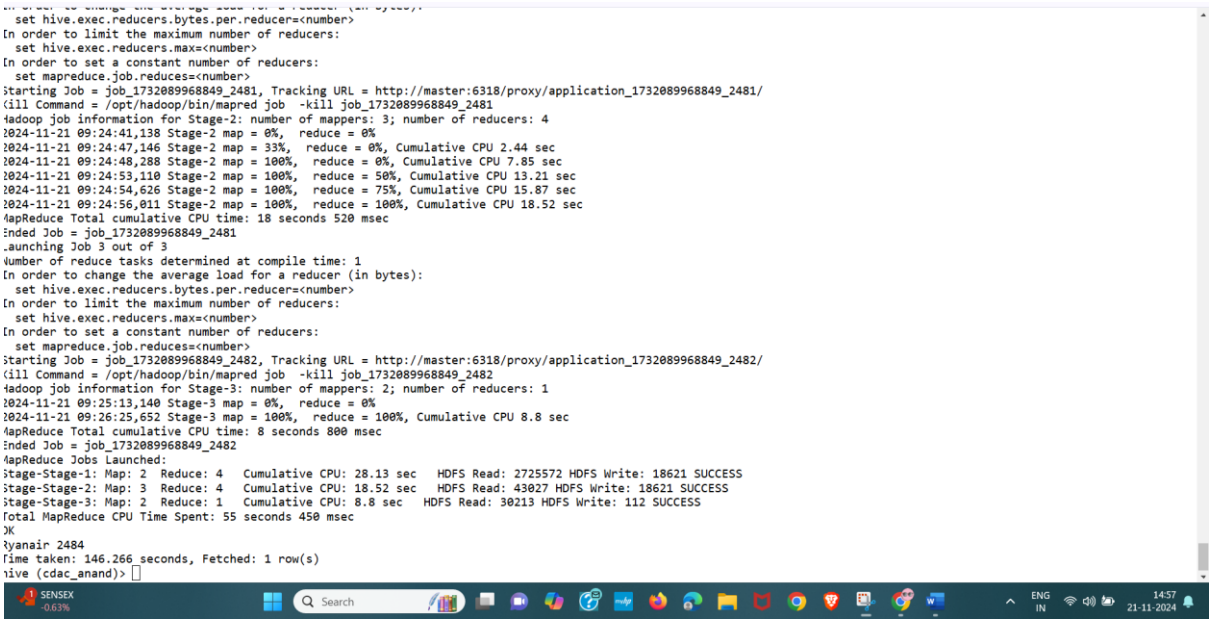
HIVE

Q1.

1.select distinct a.name from routes r join airport a on r.src_airport_id=a.airport_id join airport a1 on r.dest_airport_id=a1.airport_id limit 10;

```
Total MapReduce CPU Time Spent: 1 minutes 22 seconds 300 msec
OK
Agri Airport
Aizawl
Akiak Airport
Akita
Abu Dhabi Intl
Achmad Yani
Adnan Menderes
Aasiaat
Aberdeen Regional Airport
Agatti
Time taken: 85.696 seconds, Fetched: 10 row(s)
```

3. select a.name, count(*) from routes r join airlines a on r.airline_id=a.airline_id group by a.name order by count(*) desc limit 1;



2.

Select equipment ,count(*) from routes where equipment="U" group by equipment order by count(*) desc limit 1;

Q.2

1. **create table source_airport(airline_iata string,airline_id int,src_airport_id int,dest_airport_iata string,dest_airport_id int,codeshare string,**

stops int,equipment string) partitioned by (src_airport_iata string) row format delimited fields terminated by "," stored as textfile;

insert overwrite table source_airport partition(src_airport_iata) select r.airline_iata, r.airline_id, r.dest_airport_iata,

r.dest_airport_id, r.codeshare, r.stops, r.equipment, r.src_airport_iata from routes r distribute by src_airport_iata;

server down that why not taken screenshot.

2. insert overwrite table routes_partition(src_airport_iata) select r.airline_iata ,r.airline_id ,r.src_airport_id,r.dest_airport_iata ,r.dest_airport_i

d ,r.codeshare ,r.stops,r.equipment,r.src_airport_iata from routes r where src_airport_iata='JFK' distribute by src_airport_iata;

server down

3.select * from source_airport where src_airport_iata=”LAX”;

4.select * from routes_partition;

Spark

Q.1

1. from pyspark import SparkConf,SparkContext

```
airlinerdd=sc.textFile("/user/cdacuser882112/airlines.csv")
header=airlinerdd.first()
el=airlinerdd.filter(lambda a: a!=header)
split=el.map(lambda a:a.split(","))
combine=split.map(lambda a:(a[0],int(a[3])))
total=combine.filter(lambda a:(a[3] >40000))
total.count()
```

2.

```
combine1=split.map(lambda a:a[0] )
 year=combine1.map(lambda a:(a,1))
  arrange=year.reduceByKey(lambda a,b:a+b)
  for a in arrange.collect():
          print(a[0])
```

distictyear=print(a[0])

Q.2

Server down that why written programm.

5. ywiserev=combine.map(lambda a:(a[0],(float(a[2])*int(a[3]))))

rev=ywiserev.reduceByKey(lambda a,b:a+b)

rev.take(5)

4. for line in rev.collect():

Print(line[0])

3. total=split.map(lambda a: a[2]).sum()

1. combine2=split.map(lambda a:(a[0],int(a[2])))
   avg=combine2..sortBy(lambda a:-a[1]).mean()

   max= combine2.max(lambda a:a[1])
   min=combine2.min(lambda a:a[1])

   2.

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 1586, in first
    rs = self.take(1)
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 1533, in take
    totalParts = self.getNumPartitions()
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 465, in getNumPartitions
    return self._jrdd.partitions().size()
  File "/opt/spark-3.1.2/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py", line 1304, in __call__
  File "/opt/spark-3.1.2/python/pyspark/sql/utils.py", line 111, in deco
    return f(*a, **kw)
  File "/opt/spark-3.1.2/python/lib/py4j-0.10.9-src.zip/py4j/protocol.py", line 326, in get_return_value
py4j.protocol.Py4JJavaError: An error occurred while calling o46.partitions.
: org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: hdfs://master:9000/user/cdacuser/airlines.csv
        at org.apache.hadoop.mapred.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:297)
        at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:239)
        at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:325)
        at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:205)
        at org.apache.spark.rdd.RDD.$anonfun$partitions$2(RDD.scala:300)
        at scala.Option.getOrElse(Option.scala:189)
        at org.apache.spark.rdd.RDD.partitions(RDD.scala:296)
        at org.apache.spark.rdd.MapPartitionsRDD.getPartitions(MapPartitionsRDD.scala:49)
        at org.apache.spark.rdd.RDD.$anonfun$partitions$2(RDD.scala:300)
        at scala.Option.getOrElse(Option.scala:189)
        at org.apache.spark.rdd.RDD.partitions(RDD.scala:296)
        at org.apache.spark.api.java.JavaRDDLike.partitions(JavaRDDLike.scala:61)
        at org.apache.spark.api.java.JavaRDDLike.partitions$(JavaRDDLike.scala:61)
        at org.apache.spark.api.java.AbstractJavaRDDLike.partitions(JavaRDDLike.scala:45)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:750)
```