

BIG DATA MODULE END EXAM

Name : Atharva Nagesh Jade

ID : 240840325017

Q.1

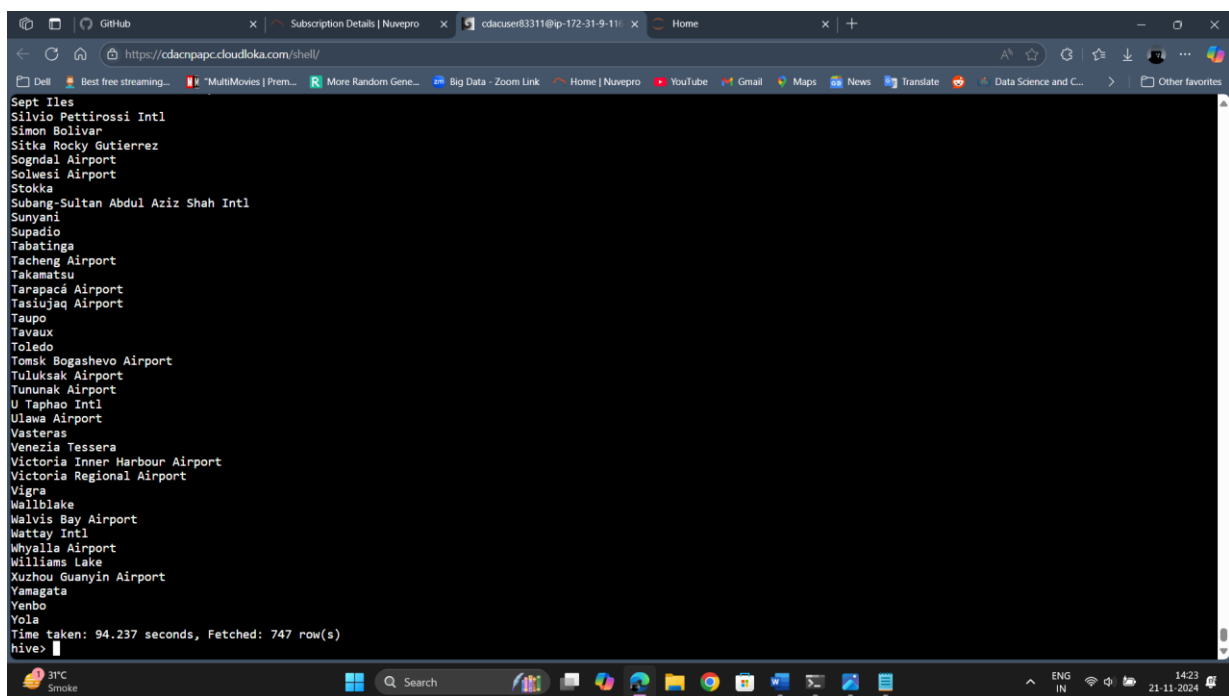
1]

select a.name from routes r join airports a

> on r.src_airport_id = a.airport_id

> join airports a2

> on r.dest_airport_id = a2.airport_id;



```
Sept Iles
Silvio Petrirossi Intl
Simon Bolivar
Sitka Rocky Gutierrez
Sogndal Airport
Solwesi Airport
Stokka
Subang-Sultan Abdul Aziz Shah Intl
Sunyani
Supadio
Tabatinga
Tacheng Airport
Takamatsu
Tanapacá Airport
Tasiuq Airport
Taupo
Tavaux
Toledo
Tomsok Bogashevo Airport
Tulaksak Airport
Tununak Airport
U Taphao Intl
Uluwatu Airport
Vasteras
Venezia Tessera
Victoria Inner Harbour Airport
Victoria Regional Airport
Vigra
Wallblake
Walvis Bay Airport
Wattay Intl
Whyllis Airport
Williams Lake
Xuzhou Guanyin Airport
Yamagata
Yenbo
Yola
Yola
Time taken: 94.237 seconds, Fetched: 747 row(s)
hive>
```

2]

select equipment,count(*),src_airport_iata,dest_airport_iata

> from routes group by src_airport_iata,dest_airport_iata,equipment

> limit 10;

Used limit 10 because output was lengthy

```
https://cdacnpapc.cloudloka.com/shell/

> select equipment,count(*),src_airport_iata,dest_airport_iata from routes group by src_airport_iata,dest_airport_iata;
FAILED: SemanticException [Error 10025]: Line 1:7 Expression not in GROUP BY key 'equipment'
hive> select equipment,count(*),src_airport_iata,dest_airport_iata from routes group by src_airport_iata,dest_airport_iata,equipment limit 10;
Query ID = cdacuser83311_20241121101853_a68167b2-e910-4d00-9231-82c7b93e72ba
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_2659, Tracking URL = http://master:6318/proxy/application_1732089968849_2659/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2659
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 10:19:09,610 Stage-1 map = 0%, reduce = 0%, Cumulative CPU 6.16 sec
2024-11-21 10:19:11,805 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.47 sec
2024-11-21 10:19:18,957 Stage-1 map = 100%, reduce = 25%, Cumulative CPU 15.91 sec
2024-11-21 10:19:19,978 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 19.25 sec
2024-11-21 10:19:20,999 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 19.25 sec
MapReduce Total cumulative CPU time: 19 seconds 250 msec
Ended Job = job_1732089968849_2659
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 4 Cumulative CPU: 19.25 sec HDFS Read: 2415859 HDFS Write: 616 SUCCESS
Total MapReduce CPU Time Spent: 19 seconds 250 msec
OK
FRJ 1 AAL AAR
ATR 1 AAE ORN
736 1 AAE ORY
738 ATR 736 1 AAE ALG
319 1 AAE MRS
319 1 AAE ORY
738 1 AAE CDG
738 1 AAE IST
738 1 AAE LVS
738 1 AAE MRS
Time taken: 29.145 seconds, Fetched: 10 row(s)
hive>
```

3]

```
select count(al.airline_id),al.name from airlines al join routes r on a.airline_id = r.airline_id
order by al.airline_id desc limit 1
```

Question 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 * from Routes r
DISTRIBUTE By src_airport_iata;
```

cdacnpapc.doudloka.com8132/hue/filebrowser/view=%2Fuser%2Fcdacuser83311%2Fuser/hive/warehouse/cdac_atharva.db/source_airport

Best free streaming...MultiMovies | Prem...More Random Gene...Big Data - Zoom LinkHome | NoveproYouTubeGmailMapsNewsTranslateData Science and C...Other favorites

Search data and saved documents...

Jobs

File Browser

Search for file name

ActionsDelete forever

UploadNew

Home / user / hive / warehouse / cdac_atharva.db / source_airport

	Name	Size	User	Group	Permissions	Date
	↑		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:49 AM
	.		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:58 AM
	src_airport_lata=100 318		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:54 AM
	src_airport_lata=100 DH4		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:54 AM
	src_airport_lata=310 330		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:53 AM
	src_airport_lata=318 733 AT7		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:54 AM
	src_airport_lata=319 100 ER4		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:57 AM
	src_airport_lata=319 318 320 321		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:55 AM
	src_airport_lata=319 320 AR1 321 32A		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:55 AM
	src_airport_lata=319 321 320 E90		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:55 AM
	src_airport_lata=319 321 320 E95		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:53 AM
	src_airport_lata=319 321 32B		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:56 AM
	src_airport_lata=319 32S 321 E75 320		cdacuser83311	hive	drwxr-xr-x	November 21, 2024 01:56 AM

Hive

Databases (480)

Filter databases...

01sept

1sept2024_anup

02oct2024

2sep2024

2sep_abhi

2sep_abhi_new

2sept2024_postoffice

3octad

3sept_2024

7sep24

9thsept

10october

24aug2024

27_aug_2024_devang

27_aug_2024_pavan

27_aug_2024_prawn

27_aug_2024_sumedh

27_aug_chirag

27aug24

27aug24_avinash

27aug24_nn

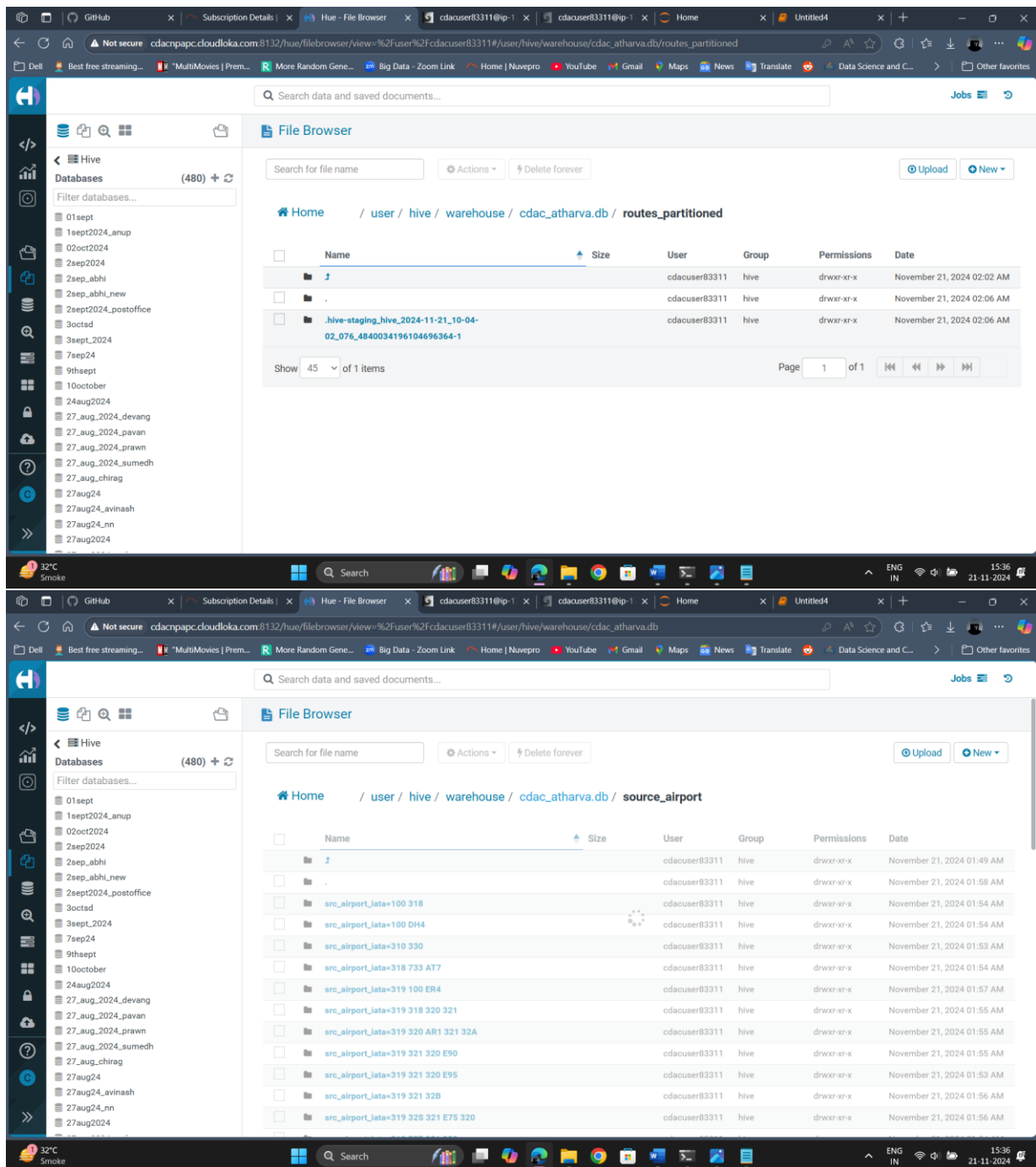
27aug2024

32°C Smoke

Search

ENG IN

15:29 21-11-2024



Hive and Hue were taking time that's why I only added queries

2]

create table routes_partitioned(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 routes_partitioned PARTITION(src_airport_iata) select * from Routes r

where src_airport_iata='JFK';

1]

only showing top 20 rows

```
[123]: exceed=AirlineDF.groupby("Year").agg(sum("booked_seats").alias("total"))
exceed.show()
```

Year	total
2003	156153
2007	176299
2015	165438
2006	153789
2013	173676
1997	157972
2014	159823
2004	164800
1996	167223
1998	135678
2012	166076
2009	150308
1995	148520
2001	173598
2005	150610
2000	154376
2010	163741
2011	142647
2008	166897
1999	150000

only showing top 20 rows

```
[133]: exceed.filter(col("total")>40000).show()
```

only showing top 20 rows

```
[133]: exceed.filter(col("total")>40000).show()
```

Year	total
2003	156153
2007	176299
2015	165438
2006	153789
2013	173676
1997	157972
2014	159823
2004	164800
1996	167223
1998	135678
2012	166076
2009	150308
1995	148520
2001	173598
2005	150610
2000	154376
2010	163741
2011	142647
2008	166897
1999	150000

only showing top 20 rows

```
[ ]:
```

2]

Question 2.

1]

localhost:8888/notebooks/Untitled4.ipynb

Jupyter Untitled4 Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[19]: from pyspark.sql.functions import *
[33]: min=AirlineDF.orderBy(col("Avg_rev_per_seat").asc()).limit(1)
[35]: min.show()

+-----+
|Year|Quarter|Avg_rev_per_seat|booked_seats|
+-----+
|1996|    3|      269.49|      38952|
+-----+

[37]: max=AirlineDF.orderBy(col("Avg_rev_per_seat").desc()).limit(1)
max.show()

+-----+
|Year|Quarter|Avg_rev_per_seat|booked_seats|
+-----+
|2014|    3|      396.37|      40257|
+-----+

[43]: max=AirlineDF.agg(avg("Avg_rev_per_seat").alias("Avg_rev"))
max.show()

+-----+
|          Avg_rev|
+-----+
|329.7475000000000|
+-----+

[ ]:
```

31°C Smoke 14:51 21-11-2024

2]

localhost:8888/notebooks/Untitled4.ipynb

Jupyter Untitled4 Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

only showing top 20 rows

```
[167]: more=AirlineDF.groupBy("Year").agg(avg("Avg_rev_per_seat").alias("avg"))
more.show()

+-----+
|Year|          avg|
+-----+
|2003|    315.4675|
|2007|     325.14|
|2015|    377.1275|
|2006|     328.31|
|2013|    382.0025|
|1997|    287.155|
|2014|     391.7|
|2004|    305.875|
|1996|    276.8925|
|1998|    309.205|
|2012|    374.675|
|2009|    310.61|
|1995|    292.2475|
|2001|    319.7975|
|2005|    307.185|
|2000|    339.0325|
|2010|    339.8325|
|2011|363.63250000000005|
|2008|    346.1575|
|1999|    324.0575|
+-----+
only showing top 20 rows

[165]: more.filter(col("avg")>200).show()

+-----+
|Year|          avg|
+-----+
|2003|    315.4675|
|2007|     325.14|
|2015|    377.1275|
|2006|     328.31|
|2013|    382.0025|
|1997|    287.155|
|2014|     391.7|
|2004|    305.875|
|1996|    276.8925|
|1998|    309.205|
|2012|    374.675|
|2009|    310.61|
|1995|    292.2475|
|2001|    319.7975|
|2005|    307.185|
|2000|    339.0325|
|2010|    339.8325|
|2011|363.63250000000005|
|2008|    346.1575|
|1999|    324.0575|
+-----+

[ ]:
```

32°C Smoke 15:46 21-11-2024

The screenshot shows a JupyterLab interface with a notebook titled 'Untitled4'. The code cell [165] contains the following Python code:

```
more.filter(col("avg")>290).show()
```

The output shows a DataFrame with two columns: 'Year' and 'avg'. The data is as follows:

Year	avg
2003	315.4675
2007	325.14
2015	377.1275
2006	328.3
2013	382.0025
2014	391.7
2004	305.875
1998	309.285
2012	374.675
2009	310.61
1995	292.2475
2001	319.7975
2005	307.185
2000	335.0325
2010	335.8325
2011	363.63250000000005
2008	346.1575
1999	324.0575
2002	312.525

The code cell [73] contains the following Python code:

```
more=AirlineDF.filter(("Avg_rev_per_seat">290))
```

3]

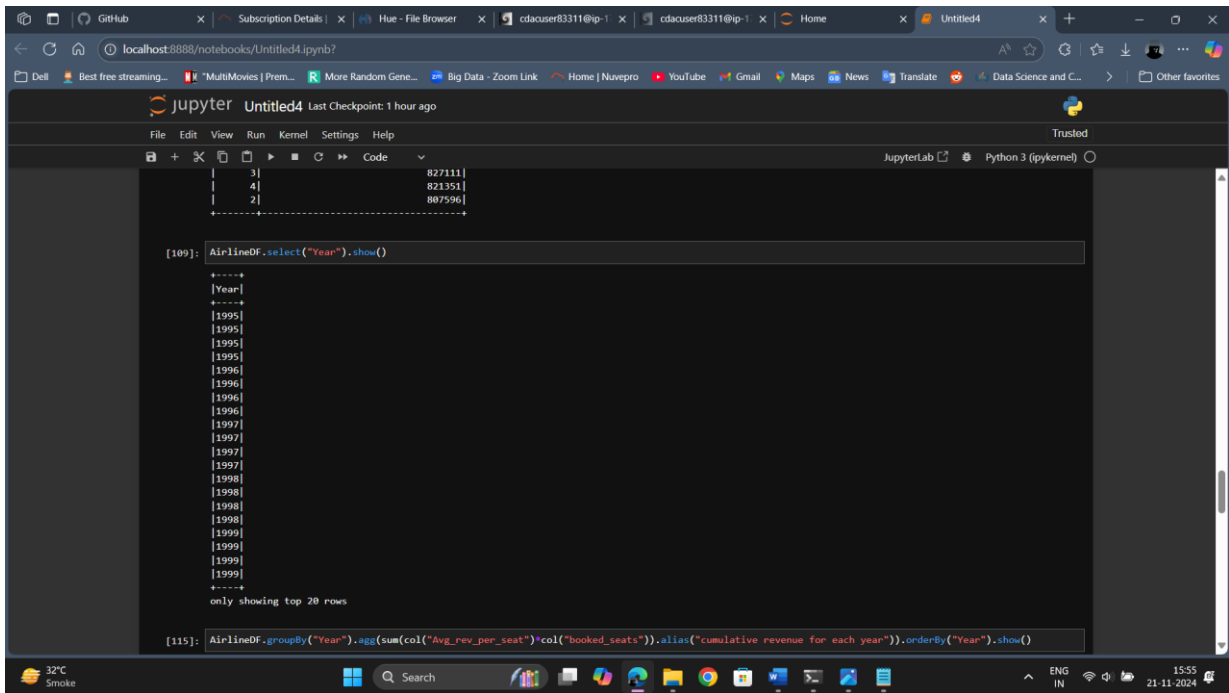
The screenshot shows a JupyterLab interface with a notebook titled 'Untitled4'. The code cell [81] contains the following Python code:

```
AirlineDF.groupBy("Quarter").agg(sum("booked_seats").alias("Total booked seats for each quarter")).show()
```

The output shows a DataFrame with two columns: 'Quarter' and 'Total booked seats for each quarter'. The data is as follows:

Quarter	Total booked seats for each quarter
1	873761
3	827111
4	821351
2	807596

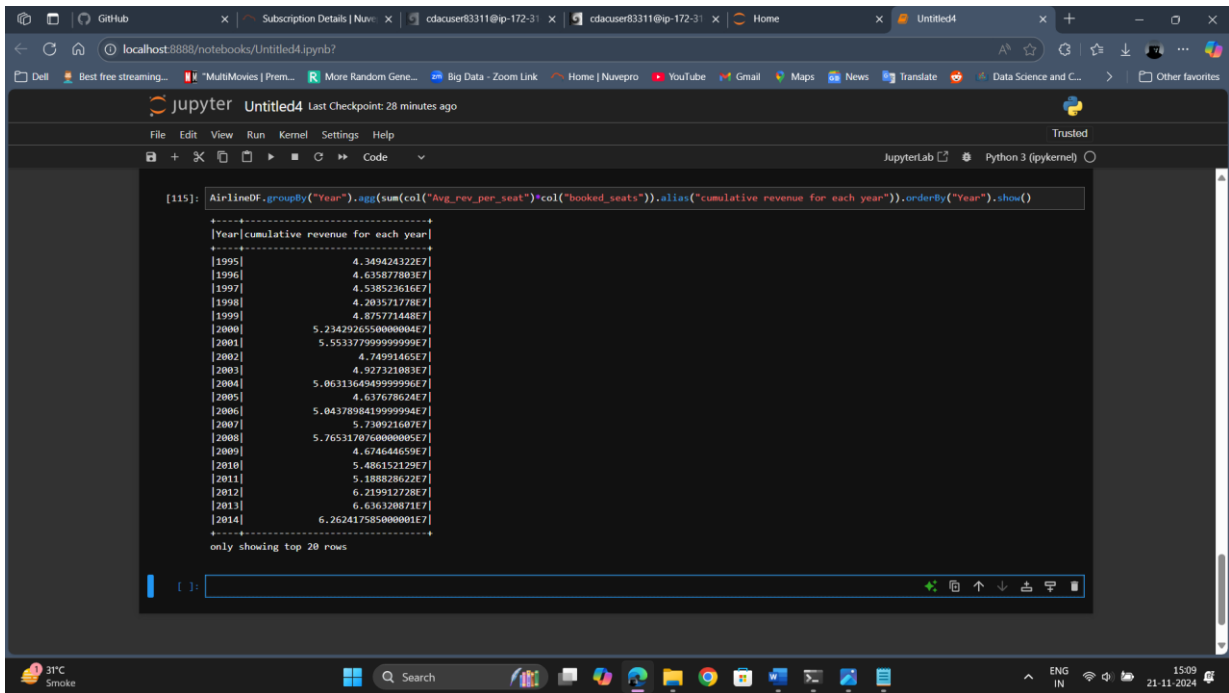
4]



```
[109]: AirlineDF.select("Year").show()
+----+
|Year|
+----+
|1995|
|1995|
|1995|
|1995|
|1996|
|1996|
|1996|
|1996|
|1997|
|1997|
|1997|
|1998|
|1998|
|1998|
|1999|
|1999|
|1999|
+----+
only showing top 20 rows

[115]: AirlineDF.groupBy("Year").agg(sum(col("Avg_rev_per_seat")*col("booked_seats")).alias("cumulative revenue for each year")).orderBy("Year").show()
+----+
|Year|cumulative revenue for each year|
+----+
|1995|4.349424322E7|
|1996|4.635877803E7|
|1997|4.538522615E7|
|1998|4.203571778E7|
|1999|4.875771448E7|
|2000|5.2342926550000004E7|
|2001|5.553779999999999E7|
|2002|4.74991465E7|
|2003|4.927321083E7|
|2004|5.063136499999999E7|
|2005|4.637678624E7|
|2006|5.0437898419999994E7|
|2007|5.730921607E7|
|2008|5.7653178760000005E7|
|2009|4.674644659E7|
|2010|5.486152129E7|
|2011|5.188828622E7|
|2012|6.219912728E7|
|2013|6.636320871E7|
|2014|6.262417585000001E7|
+----+
only showing top 20 rows
```

5]



```
[115]: AirlineDF.groupBy("Year").agg(sum(col("Avg_rev_per_seat")*col("booked_seats")).alias("cumulative revenue for each year")).orderBy("Year").show()
+----+
|Year|cumulative revenue for each year|
+----+
|1995|4.349424322E7|
|1996|4.635877803E7|
|1997|4.538522615E7|
|1998|4.203571778E7|
|1999|4.875771448E7|
|2000|5.2342926550000004E7|
|2001|5.553779999999999E7|
|2002|4.74991465E7|
|2003|4.927321083E7|
|2004|5.063136499999999E7|
|2005|4.637678624E7|
|2006|5.0437898419999994E7|
|2007|5.730921607E7|
|2008|5.7653178760000005E7|
|2009|4.674644659E7|
|2010|5.486152129E7|
|2011|5.188828622E7|
|2012|6.219912728E7|
|2013|6.636320871E7|
|2014|6.262417585000001E7|
+----+
only showing top 20 rows
```