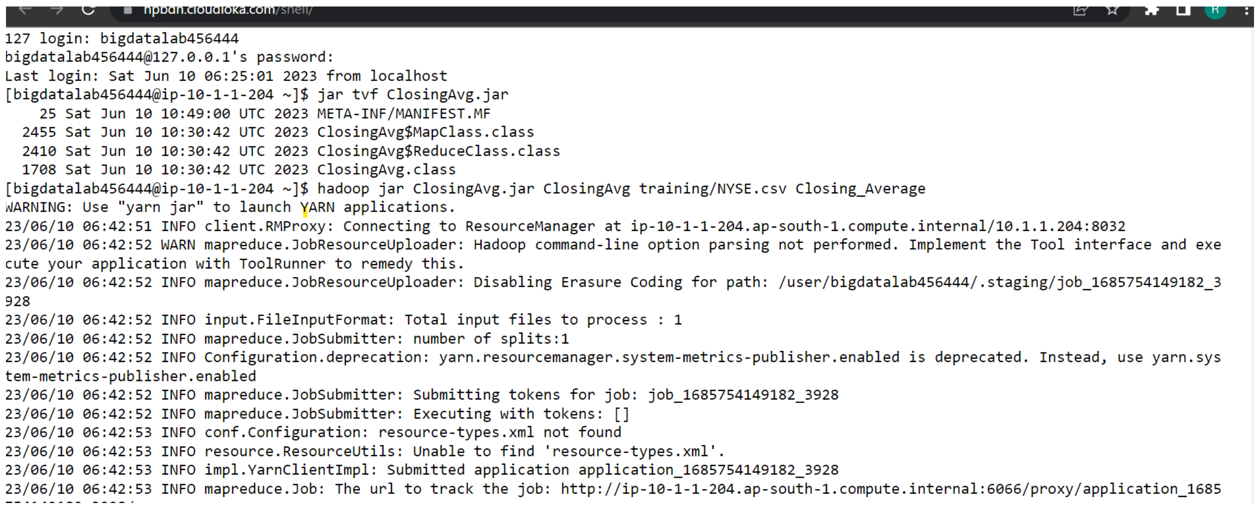
SET= A

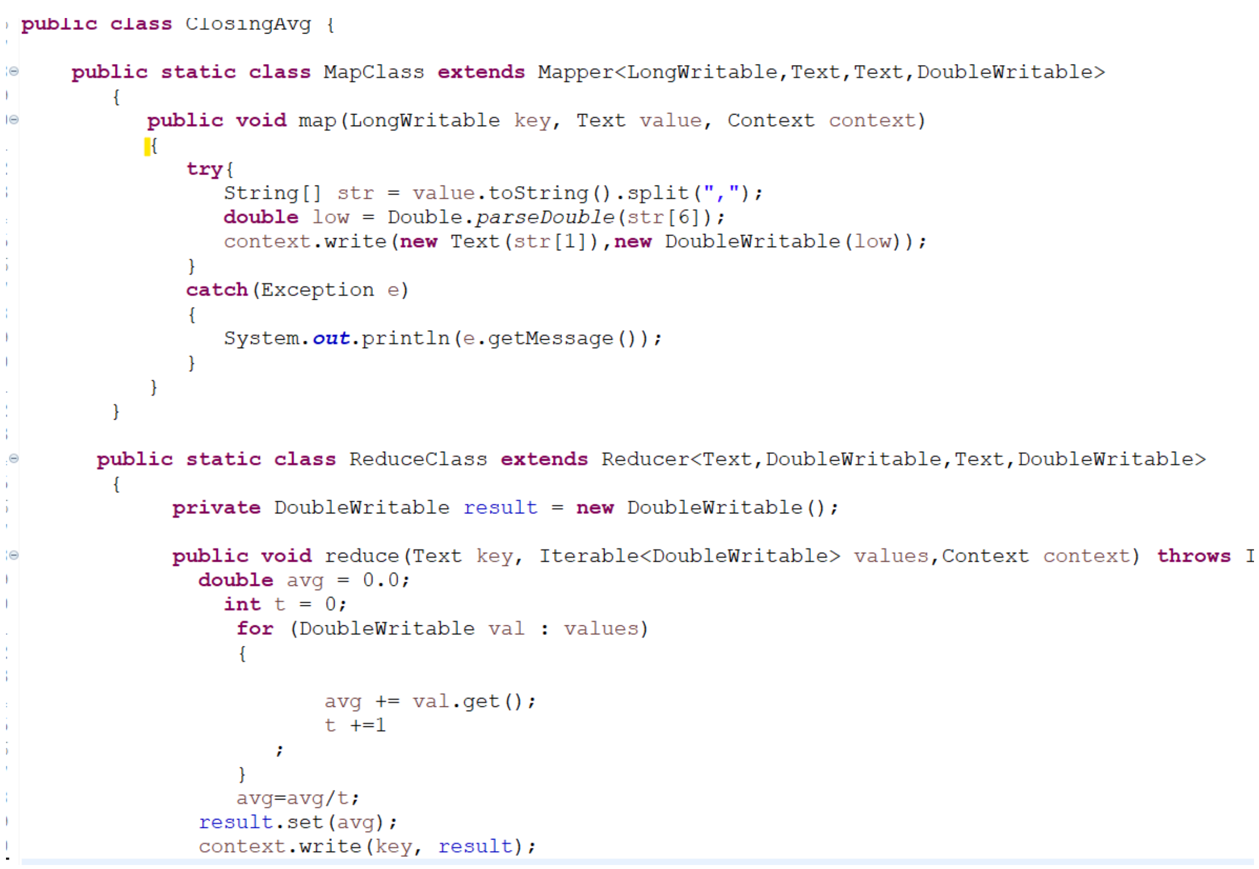
Q.1) Map Reduce Use NYSE DATA SET FOR THE BELOW QUESTION:

1) Find out the average closing price for each stock.

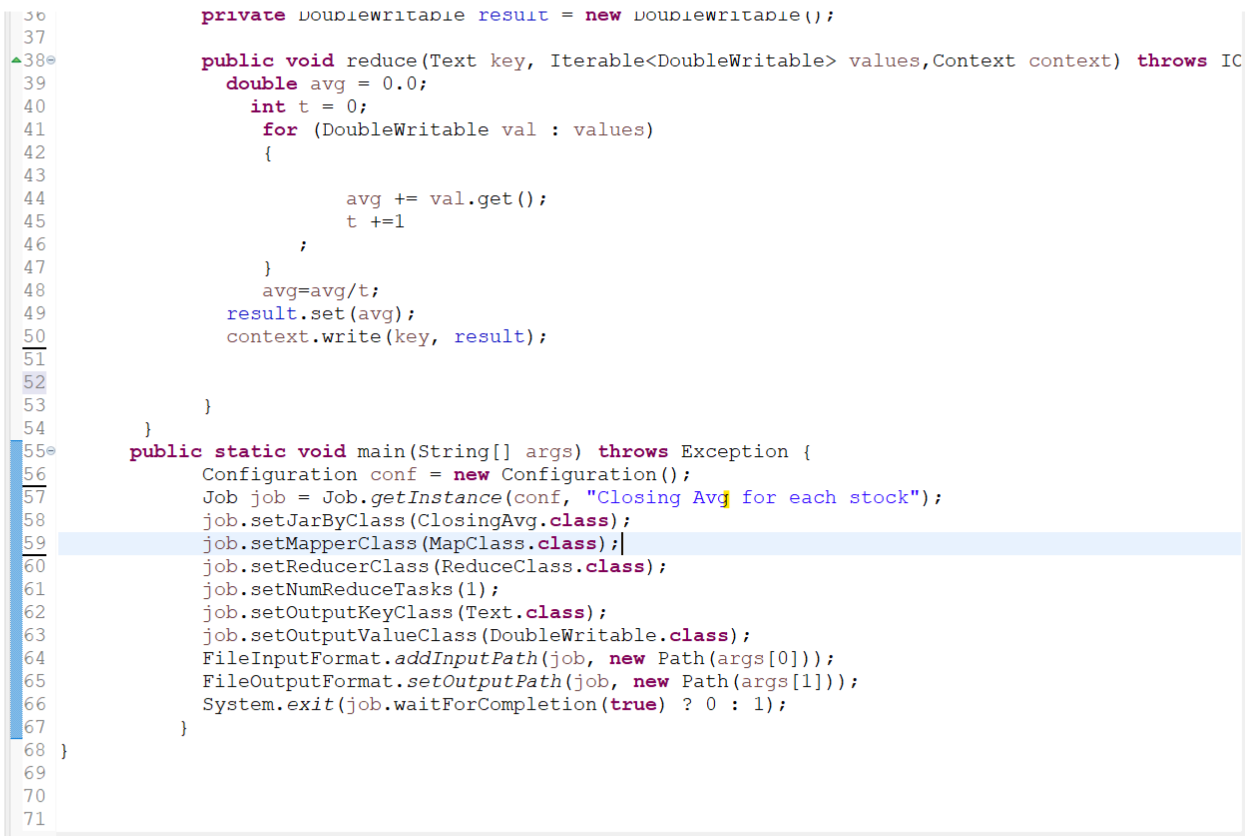
= `hadoop jar ClosingAvg.jar ClosingAvg training/NYSE.csv Closing_Averag`



```
127 login: bigdatalab456444
bigdatalab456444@127.0.0.1's password:
Last login: Sat Jun 10 06:25:01 2023 from localhost
[bigdatalab456444@ip-10-1-1-204 ~]$ jar tvf ClosingAvg.jar
    25 Sat Jun 10 10:49:00 UTC 2023 META-INF/MANIFEST.MF
  2455 Sat Jun 10 10:30:42 UTC 2023 ClosingAvg$MapClass.class
  2410 Sat Jun 10 10:30:42 UTC 2023 ClosingAvg$ReduceClass.class
  1708 Sat Jun 10 10:30:42 UTC 2023 ClosingAvg.class
[bigdatalab456444@ip-10-1-1-204 ~]$ hadoop jar ClosingAvg.jar ClosingAvg training/NYSE.csv Closing_Average
WARNING: Use "yarn jar" to launch YARN applications.
23/06/10 06:42:51 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 06:42:52 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and exe
cute your application with ToolRunner to remedy this.
23/06/10 06:42:52 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/bigdatalab456444/.staging/job_1685754149182_3
928
23/06/10 06:42:52 INFO input.FileInputFormat: Total input files to process : 1
23/06/10 06:42:52 INFO mapreduce.JobSubmitter: number of splits:1
23/06/10 06:42:52 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.sys
tem-metrics-publisher.enabled
23/06/10 06:42:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1685754149182_3928
23/06/10 06:42:52 INFO mapreduce.JobSubmitter: Executing with tokens: []
23/06/10 06:42:53 INFO conf.Configuration: resource-types.xml not found
23/06/10 06:42:53 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/06/10 06:42:53 INFO impl.YarnClientImpl: Submitted application application_1685754149182_3928
23/06/10 06:42:53 INFO mapreduce.Job: The url to track the job: http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685
```



```
public class ClosingAvg {

    public static class MapClass extends Mapper<LongWritable,Text,Text,DoubleWritable>
       {
          public void map(LongWritable key, Text value, Context context)
          {
             try{
                String[] str = value.toString().split(",");
                double low = Double.parseDouble(str[6]);
                context.write(new Text(str[1]),new DoubleWritable(low));
             }
             catch(Exception e)
             {
                System.out.println(e.getMessage());
             }
          }
       }

      public static class ReduceClass extends Reducer<Text,DoubleWritable,Text,DoubleWritable>
       {
            private DoubleWritable result = new DoubleWritable();

            public void reduce(Text key, Iterable<DoubleWritable> values,Context context) throws I
              double avg = 0.0;
                int t = 0;
                 for (DoubleWritable val : values)
                 {

                        avg += val.get();
                        t +=1
                    ;
                 }
                 avg=avg/t;
              result.set(avg);
              context.write(key, result);
```

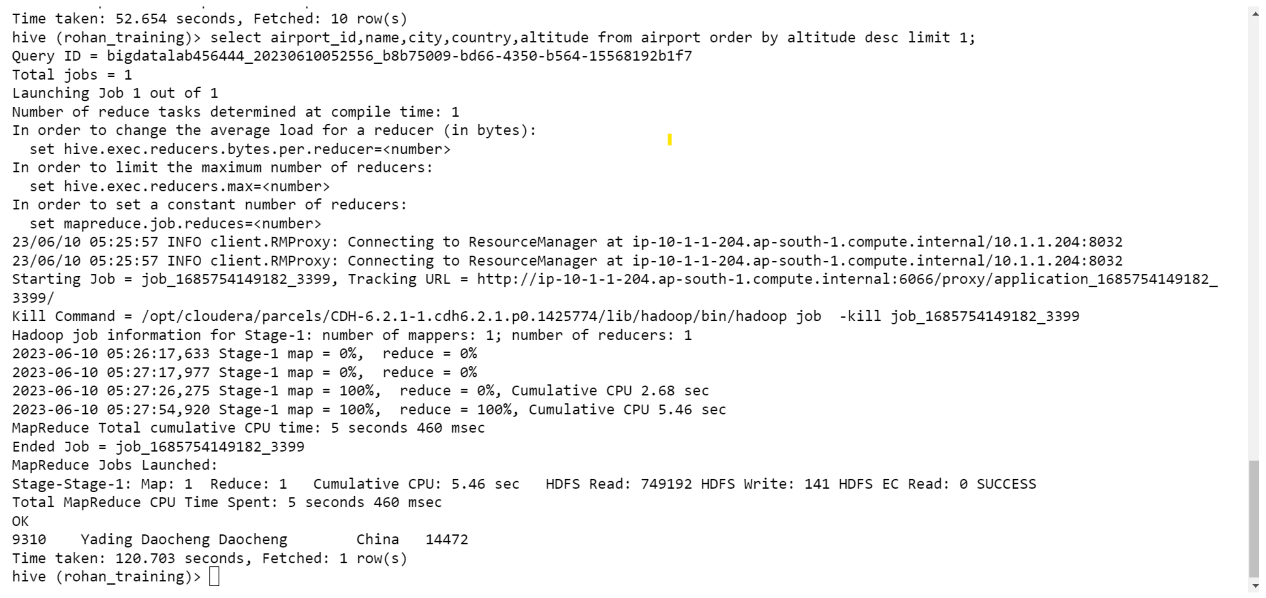


```
            private DoubleWritable result = new DoubleWritable();

            public void reduce(Text key, Iterable<DoubleWritable> values,Context context) throws IO
              double avg = 0.0;
                int t = 0;
                 for (DoubleWritable val : values)
                 {

                        avg += val.get();
                        t +=1
                    ;
                 }
                 avg=avg/t;
              result.set(avg);
              context.write(key, result);


            }
      }
      public static void main(String[] args) throws Exception {
            Configuration conf = new Configuration();
            Job job = Job.getInstance(conf, "Closing Avg for each stock");
            job.setJarByClass(ClosingAvg.class);
            job.setMapperClass(MapClass.class);
            job.setReducerClass(ReduceClass.class);
            job.setNumReduceTasks(1);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(DoubleWritable.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));
            System.exit(job.waitForCompletion(true) ? 0 : 1);
          }
}
```

Q2.) Hive:

1) Which airports have the highest altitude ?

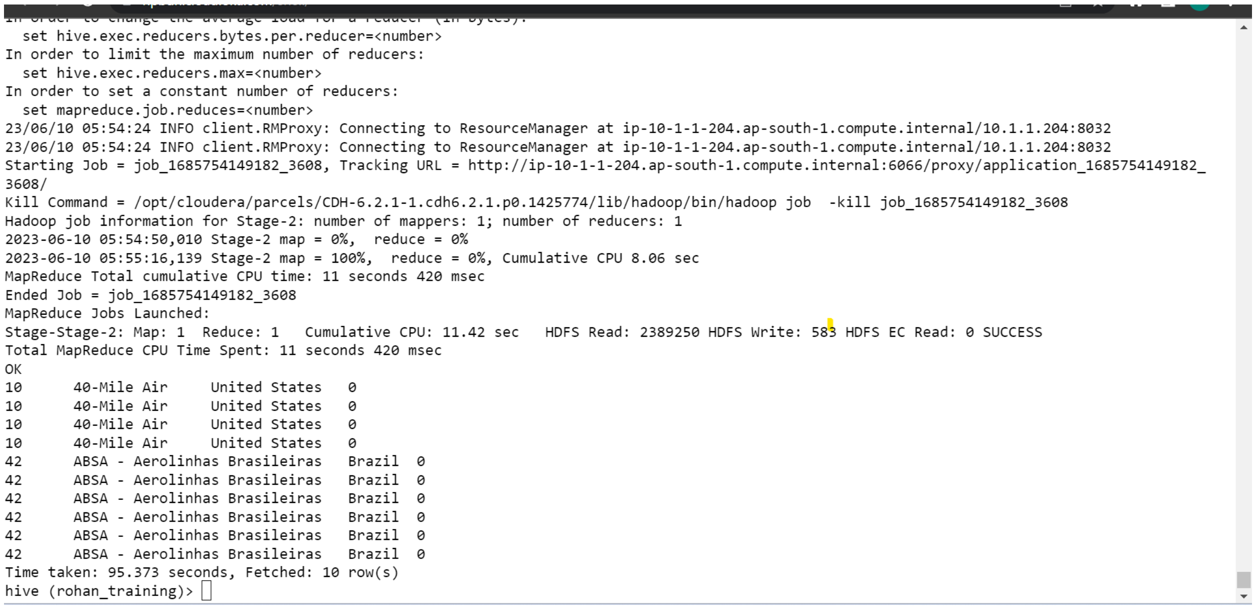
= select airport_id,name,city,country,altitude from airport order by altitude desc limit 1;



```
Time taken: 52.654 seconds, Fetched: 10 row(s)
hive (rohan_training)> select airport_id,name,city,country,altitude from airport order by altitude desc limit 1;
Query ID = bigdatalab456444_20230610052556_b8b75009-bd66-4350-b564-15568192b1f7
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
23/06/10 05:25:57 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 05:25:57 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1685754149182_3399, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685754149182_
3399/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1685754149182_3399
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-06-10 05:26:17,633 Stage-1 map = 0%,  reduce = 0%
2023-06-10 05:27:17,977 Stage-1 map = 0%,  reduce = 0%
2023-06-10 05:27:26,275 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.68 sec
2023-06-10 05:27:54,920 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.46 sec
MapReduce Total cumulative CPU time: 5 seconds 460 msec
Ended Job = job_1685754149182_3399
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.46 sec   HDFS Read: 749192 HDFS Write: 141 HDFS EC Read: 0 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 460 msec
OK
9310    Yading Daocheng Daocheng        China   14472
Time taken: 120.703 seconds, Fetched: 1 row(s)
hive (rohan_training)> 
```

2) How many routes are operated by active airlines from the United States ?
= select distinct a.name from airlines a join routes r
on a.airline_id = r.airline_id join airport ar1 on ar1.airport_id = src_airport_id
join
airport ar2
on
ar2.airport_id = dest_airport_id where trim(upper(ar1.country)) = "UNITED STATES" or
trim(upper(ar2.country)) = "UNITED STATES";

3) Which airlines operate routes that have less than 3 stops number of stops
top 10 alphabetically?
= select air.airline_id,air.name,country,r.stops from airlines as air inner join routes r on
air.airline_id = r.airline_id where stops<3 order by air.name limit 10;


```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
23/06/10 05:54:24 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 05:54:24 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1685754149182_3608, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685754149182_
3608/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1685754149182_3608
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2023-06-10 05:54:50,010 Stage-2 map = 0%,  reduce = 0%
2023-06-10 05:55:16,139 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 8.06 sec
MapReduce Total cumulative CPU time: 11 seconds 420 msec
Ended Job = job_1685754149182_3608
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.42 sec   HDFS Read: 2389250 HDFS Write: 583 HDFS EC Read: 0 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 420 msec
OK
10      40-Mile Air     United States   0
10      40-Mile Air     United States   0
10      40-Mile Air     United States   0
10      40-Mile Air     United States   0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
42      ABSA - Aerolinhas Brasileiras   Brazil  0
Time taken: 95.373 seconds, Fetched: 10 row(s)
hive (rohan_training)> 
```

4) How many airlines have a specific IATA code ‘W9’?
= Select count(distinct(airline_id)) from airlines where iata='W9';

5) Find the airlines that operate routes with a specific equipment as ‘AN4’
and codeshare enabled.
= Select distinct(a.name) from airlines a join routes r on a.airline_id=r.airline_id where equipment='AN4' and codeshare='Y';

Q.3) PYSPARK
Answers=

Spark
from pyspark.sql.types import StructType,Integer Type,StringType,DoubleType,LongType

```
schema =
StructType().add("Year",IntegerType(),True).add("Quarter",IntegerType(),True).add("Avg_rev",D
oubleType(),True).add(total_seats,IntegerType(),True)
```

```
df =
spark.read.format("csv").option("header","True").schema(schema).load("hdfs://nameservice1/us
er/bigdatalab456444/training/airlines.csv")
```

```
df.registerTempTable("air”)
```

1) What is the total revenue generated in each year?
= df1 = spark.sql("select year,((sum(Avg_rev*total_seats))/1000000) as tot from air group by Year")

2) Which year had the highest average revenue per seat?
= df2 = spark.sql("select Year,Avg_rev from air order by Avg_rev desc limit 1")

3) What is the total number of booked seats for each quarter in a given year?
= df3 = spark.sql("select Year, Quarter,total_seats from air where Year = 2001")