Module exam

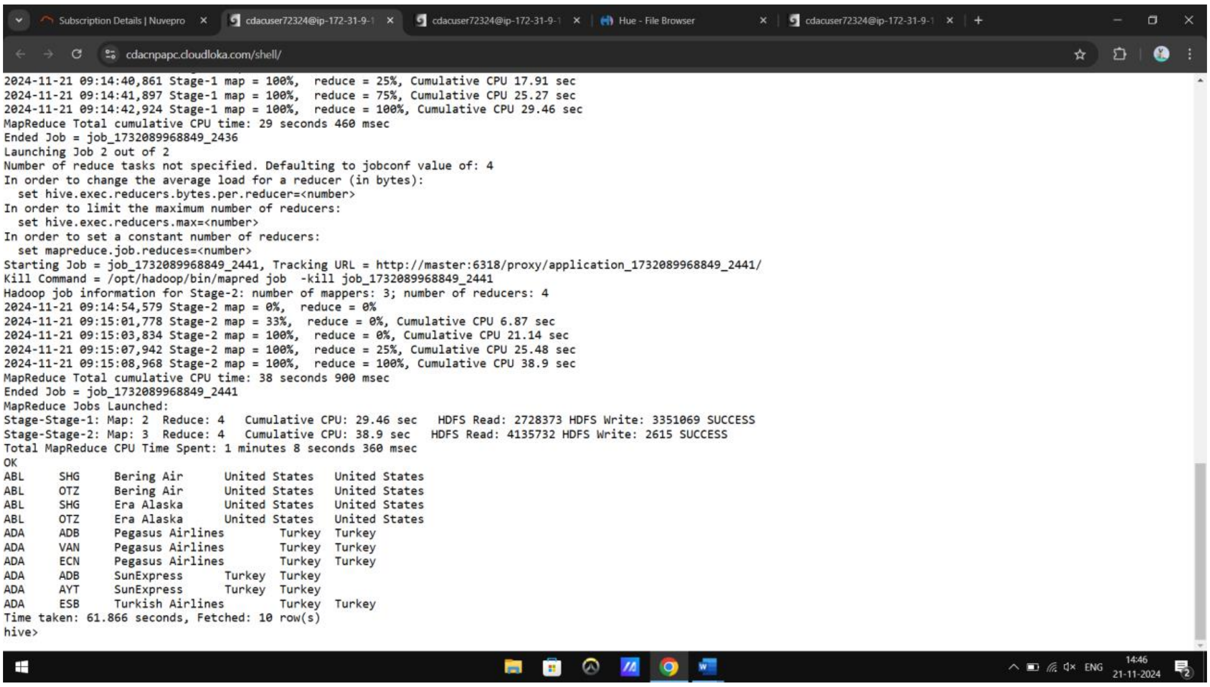
Name: Neelanjan Nandkumar Dalavi

Roll no. 240840325034

Hive

Question 1.

1. ans. select r.src_airport_iata,r.dest_airport_iata,a.name,a.country,a1.country from routes r join airlines a on r.airline_id=a.airline_id join airport a1 on r.src_airport_iata=a1.iata where a.country=a1.country limit 10;



```
2024-11-21 09:14:40,861 Stage-1 map = 100%,  reduce = 25%, Cumulative CPU 17.91 sec
2024-11-21 09:14:41,897 Stage-1 map = 100%,  reduce = 75%, Cumulative CPU 25.27 sec
2024-11-21 09:14:42,924 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 29.46 sec
MapReduce Total cumulative CPU time: 29 seconds 460 msec
Ended Job = job_1732089968849_2436
Launching Job 2 out of 2
Number of reduce tasks not specified. Defaulting to jobconf value of: 4
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_2441, Tracking URL = http://master:6318/proxy/application_1732089968849_2441/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2441
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 4
2024-11-21 09:14:54,579 Stage-2 map = 0%,  reduce = 0%
2024-11-21 09:15:01,778 Stage-2 map = 33%,  reduce = 0%, Cumulative CPU 6.87 sec
2024-11-21 09:15:03,834 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 21.14 sec
2024-11-21 09:15:07,942 Stage-2 map = 100%,  reduce = 25%, Cumulative CPU 25.48 sec
2024-11-21 09:15:08,968 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 38.9 sec
MapReduce Total cumulative CPU time: 38 seconds 900 msec
Ended Job = job_1732089968849_2441
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 29.46 sec   HDFS Read: 2728373 HDFS Write: 3351069 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 4   Cumulative CPU: 38.9 sec   HDFS Read: 4135732 HDFS Write: 2615 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 8 seconds 360 msec
OK
ABL     SHG     Bering Air      United States   United States
ABL     OTZ     Bering Air      United States   United States
ABL     SHG     Era Alaska      United States   United States
ABL     OTZ     Era Alaska      United States   United States
ADA     ADB     Pegasus Airlines        Turkey  Turkey
ADA     VAN     Pegasus Airlines        Turkey  Turkey
ADA     ECN     Pegasus Airlines        Turkey  Turkey
ADA     ADB     SunExpress      Turkey  Turkey
ADA     AYT     SunExpress      Turkey  Turkey
ADA     ESB     Turkish Airlines        Turkey  Turkey
Time taken: 61.866 seconds, Fetched: 10 row(s)
hive>
```

2.

3.

Question 2.

1.

2.

3.

Spark

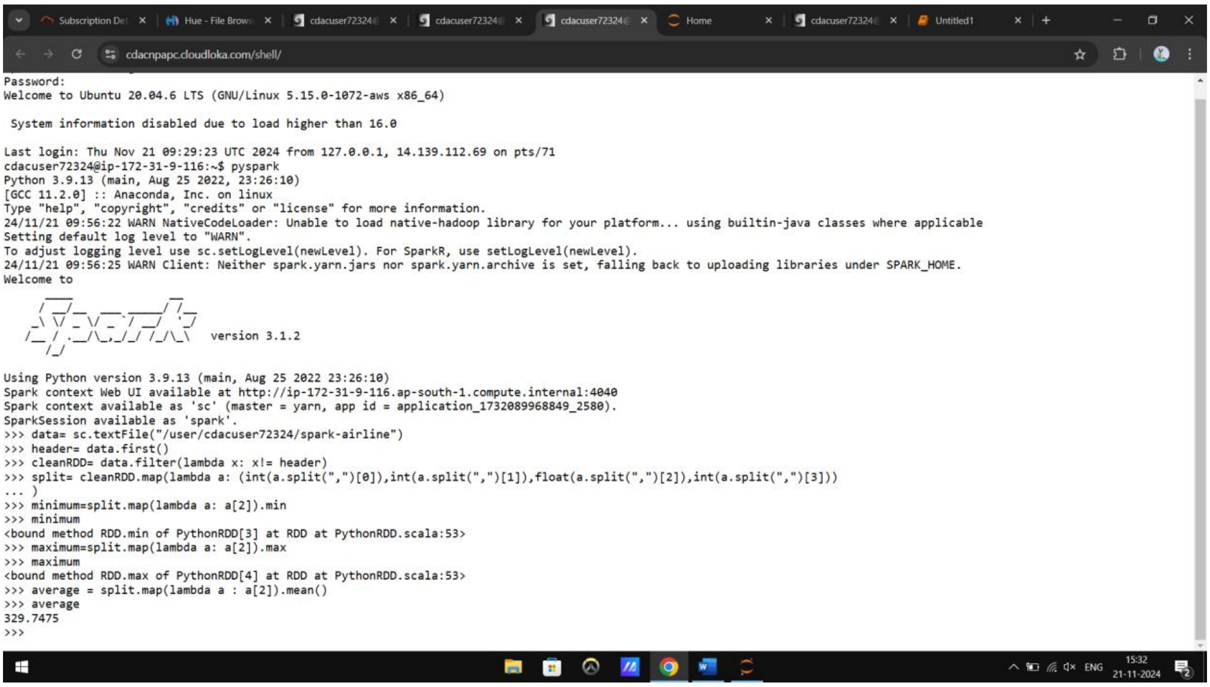
Question 1.

1. more = split.map(lambda a: a[3]>40000)
   counter= more.count()
   counter

2.dist= split.map(lambda a: a[0]).distinct

Question 2. (every query in RDD)

1. minimum=split.map(lambda a: a[2]).min
   maximum=split.map(lambda a: a[2]).max
   average = split.map(lambda a : a[2]).mean()



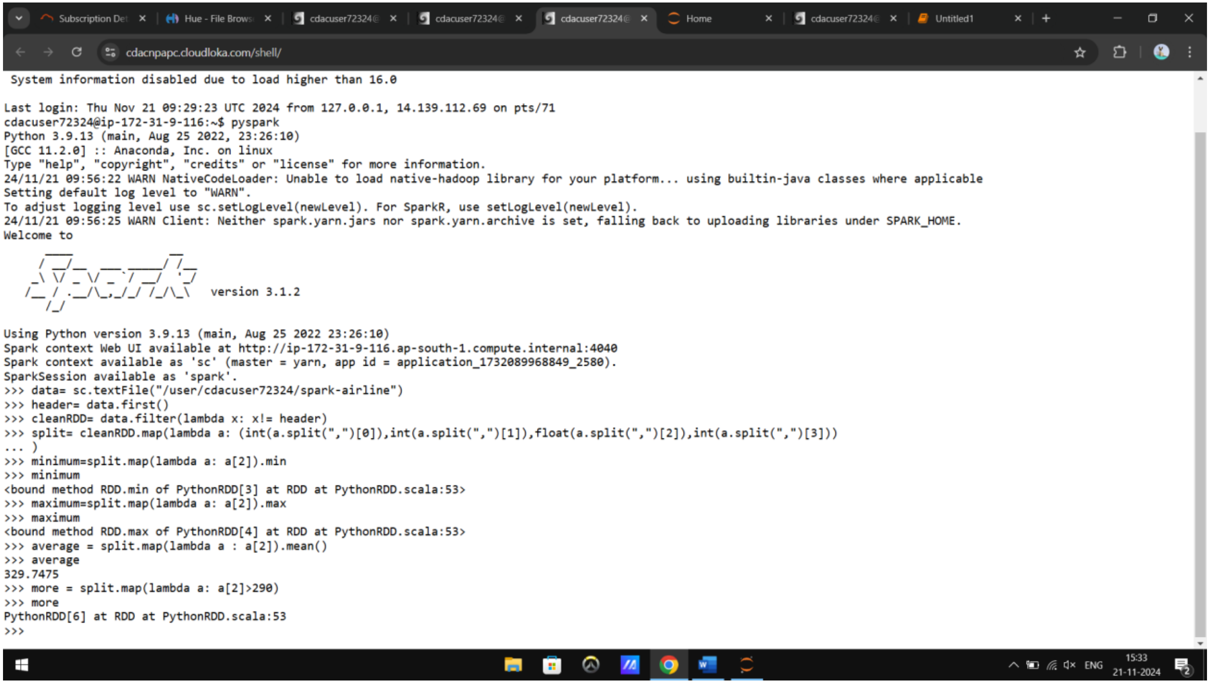
```
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1072-aws x86_64)

 System information disabled due to load higher than 16.0

Last login: Thu Nov 21 09:29:23 UTC 2024 from 127.0.0.1, 14.139.112.69 on pts/71
cdacuser72324@ip-172-31-9-116:~$ pyspark
Python 3.9.13 (main, Aug 25 2022, 23:26:10)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
24/11/21 09:56:22 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/11/21 09:56:25 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.9.13 (main, Aug 25 2022 23:26:10)
Spark context Web UI available at http://ip-172-31-9-116.ap-south-1.compute.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2580).
SparkSession available as 'spark'.
>>> data= sc.textFile("/user/cdacuser72324/spark-airline")
>>> header= data.first()
>>> cleanRDD= data.filter(lambda x: x!= header)
>>> split= cleanRDD.map(lambda a: (int(a.split(",")[0]),int(a.split(",")[1]),float(a.split(",")[2]),int(a.split(",")[3]))
... )
>>> minimum=split.map(lambda a: a[2]).min
>>> minimum
<bound method RDD.min of PythonRDD[3] at RDD at PythonRDD.scala:53>
>>> maximum=split.map(lambda a: a[2]).max
>>> maximum
<bound method RDD.max of PythonRDD[4] at RDD at PythonRDD.scala:53>
>>> average = split.map(lambda a : a[2]).mean()
>>> average
329.7475
>>>
```

2. more = split.map(lambda a: a[2]>290)



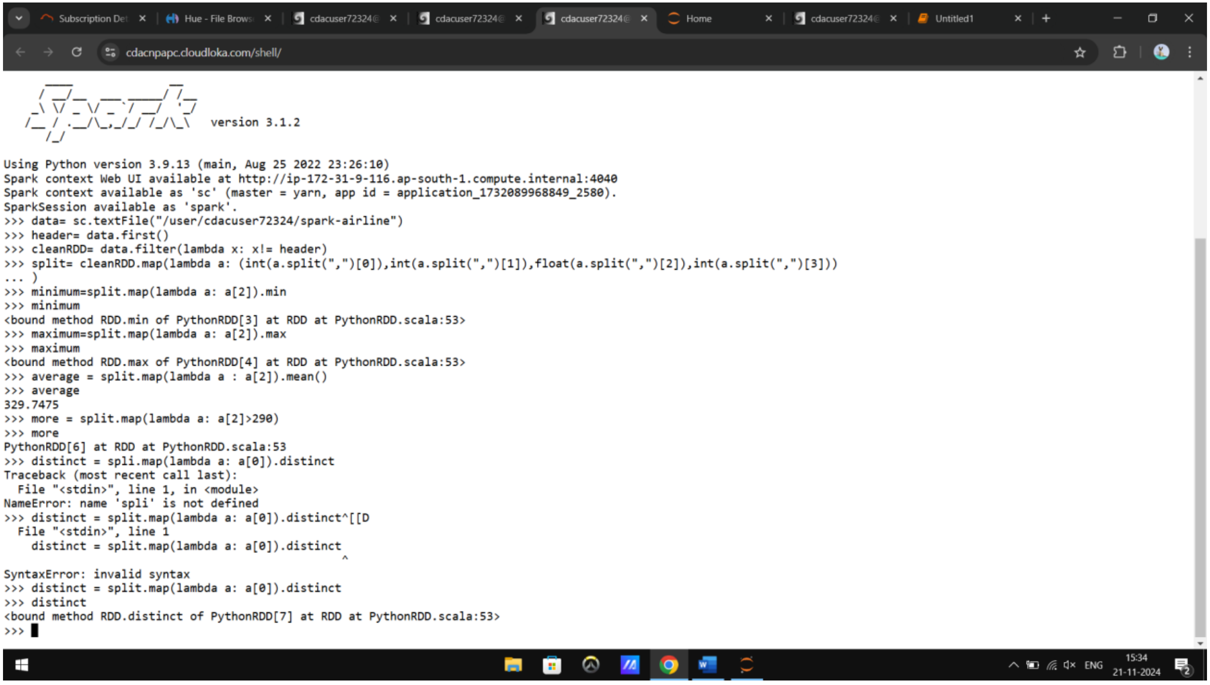
```
 System information disabled due to load higher than 16.0

Last login: Thu Nov 21 09:29:23 UTC 2024 from 127.0.0.1, 14.139.112.69 on pts/71
cdacuser72324@ip-172-31-9-116:~$ pyspark
Python 3.9.13 (main, Aug 25 2022, 23:26:10)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
24/11/21 09:56:22 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/11/21 09:56:25 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.9.13 (main, Aug 25 2022 23:26:10)
Spark context Web UI available at http://ip-172-31-9-116.ap-south-1.compute.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2580).
SparkSession available as 'spark'.
>>> data= sc.textFile("/user/cdacuser72324/spark-airline")
>>> header= data.first()
>>> cleanRDD= data.filter(lambda x: x!= header)
>>> split= cleanRDD.map(lambda a: (int(a.split(",")[0]),int(a.split(",")[1]),float(a.split(",")[2]),int(a.split(",")[3]))
... )
>>> minimum=split.map(lambda a: a[2]).min
>>> minimum
<bound method RDD.min of PythonRDD[3] at RDD at PythonRDD.scala:53>
>>> maximum=split.map(lambda a: a[2]).max
>>> maximum
<bound method RDD.max of PythonRDD[4] at RDD at PythonRDD.scala:53>
>>> average = split.map(lambda a : a[2]).mean()
>>> average
329.7475
>>> more = split.map(lambda a: a[2]>290)
>>> more
PythonRDD[6] at RDD at PythonRDD.scala:53
>>>
```

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

4.distinct = spli.map(lambda a: a[0]).distinct



```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.9.13 (main, Aug 25 2022 23:26:10)
Spark context Web UI available at http://ip-172-31-9-116.ap-south-1.compute.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2580).
SparkSession available as 'spark'.
>>> data= sc.textFile("/user/cdacuser72324/spark-airline")
>>> header= data.first()
>>> cleanRDD= data.filter(lambda x: x!= header)
>>> split= cleanRDD.map(lambda a: (int(a.split(",")[0]),int(a.split(",")[1]),float(a.split(",")[2]),int(a.split(",")[3]))
... )
>>> minimum=split.map(lambda a: a[2]).min
>>> minimum
<bound method RDD.min of PythonRDD[3] at RDD at PythonRDD.scala:53>
>>> maximum=split.map(lambda a: a[2]).max
>>> maximum
<bound method RDD.max of PythonRDD[4] at RDD at PythonRDD.scala:53>
>>> average = split.map(lambda a : a[2]).mean()
>>> average
329.7475
>>> more = split.map(lambda a: a[2]>290)
>>> more
PythonRDD[6] at RDD at PythonRDD.scala:53
>>> distinct = spli.map(lambda a: a[0]).distinct
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'spli' is not defined
>>> distinct = split.map(lambda a: a[0]).distinct^[[D
  File "<stdin>", line 1
    distinct = split.map(lambda a: a[0]).distinct
                                                 ^

SyntaxError: invalid syntax
>>> distinct = split.map(lambda a: a[0]).distinct
>>> distinct
<bound method RDD.distinct of PythonRDD[7] at RDD at PythonRDD.scala:53>
>>> █
```

5. combine = split.map(lambda a : (str(a[0]), a[2]))

average = combine.map(lambda a : a[1]).mean()