2.1\etc\hadoop;C:\hadoop-3.2.1\share\hadoop\common;C:\hadoop-3.2.1\share\hadoop\common\lib\*;C:\hadoop-3.2.1
\share\hadoop\common\*;C:\hadoop-3.2.1\share\hadoop\hdfs;C:\hadoop-3.2.1\share\hadoop\hdfs\lib\*;C:\hadoop-3.2.1\share\h
adoop\hdfs\*;C:\hadoop-3.2.1\share\hadoop\yarn;C:\hadoop-3.2.1\share\hadoop\yarn\lib\*;C:\hadoop-3.2.1\share\hadoop\yarn
\*;C:\hadoop-3.2.1\share\hadoop\mapreduce\lib\*;C:\hadoop-3.2.1\share\hadoop\mapreduce\*
PS C:\DBMS\Project> |
```

First we will go the directory of our project to compile our java files using following commands.



```
C:\DBMS\Project\MapReduceDirectory>javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java -cp "C:\
hadoop-3.2.1\etc\hadoop;C:\hadoop-3.2.1\share\hadoop\common;C:\hadoop-3.2.1\share\hadoop\common\lib\*;C:\hadoop-3.2.1\sh
are\hadoop\common\*;C:\hadoop-3.2.1\share\hadoop\hdfs;C:\hadoop-3.2.1\share\hadoop\hdfs\lib\*;C:\hadoop-3.2.1\share\hado
op\hdfs\*;C:\hadoop-3.2.1\share\hadoop\yarn;C:\hadoop-3.2.1\share\hadoop\yarn\lib\*;C:\hadoop-3.2.1\share\hadoop\yarn\*;
C:\hadoop-3.2.1\share\hadoop\mapreduce\lib\*;C:\hadoop-3.2.1\share\hadoop\mapreduce\*"

C:\DBMS\Project\MapReduceDirectory>
```

After this, we create a new file Manifest.txt and add the class of driver file.



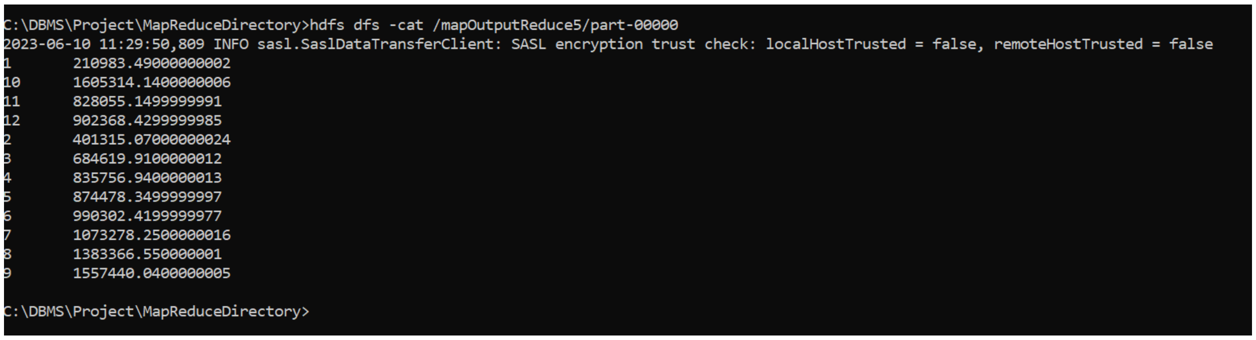
```
:\DBMS\Project\MapReduceDirectory>jar cfm ProductSalePerCountry.jar Manifest.txt SalesCountry/*.class

:\DBMS\Project\MapReduceDirectory>
```

Now, we will copy the file hotel-booking.csv into ~/inputMapReduce.

After this, we run the following command to run the MapReduce job:

hadoop jar ProductSalePerCountry.jar /inputMapReduce /mapreduce_output1

This will return a file mapreduce_output1.



```
C:\DBMS\Project\MapReduceDirectory>hdfs dfs -cat /mapOutputReduce5/part-00000
2023-06-10 11:29:50,809 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
1       210983.49000000002
10      1605314.1400000006
11      828055.1499999991
12      902368.4299999985
2       401315.07000000024
3       684619.9100000012
4       835756.9400000013
5       874478.3499999997
6       990302.4199999977
7       1073278.2500000016
8       1383366.550000001
9       1557440.0400000005

C:\DBMS\Project\MapReduceDirectory>
```

Now we will use the same steps for the second map reduce in the new directory “C:\DBMS\Project\MapReduceDirectory2” to perform the task on our second dataset.

We have also used one more MapReduce function that combine these values for each month and we get the month with most revenue generated.