

Abhay Patil Big Data Exam

ID : 240840325002

HIVE : (20 mrks)

Q1.

A]

```
Subscription Details | Nuvepro x cdacuser90115@ip-172-31-9-1 x cdacuser90115@ip-172-31-9-1 x +
cdacnpac.cloudloka.com/shell/
file des rns Airport
Time taken: 99.038 seconds, Fetched: 3254 row(s)
hive (abyyairproject)> select distinct(ap.name)
>
> from airport ap
>
> join routes r1
>
> on r1.src_id = ap.airport_id
>
> join routes r2
>
> on r2.dest_id = ap.airport_id
>
> where r1.src_id = r2.dest_id
>
> limit 10;
No Stats for abyyairproject@airport, Columns: airport_id, name
No Stats for abyyairproject@routes, Columns: src_id
No Stats for abyyairproject@routes, Columns: dest_id
Query ID = cdacuser90115_20241121084703_a2b7d650-23c7-43fe-a0d1-0a427154ea4c
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_2277, Tracking URL = http://master:6318/proxy/application_1732089968849_2277/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2277
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 4
2024-11-21 08:47:17,477 Stage-1 map = 0%, reduce = 0%
2024-11-21 08:47:26,704 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 6.05 sec
2024-11-21 08:47:27,728 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 13.06 sec
2024-11-21 08:47:32,841 Stage-1 map = 100%, reduce = 50%, Cumulative CPU 20.71 sec
2024-11-21 08:47:34,887 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 24.41 sec
2024-11-21 08:47:34,887 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 28.38 sec
master:6318/proxy/application_1732089968849_2277/
```

```
Subscription Details | Nuvepro x cdacuser90115@ip-172-31-9-1 x cdacuser90115@ip-172-31-9-1 x +
cdacnpac.cloudloka.com/shell/
2024-11-21 08:48:08,518 Stage-2 map = 100%, reduce = 50%, Cumulative CPU 26.69 sec
2024-11-21 08:48:10,565 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 34.7 sec
MapReduce Total cumulative CPU time: 34 seconds 700 msec
Ended Job = job_1732089968849_2282
Launching Job 3 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_2285, Tracking URL = http://master:6318/proxy/application_1732089968849_2285/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2285
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 4
2024-11-21 08:48:21,992 Stage-3 map = 0%, reduce = 0%
2024-11-21 08:48:28,143 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.1 sec
2024-11-21 08:48:34,278 Stage-3 map = 100%, reduce = 25%, Cumulative CPU 6.22 sec
2024-11-21 08:48:36,322 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 15.67 sec
MapReduce Total cumulative CPU time: 15 seconds 670 msec
Ended Job = job_1732089968849_2285
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 28.38 sec HDFS Read: 3148610 HDFS Write: 2543075 SUCCESS
Stage-Stage-2: Map: 3 Reduce: 4 Cumulative CPU: 34.7 sec HDFS Read: 4954822 HDFS Write: 105775 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 4 Cumulative CPU: 15.67 sec HDFS Read: 124648 HDFS Write: 587 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 18 seconds 750 msec
OK
Aarhus
Abakan
Abbotsford
A Coruna
Abadan
Abdul Rachman Saleh
Abel Santamaria
Aalborg
Aasiaat
Aberdeen Regional Airport
Time taken: 95.573 seconds, Fetched: 10 row(s)
hive (abyyairproject)>
```

select distinct(ap.name)

```
from airport ap
join routes r1
on r1.src_id = ap.airport_id
join routes r2
on r2.dest_id = ap.airport_id
where r1.src_id = r2.dest_id
limit 10;
```

B]

```
with (select max(equipment) from routes) as max_count,
(select equipment, count(equipment)
from routes
group by equipment
having count(equipment) = max_count);
```

“Can't get the output due to crashing of the hive terminal.”

C]

```
select a.name, count(*)
from airlines a
join routes r
on a.airline_id = r.airline_id
group by a.name
having max(count(*)) = count(*);
```

“Can't get the output due to crashing of the hive terminal.”

Q2.

A]

```
stops, r.equipment, r.src_ap from routes r distribute by src_ap;
FAILED: SemanticException [Error 10002]: Line 1:120 Invalid column reference 'desc_id'
hive (abyyairproject)> insert overwrite table partitioned_routes partition(src_ap) select r.airline_iata, r.airline_id, r.src_id, r.dest_ap, r.dest_id, r.codeshare, r.
stops, r.equipment, r.src_ap from routes r distribute by src_ap;
Query ID = cdacuser90115_20241121094037_7ddae755-e2b6-4856-ade2-cdac3dd2278a
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.reducers=<number>
Starting Job = job_1732089968849_2527, Tracking URL = http://master:6318/proxy/application_1732089968849_2527/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2527
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 09:40:46,944 Stage-1 map = 0%, reduce = 0%
2024-11-21 09:40:55,153 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.19 sec
2024-11-21 09:41:12,544 Stage-1 map = 100%, reduce = 33%, Cumulative CPU 20.45 sec
2024-11-21 09:41:19,563 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 37.15 sec
2024-11-21 09:41:23,790 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 51.58 sec
2024-11-21 09:41:25,833 Stage-1 map = 100%, reduce = 83%, Cumulative CPU 54.34 sec
2024-11-21 09:41:29,918 Stage-1 map = 100%, reduce = 92%, Cumulative CPU 55.97 sec
2024-11-21 09:41:31,963 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 57.64 sec
2024-11-21 09:42:32,161 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 82.08 sec
2024-11-21 09:43:32,344 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 94.25 sec
MapReduce Total cumulative CPU time: 1 minutes 34 seconds 250 msec
Ended Job = job_1732089968849_2527
Loading data to table abyyairproject.partitioned_routes partition (src_ap=null)

Time taken to load dynamic partitions: 147.811 seconds
Time taken for adding to write entity : 0.07 seconds
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive ql.exec.StatsTask
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduces: 4 Cumulative CPU: 94.25 sec HDFS Read: 2435036 HDFS Write: 10398240 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 34 seconds 250 msec
hive (abyyairproject)> |
```

File Name	Owner	Group	Permissions	Last Modified
.	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:46 AM
src_ap=AAE	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAL	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAN	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAQ	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAR	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAT	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAX	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=AAY	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABA	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABB	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABD	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABE	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABI	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABJ	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABL	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABM	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM
src_ap=ABQ	cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:43 AM

create table partitioned_routes (airline_iata string, airline_id int, src_id int, dest_ap string, dest_id int,

codeshare string, stops int, equipment string)

partitioned by (src_ap string)

row format delimited

fields terminated by ','

stored as textfile;

insert overwrite table partitioned_routes partition(src_ap) select r.airline_iata, r.airline_id, r.src_id, r.dest_ap, r.dest_id, r.codeshare, r.stops, r.equipment, r.src_ap from routes r distribute by src_ap;

B]

```

FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.StatsTask
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 4 Cumulative CPU: 94.25 sec HDFS Read: 2435036 HDFS Write: 10398240 SUCCESS
Total MapReduce CPU Time
> insert overwrite table partitioned_routes partition(src_ap) select r.airline_iata, r.airline_id, r.src_id, r.dest_ap, r.dest_id, r.codeshare, r.
stops, r.equipment, r.src_ap from routes r where r.src_ap = "JFK" distribute by src_ap;
Query ID = cdacuser90115_20241121095231_dc36c455-0857-45c1-803f-1f28c9e24251
Total jobs = 2
Launching Job 1 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_2570, Tracking URL = http://master:6318/proxy/application_1732089968849_2570/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2570
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 09:52:44,532 Stage-1 map = 0%, reduce = 0%
2024-11-21 09:52:52,163 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.46 sec
2024-11-21 09:53:46,204 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 21.59 sec
MapReduce Total cumulative CPU time: 21 seconds 590 msec
Ended Job = job_1732089968849_2570
Launching Job 2 out of 2
Loading data to table abyyyyairproject.partitioned_routes partition (src_ap=null)
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>

Time taken to load dynamic partitions: 0.161 seconds
Time taken for adding to write entity : 0.0 seconds
Starting Job = job_1732089968849_2572, Tracking URL = http://master:6318/proxy/application_1732089968849_2572/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2572
  
```

Search data and saved documents...

File Browser

Search for file name

Actions Delete forever

Upload New

Home / user / hive / warehouse / abyyyyairproject.db / partitioned_routes / src_ap=JFK

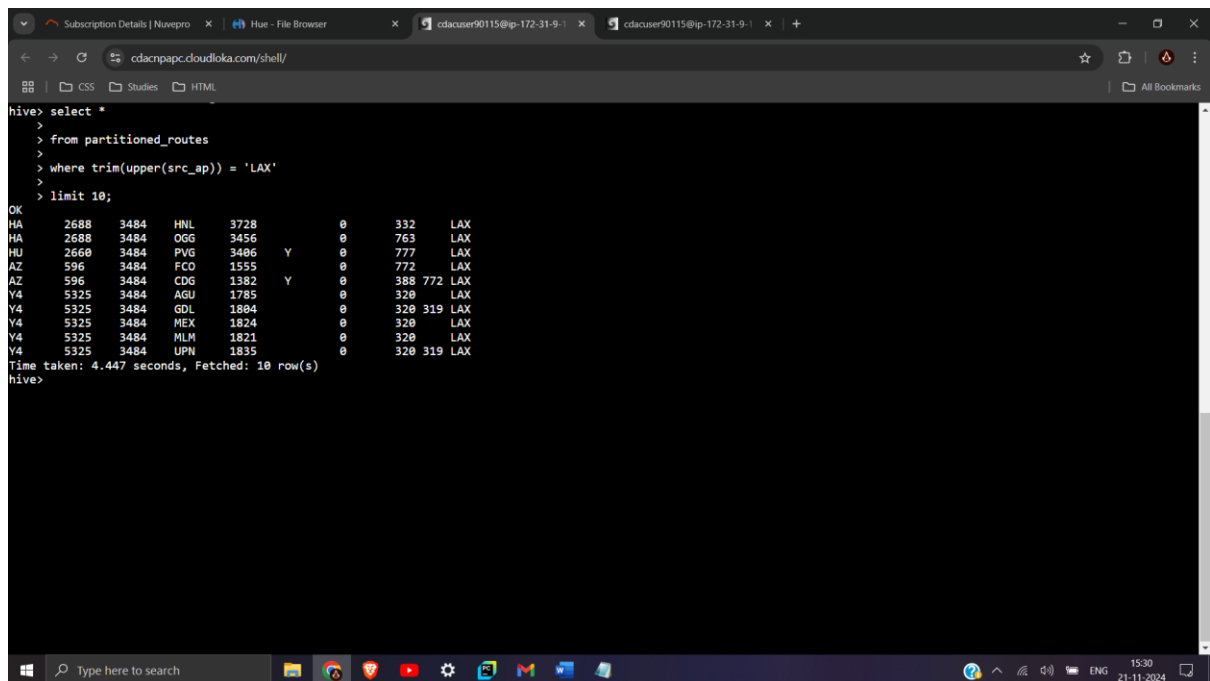
Name	Size	User	Group	Permissions	Date
f		cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:54 AM
.		cdacuser90115	hive	drwxr-xr-x	November 21, 2024 01:52 AM
000003_0	13.5 KB	cdacuser90115	hive	-rw-r--r--	November 21, 2024 01:52 AM

Show 45 of 1 items

Page 1 of 1

insert overwrite table partitioned_routes partition(src_ap) select r.airline_iata, r.airline_id, r.src_id, r.dest_ap, r.dest_id, r.codeshare, r.stops, r.equipment, r.src_ap from routes r where r.src_ap = "JFK" distribute by src_ap;

C]



```
hive> select *
>
> from partitioned_routes
>
> where trim(upper(src_ap)) = 'LAX'
>
> limit 10;
OK
2688 3484 HNL 3728 0 332 LAX
2688 3484 OGG 3456 0 763 LAX
2660 3484 PVG 3406 Y 0 777 LAX
596 3484 FCO 1555 0 772 LAX
596 3484 CDG 1382 Y 0 388 772 LAX
5325 3484 AGU 1785 0 320 LAX
5325 3484 GDL 1804 0 320 319 LAX
5325 3484 MEX 1824 0 320 LAX
5325 3484 MLM 1821 0 320 LAX
5325 3484 UPN 1835 0 320 319 LAX
Time taken: 4.447 seconds, Fetched: 10 row(s)
hive>
```

```
select *
from partitioned_routes
where trim(upper(src_ap)) = 'LAX'
limit 10;
```

D]

SPARK : (20 mrks)

Q1.

A]

```
Subscription Details | Nuvepro x cdacuser90115@ip-172-31-9-1 x cdacuser90115@ip-172-31-9-1 x +
cdacnpac.cloudloka.com/shell/
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4043. Attempting port 4044.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4044. Attempting port 4045.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4045. Attempting port 4046.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4046. Attempting port 4047.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4047. Attempting port 4048.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4048. Attempting port 4049.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4049. Attempting port 4050.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4050. Attempting port 4051.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4051. Attempting port 4052.
24/11/21 08:40:30 WARN Utils: Service 'SparkUI' could not bind on port 4052. Attempting port 4053.
24/11/21 08:40:30 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.

>>> data = sc.textFile("/user/cdacuser90115/airlines.csv")
>>> header = data.first()
>>> clean = data.filter(lambda a : a != header)
>>> for i in clean.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
>>>
>>> split = clean.map(lambda a : (a.split(",")[0], a.split(",")[1], a.split(",")[2], a.split(",")[3]))
>>> intRDD = split.map(lambda a : (a[0], a[1], float(a[2]), int(a[3])))
>>>
>>> for i in intRDD.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)
>>> count_no_rows = intRDD.filter(lambda a : a[3] > 40000).count()
>>> print(count_no_rows)
88
>>>
```

```
>>> count_no_rows = intRDD.filter(lambda a : a[3] > 40000).count()
```

```
>>> print(count_no_rows)
```

B)

```
Subscription Details | Nuvepro x cdacuser90115@ip-172-31-9-1 x cdacuser90115@ip-172-31-9-1 x +
cdacnpac.cloudloka.com/shell/
...     print(line)
...
1995
1995
1995
1995
1995
1996
>>>
>>>
>>>
>>>
>>> unique_years = intRDD.map(lambda a : a[0]).distinct()
>>>
>>> for line in unique_years.collect():
...     print(line)
...
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
1995
1997
1999
2001
2003
2005
2007
2009
2011
2013
2015
>>>
```

```
>>> unique_years = intRDD.map(lambda a : a[0]).distinct()
```

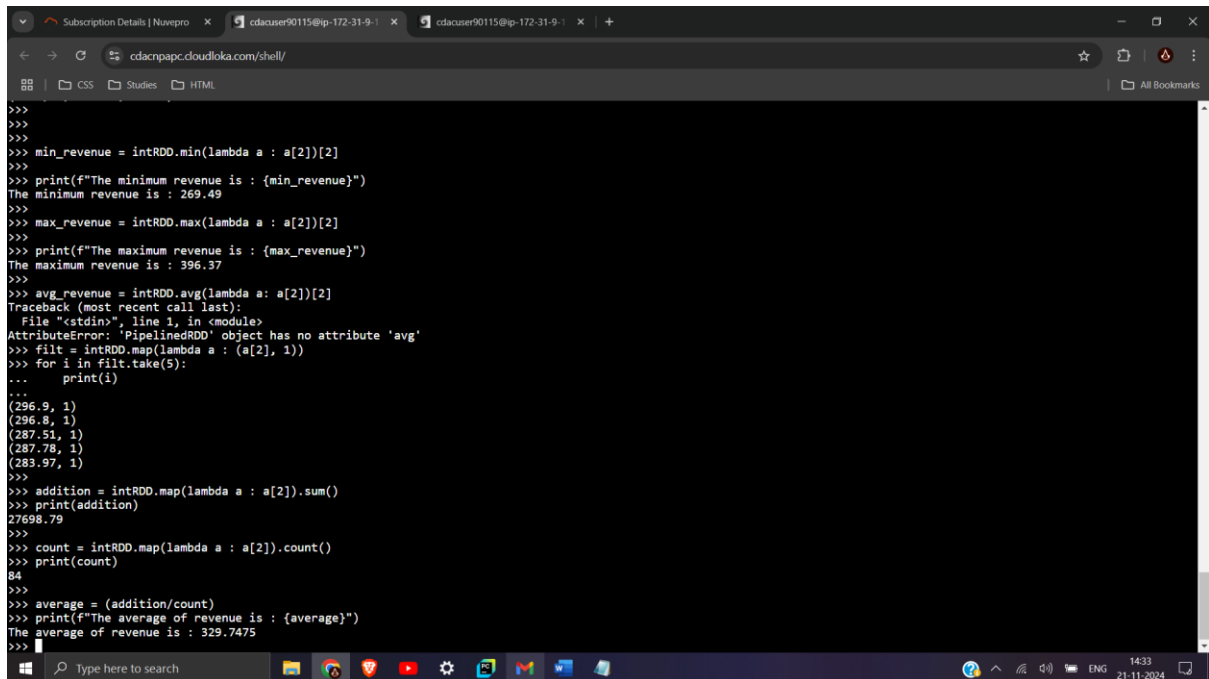
```
>>>
```

```
>>> for line in unique_years.collect():
```

```
...     print(line)
```

Q2.

A]



The screenshot shows a Jupyter Notebook terminal window with the following code and output:

```
>>>
>>>
>>> min_revenue = intrRDD.min(lambda a : a[2])[2]
>>> print(f"The minimum revenue is : {min_revenue}")
The minimum revenue is : 269.49
>>>
>>> max_revenue = intrRDD.max(lambda a : a[2])[2]
>>>
>>> print(f"The maximum revenue is : {max_revenue}")
The maximum revenue is : 396.37
>>>
>>> avg_revenue = intrRDD.avg(lambda a : a[2])[2]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'PipelinedRDD' object has no attribute 'avg'
>>> filt = intrRDD.map(lambda a : (a[2], 1))
>>> for i in filt.take(5):
...     print(i)
...
(296.9, 1)
(296.8, 1)
(287.51, 1)
(287.78, 1)
(283.97, 1)
>>>
>>> addition = intrRDD.map(lambda a : a[2]).sum()
>>> print(addition)
27698.79
>>>
>>> count = intrRDD.map(lambda a : a[2]).count()
>>> print(count)
84
>>>
>>> average = (addition/count)
>>> print(f"The average of revenue is : {average}")
The average of revenue is : 329.7475
>>>
```

```
>>> min_revenue = intrRDD.min(lambda a : a[2])[2]

>>>

>>> print(f"The minimum revenue is : {min_revenue}")

The minimum revenue is : 269.49

>>>

>>> max_revenue = intrRDD.max(lambda a : a[2])[2]

>>>

>>> print(f"The maximum revenue is : {max_revenue}")

The maximum revenue is : 396.37

>>> addition = intrRDD.map(lambda a : a[2]).sum()

>>> print(addition)

27698.79

>>>

>>> count = intrRDD.map(lambda a : a[2]).count()

>>> print(count)
```

84

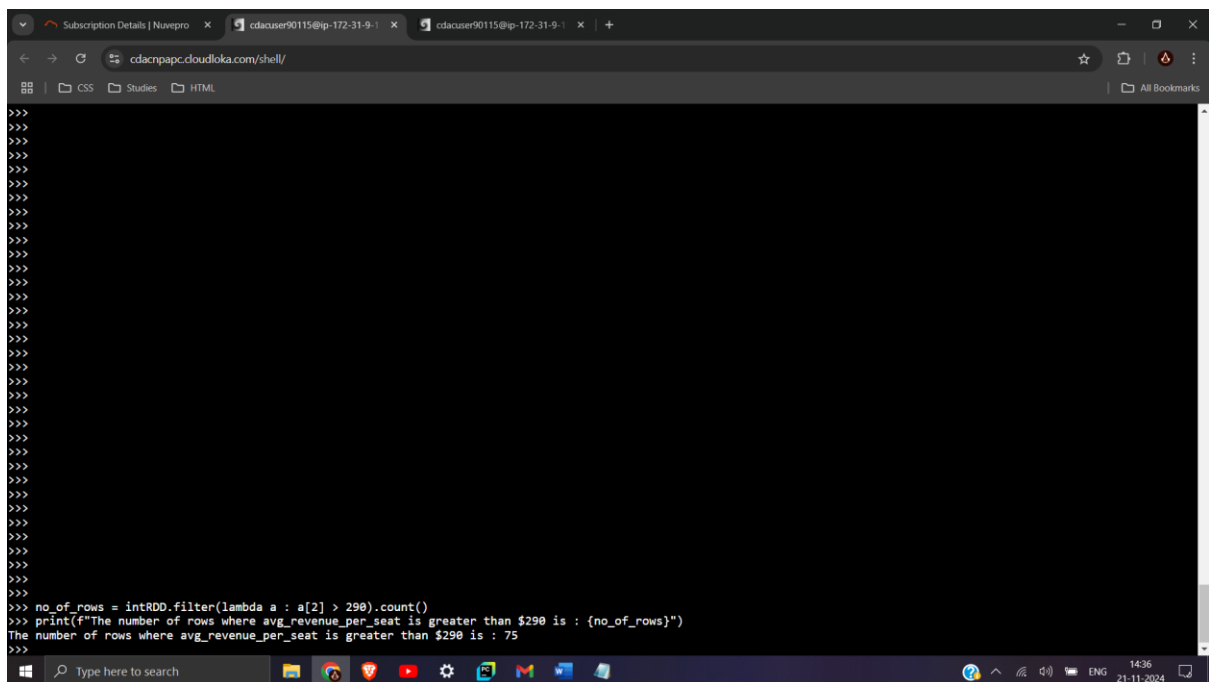
```
>>>
```

```
>>> average = (addition/count)
```

```
>>> print(f"The average of revenue is : {average}")
```

The average of revenue is : 329.7475

B]

A screenshot of a web browser window displaying a Jupyter Notebook. The browser has three tabs: 'Subscription Details | Nuvepro', 'cdacuser90115@ip-172-31-9-1', and 'cdacuser90115@ip-172-31-9-1'. The address bar shows 'cdacnpac.cloudloka.com/shell/'. The notebook interface shows a series of empty code cells, followed by a final cell containing the following code:

```
>>> no_of_rows = intRDD.filter(lambda a : a[2] > 290).count()
>>> print(f"The number of rows where avg_revenue_per_seat is greater than $290 is : {no_of_rows}")
The number of rows where avg_revenue_per_seat is greater than $290 is : 75
>>>
```

The Windows taskbar is visible at the bottom of the browser window.

```
>>> no_of_rows = intRDD.filter(lambda a : a[2] > 290).count()
```

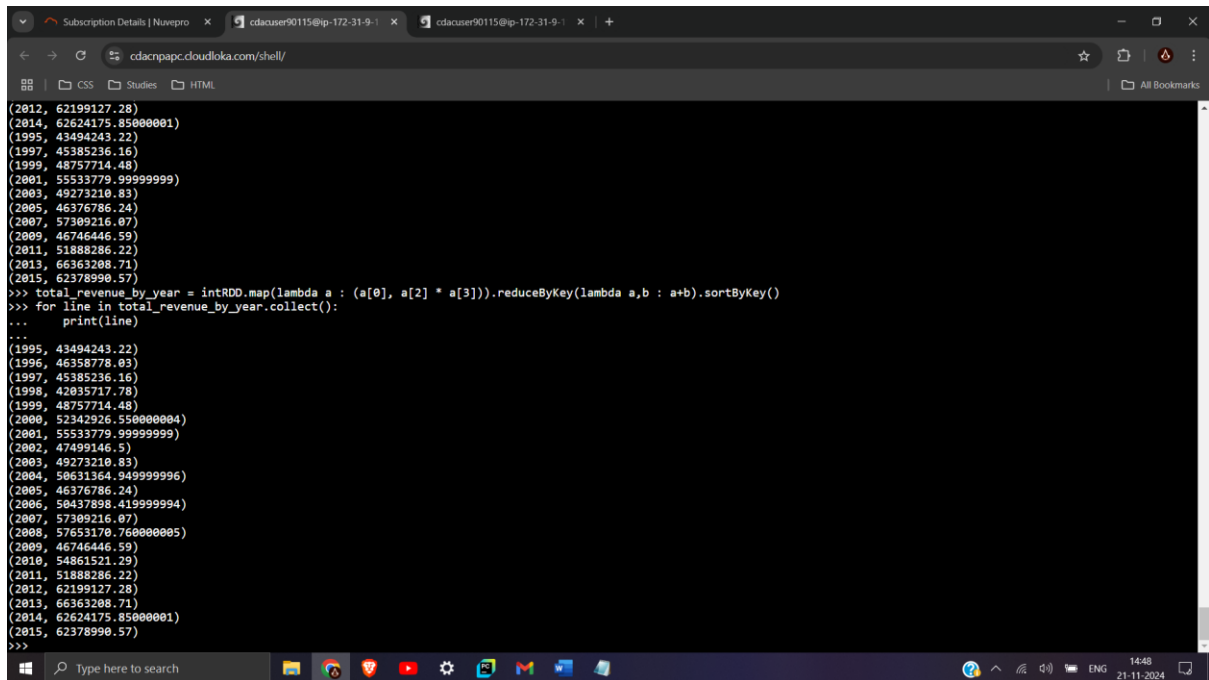
```
>>> print(f"The number of rows where avg_revenue_per_seat is greater than $290 is : {no_of_rows}")
```

The number of rows where avg_revenue_per_seat is greater than \$290 is : 75

C]

List of all distinct years in the dataset is : [1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014, 1995, 1997, 1999, 2001, 2003, 2005, 2007, 2009, 2011, 2013, 2015]

E]



The screenshot shows a terminal window with a dark background. The top part displays a list of tuples representing years and revenue values, sorted by year. The bottom part shows the execution of Spark SQL code to calculate total revenue by year.

```
(2012, 62199127.28)
(2014, 62624175.85000001)
(1995, 43494243.22)
(1997, 45385236.16)
(1999, 48757714.48)
(2001, 55533779.99999999)
(2003, 49273210.83)
(2005, 46376786.24)
(2007, 57309216.07)
(2009, 46746446.59)
(2011, 51888286.22)
(2013, 66363208.71)
(2015, 62378990.57)
>>> total_revenue_by_year = intRDD.map(lambda a : (a[0], a[2] * a[3])).reduceByKey(lambda a,b : a+b).sortByKey()
>>> for line in total_revenue_by_year.collect():
...     print(line)
...
(1995, 43494243.22)
(1996, 46358778.03)
(1997, 45385236.16)
(1998, 42035717.78)
(1999, 48757714.48)
(2000, 52342926.550000004)
(2001, 55533779.99999999)
(2002, 47490146.5)
(2003, 49273210.83)
(2004, 50631364.949999996)
(2005, 46376786.24)
(2006, 50437898.419999994)
(2007, 57309216.07)
(2008, 57653170.760000005)
(2009, 46746446.59)
(2010, 54861521.29)
(2011, 51888286.22)
(2012, 62199127.28)
(2013, 66363208.71)
(2014, 62624175.85000001)
(2015, 62378990.57)
>>>
```

```
>>> total_revenue_by_year = intRDD.map(lambda a : (a[0], a[2] * a[3])).reduceByKey(lambda a,b : a+b).sortByKey()
```

```
>>> for line in total_revenue_by_year.collect():
```

```
...     print(line)
```

```
...
```