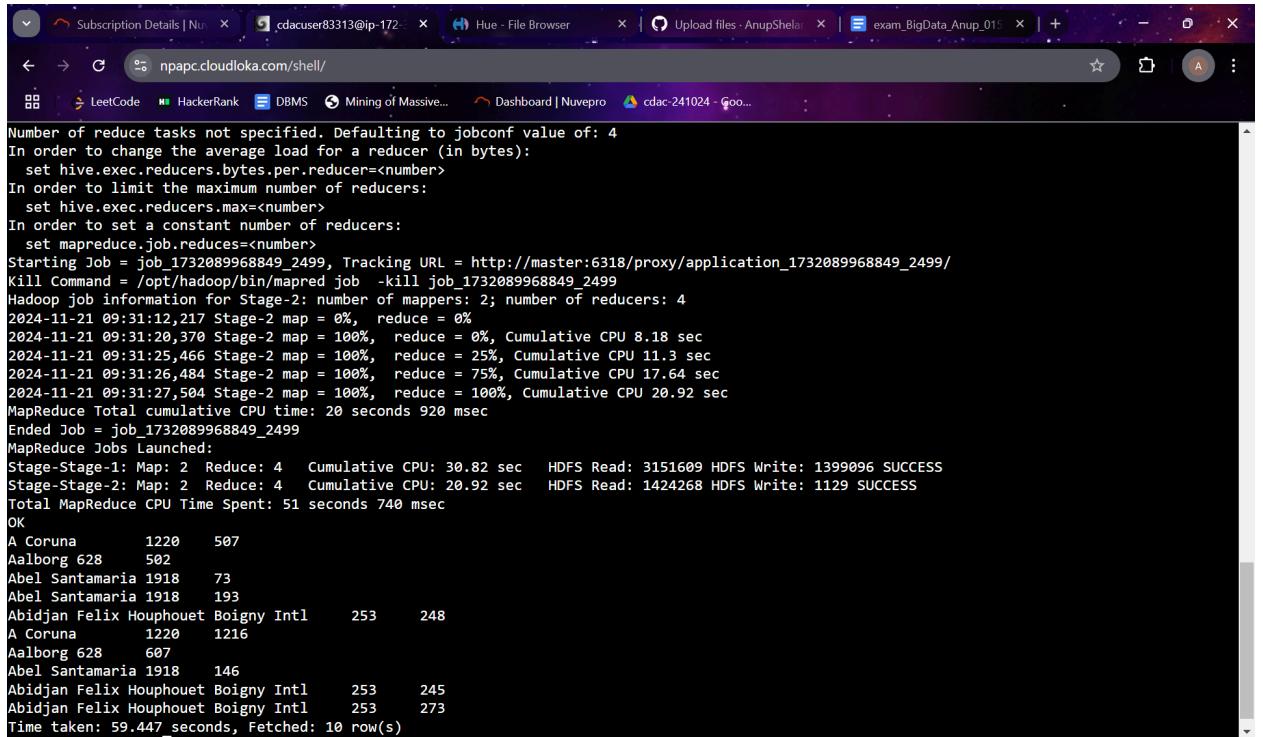


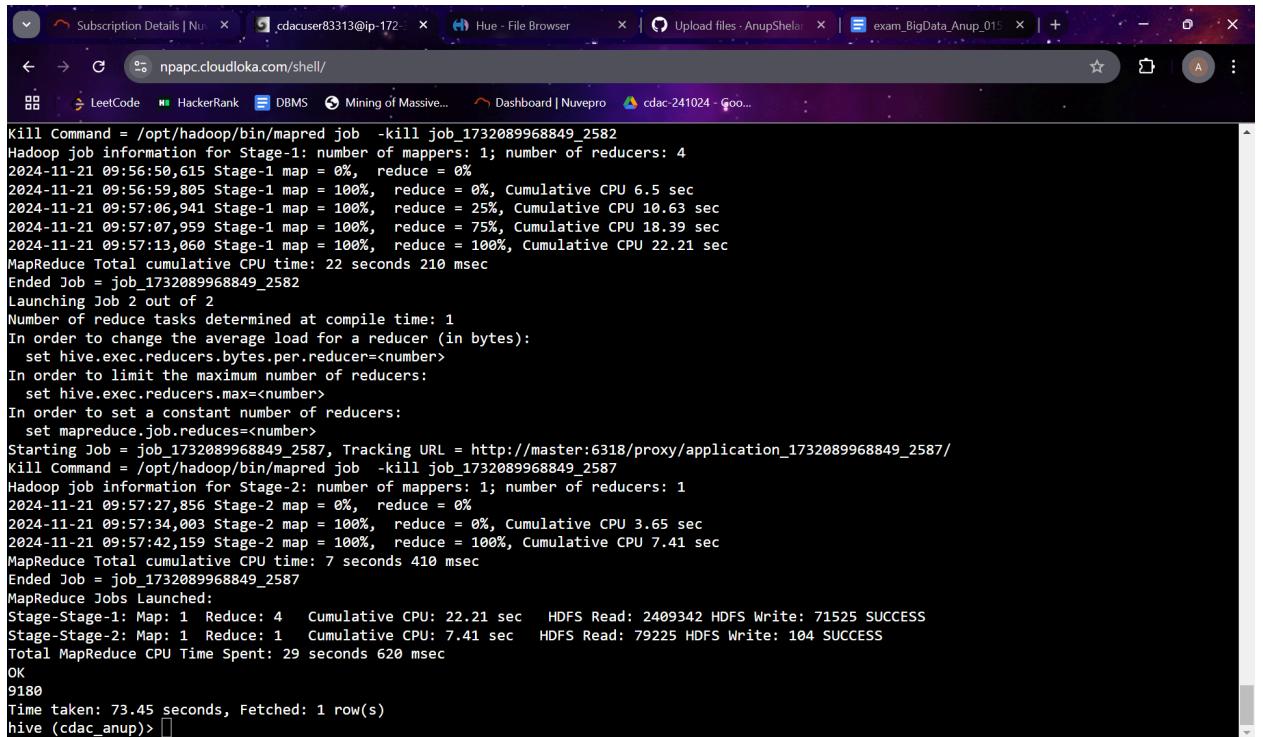
Hive

- 1) select a.name, r.src_airport_id, r.dest_airport_id from airport a join routes r on a.airport_id=r.src_airport_id limit 10;



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_2499, Tracking URL = http://master:6318/proxy/application_1732089968849_2499/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2499
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 4
2024-11-21 09:31:12,217 Stage-2 map = 0%, reduce = 0%
2024-11-21 09:31:20,370 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 8.18 sec
2024-11-21 09:31:25,466 Stage-2 map = 100%, reduce = 25%, Cumulative CPU 11.3 sec
2024-11-21 09:31:26,484 Stage-2 map = 100%, reduce = 75%, Cumulative CPU 17.64 sec
2024-11-21 09:31:27,504 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 20.92 sec
MapReduce Total cumulative CPU time: 20 seconds 920 msec
Ended Job = job_1732089968849_2499
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 30.82 sec HDFS Read: 3151609 HDFS Write: 1399096 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 4 Cumulative CPU: 20.92 sec HDFS Read: 1424268 HDFS Write: 1129 SUCCESS
Total MapReduce CPU Time Spent: 51 seconds 740 msec
OK
A Coruna 1220 507
Aalborg 628 502
Abel Santamaría 1918 73
Abel Santamaría 1918 193
Abidjan Felix Houphouët Boigny Intl 253 248
A Coruna 1220 1216
Aalborg 628 607
Abel Santamaría 1918 146
Abidjan Felix Houphouët Boigny Intl 253 245
Abidjan Felix Houphouët Boigny Intl 253 273
Time taken: 59.447 seconds, Fetched: 10 row(s)

- 2) select count(airline_id) as high from routes group by equipment order by high desc limit 1;



Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2582
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 09:56:50,615 Stage-1 map = 0%, reduce = 0%
2024-11-21 09:56:59,805 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.5 sec
2024-11-21 09:57:06,941 Stage-1 map = 100%, reduce = 25%, Cumulative CPU 10.63 sec
2024-11-21 09:57:07,959 Stage-1 map = 100%, reduce = 75%, Cumulative CPU 18.39 sec
2024-11-21 09:57:13,060 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 22.21 sec
MapReduce Total cumulative CPU time: 22 seconds 210 msec
Ended Job = job_1732089968849_2582
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_2587, Tracking URL = http://master:6318/proxy/application_1732089968849_2587/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2587
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2024-11-21 09:57:27,856 Stage-2 map = 0%, reduce = 0%
2024-11-21 09:57:34,003 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.65 sec
2024-11-21 09:57:42,159 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 7.41 sec
MapReduce Total cumulative CPU time: 7 seconds 410 msec
Ended Job = job_1732089968849_2587
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 4 Cumulative CPU: 22.21 sec HDFS Read: 2409342 HDFS Write: 71525 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 7.41 sec HDFS Read: 79225 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 29 seconds 620 msec
OK
9180
Time taken: 73.45 seconds, Fetched: 1 row(s)
hive (cdac_anup)>

- 3) select count(r.airline_id) as high from routes r join airlines l on r.airline_id=l.airline_id group by(r.airline_id) order by high desc limit 1;

```

2024-11-21 10:02:20,656 Stage-2 map = 0%, reduce = 0%
2024-11-21 10:02:28,807 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 5.43 sec
2024-11-21 10:02:33,904 Stage-2 map = 100%, reduce = 50%, Cumulative CPU 8.16 sec
2024-11-21 10:02:34,922 Stage-2 map = 100%, reduce = 75%, Cumulative CPU 13.6 sec
2024-11-21 10:02:36,957 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 16.32 sec
MapReduce Total cumulative CPU time: 16 seconds 320 msec
Ended Job = job_1732089968849_2616
Launching Job 3 out of 3
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_2616, Tracking URL = http://master:6318/proxy/application_1732089968849_2616/
Kill Command = /opt/hadoop/bin/mapred job -kill job_1732089968849_2616
Hadoop job information for Stage-3: number of mappers: 3; number of reducers: 1
2024-11-21 10:02:48,659 Stage-3 map = 0%, reduce = 0%
2024-11-21 10:02:54,779 Stage-3 map = 33%, reduce = 0%, Cumulative CPU 2.59 sec
2024-11-21 10:02:56,819 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 7.74 sec
2024-11-21 10:03:01,912 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 11.08 sec
MapReduce Total cumulative CPU time: 11 seconds 80 msec
Ended Job = job_1732089968849_2616
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 4 Cumulative CPU: 28.65 sec HDFS Read: 2725054 HDFS Write: 11852 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 4 Cumulative CPU: 16.32 sec HDFS Read: 32890 HDFS Write: 10287 SUCCESS
Stage-Stage-3: Map: 3 Reduce: 1 Cumulative CPU: 11.08 sec HDFS Read: 23978 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 56 seconds 50 msec
OK
2484
Time taken: 83.417 seconds, Fetched: 1 row(s)
hive (cdac_anup)> 
```

Question2:

- 1) create table routes23(airline_iata STRING, airline_id INT, src_airport_iata STRING, 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 by fields terminated by "," stored by textfile;

Pyspark

Question 1:

- 1)
- 2) Distinct year in airlines table

The screenshot shows a terminal window with the URL `cdacnppc.cloudloka.com/shell/`. The terminal output displays several lines of log messages from the Spark UI service, indicating it is attempting to bind to various ports (4045, 4046, 4047, 4048, 4049, 4050, 4051, 4052, 4053, 4054) because port 4045 is already in use. It also mentions that spark.yarn.jars or spark.yarn.archive is not set, so it will fall back to uploading libraries under SPARK_HOME. The message "Welcome to" is followed by the Apache logo and "version 3.1.2".

```

24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4045. Attempting port 4046.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4046. Attempting port 4047.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4047. Attempting port 4048.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4048. Attempting port 4049.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4049. Attempting port 4050.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4050. Attempting port 4051.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4051. Attempting port 4052.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4052. Attempting port 4053.
24/11/21 08:49:59 WARN Utils: Service 'SparkUI' could not bind on port 4053. Attempting port 4054.
24/11/21 08:50:00 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
   __/\__\_\_/\_\_/\_\_/\_\_/\_\_
  / \ / . \ \_, / / \ \ \_ \
 /_/
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:4054
Spark context available as 'sc' (master = yarn, app id = application_1732089968849_2295).
SparkSession available as 'spark'.
>>> airlinedf=spark.read.format("csv").option("header","True").option("inferSchema","True").load("/user/cdacuser83313/training/airlines_seat.csv")
>>> airlineRDD=airlinedf.rdd
>>> airlineRDD.take(2)
[Row(Year=1995, Quarter=1, Avg_rev_per_seat=296.9, booked_seats=46561), Row(Year=1995, Quarter=2, Avg_rev_per_seat=296.8, booked_seats=37443)]
>>> map1=airlineRDD.map(lambda a: (int(a["Year"]))).distinct()
>>> map1.take(5)
[1995, 1996, 1997, 1998, 1999]
>>> map1.take(10)
[1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004]
>>> 
```

Question 2:

1)

1.Min value:

```
countdf=airlinedf.agg(min("Avg_rev_per_seat").alias("min_value")).orderBy("min_value", ascending=True).limit(1)
```

2.Maxvalue

```
countdf=airlinedf.agg(max("Avg_rev_per_seat").alias("max_value")).orderBy("max_value", ascending=True).limit(1)
```

3.avgvalue

```
countdf=airlinedf.agg(avg("Avg_rev_per_seat").alias("avg_value")).orderBy("avg_value", ascending=True).limit(1)
```

```

Subscription Details | Nu... | Hue - File Browser | cdacuser83313@ip-172... | Upload files - AnupShela... | exam_BigData_Anup_015 |
cdacnppc.cloudloka.com/shell/ | LeetCode | HackerRank | DBMS | Mining of Massive... | Dashboard | Nuvepro | cdac-241024 - Go...
>>> countdf=airlinedf.agg(min("Avg_rev_per_seat").alias("max_value")).orderBy("max_value",ascending=True).limit(1)
>>> countdf.show()
+-----+
|max_value|
+-----+
| 269.49|
+-----+
>>> countdf=airlinedf.agg(min("Avg_rev_per_seat").alias("min_value")).orderBy("min_value",ascending=True).limit(1)
>>> countdf.show()
+-----+
|min_value|
+-----+
| 269.49|
+-----+
>>> countdf=airlinedf.agg(max("Avg_rev_per_seat").alias("max_value")).orderBy("max_value",ascending=True).limit(1)
>>> countdf.show()
+-----+
|max_value|
+-----+
| 396.37|
+-----+
>>> countdf=airlinedf.agg(avg("Avg_rev_per_seat").alias("avg_value")).orderBy("avg_value",ascending=True).limit(1)
>>> countdf.show()
+-----+
| avg_value|
+-----+
|329.7475000000006|
+-----+
>>> 

```

2)

3) `countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats").alias("sum_seat"))`

```

Subscription Details | Nu... | lab - JupyterLab | Hue - File Browser | Upload files - AnupShela... | exam_BigData_Anup_015 |
cdacnppc.cloudloka.com/shell/ | LeetCode | HackerRank | DBMS | Mining of Massive... | Dashboard | Nuvepro | cdac-241024 - Go...
SparkSession available as 'spark'.
>>> airlinedf=spark.read.format("csv").option("header","True").option("inferSchema","True").load("/user/cdacuser83313/training/airlines_seat.csv")
>>> from pyspark.sql.functions import count
>>> from pyspark.sql.functions import sum
>>> countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats")alias(sum))
  File "<stdin>", line 1
    countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats")alias(sum))
                                         ^
SyntaxError: invalid syntax
>>> countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats")alias(sum_seat))
  File "<stdin>", line 1
    countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats")alias(sum_seat))
                                         ^
SyntaxError: invalid syntax
>>> countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats").alias(sum_seat))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sum_seat' is not defined
>>> countdf= airlinedf.groupBy("Quarter").agg(sum("booked_seats").alias("sum_seat"))
>>> countdf.take(4)
[Row(Quarter=1, sum_seat=873761), Row(Quarter=3, sum_seat=827111), Row(Quarter=4, sum_seat=821351), Row(Quarter=2, sum_seat=807596)]
>>> countdf.show()
+-----+
|Quarter|sum_seat|
+-----+
| 1| 873761|
| 3| 827111|
| 4| 821351|
| 2| 807596|
+-----+
>>> 

```

4) countdf= airlinedf.select("Year").distinct()

```
Subscription Detail x cdacuser83313@ip x lab - JupyterLab x Hue - File Browser x Upload files - Anup x exam_BigData_Am x + 
cdacnlpapc.cloudloka.com/shell/ LeetCode HackerRank DBMS Mining of Massive... Dashboard | Nuvepro cdac-241024 - Go...
+-----+
| 1| 873761|
| 3| 827111|
| 4| 821351|
| 2| 807596|
+-----+
>>> countdf= airlinedf.select("Year").distinct()
  File "<stdin>", line 1
    countdf= airlinedf.select("Year").distinct()
IndentationError: unexpected indent
>>> countdf= airlinedf..distinct("Year")
  File "<stdin>", line 1
    countdf= airlinedf..distinct("Year")
IndentationError: unexpected indent
>>> countdf= airlinedf.distinct("Year")
  File "<stdin>", line 1
    countdf= airlinedf.distinct("Year")
IndentationError: unexpected indent
>>> countdf= airlinedf.groupBy("Year")
>>> countdf.show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'GroupedData' object has no attribute 'show'
>>> airlinrdd=airlinedf.rdd
>>> dis1=airlinerdd.map(lambda a: (int(a["Year"]))).distinct()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'airlinerdd' is not defined
>>> airlinerdd=airlinedf.rdd
>>> dis1=airlinerdd.map(lambda a: (int(a["Year"]))).distinct()
>>> dis1.show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

5)

```
year_rev=airlinedf.groupBy("Year").agg(sum("Avg_rev_per_seat")).
alias("total_rev")).orderBy("total_rev").limit(3)
```

```
Subscription Detail x cdacuser83313@ip x lab - JupyterLab x Hue - File Browser x Upload files - Anup x exam_BigData_Am x + 
cdacnlpapc.cloudloka.com/shell/ LeetCode HackerRank DBMS Mining of Massive... Dashboard | Nuvepro cdac-241024 - Go...
IndentationError: unexpected indent
>>> countdf= airlinedf..distinct("Year")
  File "<stdin>", line 1
    countdf= airlinedf..distinct("Year")
IndentationError: unexpected indent
>>> countdf= airlinedf.distinct("Year")
  File "<stdin>", line 1
    countdf= airlinedf.distinct("Year")
IndentationError: unexpected indent
>>> countdf= airlinedf.groupBy("Year")
>>> countdf.show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'GroupedData' object has no attribute 'show'
>>> airlinrdd=airlinedf.rdd
>>> dis1=airlinerdd.map(lambda a: (int(a["Year"]))).distinct()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'airlinerdd' is not defined
>>> airlinerdd=airlinedf.rdd
>>> dis1=airlinerdd.map(lambda a: (int(a["Year"]))).distinct()
>>> dis1.show()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'PipelinedRDD' object has no attribute 'show'
>>> dis1.take(3)
[1995, 1996, 1997]
>>> dis1.collect()
[1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015]
>>> year_rev=airlinedf.groupBy("Year").agg(sum("Avg_rev_per_seat")).alias("total_rev")).orderBy("total_rev").limit(3)
>>> year_rev.collect()
[Row(Year=1996, total_rev=1107.57), Row(Year=1997, total_rev=1148.62), Row(Year=1995, total_rev=1168.99)]
>>>
```