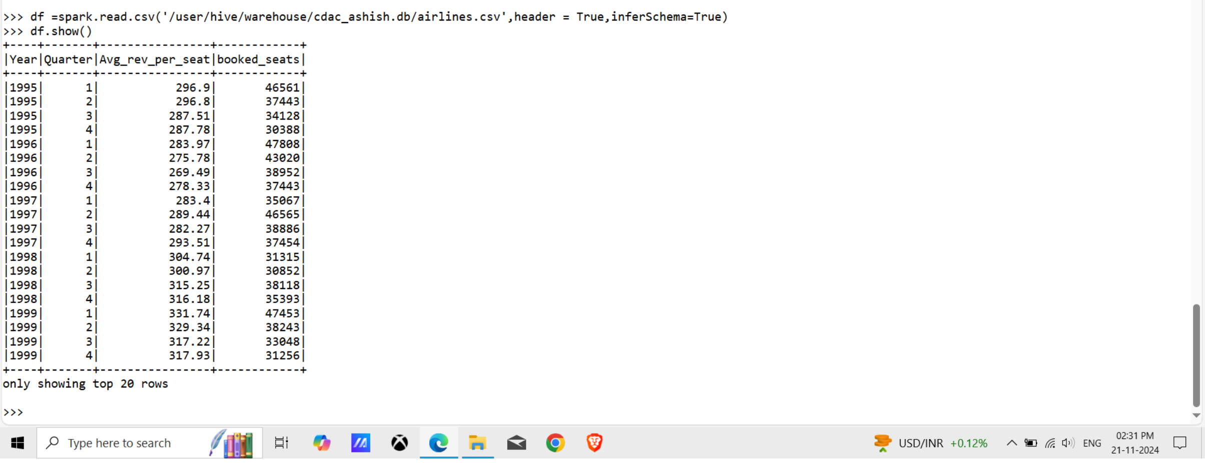
Spark Question 2 Answers:

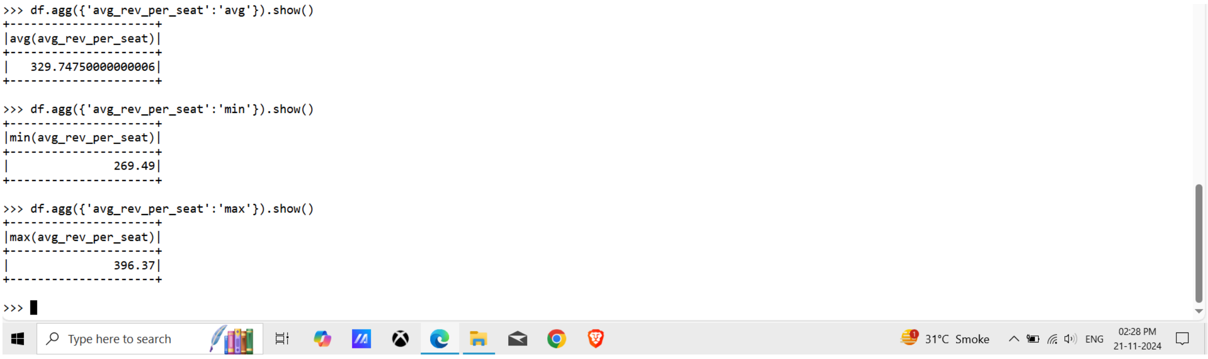
Loading data in data frame:



```
>>> df =spark.read.csv('/user/hive/warehouse/cdac_ashish.db/airlines.csv',header = True,inferSchema=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|       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



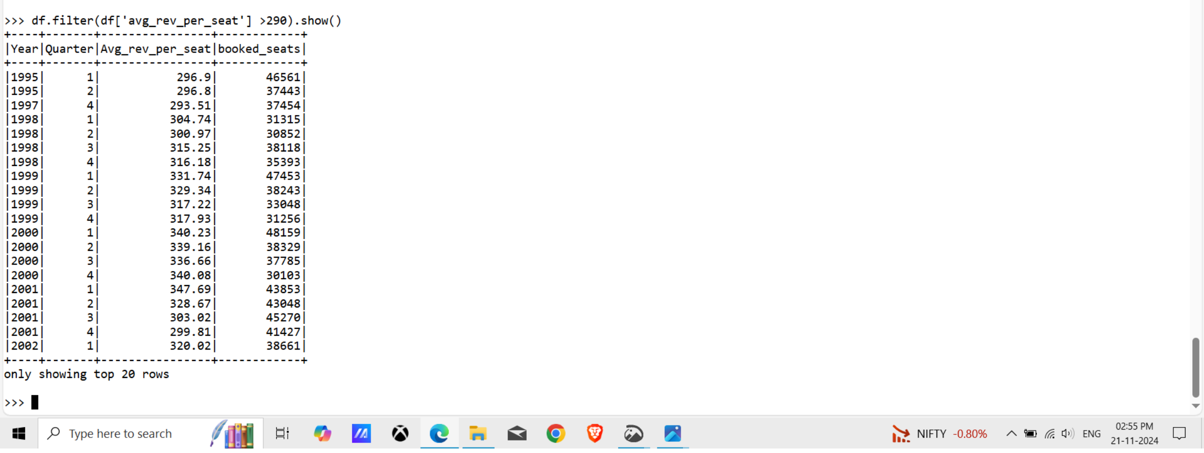
```
>>> df.agg({'avg_rev_per_seat':'avg'}).show()
+---------------------+
|avg(avg_rev_per_seat)|
+---------------------+
|   329.74750000000006|
+---------------------+

>>> df.agg({'avg_rev_per_seat':'min'}).show()
+---------------------+
|min(avg_rev_per_seat)|
+---------------------+
|               269.49|
+---------------------+

>>> df.agg({'avg_rev_per_seat':'max'}).show()
+---------------------+
|max(avg_rev_per_seat)|
+---------------------+
|               396.37|
+---------------------+

>>> █
```

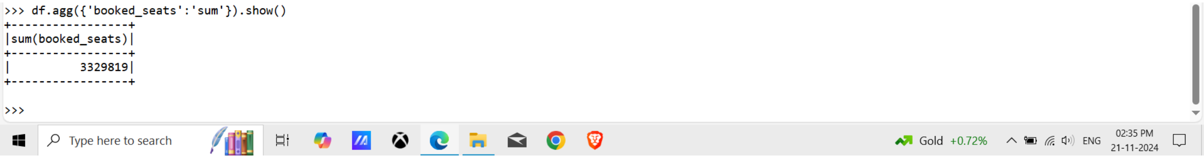
2



```
>>> df.filter(df['avg_rev_per_seat'] >290).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

>>> 
```

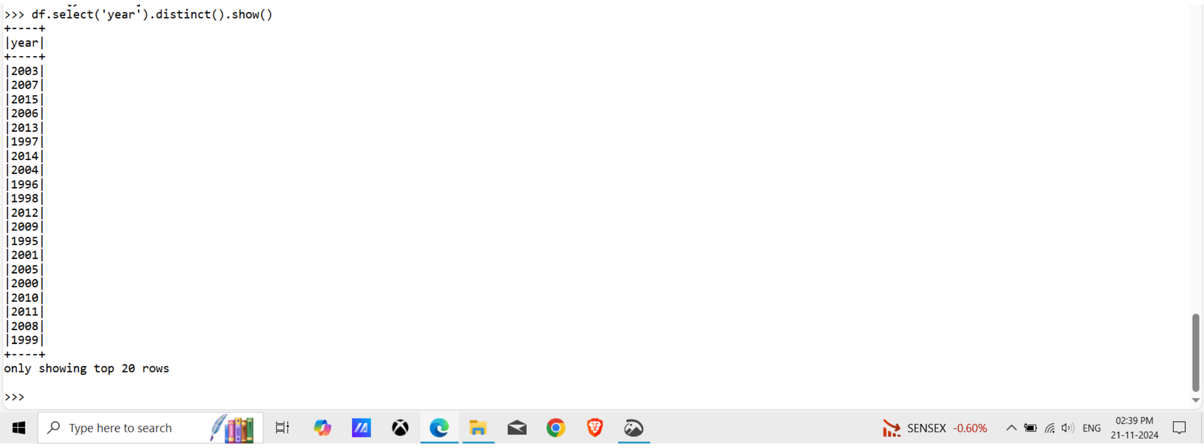
3



```
>>> df.agg({'booked_seats':'sum'}).show()
+-----------------+
|sum(booked_seats)|
+-----------------+
|          3329819|
+-----------------+

>>>
```

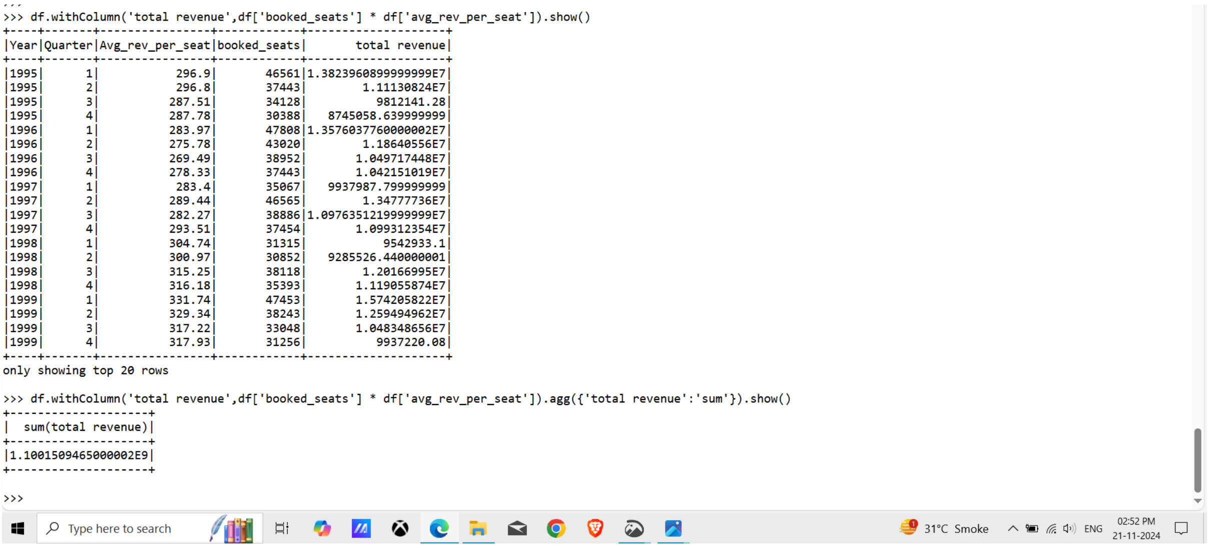
4



```
>>> 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



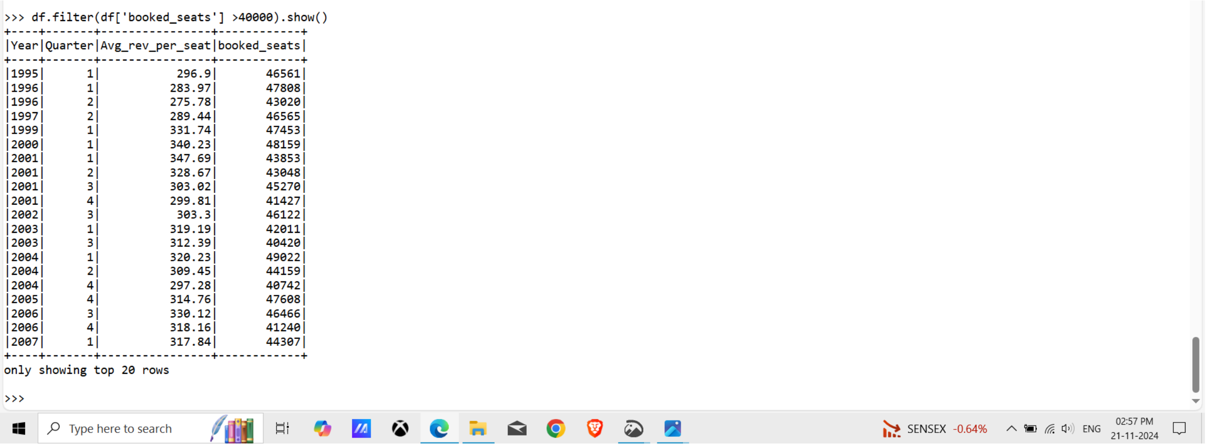
```
...
>>> df.withColumn('total revenue',df['booked_seats'] * df['avg_rev_per_seat']).show()
+----+-------+----------------+------------+--------------------+
|Year|Quarter|Avg_rev_per_seat|booked_seats|       total revenue|
+----+-------+----------------+------------+--------------------+
|1995|      1|           296.9|       46561|1.382396089999999E7|
|1995|      2|           296.8|       37443|        1.11130824E7|
|1995|      3|          287.51|       34128|          9812141.28|
|1995|      4|          287.78|       30388|   8745058.639999999|
|1996|      1|          283.97|       47808|1.357603776000002E7|
|1996|      2|          275.78|       43020|        1.18640556E7|
|1996|      3|          269.49|       38952|       1.049717448E7|
|1996|      4|          278.33|       37443|       1.042151019E7|
|1997|      1|           283.4|       35067|   9937987.799999999|
|1997|      2|          289.44|       46565|        1.34777736E7|
|1997|      3|          282.27|       38886|1.097635121999999E7|
|1997|      4|          293.51|       37454|       1.099312354E7|
|1998|      1|          304.74|       31315|           9542933.1|
|1998|      2|          300.97|       30852|   9285526.440000001|
|1998|      3|          315.25|       38118|        1.20166995E7|
|1998|      4|          316.18|       35393|       1.119055874E7|
|1999|      1|          331.74|       47453|       1.574205822E7|
|1999|      2|          329.34|       38243|       1.259494962E7|
|1999|      3|          317.22|       33048|       1.048348656E7|
|1999|      4|          317.93|       31256|          9937220.08|
+----+-------+----------------+------------+--------------------+
only showing top 20 rows

>>> df.withColumn('total revenue',df['booked_seats'] * df['avg_rev_per_seat']).agg({'total revenue':'sum'}).show()
+--------------------+
|  sum(total revenue)|
+--------------------+
|1.1001509465000002E9|
+--------------------+

>>>
```

Question 1 Answers:

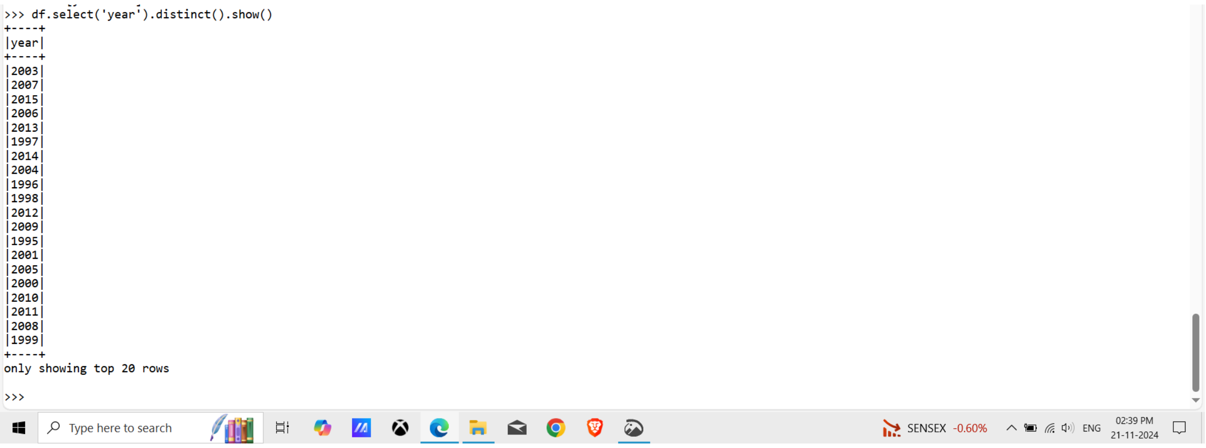
1



```
>>> df.filter(df['booked_seats'] >40000).show()
+----+-------+----------------+------------+
|Year|Quarter|Avg_rev_per_seat|booked_seats|
+----+-------+----------------+------------+
|1995|      1|           296.9|       46561|
|1996|      1|          283.97|       47808|
|1996|      2|          275.78|       43020|
|1997|      2|          289.44|       46565|
|1999|      1|          331.74|       47453|
|2000|      1|          340.23|       48159|
|2001|      1|          347.69|       43853|
|2001|      2|          328.67|       43048|
|2001|      3|          303.02|       45270|
|2001|      4|          299.81|       41427|
|2002|      3|           303.3|       46122|
|2003|      1|          319.19|       42011|
|2003|      3|          312.39|       40420|
|2004|      1|          320.23|       49022|
|2004|      2|          309.45|       44159|
|2004|      4|          297.28|       40742|
|2005|      4|          314.76|       47608|
|2006|      3|          330.12|       46466|
|2006|      4|          318.16|       41240|
|2007|      1|          317.84|       44307|
+----+-------+----------------+------------+
only showing top 20 rows

>>>
```

2



```
>>> 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

>>>
```

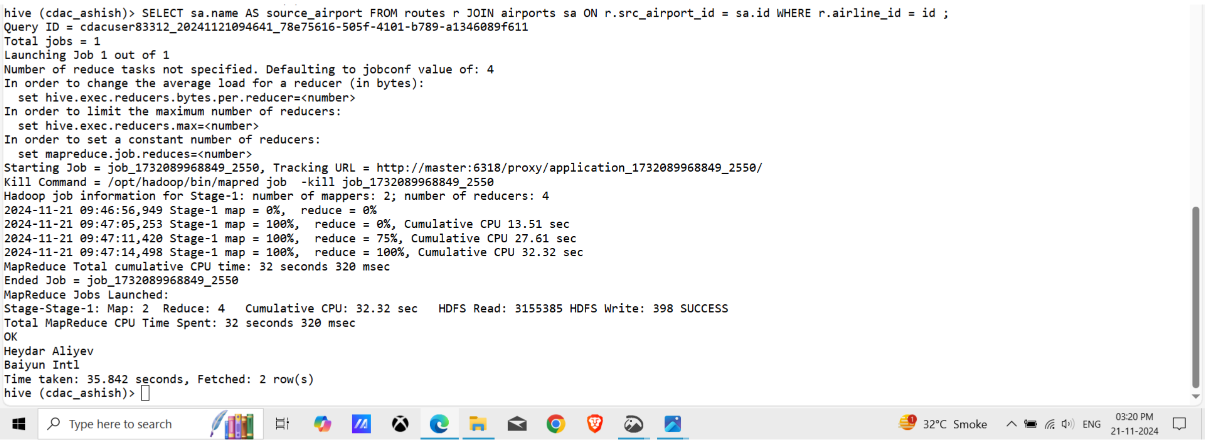
Hive

Question1

3

```
SELECT name,
COUNT(airline_id) AS
route_count FROM routes
GROUP BY airline_iata
ORDER BY route_count DESC
LIMIT 1;
```

1



```
hive (cdac_ashish)> SELECT sa.name AS source_airport FROM routes r JOIN airports sa ON r.src_airport_id = sa.id WHERE r.airline_id = id ;
Query ID = cdacuser83312_20241121094641_78e75616-505f-4101-b789-a1346089f611
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_2550, Tracking URL = http://master:6318/proxy/application_1732089968849_2550/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2550
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 4
2024-11-21 09:46:56,949 Stage-1 map = 0%,  reduce = 0%
2024-11-21 09:47:05,253 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 13.51 sec
2024-11-21 09:47:11,420 Stage-1 map = 100%,  reduce = 75%, Cumulative CPU 27.61 sec
2024-11-21 09:47:14,498 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 32.32 sec
MapReduce Total cumulative CPU time: 32 seconds 320 msec
Ended Job = job_1732089968849_2550
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 32.32 sec   HDFS Read: 3155385 HDFS Write: 398 SUCCESS
Total MapReduce CPU Time Spent: 32 seconds 320 msec
OK
Heydar Aliyev
Baiyun Intl
Time taken: 35.842 seconds, Fetched: 2 row(s)
hive (cdac_ashish)> 
```