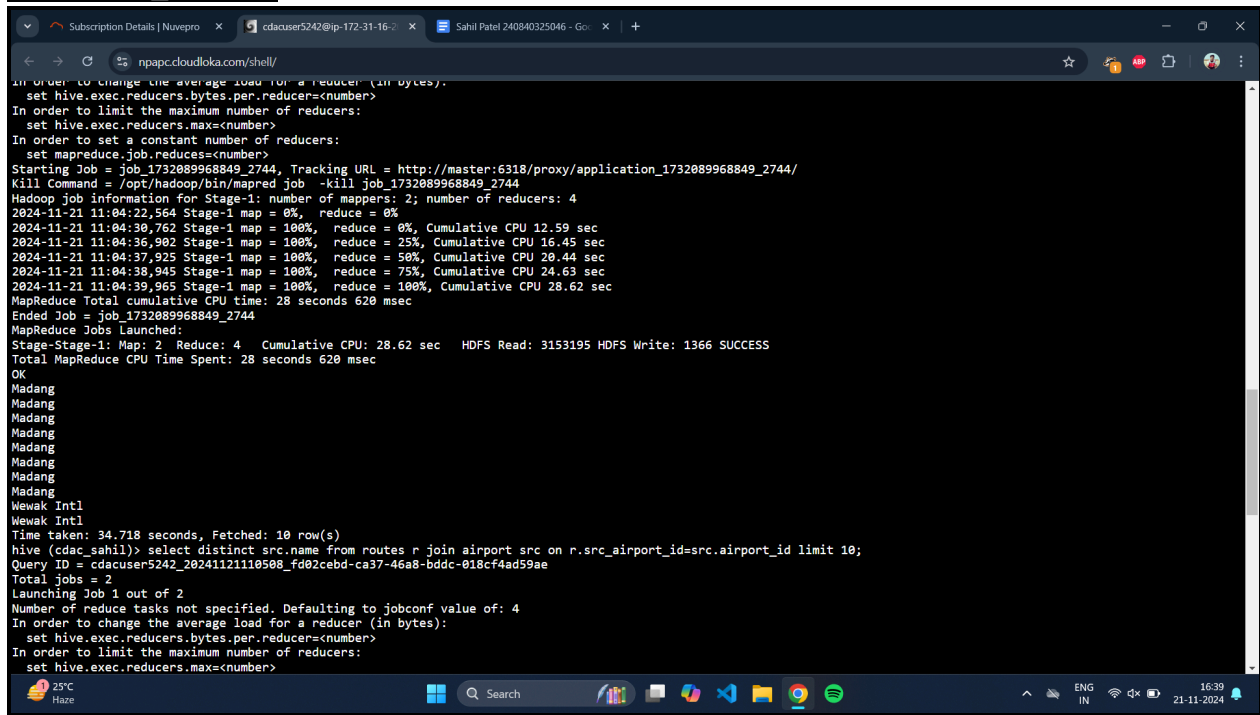


## Question 1

1.

hive

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



The screenshot shows a terminal window with a dark background. The top of the window displays browser tabs for 'Subscription Details | Nuvepro', 'cdacuser5242@ip-172-31-16-7', and 'Sahil Patel 240840325046 - Go...'. The terminal content shows Hive configuration commands, Hadoop job information for Stage-1, and the execution of a SQL query. The query is 'select distinct src.name from routes r join airport src on r.src\_airport\_id=src.airport\_id limit 10;'. The output shows the query ID, total jobs, and the launch of Job 1. The bottom of the window shows a Windows taskbar with a search bar, application icons, and system tray information including temperature (25°C), time (16:39), and date (21-11-2024).

```
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_2744, Tracking URL = http://master:6318/proxy/application_1732089968849_2744/  
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2744  
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 4  
2024-11-21 11:04:22,564 Stage-1 map = 0%, reduce = 0%  
2024-11-21 11:04:30,762 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 12.59 sec  
2024-11-21 11:04:36,982 Stage-1 map = 100%, reduce = 25%, Cumulative CPU 16.45 sec  
2024-11-21 11:04:37,925 Stage-1 map = 100%, reduce = 50%, Cumulative CPU 20.44 sec  
2024-11-21 11:04:38,945 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 24.63 sec  
2024-11-21 11:04:39,965 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 28.62 sec  
MapReduce Total cumulative CPU time: 28 seconds 620 msec  
Ended Job = job_1732089968849_2744  
MapReduce Jobs Launched:  
Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 28.62 sec HDFS Read: 3153195 HDFS Write: 1366 SUCCESS  
Total MapReduce CPU Time Spent: 28 seconds 620 msec  
OK  
Madang  
Madang  
Madang  
Madang  
Madang  
Madang  
Madang  
Wewak Intl  
Wewak Intl  
Time taken: 34.718 seconds, Fetched: 10 row(s)  
hive (cdac_sahil)> select distinct src.name from routes r join airport src on r.src_airport_id=src.airport_id limit 10;  
Query ID = cdacuser5242_20241121110508_fd02cebd-ca37-46a8-bddc-018cf4ad59ae  
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>
```

```
select distinct src.name from routes r join airport src on  
r.src_airport_id=src.airport_id limit 10;
```

```
Subscription Details | Nuvepro x cdacuser5242@ip-172-31-16-2 x Sahil Patel 240840325046 - Go x +
npapc.cloudloka.com/shell/
2024-11-21 11:05:28,323 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 13.09 sec
2024-11-21 11:05:32,480 Stage-1 map = 100%, reduce = 25%, Cumulative CPU 16.95 sec
2024-11-21 11:05:34,440 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 24.88 sec
2024-11-21 11:05:35,459 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 28.55 sec
MapReduce Total cumulative CPU time: 28 seconds 550 msec
Ended Job = job_1732089968849_2748
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_2750, Tracking URL = http://master:6318/proxy/application_1732089968849_2750/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2750
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 4
2024-11-21 11:05:47,631 Stage-2 map = 0%, reduce = 0%
2024-11-21 11:05:55,785 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 6.53 sec
2024-11-21 11:06:02,914 Stage-2 map = 100%, reduce = 75%, Cumulative CPU 16.03 sec
2024-11-21 11:06:03,932 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 19.13 sec
MapReduce Total cumulative CPU time: 19 seconds 130 msec
Ended Job = job_1732089968849_2750
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 28.55 sec HDFS Read: 3149803 HDFS Write: 106222 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 4 Cumulative CPU: 19.13 sec HDFS Read: 128076 HDFS Write: 882 SUCCESS
Total MapReduce CPU Time Spent: 47 seconds 680 msec
OK
Aarhus
Abakan
Abbotsford
Adi Sutjipto
A Coruna
Abadan
Abdul Rachman Saleh
Abel Santamaria
Abidjan Felix Houphouet Boigny Intl
Abraham Gonzalez Intl
Time taken: 57.455 seconds, Fetched: 10 row(s)
hive (cdac_sahil)>
```

3.

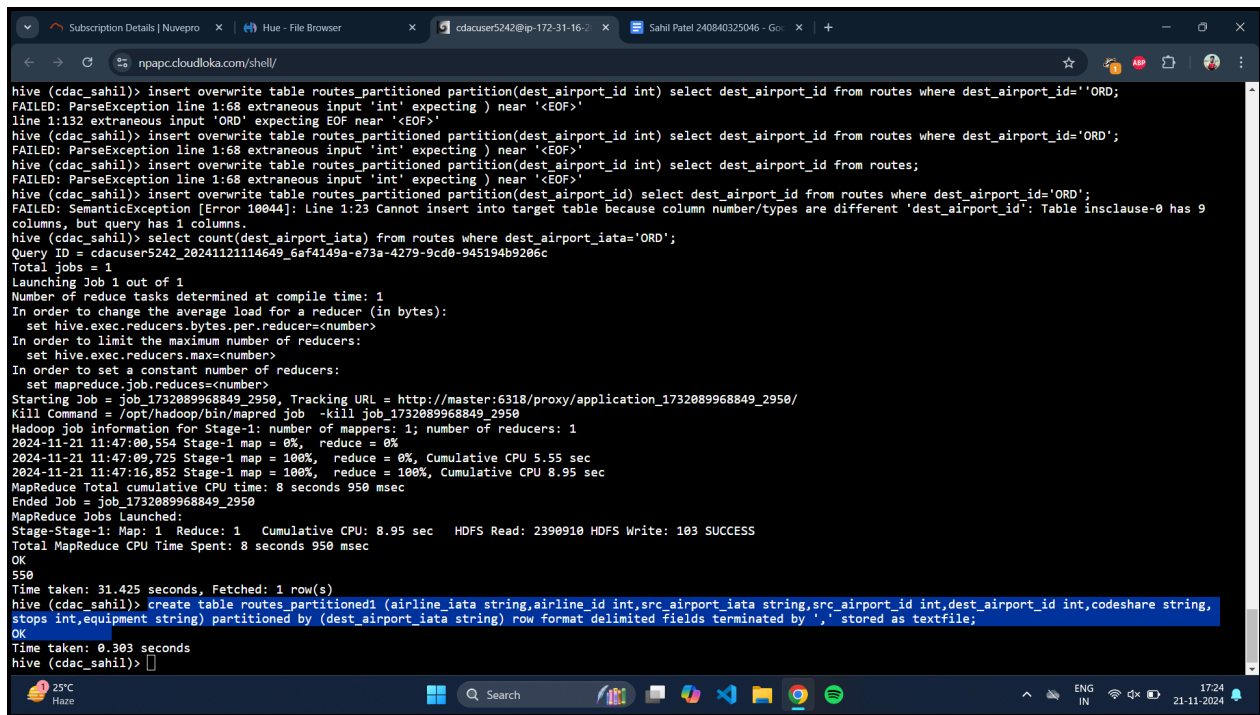
```
select count(distinct equipment) from routes;
```

```
Subscription Details | Nuvepro x Hue - File Browser x cdacuser5242@ip-172-31-16-2 x Sahil Patel 240840325046 - Go x +
npapc.cloudloka.com/shell/
OK
airport_id      int
name            string
city            string
country         string
iata            string
icao             string
latitude        double
longitude        double
altitude        int
timezone        double
dst             string
tz             string
Time taken: 0.044 seconds, Fetched: 12 row(s)
hive (cdac_sahil)> select count(distinct equipment) from routes;
Query ID = cdacuser5242_20241121111544_be32c08b-a68a-451c-a285-bb783ba131a8
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_2817, Tracking URL = http://master:6318/proxy/application_1732089968849_2817/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2817
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-11-21 11:15:55,892 Stage-1 map = 0%, reduce = 0%
2024-11-21 11:16:03,030 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.77 sec
2024-11-21 11:16:11,182 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 8.18 sec
MapReduce Total cumulative CPU time: 8 seconds 180 msec
Ended Job = job_1732089968849_2817
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.18 sec HDFS Read: 2385303 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 180 msec
OK
3946
Time taken: 29.38 seconds, Fetched: 1 row(s)
hive (cdac_sahil)> select count(equipment) from routes;
Query ID = cdacuser5242_20241121111632_5871896c-a476-4072-9d7b-f0c4d45769eb
```

## Question 2

1.

```
create table routes_partitioned1 (airline_iata string,airline_id
int,src_airport_iata string,src_airport_id int,dest_airport_id
int,codeshare string,
stops int,equipment string) partitioned by (dest_airport_iata string)
row format delimited fields terminated by ',' stored as textfile;
```



```
hive (cdac_sahil)> insert overwrite table routes_partitioned partition(dest_airport_id int) select dest_airport_id from routes where dest_airport_id='ORD';
FAILED: ParseException line 1:68 extraneous input 'int' expecting ) near '<EOF>'
line 1:132 extraneous input 'ORD' expecting EOF near '<EOF>'
hive (cdac_sahil)> insert overwrite table routes_partitioned partition(dest_airport_id int) select dest_airport_id from routes where dest_airport_id='ORD';
FAILED: ParseException line 1:68 extraneous input 'int' expecting ) near '<EOF>'
hive (cdac_sahil)> insert overwrite table routes_partitioned partition(dest_airport_id int) select dest_airport_id from routes;
FAILED: ParseException line 1:68 extraneous input 'int' expecting ) near '<EOF>'
hive (cdac_sahil)> insert overwrite table routes_partitioned partition(dest_airport_id) select dest_airport_id from routes where dest_airport_id='ORD';
FAILED: SemanticException [Error 10044]: Line 1:23 Cannot insert into target table because column number/types are different 'dest_airport_id': Table insclause-0 has 9
columns, but query has 1 columns.
hive (cdac_sahil)> select count(dest_airport_iata) from routes where dest_airport_iata='ORD';
Query ID = cdacuser5242_20241121114649_6af4149a-e73a-4279-9cd0-945194b9206c
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_2950, Tracking URL = http://master:6318/proxy/application_1732089968849_2950/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2950
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-11-21 11:47:00,554 Stage-1 map = 0%, reduce = 0%
2024-11-21 11:47:09,725 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.55 sec
2024-11-21 11:47:16,852 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 8.95 sec
MapReduce Total cumulative CPU time: 8 seconds 950 msec
Ended Job = job_1732089968849_2950
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.95 sec HDFS Read: 2390910 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 950 msec
OK
550
Time taken: 31.425 seconds, Fetched: 1 row(s)
hive (cdac_sahil)> create table routes_partitioned1 (airline_iata string,airline_id int,src_airport_iata string,src_airport_id int,dest_airport_id int,codeshare string,
stops int,equipment string) partitioned by (dest_airport_iata string) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.303 seconds
hive (cdac_sahil)>
```

2.

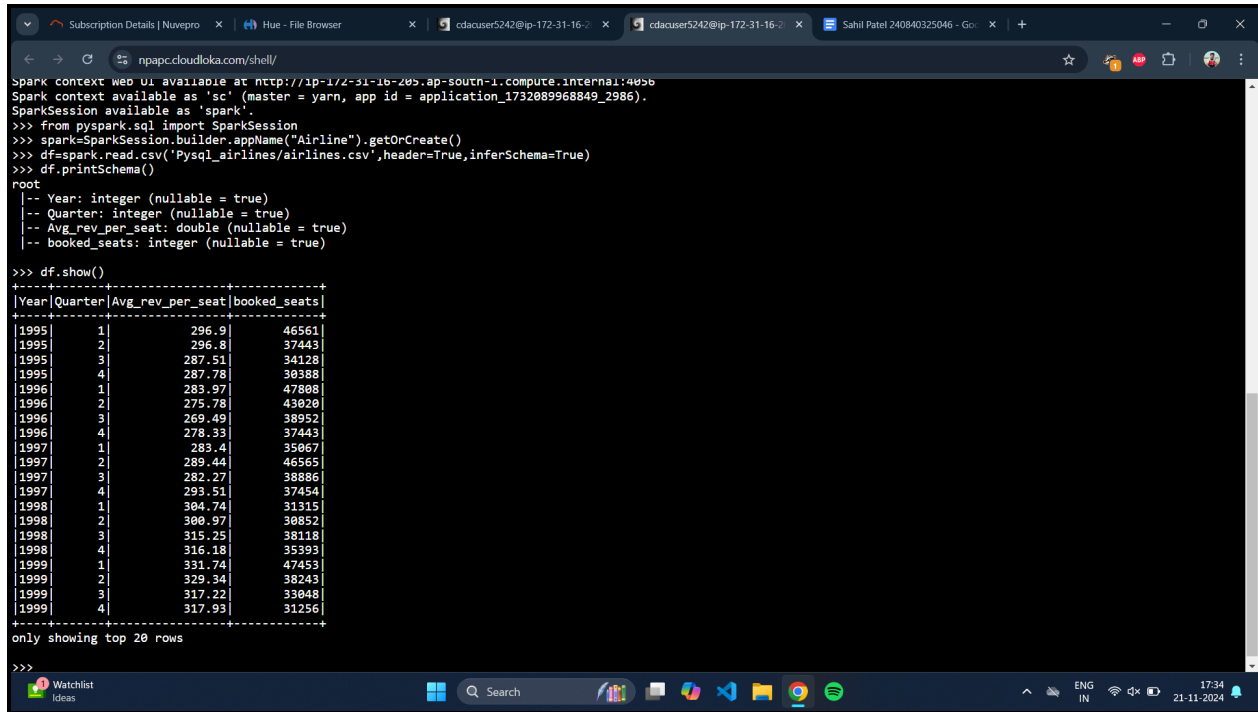
```
insert overwrite table routes_partitioned1
partition(dest_airport_iata) select dest_airport_iata from routes;
```

Spark

Question 2:

1.

```
from pyspark.sql import SparkSession
>>> spark=SparkSession.builder.appName("Airline").getOrCreate()
>>>
df=spark.read.csv('Pysql_airlines/airlines.csv',header=True,inferSchema=True)
>>> df.printSchema()
df.show()
```



The screenshot shows a terminal window with the following content:

```
Spark context web UI available at http://1p-172-31-16-205.ap-south-1.compute.internal:4056
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2986).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark=SparkSession.builder.appName("Airline").getOrCreate()
>>> df=spark.read.csv('Pysql_airlines/airlines.csv',header=True,inferSchema=True)
>>> df.printSchema()
root
 |-- Year: integer (nullable = true)
 |-- Quarter: integer (nullable = true)
 |-- Avg_rev_per_seat: double (nullable = true)
 |-- booked_seats: integer (nullable = true)
>>> 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|38388|
|1996|1|283.97|47888|
|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|38852|
|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
>>>
```

Question 2:

3.

```
df.groupby("quarter").agg(avg("booked_seats")).show()
```

```
Subscription Details | Nuvepro x Hue - File Browser x cdacuser5242@ip-172-31-16-2 x cdacuser5242@ip-172-31-16-2 x Sahil Patel 240840325046 - Go x +
npapc.cloudloka.com/shell/
File "<stdin>", line 1, in <module>
NameError: name 'col' is not defined
>>> from pyspark.sql.functions import col
>>> df.filter(min(col("booked_seats")))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/opt/spark-3.1.2/python/pyspark/sql/column.py", line 460, in __iter__
raise TypeError("Column is not iterable")
TypeError: Column is not iterable
>>> df.min(col("booked_seats"))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/opt/spark-3.1.2/python/pyspark/sql/dataframe.py", line 1643, in __getattr__
raise AttributeError(
AttributeError: 'DataFrame' object has no attribute 'min'
>>> from pyspark.sql.functions import *
>>> df.min(col("booked_seats"))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/opt/spark-3.1.2/python/pyspark/sql/dataframe.py", line 1643, in __getattr__
raise AttributeError(
AttributeError: 'DataFrame' object has no attribute 'min'
>>> from pyspark.sql.functions import min,max,avg;
>>> df.min(col("booked_seats"))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/opt/spark-3.1.2/python/pyspark/sql/dataframe.py", line 1643, in __getattr__
raise AttributeError(
AttributeError: 'DataFrame' object has no attribute 'min'
>>> df.groupby("quarter").agg(avg("booked_seats")).show()
+-----+
|quarter| avg(booked_seats)|
+-----+
1|41607.666666666664|
3| 39386.23809523809|
4| 39111.95238095238|
2| 38456.95238095238|
+-----+
>>> 
```

4.

```
df.groupby("year").distinct().count().show()
```

5.

```
df.groupBy("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1).show()
```

```
Subscription Details | Nuvepro x Hue - File Browser x cdacuser5242@ip-172-31-16-2 x cdacuser5242@ip-172-31-16-2 x Sahil Patel 240840325046 - Go x +
npapc.cloudloka.com/shell/
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/opt/spark-3.1.2/python/pyspark/sql/dataframe.py", line 1643, in __getattr__
    raise AttributeError(
AttributeError: 'DataFrame' object has no attribute 'col'
>>> df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).show()
+-----+-----+
|Quarter|Highest Total Revenue|
+-----+-----+
|1|2.8886029941999996E8|
|3|2.7197394587999994E8|
|4|2.7098627115999997E8|
|2|2.6833043004000002E8|
+-----+-----+

>>> df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1)
show()
File "<stdin>", line 1
df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1)
show()
^
SyntaxError: invalid syntax
>>> df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1)
show()
File "<stdin>", line 1
df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1)
show()
^
SyntaxError: invalid syntax
>>> df.groupby("Quarter").agg(sum(col("avg_rev_per_seat")*col("booked_seats")).alias("Highest Total Revenue")).orderBy("Highest Total Revenue",ascending=False).limit(1)
.show()
+-----+-----+
|Quarter|Highest Total Revenue|
+-----+-----+
|1|2.8886029941999996E8|
+-----+-----+

>>> 
```

2.

```
(df.avg_rev_per_seat < 290).count()
```