

# 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;**

```
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>
Starting Job = job_1732089968849_2481, Tracking URL = http://master:6318/proxy/application_1732089968849_2481/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2481
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 4
2024-11-21 09:24:41,138 Stage-2 map = 0%, reduce = 0%
2024-11-21 09:24:47,146 Stage-2 map = 33%, reduce = 0%, Cumulative CPU 2.44 sec
2024-11-21 09:24:48,288 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 7.85 sec
2024-11-21 09:24:53,110 Stage-2 map = 100%, reduce = 50%, Cumulative CPU 13.21 sec
2024-11-21 09:24:54,626 Stage-2 map = 100%, reduce = 75%, Cumulative CPU 15.87 sec
2024-11-21 09:24:56,011 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 18.52 sec
4mapReduce Total cumulative CPU time: 18 seconds 520 msec
Ended Job = job_1732089968849_2481
Launching Job 3 out of 3
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>
Starting Job = job_1732089968849_2482, Tracking URL = http://master:6318/proxy/application_1732089968849_2482/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2482
Hadoop job information for Stage-3: number of mappers: 2; number of reducers: 1
2024-11-21 09:25:13,140 Stage-3 map = 0%, reduce = 0%
2024-11-21 09:26:25,652 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 8.8 sec
4mapReduce Total cumulative CPU time: 8 seconds 800 msec
Ended Job = job_1732089968849_2482
4mapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 28.13 sec HDFS Read: 2725572 HDFS Write: 18621 SUCCESS
Stage-Stage-2: Map: 3 Reduce: 4 Cumulative CPU: 18.52 sec HDFS Read: 43027 HDFS Write: 18621 SUCCESS
Stage-Stage-3: Map: 2 Reduce: 1 Cumulative CPU: 8.8 sec HDFS Read: 30213 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 55 seconds 450 msec
%
ryanair 2484
Time taken: 146.266 seconds, Fetched: 1 row(s)
hive (cdac_anand)>
```

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._jrdm.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)
```

