Name : Ninad Khune
Exam : Big Data Module Exam
PRN : 240840325036

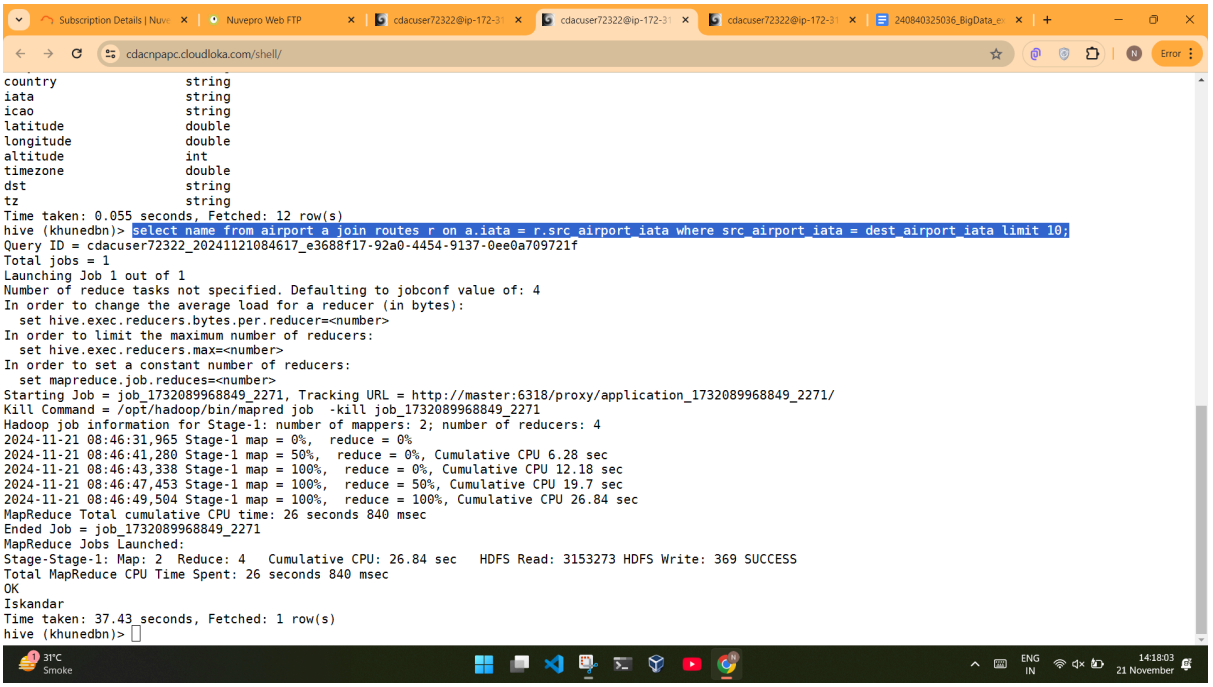
HIVE:

Question 1

1. **Query :**

```
select name from airport a join routes r on a.iata =
r.src_airport_iata where src_airport_iata = dest_airport_iata
limit 10;
```

Output : Iskandar



```
country                 string
iata                    string
icao                    string
latitude                double
longitude               double
altitude                int
timezone                double
dst                     string
tz                      string
Time taken: 0.055 seconds, Fetched: 12 row(s)
hive (khunedbn)> select name from airport a join routes r on a.iata = r.src_airport_iata where src_airport_iata = dest_airport_iata limit 10;
Query ID = cdacuser72322_20241121084617_e3688f17-92a0-4454-9137-0ee0a709721f
Total jobs = 1
Launching Job 1 out of 1
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_2271, Tracking URL = http://master:6318/proxy/application_1732089968849_2271/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2271
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 4
2024-11-21 08:46:31,965 Stage-1 map = 0%,  reduce = 0%
2024-11-21 08:46:41,280 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 6.28 sec
2024-11-21 08:46:43,338 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 12.18 sec
2024-11-21 08:46:47,453 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 19.7 sec
2024-11-21 08:46:49,504 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 26.84 sec
MapReduce Total cumulative CPU time: 26 seconds 840 msec
Ended Job = job_1732089968849_2271
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 26.84 sec   HDFS Read: 3153273 HDFS Write: 369 SUCCESS
Total MapReduce CPU Time Spent: 26 seconds 840 msec
OK
Iskandar
Time taken: 37.43 seconds, Fetched: 1 row(s)
hive (khunedbn)>
```

2. **Query :**

```
select max(equipment) from routes limit 5;
```

Output :

YN7

```
2024-11-21 08:46:47,453 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 19.7 sec
2024-11-21 08:46:49,504 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 26.84 sec
MapReduce Total cumulative CPU time: 26 seconds 840 msec
Ended Job = job_1732089968849_2271
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 26.84 sec   HDFS Read: 3153273 HDFS Write: 369 SUCCESS
Total MapReduce CPU Time Spent: 26 seconds 840 msec
OK
Iskandar
Time taken: 37.43 seconds, Fetched: 1 row(s)
hive (khunedbn)> select equipment from routes where max(airline_iata) limit 5;
FAILED: SemanticException [Error 10128]: Line 1:35 Not yet supported place for UDAF 'max'
hive (khunedbn)> select max(equipment) from routes limit 5;
Query ID = cdacuser72322_20241121085057_0cd9a89a-9dd3-442e-a1a8-ac0fee7afb6a
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>
Starting Job = job_1732089968849_2298, Tracking URL = http://master:6318/proxy/application_1732089968849_2298/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2298
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-11-21 08:51:08,744 Stage-1 map = 0%,  reduce = 0%
2024-11-21 08:51:14,906 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.26 sec
2024-11-21 08:51:23,114 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.52 sec
MapReduce Total cumulative CPU time: 6 seconds 520 msec
Ended Job = job_1732089968849_2298
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.52 sec   HDFS Read: 2389605 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 520 msec
OK
YN7
Time taken: 27.238 seconds, Fetched: 1 row(s)
hive (khunedbn)>
```

3. **Query :**

```
select a.name, count(max airline_iata) from airlines a join
routes r on a.airline_id = r.airline_id;
```

Output : Giving error

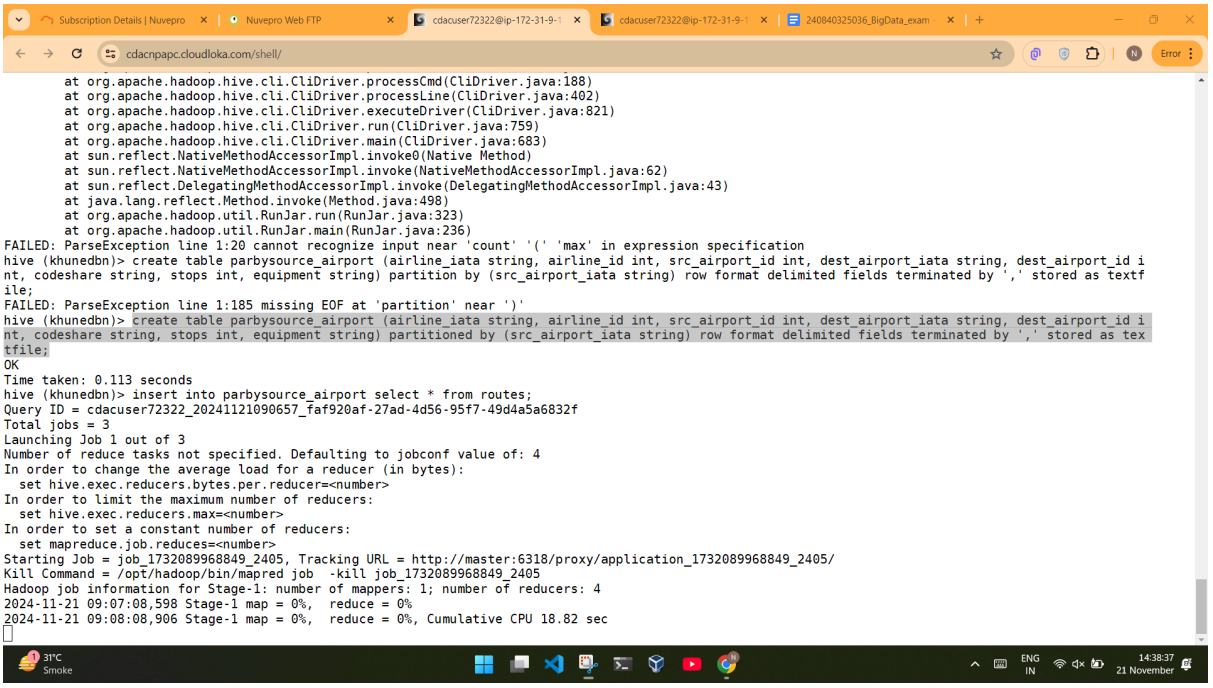
```
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
FAILED: ParseException line 1:20 cannot recognize input near 'count' '(' 'max' in expression specification
hive (khunedbn)> select a.name, count(max airline_iata) from airlines a join routes r on a.airline_id = r.airline_id groupby a.name;
NoViableAltException(361@[423:1: atomExpression : ( constant | ( intervalExpression )=> intervalExpression | castExpression | extractExpression | floor
Expression | caseExpression | whenExpression | ( subQueryExpression )=> ( subQueryExpression ) -> ^( TOK_SUBQUERY_EXPR TOK_SUBQUERY_OP subQueryExpressi
on ) | ( function )=> function | tableOrColumn | expressionsInParenthesis[true, false] );])
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser$DFA36.specialStateTransition(HiveParser_IdentifiersParser.java:34557)
        at org.antlr.runtime.DFA.predict(DFA.java:80)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.atomExpression(HiveParser_IdentifiersParser.java:6930)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceFieldExpression(HiveParser_IdentifiersParser.java:7172)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceUnaryPrefixExpression(HiveParser_IdentifiersParser.java:7671)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceUnarySuffixExpression(HiveParser_IdentifiersParser.java:7727)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceBitwiseXorExpression(HiveParser_IdentifiersParser.java:7889)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceStarExpression(HiveParser_IdentifiersParser.java:8032)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedencePlusExpression(HiveParser_IdentifiersParser.java:8175)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceConcatenateExpression(HiveParser_IdentifiersParser.java:8314)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceAmpersandExpression(HiveParser_IdentifiersParser.java:8524)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceBitwiseOrExpression(HiveParser_IdentifiersParser.java:8661)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceSimilarExpressionMain(HiveParser_IdentifiersParser.java:9144)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceSimilarExpression(HiveParser_IdentifiersParser.java:9040)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceEqualExpression(HiveParser_IdentifiersParser.java:10254)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceNotExpression(HiveParser_IdentifiersParser.java:10541)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceAndExpression(HiveParser_IdentifiersParser.java:10650)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.precedenceOrExpression(HiveParser_IdentifiersParser.java:10791)
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.expression(HiveParser_IdentifiersParser.java:6870)
        at org.apache.hadoop.hive.ql.parse.HiveParser.expression(HiveParser.java:45244)
        at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectItem(HiveParser_SelectClauseParser.java:1720)
        at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectList(HiveParser_SelectClauseParser.java:1234)
        at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectClause(HiveParser_SelectClauseParser.java:1004)
        at org.apache.hadoop.hive.ql.parse.HiveParser.selectClause(HiveParser.java:45290)
        at org.apache.hadoop.hive.ql.parse.HiveParser.atomSelectStatement(HiveParser.java:39777)
        at org.apache.hadoop.hive.ql.parse.HiveParser.selectStatement(HiveParser.java:40044)
        at org.apache.hadoop.hive.ql.parse.HiveParser.regularBody(HiveParser.java:39690)
        at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatementExpressionBody(HiveParser.java:38900)
        at org.apache.hadoop.hive.ql.parse.HiveParser.queryStatementExpression(HiveParser.java:38788)
        at org.apache.hadoop.hive.ql.parse.HiveParser.execStatement(HiveParser.java:2396)
        at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1420)
```

Question 2 :

1. Query :

```
create table parbysource_airport (airline_iata string, airline_id int, src_airport_id int, dest_airport_iata string, dest_airport_id i
nt, codeshare string, stops int, equipment string) partitioned by (src_airport_iata string) row format delimited fields terminated by ',' stored as textfile;
```

Output :



```
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:188)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:402)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        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 org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
FAILED: ParseException line 1:20 cannot recognize input near 'count' '(' 'max' in expression specification
hive (khunedbn)> create table parbysource_airport (airline_iata string, airline_id int, src_airport_id int, dest_airport_iata string, dest_airport_id i
nt, codeshare string, stops int, equipment string) partition by (src_airport_iata string) row format delimited fields terminated by ',' stored as textf
ile;
FAILED: ParseException line 1:185 missing EOF at 'partition' near ')'
hive (khunedbn)> create table parbysource_airport (airline_iata string, airline_id int, src_airport_id int, dest_airport_iata string, dest_airport_id i
nt, codeshare string, stops int, equipment string) partitioned by (src_airport_iata string) row format delimited fields terminated by ',' stored as tex
tfile;
OK
Time taken: 0.113 seconds
hive (khunedbn)> insert into parbysource_airport select * from routes;
Query ID = cdacuser72322_20241121090657_faf920af-27ad-4d56-95f7-49d4a5a6832f
Total jobs = 3
Launching Job 1 out of 3
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_2405, Tracking URL = http://master:6318/proxy/application_1732089968849_2405/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2405
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 09:07:08,598 Stage-1 map = 0%,  reduce = 0%
2024-11-21 09:08:08,906 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 18.82 sec
```

3. Query :

```
select distinct airline_iata from routes_partitioned where src_airport_iata = "LAX";
```

4. Query :

```
describe extended routes_partitioned;
```

SPARK :

Question 1 :

1. RDD command:

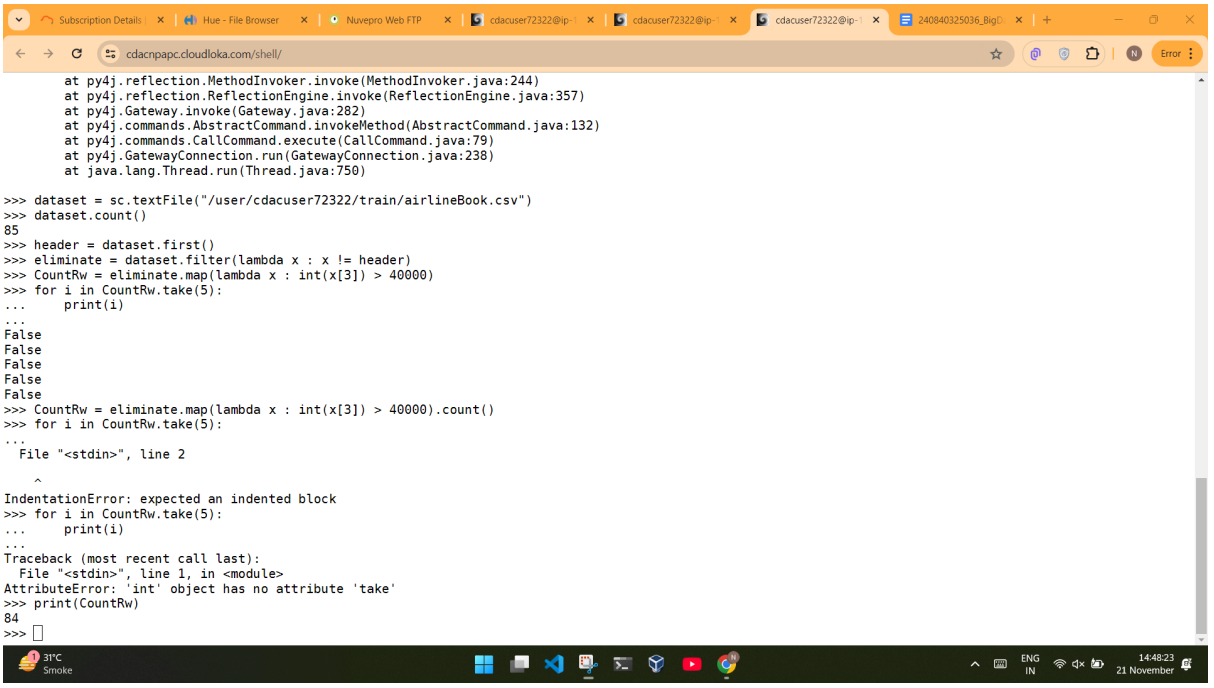
   dataset = sc.textFile("/user/cdacuser72322/train/airlineBook.csv")

- **dataset.count()**
- **header = dataset.first()**
- **eliminate = dataset.filter(lambda x : x != header)**
- **CountRw = eliminate.map(lambda x : int(x[3]) > 40000).count()**

Output :

print(CountRw)

84



```
        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)

>>> dataset = sc.textFile("/user/cdacuser72322/train/airlineBook.csv")
>>> dataset.count()
85
>>> header = dataset.first()
>>> eliminate = dataset.filter(lambda x : x != header)
>>> CountRw = eliminate.map(lambda x : int(x[3]) > 40000)
>>> for i in CountRw.take(5):
...     print(i)
...
False
False
False
False
False
>>> CountRw = eliminate.map(lambda x : int(x[3]) > 40000).count()
>>> for i in CountRw.take(5):
...
  File "<stdin>", line 2

    ^
IndentationError: expected an indented block
>>> for i in CountRw.take(5):
...     print(i)
...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'int' object has no attribute 'take'
>>> print(CountRw)
84
>>>
```

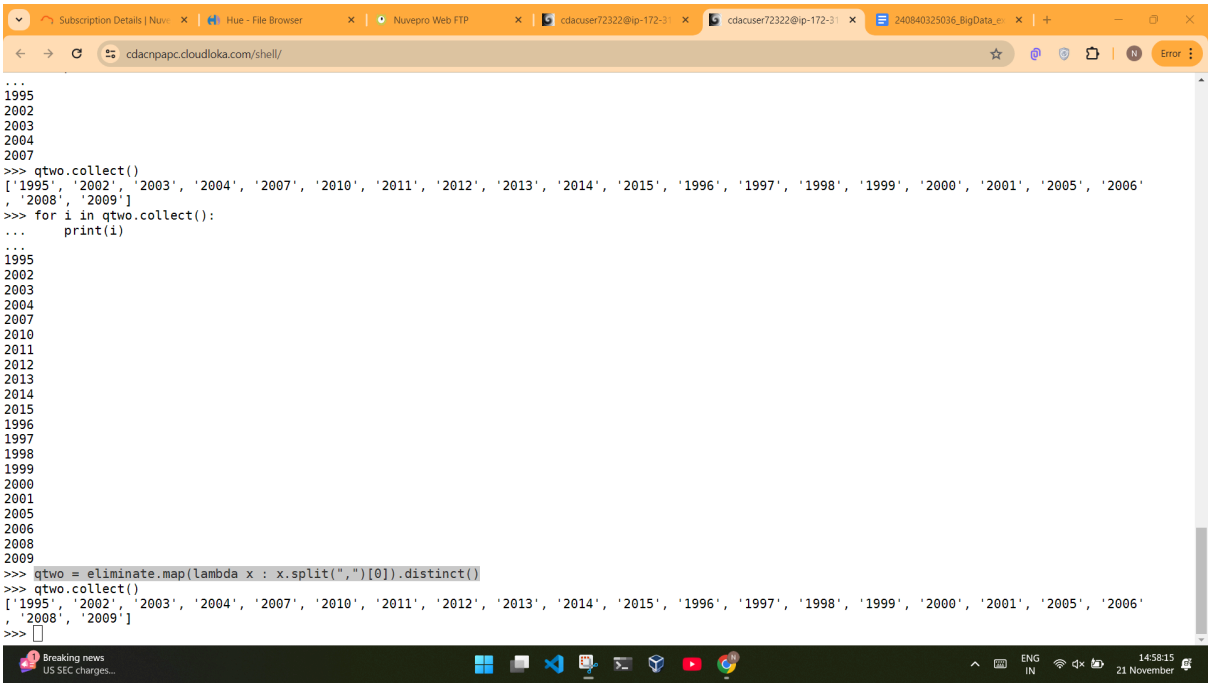
2. **Query :**

qtwo = eliminate.map(lambda x : x.split(",")[0]).distinct()

qtwo.collect()

```
['1995', '2002', '2003', '2004', '2007', '2010',
'2011', '2012', '2013', '2014', '2015', '1996',
'1997', '1998', '1999', '2000', '2001', '2005',
'2006'
, '2008', '2009']
```

Output :



Question 2 :

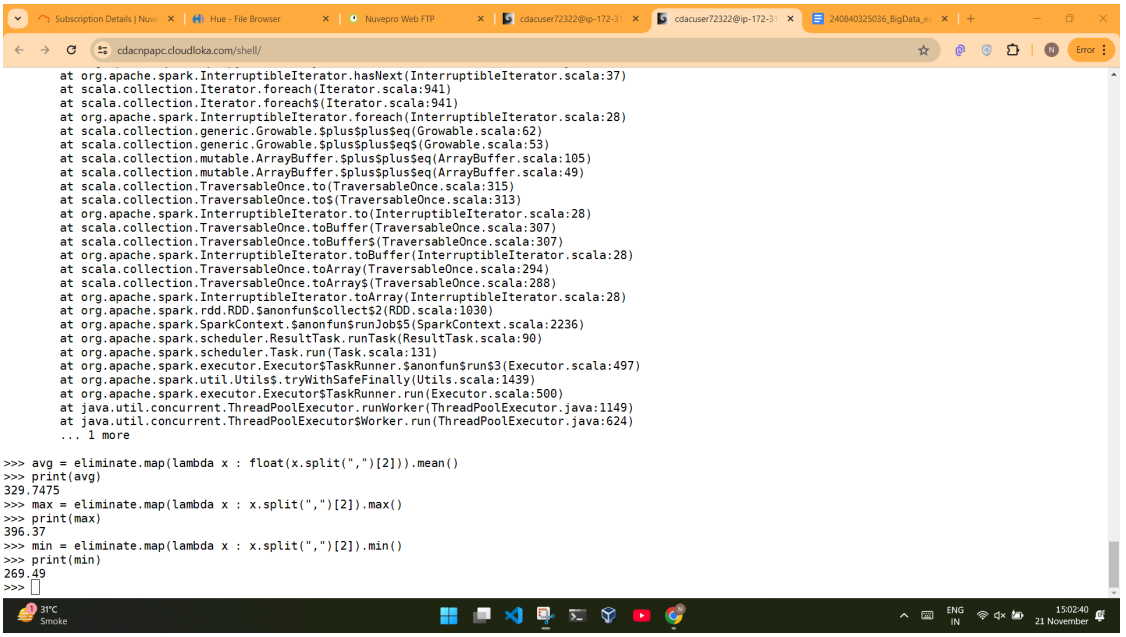
1. **Query :**

```
avg = eliminate.map(lambda x  :
float(x.split(",")[2])).mean()
print(avg)
329.7475

max = eliminate.map(lambda x : x.split(",")[2]).max()
>>> print(max)
396.37

min = eliminate.map(lambda x : x.split(",")[2]).min()
>>> print(min)
269.49
```

Output :



```
        at org.apache.spark.InterruptibleIterator.hasNext(InterruptibleIterator.scala:37)
        at scala.collection.Iterator.foreach(Iterator.scala:941)
        at scala.collection.Iterator.foreach$(Iterator.scala:941)
        at org.apache.spark.InterruptibleIterator.foreach(InterruptibleIterator.scala:28)
        at scala.collection.generic.Growable.$plus$plus$eq(Growable.scala:62)
        at scala.collection.generic.Growable.$plus$plus$eq$(Growable.scala:53)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:105)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:49)
        at scala.collection.TraversableOnce.to(TraversableOnce.scala:315)
        at scala.collection.TraversableOnce.to$(TraversableOnce.scala:313)
        at org.apache.spark.InterruptibleIterator.to(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toBuffer(TraversableOnce.scala:307)
        at scala.collection.TraversableOnce.toBuffer$(TraversableOnce.scala:307)
        at org.apache.spark.InterruptibleIterator.toBuffer(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toArray(TraversableOnce.scala:294)
        at scala.collection.TraversableOnce.toArray$(TraversableOnce.scala:288)
        at org.apache.spark.InterruptibleIterator.toArray(InterruptibleIterator.scala:28)
        at org.apache.spark.rdd.RDD.$anonfun$collect$2(RDD.scala:1030)
        at org.apache.spark.SparkContext.$anonfun$runJob$5(SparkContext.scala:2236)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:131)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Executor.scala:497)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> avg = eliminate.map(lambda x : float(x.split(",")[2])).mean()
>>> print(avg)
329.7475
>>> max = eliminate.map(lambda x : x.split(",")[2]).max()
>>> print(max)
396.37
>>> min = eliminate.map(lambda x : x.split(",")[2]).min()
>>> print(min)
269.49
>>>
```

2. **Query :**

```
countrow = eliminate.filter(lambda x :
float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
75
```

Output :

```
>>> print(avg)
329.7475
>>> max = eliminate.map(lambda x : x.split(",")[2]).max()
>>> print(max)
396.37
>>> min = eliminate.map(lambda x : x.split(",")[2]).min()
>>> print(min)
269.49
>>> for i in eliminate.take(5):
...     print(i)
...
1995,1,296.9,46561
1995,2,296.8,37443
1995,3,287.51,34128
1995,4,287.78,30388
1996,1,283.97,47808
>>> countrow = eliminate.map(lambda x : float(x.split(",")[2]) > 290).count()
>>> print(countrow)
84
>>> countrow = eliminate.map(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.map(lambda x : float(x.split(",")[2]) > 2.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
75
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 29.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 300).count()
>>> print(countrow)
69
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
75
>>>
```

3. Query :

```
qthree = eliminate.map(lambda x : (x.split(",")[0],
x.split(",")[3]))

reduce = qthree.reduceByKey(lambda x,y : x + y)

for i in reduce.take(5):
...     print(i)
```

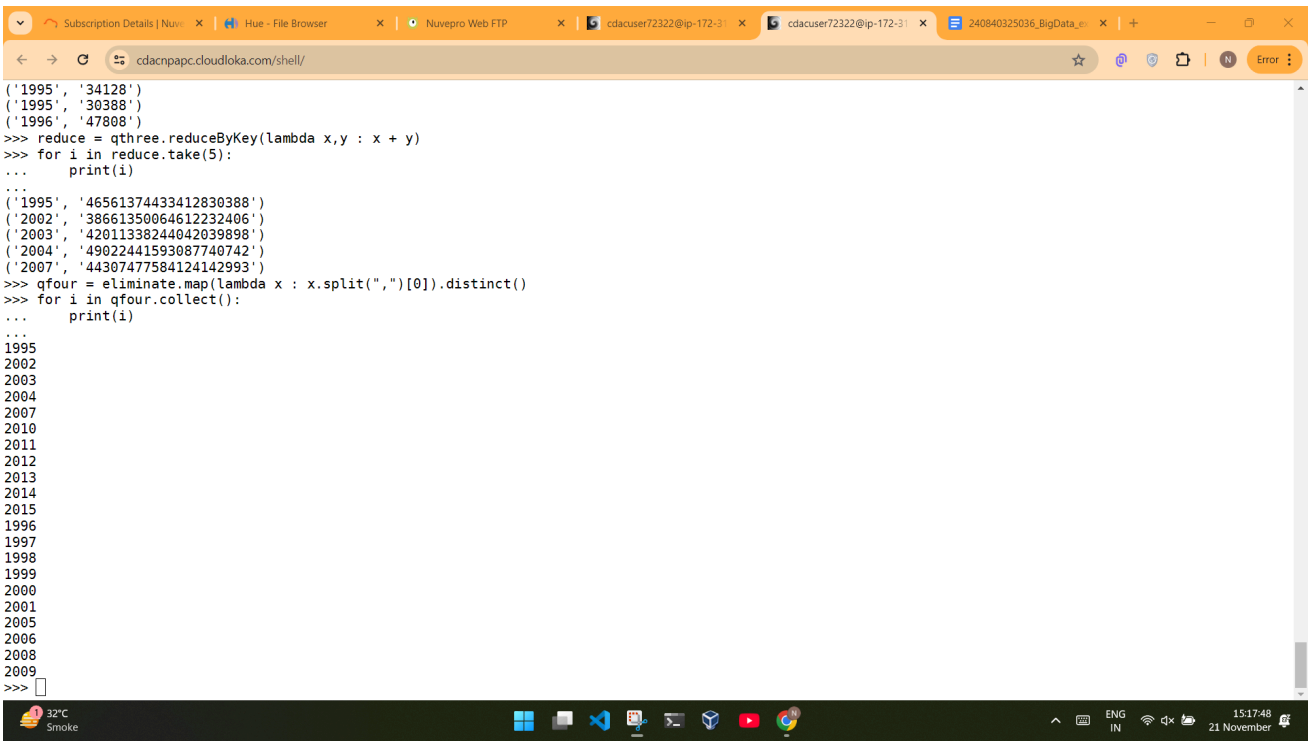
```
84
>>> countrow = eliminate.map(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.map(lambda x : float(x.split(",")[2]) > 2.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
75
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 29.00).count()
>>> print(countrow)
84
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 300).count()
>>> print(countrow)
69
>>> countrow = eliminate.filter(lambda x : float(x.split(",")[2]) > 290.00).count()
>>> print(countrow)
75
>>> qthree = eliminate.map(lambda x : (x.split(",")[0], x.split(",")[3]))
>>> for i in qthree.take(5):
...     print(i)
...
('1995', '46561')
('1995', '37443')
('1995', '34128')
('1995', '30388')
('1996', '47808')
>>> reduce = qthree.reduceByKey(lambda x,y : x + y)
>>> for i in reduce.take(5):
...     print(i)
...
('1995', '46561374433412830388')
('2002', '38661350064612232406')
('2003', '42011338244042039898')
('2004', '49022441593087740742')
('2007', '44307477584124142993')
>>>
```

4. Query :

```
qfour = eliminate.map(lambda x :
x.split(",")[0]).distinct()

for i in qfour.collect():
...     print(i)
```

Output :



```
('1995', '34128')
('1995', '30388')
('1996', '47808')
>>> reduce = qthree.reduceByKey(lambda x,y : x + y)
>>> for i in reduce.take(5):
...     print(i)
...
('1995', '46561374433412830388')
('2002', '38661350064612232406')
('2003', '42011338244042039898')
('2004', '49022441593087740742')
('2007', '44307477584124142993')
>>> qfour = eliminate.map(lambda x : x.split(",")[0]).distinct()
>>> for i in qfour.collect():
...     print(i)
...
1995
2002
2003
2004
2007
2010
2011
2012
2013
2014
2015
1996
1997
1998
1999
2000
2001
2005
2006
2008
2009
>>>
```

5. Query :

```
qfive = eliminate.map(lambda x : (x.split(",")[0],
x.split(",")[2], x.split(",")[3]))

combine = qfive.map(lambda x : (x[0], x[2] * x[3]))

cummulative = combine.reduceByKey(lambda x,y : x + y)
```

Output : giving error

```
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> 24/11/21 09:57:47 WARN TaskSetManager: Lost task 1.3 in stage 45.0 (TID 90) (dn2.cloudloka.com executor 1): TaskKilled (Stage cancelled)

>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> qfive = eliminate.map(lambda x : (x.split(",")[0], x.split(",")[2], x.split(",")[3]))
>>> combine = qfive.map(lambda x : (x[0], x[2] * x[3]))
>>> cummulative = combine.reduceByKey(lambda x,y : x + y)
>>> for i in cummulative.take(5):
...     print(i)
...
24/11/21 10:01:49 WARN TaskSetManager: Lost task 0.0 in stage 47.0 (TID 91) (dn2.cloudloka.com executor 1): org.apache.spark.api.python.PythonException
: Traceback (most recent call last):
  File "/opt/spark-3.1.2/python/pyspark/worker.py", line 604, in main
    process()
  File "/opt/spark-3.1.2/python/pyspark/worker.py", line 594, in process
    out_iter = func(split_index, iterator)
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 2916, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 2916, in pipeline_func
    return func(split, prev_func(split, iterator))
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 418, in func
    return f(iterator)
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 2144, in combineLocally
    merger.mergeValues(iterator)
  File "/opt/spark-3.1.2/python/pyspark/shuffle.py", line 240, in mergeValues
    for k, v in iterator:
  File "/opt/spark-3.1.2/python/pyspark/util.py", line 73, in wrapper
```