```
>>> airline_df = spark.read.csv("/user/cdacuser5233/airlines.csv",inferSchema=True,header=True)
>>> ariline_df.printSchema()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ariline_df' is not defined
>>> airline_df.printSchema()
root
 |-- Year: integer (nullable = true)
 |-- Quarter: integer (nullable = true)
 |-- Avg_rev_per_seat: double (nullable = true)
 |-- booked_seats: integer (nullable = true)

>>> airline_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|
```

```
>>> airline_df.printSchema()
root
 |-- Year: integer (nullable = true)
 |-- Quarter: integer (nullable = true)
 |-- Avg_rev_per_seat: double (nullable = true)
 |-- booked_seats: integer (nullable = true)

>>> airline_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|
```

SPARK QUESTION 2, USing DataFrame.

Q.1)

```
>>> airline_df.groupBy('year').agg(max('booked_seats'),min('booked_seats'),avg('booked_seats')).orderBy('year').show()
+----+----------------+----------------+----------------+
|year|max(booked_seats)|min(booked_seats)|avg(booked_seats)|
+----+----------------+----------------+----------------+
|1995|           46561|           30388|         37130.0|
|1996|           47808|           37443|        41805.75|
|1997|           46565|           35067|         39493.0|
|1998|           38118|           30852|         33919.5|
|1999|           47453|           31256|         37500.0|
|2000|           48159|           30103|         38594.0|
|2001|           45270|           41427|         43399.5|
|2002|           46122|           32406|        38048.75|
|2003|           42011|           33824|        39038.25|
|2004|           49022|           30877|         41200.0|
|2005|           47608|           32003|         37652.5|
|2006|           46466|           32621|        38447.25|
|2007|           47758|           41241|        44074.75|
|2008|           46885|           37217|        41724.25|
|2009|           44186|           32491|         37577.0|
|2010|           49678|           33099|        40935.25|
|2011|           41927|           30562|        35661.75|
|2012|           43456|           39474|         41519.0|
|2013|           49143|           38613|         43419.0|
|2014|           47928|           35014|        39955.75|
+----+----------------+----------------+----------------+
only showing top 20 rows

>>>
```

Q.3)

```
|2006|           46466|           32621|        38447.25|
|2007|           47758|           41241|        44074.75|
|2008|           46885|           37217|        41724.25|
|2009|           44186|           32491|         37577.0|
|2010|           49678|           33099|        40935.25|
|2011|           41927|           30562|        35661.75|
|2012|           43456|           39474|         41519.0|
|2013|           49143|           38613|         43419.0|
|2014|           47928|           35014|        39955.75|
+----+----------------+----------------+----------------+
only showing top 20 rows

>>> airline_df.groupBy('quarter').agg('avg('booked_seats').alias('avg_booked_seats')).orderBy('quarter').show()
  File "<stdin>", line 1
    airline_df.groupBy('quarter').agg('avg('booked_seats').alias('avg_booked_seats')).orderBy('quarter').show()
                                      ^
SyntaxError: invalid syntax
>>> airline_df.groupBy('quarter').agg(avg('booked_seats').alias('avg_booked_seats')).orderBy('quarter').show()
+-------+------------------+
|quarter|  avg_booked_seats|
+-------+------------------+
|      1|41607.666666666664|
|      2| 38456.95238095238|
|      3| 39386.23809523809|
|      4| 39111.95238095238|
+-------+------------------+

>>>
```

2Q)

```
|      4| 39111.95238095238|
+-------+------------------+

>>> airline_df.withColumn('avg_revenue',expr('avg_rev_per_seat<290')).count('avg_revenue').show()
  File "<stdin>", line 1
    airline_df..withColumn('avg_revenue',expr('avg_rev_per_seat<290')).count('avg_revenue').show()
              ^
SyntaxError: invalid syntax
>>> airline_df.withColumn('avg_revenue',expr('avg_rev_per_seat<290')).count('avg_revenue').show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: count() takes 1 positional argument but 2 were given
>>> airline_df.withColumn('avg_revenue',expr('avg_rev_per_seat<290')).count().show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'int' object has no attribute 'show'
>>> airline_df.withColumn('avg_revenue',expr('avg_rev_per_seat<290')).count('avg_revenue')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: count() takes 1 positional argument but 2 were given
>>> airline_df.withColumn('avg_revenue',expr('avg_rev_per_seat<290')).agg(count('avg_revenue')).show()
+------------------+
|count(avg_revenue)|
+------------------+
|                84|
+------------------+

>>>
```

4)

```
>>> airline_df.groupBy('year').agg(count('year')).orderBy('year').show()
+----+-----------+
|year|count(year)|
+----+-----------+
|1995|          4|
|1996|          4|
|1997|          4|
|1998|          4|
|1999|          4|
|2000|          4|
|2001|          4|
|2002|          4|
|2003|          4|
|2004|          4|
|2005|          4|
|2006|          4|
|2007|          4|
|2008|          4|
|2009|          4|
|2010|          4|
|2011|          4|
|2012|          4|
|2013|          4|
|2014|          4|
+----+-----------+
only showing top 20 rows

>>>
```

5)

```
>>> airline_df.withColumn('total_revenue',expr('avg_rev_per_seat*booked_seats')).groupBy('year','quarter').agg(sum('total_revenue').alias('t
otal_revenue')).orderBy('year','quarter').show()
+----+-------+--------------------+
|year|quarter|       total_revenue|
+----+-------+--------------------+
|1995|      1|1.3823960899999999E7|
|1995|      2|        1.11130824E7|
|1995|      3|          9812141.28|
|1995|      4|   8745058.639999999|
|1996|      1|1.3576037760000002E7|
|1996|      2|        1.18640556E7|
|1996|      3|       1.049717448E7|
|1996|      4|       1.042151019E7|
|1997|      1|   9937987.799999999|
|1997|      2|        1.34777736E7|
|1997|      3|1.0976351219999999E7|
|1997|      4|       1.099312354E7|
|1998|      1|          9542933.1|
|1998|      2|   9285526.440000001|
|1998|      3|        1.20166995E7|
|1998|      4|       1.119055874E7|
|1999|      1|       1.574205822E7|
|1999|      2|       1.259494962E7|
|1999|      3|       1.048348656E7|
|1999|      4|          9937220.08|
+----+-------+--------------------+
only showing top 20 rows
```

HIVE Question 2

Q3)

```
nyse
rank_table
routes
stockdata
txn_orc
txn_parquet
txnrecords
txnrecsbycat
txnrecsbycat2
txnrecsbycat3
txnrecsbycat4
txnrecsbycat6
txnrecsbystate
Time taken: 0.112 seconds, Fetched: 20 row(s)
hive (cdac_sahilj)> select * from routes where dest_airport_iata = 'ORD' limit 10;
OK
3E      10739   BRL     5726    ORD     3830            0       CNC
3E      10739   DEC     4042    ORD     3830            0       CNC
AA      24      ABQ     4019    ORD     3830    Y       0       E75
AA      24      ALO     5718    ORD     3830    Y       0       ERD
AA      24      AMM     2170    ORD     3830    Y       0       340
AA      24      ART     3838    ORD     3830    Y       0       ERD
AA      24      ATL     3682    ORD     3830    Y       0       CR7 E75
AA      24      AUH     2179    ORD     3830    Y       0       777
AA      24      AUS     3673    ORD     3830            0       M83 M80
AA      24      AZO     4039    ORD     3830    Y       0       ER4 ERD
Time taken: 2.501 seconds, Fetched: 10 row(s)
hive (cdac_sahilj)>
```

Question 1) in

3)

```
AA      24      AZO     4039    ORD     3830    Y       0       ER4 ERD
Time taken: 2.501 seconds, Fetched: 10 row(s)
hive (cdac_sahilj)> select count(distinct(equipment)) as total_equip from routes;
Query ID = cdacuser5233_20241121121919_59d76355-129a-454a-a164-5067e830cf74
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_3041, Tracking URL = http://master:6318/proxy/application_1732089968849_3041/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_3041
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-11-21 12:19:35,880 Stage-1 map = 0%,  reduce = 0%
2024-11-21 12:19:42,032 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.68 sec
2024-11-21 12:19:50,203 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.64 sec
MapReduce Total cumulative CPU time: 7 seconds 640 msec
Ended Job = job_1732089968849_3041
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.64 sec   HDFS Read: 2385230 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 640 msec
OK
3946
Time taken: 33.77 seconds, Fetched: 1 row(s)
hive (cdac_sahilj)> select count(distinct(equipment)) as total_equip from routes;
```

Question 1)

2)

```
2024-11-21 12:25:12,677 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 29.46 sec
MapReduce Total cumulative CPU time: 29 seconds 460 msec
Ended Job = job_1732089968849_3053
Launching Job 2 out of 2
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_3055, Tracking URL = http://master:6318/proxy/application_1732089968849_3055/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_3055
Hadoop job information for Stage-2: number of mappers: 3; number of reducers: 1
2024-11-21 12:25:26,335 Stage-2 map = 0%,  reduce = 0%
2024-11-21 12:25:33,474 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 12.0 sec
2024-11-21 12:25:39,587 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 15.31 sec
MapReduce Total cumulative CPU time: 15 seconds 310 msec
Ended Job = job_1732089968849_3055
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 4   Cumulative CPU: 29.46 sec   HDFS Read: 2727314 HDFS Write: 2701705 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 15.31 sec   HDFS Read: 2716726 HDFS Write: 198 SUCCESS
Total MapReduce CPU Time Spent: 44 seconds 770 msec
OK
Rainbow Air Polynesia   ZYL     DAC
Viking Hellas   ZYL     DAC
United Airways  ZYL     DAC
Time taken: 57.407 seconds, Fetched: 3 row(s)
hive (cdac_sahilj)>
```

1)

```
hive> select a.name,r.src_airport_iata from airport a join routes r on a.airport_id = r.src_airport_id where r.src_airport_iata!=r.dest_airp
ort_iata limit 10;
Query ID = cdacuser5233_20241121123317_09c2c9f4-cafe-4b25-9330-ffb3177c9685
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_3068, Tracking URL = http://master:6318/proxy/application_1732089968849_3068/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_3068
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 12:33:30,198 Stage-1 map = 0%,  reduce = 0%
2024-11-21 12:33:39,413 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.45 sec
2024-11-21 12:33:45,569 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 13.77 sec
2024-11-21 12:33:47,612 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 21.09 sec
MapReduce Total cumulative CPU time: 21 seconds 90 msec
Ended Job = job_1732089968849_3068
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 4   Cumulative CPU: 21.09 sec   HDFS Read: 773176 HDFS Write: 348 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 90 msec
OK
Time taken: 33.012 seconds
hive> select a.name,r.src_airport_iata from airport a join routes r on a.airport_id = r.src_airport_id where r.src_airport_iata!=r.dest_airp
ort_iata limit 20;
```

```
hive> select a.name,r.src_airport_iata from airport a join routes r on a.airport_id = r.src_airport_id where r.src_airport_iata!=r.dest_airp
ort_iata limit 20;
Query ID = cdacuser5233_20241121123433_ac8db76b-d42a-4f10-ad15-b2ac0e946332
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_3074, Tracking URL = http://master:6318/proxy/application_1732089968849_3074/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_3074
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 12:34:45,111 Stage-1 map = 0%,  reduce = 0%
2024-11-21 12:34:52,268 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.61 sec
2024-11-21 12:34:58,389 Stage-1 map = 100%,  reduce = 25%, Cumulative CPU 10.36 sec
2024-11-21 12:34:59,410 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 21.21 sec
MapReduce Total cumulative CPU time: 21 seconds 210 msec
Ended Job = job_1732089968849_3074
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 4   Cumulative CPU: 21.21 sec   HDFS Read: 773194 HDFS Write: 348 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 210 msec
OK
Time taken: 29.37 seconds
hive> select a.name,r.src_airport_iata from airport a join routes r on a.airport_id = r.src_airport_id where r.src_airport_iata!=r.dest_airp
ort_iata;
```