Question 1

1.

```
select name
from airport ap join routes r
on ap.iata=r.src_airport_iata
join airport ap2
on ap2.iata=r.dest_airport_iata limit 6;
```

2.

Question 2

1.partition table

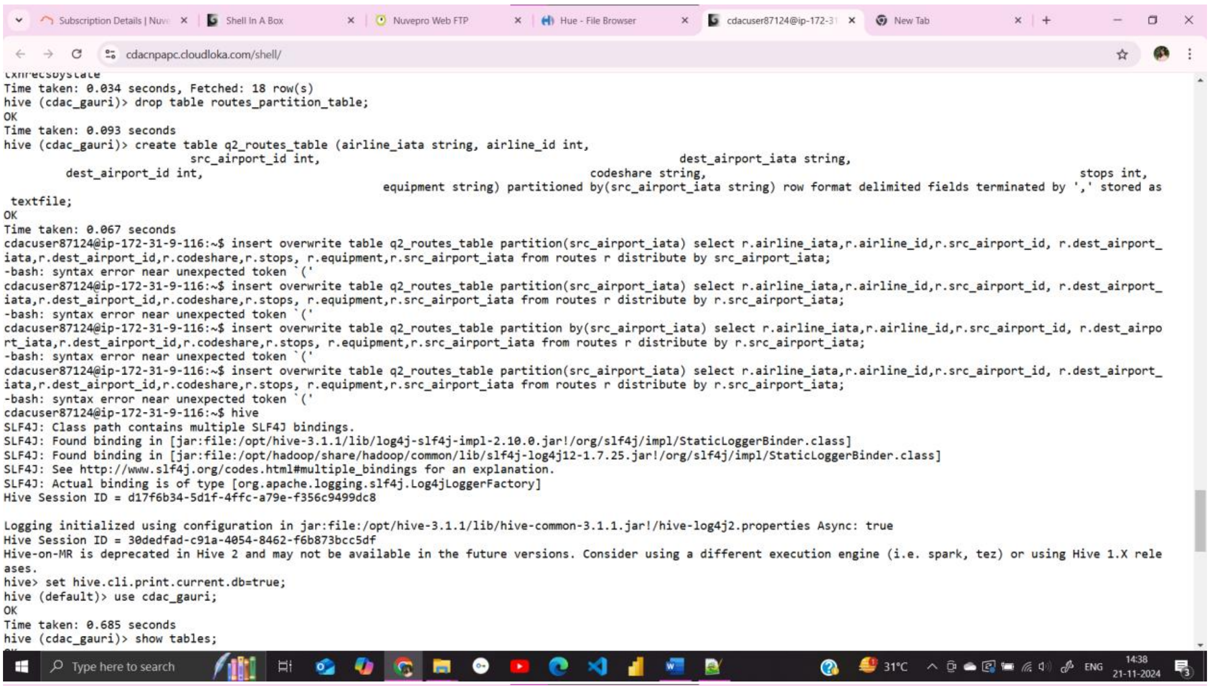
```
hive
set hive.cli.print.current.db=true;
use cdac_gauri;

create table q2_routes_table
(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 q2_routes_table partition(src_airport_iata)

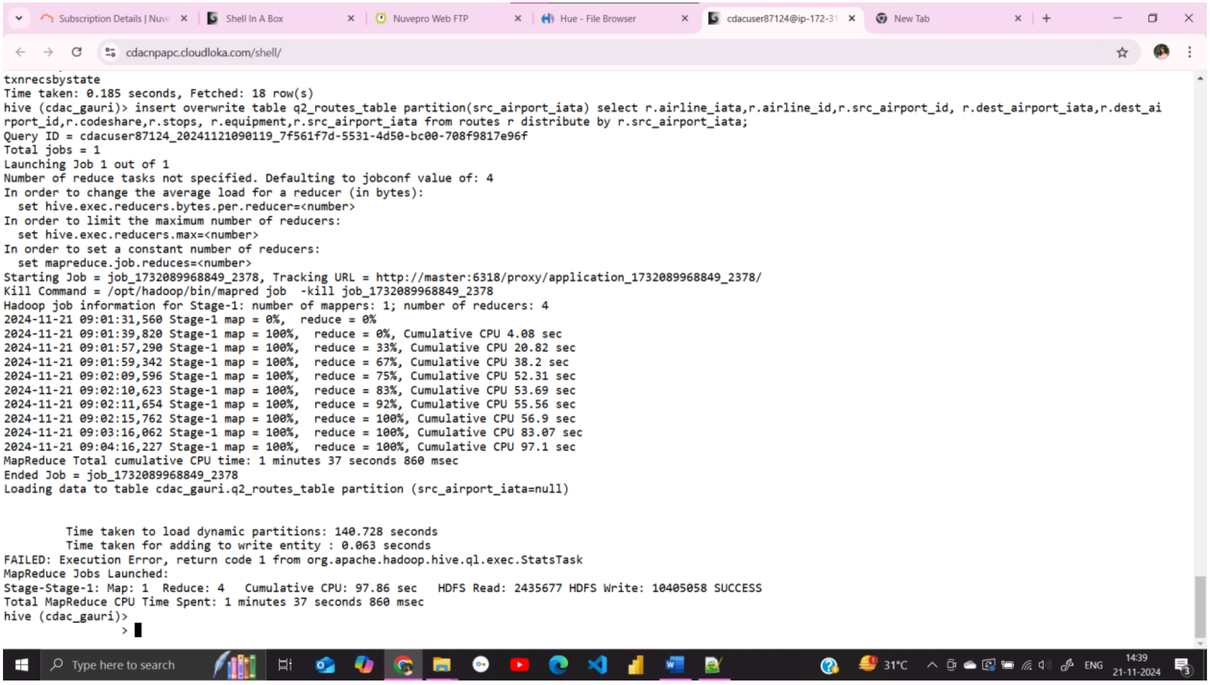
select r.airline_iata,r.airline_id,r.src_airport_id,

r.dest_airport_iata,r.dest_airport_id,r.codeshare,r.stops,

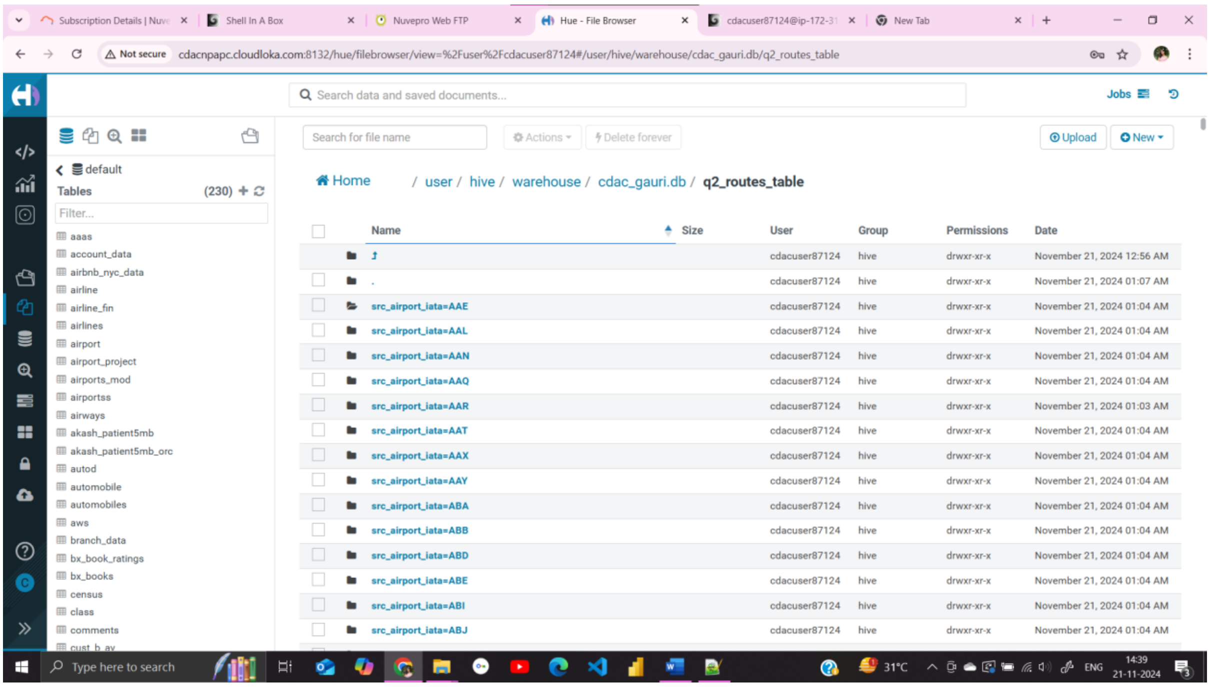
r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata;



```
txnrecsbystate
Time taken: 0.034 seconds, Fetched: 18 row(s)
hive (cdac_gauri)> drop table routes_partition_table;
OK
Time taken: 0.093 seconds
hive (cdac_gauri)> create table q2_routes_table (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;
OK
Time taken: 0.067 seconds
cdacuser87124@ip-172-31-9-116:~$ insert overwrite table q2_routes_table partition(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_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;
-bash: syntax error near unexpected token `('
cdacuser87124@ip-172-31-9-116:~$ insert overwrite table q2_routes_table partition(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_id, r.dest_airport_
iata,r.dest_airport_id,r.codeshare,r.stops, r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata;
-bash: syntax error near unexpected token `('
cdacuser87124@ip-172-31-9-116:~$ insert overwrite table q2_routes_table partition by(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_id, r.dest_airpo
rt_iata,r.dest_airport_id,r.codeshare,r.stops, r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata;
-bash: syntax error near unexpected token `('
cdacuser87124@ip-172-31-9-116:~$ insert overwrite table q2_routes_table partition(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_id, r.dest_airport_
iata,r.dest_airport_id,r.codeshare,r.stops, r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata;
-bash: syntax error near unexpected token `('
cdacuser87124@ip-172-31-9-116:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive-3.1.1/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = d17f6b34-5d1f-4ffc-a79e-f356c9499dc8

Logging initialized using configuration in jar:file:/opt/hive-3.1.1/lib/hive-common-3.1.1.jar!/hive-log4j2.properties Async: true
Hive Session ID = 30dedfad-c91a-4054-8462-f6b873bcc5df
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X rele
ases.
hive> set hive.cli.print.current.db=true;
hive (default)> use cdac_gauri;
OK
Time taken: 0.685 seconds
hive (cdac_gauri)> show tables;
```



```
txnrecsbystate
Time taken: 0.185 seconds, Fetched: 18 row(s)
hive (cdac_gauri)> insert overwrite table q2_routes_table partition(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_id, r.dest_airport_iata,r.dest_ai
rport_id,r.codeshare,r.stops, r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata;
Query ID = cdacuser87124_20241121090119_7f561f7d-5531-4d50-bc00-708f9817e96f
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_2378, Tracking URL = http://master:6318/proxy/application_1732089968849_2378/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2378
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 09:01:31,560 Stage-1 map = 0%,  reduce = 0%
2024-11-21 09:01:39,820 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.08 sec
2024-11-21 09:01:57,290 Stage-1 map = 100%,  reduce = 33%, Cumulative CPU 20.82 sec
2024-11-21 09:01:59,342 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 38.2 sec
2024-11-21 09:02:09,596 Stage-1 map = 100%,  reduce = 75%, Cumulative CPU 52.31 sec
2024-11-21 09:02:10,623 Stage-1 map = 100%,  reduce = 83%, Cumulative CPU 53.69 sec
2024-11-21 09:02:11,654 Stage-1 map = 100%,  reduce = 92%, Cumulative CPU 55.56 sec
2024-11-21 09:02:15,762 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 56.9 sec
2024-11-21 09:03:16,062 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 83.07 sec
2024-11-21 09:04:16,227 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 97.1 sec
MapReduce Total cumulative CPU time: 1 minutes 37 seconds 860 msec
Ended Job = job_1732089968849_2378
Loading data to table cdac_gauri.q2_routes_table partition (src_airport_iata=null)


         Time taken to load dynamic partitions: 140.728 seconds
         Time taken for adding to write entity : 0.063 seconds
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: 97.86 sec   HDFS Read: 2435677 HDFS Write: 10405058 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 37 seconds 860 msec
hive (cdac_gauri)>
                 >
```



cdacnpapc.cloudloka.com:8132/hue/filebrowser/view=%2Fuser%2Fcdacuser87124#/user/hive/warehouse/cdac_gauri.db/q2_routes_table

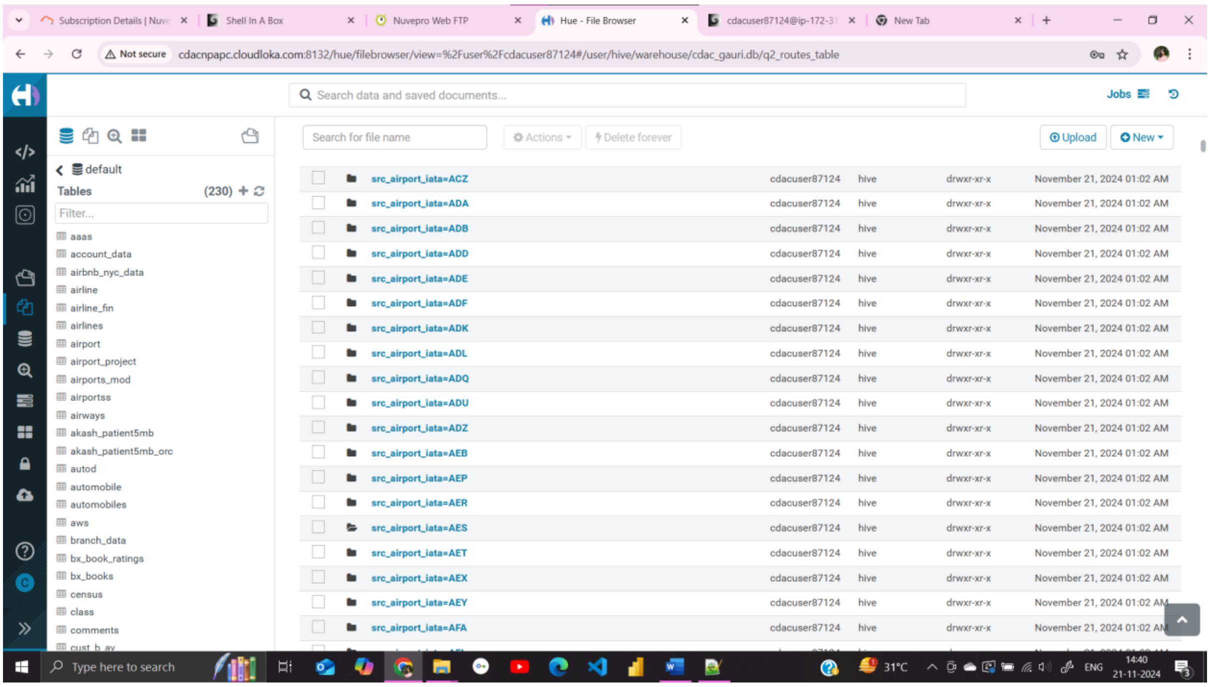
default

Tables (230)

aaas, account_data, airbnb_nyc_data, airline, airline_fin, airlines, airport, airport_project, airports_mod, airportss, airways, akash_patient5mb, akash_patient5mb_orc, autod, automobile, automobiles, aws, branch_data, bx_book_ratings, bx_books, census, class, comments

Home / user / hive / warehouse / cdac_gauri.db / q2_routes_table

Name	Size	User	Group	Permissions	Date
↥		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 12:56 AM
.		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:07 AM
src_airport_iata=AAE		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAL		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAN		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAQ		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAR		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:03 AM
src_airport_iata=AAT		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAX		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=AAY		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABA		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABB		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABD		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABE		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABI		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM
src_airport_iata=ABJ		cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:04 AM



cdacnpapc.cloudloka.com:8132/hue/filebrowser/view=%2Fuser%2Fcdacuser87124#/user/hive/warehouse/cdac_gauri.db/q2_routes_table

default

Tables (230)

aaas, account_data, airbnb_nyc_data, airline, airline_fin, airlines, airport, airport_project, airports_mod, airportss, airways, akash_patient5mb, akash_patient5mb_orc, autod, automobile, automobiles, aws, branch_data, bx_book_ratings, bx_books, census, class, comments

src_airport_iata=ACZ	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADA	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADB	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADD	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADE	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADF	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADK	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADL	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADQ	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADU	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=ADZ	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AEB	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AEP	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AER	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AES	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AET	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AEX	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AEY	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM
src_airport_iata=AFA	cdacuser87124	hive	drwxr-xr-x	November 21, 2024 01:02 AM

2.Insert data into partition table for source airport = “JFK”

create table q2_routes_table2

(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 ','

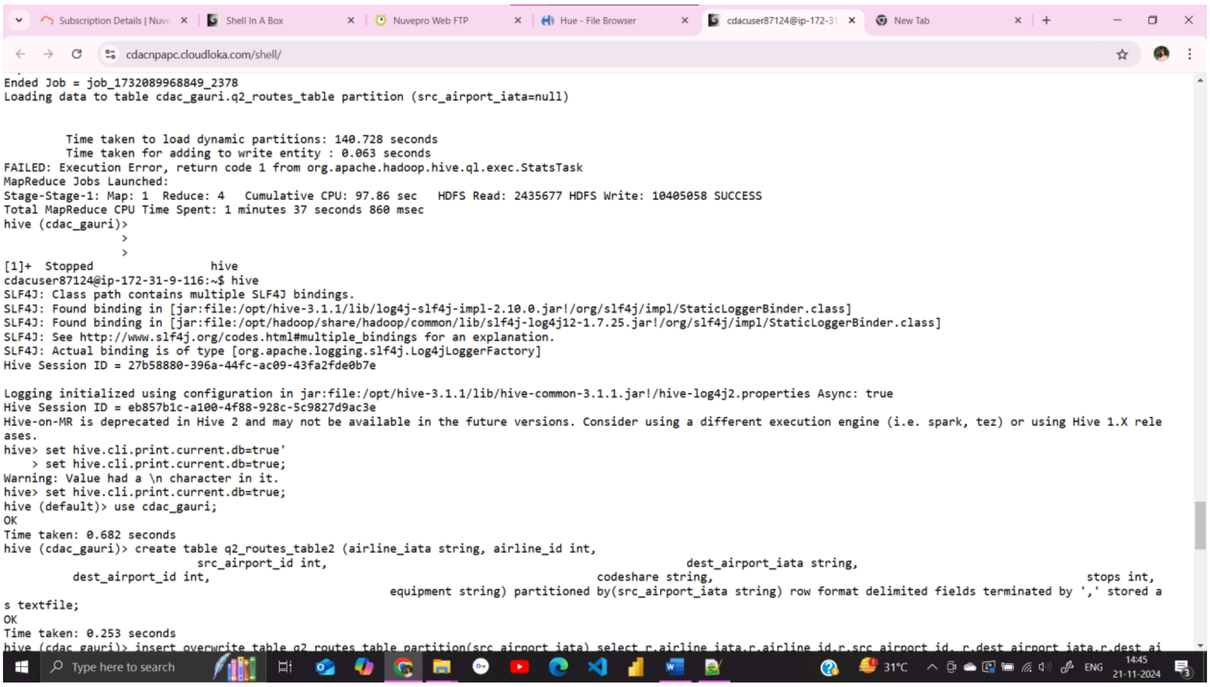
stored as textfile;

insert overwrite table q2_routes_table partition(src_airport_iata)

select r.airline_iata,r.airline_id,r.src_airport_id,

r.dest_airport_iata,r.dest_airport_id,r.codeshare,r.stops,

r.equipment,r.src_airport_iata from routes r distribute by r.src_airport_iata="JFK";

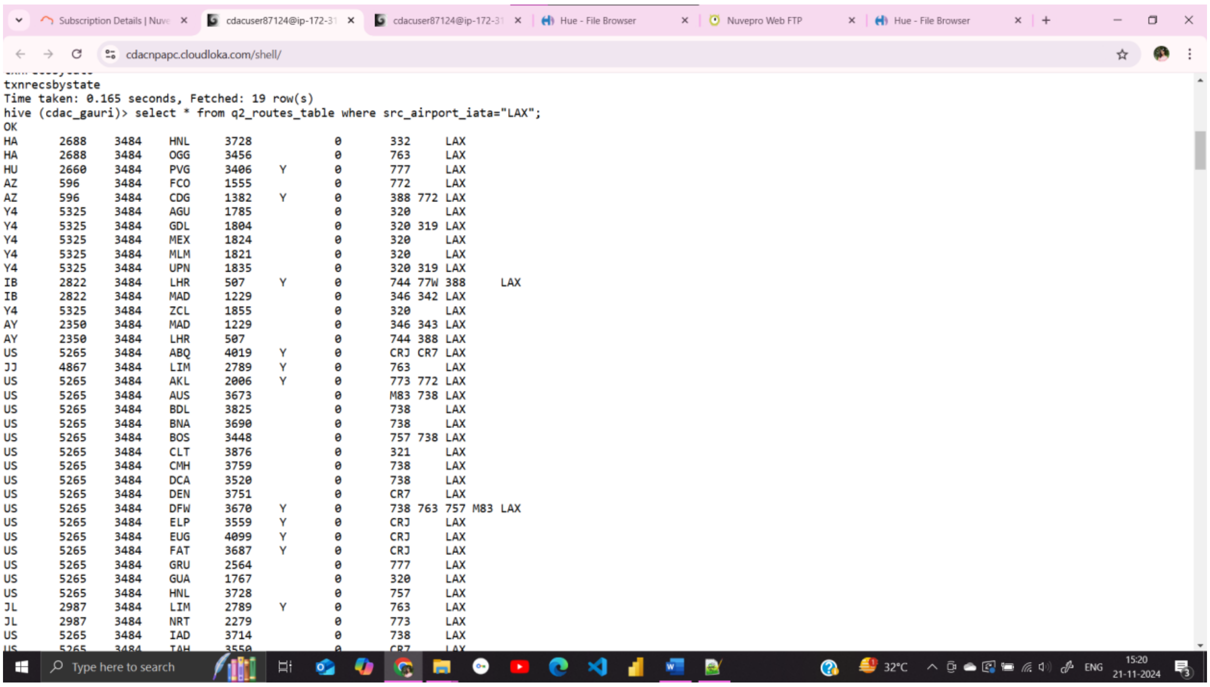


```
Ended Job = job_1732089968849_2378
Loading data to table cdac_gauri.q2_routes_table partition (src_airport_iata=null)


         Time taken to load dynamic partitions: 140.728 seconds
         Time taken for adding to write entity : 0.063 seconds
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: 97.86 sec   HDFS Read: 2435677 HDFS Write: 10405058 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 37 seconds 860 msec
hive (cdac_gauri)>
                 >
                 >
[1]+  Stopped                 hive
cdacuser87124@ip-172-31-9-116:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive-3.1.1/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 27b58880-396a-44fc-ac09-43fa2fde0b7e

Logging initialized using configuration in jar:file:/opt/hive-3.1.1/lib/hive-common-3.1.1.jar!/hive-log4j2.properties Async: true
Hive Session ID = eb857b1c-a100-4f88-928c-5c9827d9ac3e
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X rele
ases.
hive> set hive.cli.print.current.db=true'
    > set hive.cli.print.current.db=true;
Warning: Value had a \n character in it.
hive> set hive.cli.print.current.db=true;
hive (default)> use cdac_gauri;
OK
Time taken: 0.682 seconds
hive (cdac_gauri)> create table q2_routes_table2 (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 a
s textfile;
OK
Time taken: 0.253 seconds
hive (cdac_gauri)> insert overwrite table q2_routes_table partition(src_airport_iata) select r.airline_iata,r.airline_id,r.src_airport_id, r.dest_airport_iata,r.dest_ai
```

Shell is not working

3.

select * from q2_routes_table where src_airport_iata="LAX";



```
txnrecsbystate
Time taken: 0.165 seconds, Fetched: 19 row(s)
hive (cdac_gauri)> select * from q2_routes_table where src_airport_iata="LAX";
OK
HA      2688    3484    HNL     3728            0       332     LAX
HA      2688    3484    OGG     3456            0       763     LAX
HU      2660    3484    PVG     3406    Y       0       777     LAX
AZ      596     3484    FCO     1555            0       772     LAX
AZ      596     3484    CDG     1382    Y       0       388 772 LAX
Y4      5325    3484    AGU     1785            0       320     LAX
Y4      5325    3484    GDL     1804            0       320 319 LAX
Y4      5325    3484    MEX     1824            0       320     LAX
Y4      5325    3484    MLM     1821            0       320     LAX
Y4      5325    3484    UPN     1835            0       320 319 LAX
IB      2822    3484    LHR     507     Y       0       744 77W 388     LAX
IB      2822    3484    MAD     1229            0       346 342 LAX
Y4      5325    3484    ZCL     1855            0       320     LAX
AY      2350    3484    MAD     1229            0       346 343 LAX
AY      2350    3484    LHR     507             0       744 388 LAX
US      5265    3484    ABQ     4019    Y       0       CRJ CR7 LAX
JJ      4867    3484    LIM     2789    Y       0       763     LAX
US      5265    3484    AKL     2006    Y       0       773 772 LAX
US      5265    3484    AUS     3673            0       M83 738 LAX
US      5265    3484    BDL     3825            0       738     LAX
US      5265    3484    BNA     3690            0       738     LAX
US      5265    3484    BOS     3448            0       757 738 LAX
US      5265    3484    CLT     3876            0       321     LAX
US      5265    3484    CMH     3759            0       738     LAX
US      5265    3484    DCA     3520            0       738     LAX
US      5265    3484    DEN     3751            0       CR7     LAX
US      5265    3484    DFW     3670    Y       0       738 763 757 M83 LAX
US      5265    3484    ELP     3559    Y       0       CRJ     LAX
US      5265    3484    EUG     4099    Y       0       CRJ     LAX
US      5265    3484    FAT     3687    Y       0       CRJ     LAX
US      5265    3484    GRU     2564            0       777     LAX
US      5265    3484    GUA     1767            0       320     LAX
US      5265    3484    HNL     3728            0       757     LAX
JL      2987    3484    LIM     2789    Y       0       763     LAX
JL      2987    3484    NRT     2279            0       773     LAX
US      5265    3484    IAD     3714            0       738     LAX
US      5265    3484    IAH     3550            0       CR7     LAX
```

SPARK

Q1.

q1

data=spark.textFile("/user/cdacuser87124/spark/airlines_data.csv")

header=data.first()

eliminate=data.map(lambda a: a!= header)

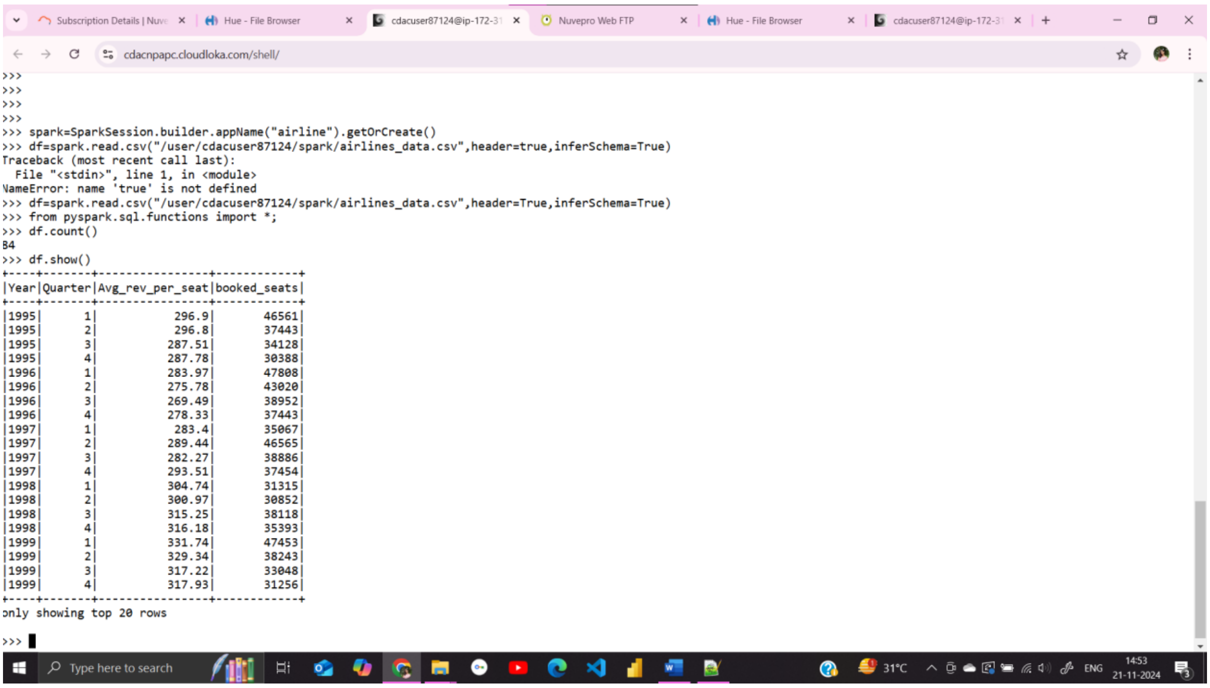
map_data=eliminate.map(lambda a :a.split(","))

Q2 dataframe

spark=SparkSession.builder.appName("airline").getOrCreate()

df=spark.read.csv("/user/cdacuser87124/spark/airlines_data.csv",header=True,inferSchema=True)

from pyspark.sql.functions import *;



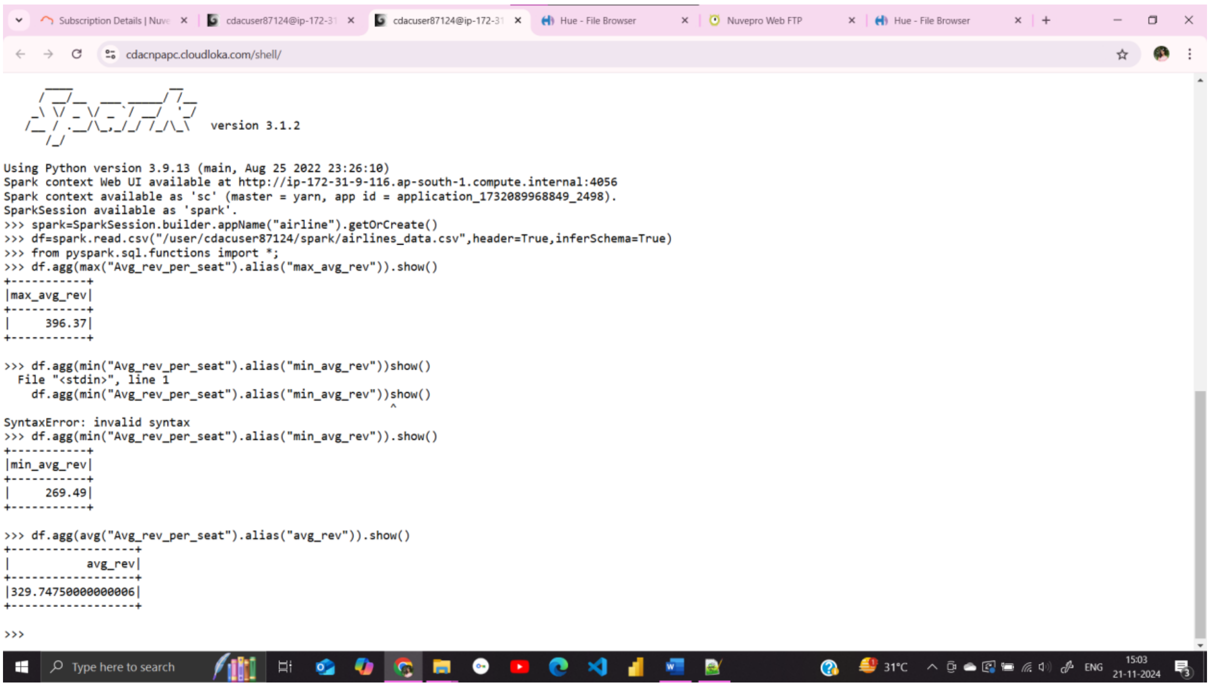
```
>>>
>>>
>>>
>>>
>>> spark=SparkSession.builder.appName("airline").getOrCreate()
>>> df=spark.read.csv("/user/cdacuser87124/spark/airlines_data.csv",header=true,inferSchema=True)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'true' is not defined
>>> df=spark.read.csv("/user/cdacuser87124/spark/airlines_data.csv",header=True,inferSchema=True)
>>> from pyspark.sql.functions import *;
>>> df.count()
84
>>> df.show()
+----+-------+---------------+------------+
|Year|Quarter|Avg_rev_per_seat|booked_seats|
+----+-------+---------------+------------+
|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|
|1996|      2|         275.78|       43020|
|1996|      3|         269.49|       38952|
|1996|      4|         278.33|       37443|
|1997|      1|          283.4|       35067|
|1997|      2|         289.44|       46565|
|1997|      3|         282.27|       38886|
|1997|      4|         293.51|       37454|
|1998|      1|         304.74|       31315|
|1998|      2|         300.97|       30852|
|1998|      3|         315.25|       38118|
|1998|      4|         316.18|       35393|
|1999|      1|         331.74|       47453|
|1999|      2|         329.34|       38243|
|1999|      3|         317.22|       33048|
|1999|      4|         317.93|       31256|
+----+-------+---------------+------------+
only showing top 20 rows

>>>
```

1.

df.agg(max("Avg_rev_per_seat").alias("max_avg_rev")).show()

df.agg(min("Avg_rev_per_seat").alias("min_avg_rev")).show()

df.agg(avg("Avg_rev_per_seat").alias("avg_rev")).show()x

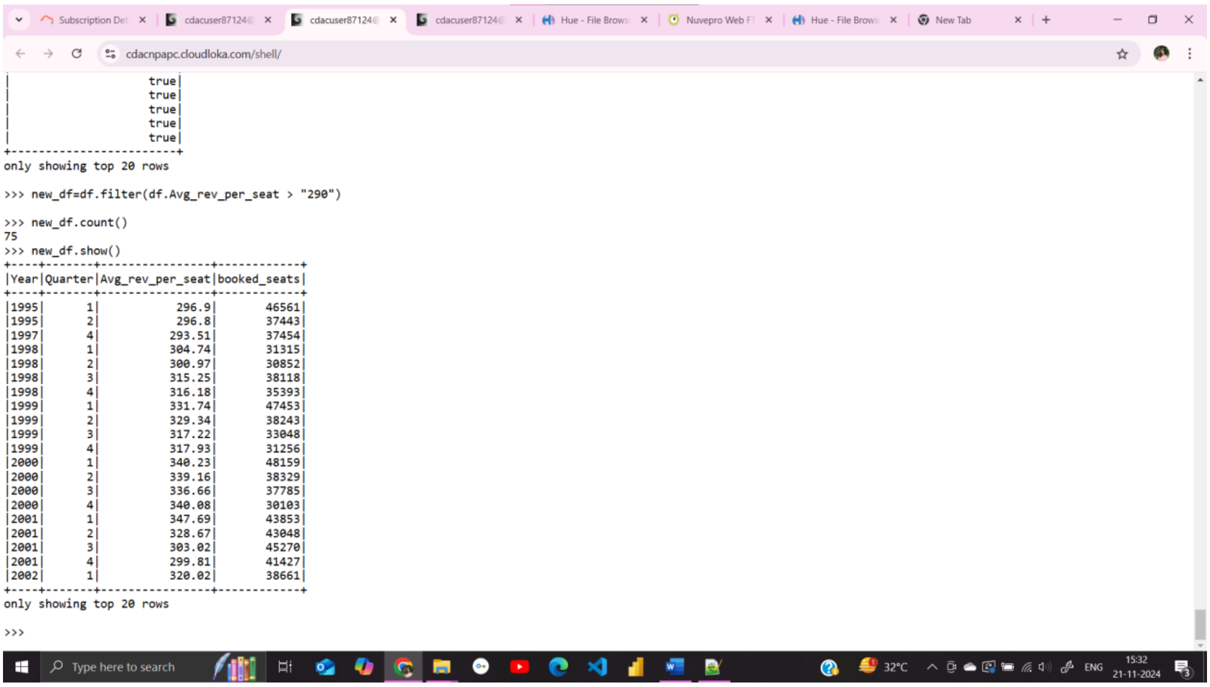


```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

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:4056
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2498).
SparkSession available as 'spark'.
>>> spark=SparkSession.builder.appName("airline").getOrCreate()
>>> df=spark.read.csv("/user/cdacuser87124/spark/airlines_data.csv",header=True,inferSchema=True)
>>> from pyspark.sql.functions import *;
>>> df.agg(max("Avg_rev_per_seat").alias("max_avg_rev")).show()
+-----------+
|max_avg_rev|
+-----------+
|     396.37|
+-----------+

>>> df.agg(min("Avg_rev_per_seat").alias("min_avg_rev"))show()
  File "<stdin>", line 1
    df.agg(min("Avg_rev_per_seat").alias("min_avg_rev"))show()
                                                        ^
SyntaxError: invalid syntax
>>> df.agg(min("Avg_rev_per_seat").alias("min_avg_rev")).show()
+-----------+
|min_avg_rev|
+-----------+
|     269.49|
+-----------+

>>> df.agg(avg("Avg_rev_per_seat").alias("avg_rev")).show()
+------------------+
|           avg_rev|
+------------------+
|329.74750000000006|
+------------------+

>>>
```

2.

new_df=df.filter(df.Avg_rev_per_seat > "290")

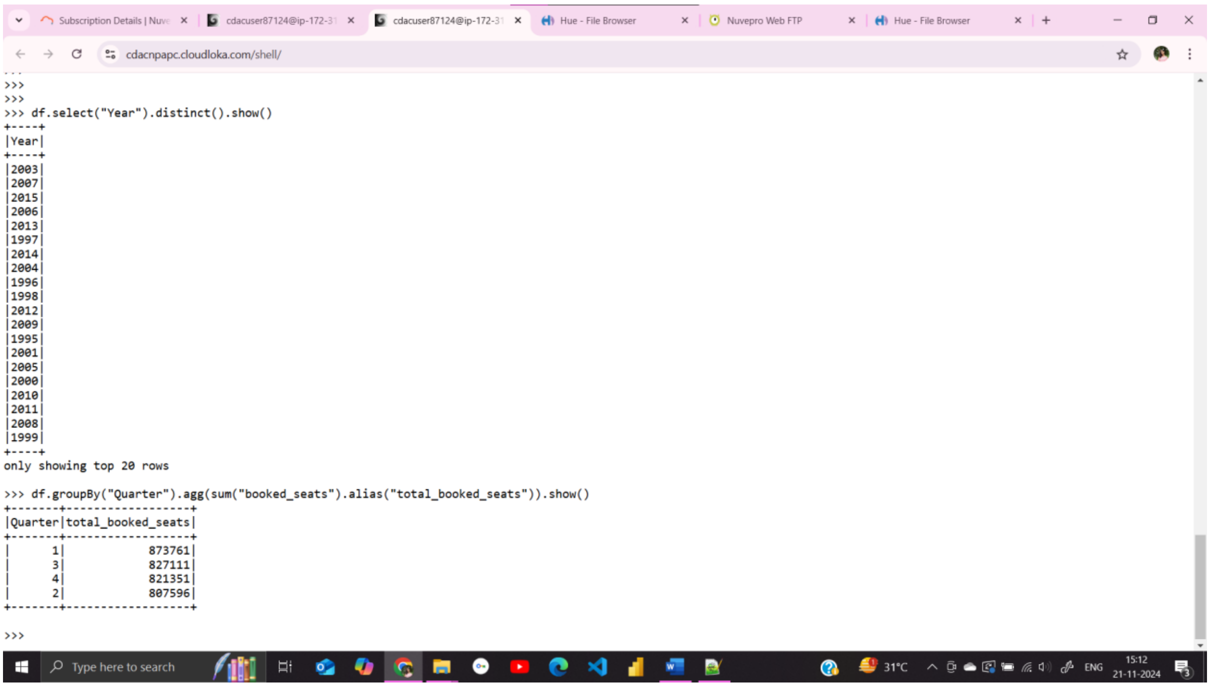
new_df.count() ----75



```
|                    true|
|                    true|
|                    true|
|                    true|
|                    true|
+------------------------+
only showing top 20 rows

>>> new_df=df.filter(df.Avg_rev_per_seat > "290")

>>> new_df.count()
75
>>> new_df.show()
+----+-------+----------------+------------+
|Year|Quarter|Avg_rev_per_seat|booked_seats|
+----+-------+----------------+------------+
|1995|      1|           296.9|       46561|
|1995|      2|           296.8|       37443|
|1997|      4|          293.51|       37454|
|1998|      1|          304.74|       31315|
|1998|      2|          300.97|       30852|
|1998|      3|          315.25|       38118|
|1998|      4|          316.18|       35393|
|1999|      1|          331.74|       47453|
|1999|      2|          329.34|       38243|
|1999|      3|          317.22|       33048|
|1999|      4|          317.93|       31256|
|2000|      1|          340.23|       48159|
|2000|      2|          339.16|       38329|
|2000|      3|          336.66|       37785|
|2000|      4|          340.08|       30103|
|2001|      1|          347.69|       43853|
|2001|      2|          328.67|       43048|
|2001|      3|          303.02|       45270|
|2001|      4|          299.81|       41427|
|2002|      1|          320.02|       38661|
+----+-------+----------------+------------+
only showing top 20 rows

>>>
```

3.

df.groupBy("Quarter").agg(sum("booked_seats").alias("total_booked_seats")).show()

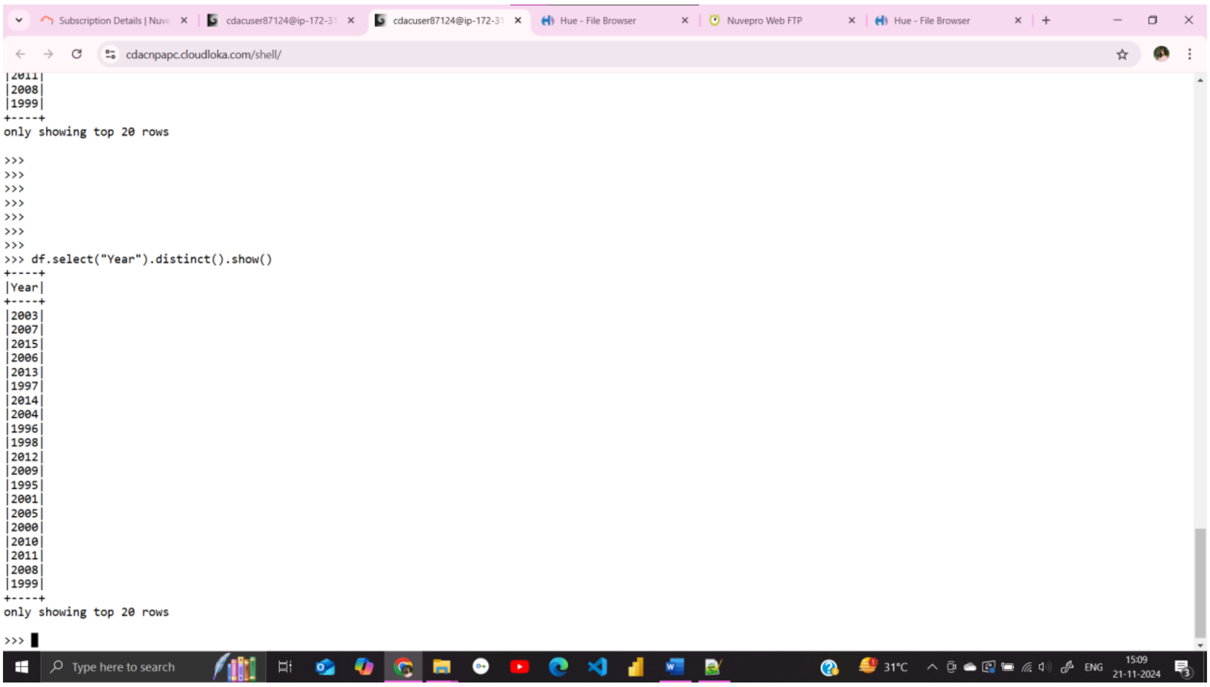


```
>>>
>>>
>>> df.select("Year").distinct().show()
+----+
|Year|
+----+
|2003|
|2007|
|2015|
|2006|
|2013|
|1997|
|2014|
|2004|
|1996|
|1998|
|2012|
|2009|
|1995|
|2001|
|2005|
|2000|
|2010|
|2011|
|2008|
|1999|
+----+
only showing top 20 rows

>>> df.groupBy("Quarter").agg(sum("booked_seats").alias("total_booked_seats")).show()
+-------+------------------+
|Quarter|total_booked_seats|
+-------+------------------+
|      1|            873761|
|      3|            827111|
|      4|            821351|
|      2|            807596|
+-------+------------------+

>>>
```

4.

5.

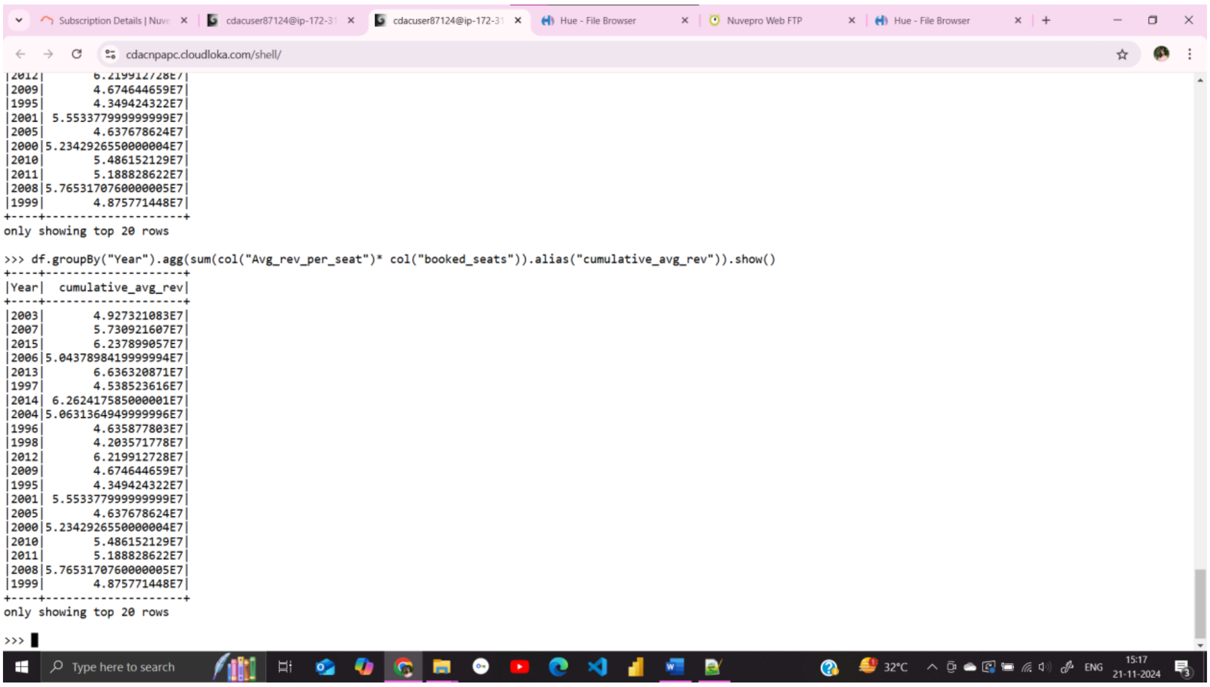
df.select("Year").distinct().show()



```
|2011|
|2008|
|1999|
+----+
only showing top 20 rows

>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> df.select("Year").distinct().show()
+----+
|Year|
+----+
|2003|
|2007|
|2015|
|2006|
|2013|
|1997|
|2014|
|2004|
|1996|
|1998|
|2012|
|2009|
|1995|
|2001|
|2005|
|2000|
|2010|
|2011|
|2008|
|1999|
+----+
only showing top 20 rows

>>> 
```

5.

df.groupBy("Year").agg(sum(col("Avg_rev_per_seat")*col("booked_seats")).alias("cumulative_avg_rev")).show()



```
|2012|         6.219912728E7|
|2009|         4.674644659E7|
|1995|         4.349424322E7|
|2001| 5.553377999999999E7|
|2005|         4.637678624E7|
|2000|5.2342926550000004E7|
|2010|         5.486152129E7|
|2011|         5.188828622E7|
|2008|5.7653170760000005E7|
|1999|         4.875771448E7|
+----+--------------------+
only showing top 20 rows

>>> df.groupBy("Year").agg(sum(col("Avg_rev_per_seat")* col("booked_seats")).alias("cumulative_avg_rev")).show()
+----+--------------------+
|Year|  cumulative_avg_rev|
+----+--------------------+
|2003|         4.927321083E7|
|2007|         5.730921607E7|
|2015|         6.237899057E7|
|2006|5.0437898419999994E7|
|2013|         6.636320871E7|
|1997|         4.538523616E7|
|2014| 6.262417585000001E7|
|2004|5.0631364949999996E7|
|1996|         4.635877803E7|
|1998|         4.203571778E7|
|2012|         6.219912728E7|
|2009|         4.674644659E7|
|1995|         4.349424322E7|
|2001| 5.553377999999999E7|
|2005|         4.637678624E7|
|2000|5.2342926550000004E7|
|2010|         5.486152129E7|
|2011|         5.188828622E7|
|2008|5.7653170760000005E7|
|1999|         4.875771448E7|
+----+--------------------+
only showing top 20 rows

>>> 
```