CS598 PDF Report - Task 1

Task 1 Overview

Table of Contents

1) Overview - Data Extract and Cleaning

I copied the snapshot, renamed as "CS598 CCC - Copied Snapshot - Transportation Databases (Linux)", to retrieve the dataset. I created a new EBS volume from the snapshot, and a new EC2 instance to be attached to the EBS volume. After mounting EBS to your EC2 instance, I copied files from EBS to my local machine.

Once I had the entire airline_ontime directory on my local, I unzipped all the CSV files, expecting and finding 240 files, ignoring 2008_11 and 2008_12 zips. I then moved all CSV to one directory and converted them to utf-8. From here, I used Jupyter for a data cleaning script. This was a two-part optimization because a) smaller file sizes needed to be uploaded to S3, saving on storage costs, and b) before doing this, I ran into issues with many of the columns were null even though they were in the CSVs in S3. It seems that these were float integers, and I had to go back to convert those columns.

After the script completed, I compressed each cleaned csv into their .gz, and upserted all the .gz to S3. From there, I used AWS Glue to clean, organize, and view the data by creating and running a crawler in AWS Glue. After the crawler completed, I cleaned and updated the table schema to confirm the wanted columns and data types. Using AWS Athena, I validated my cleaned data in S3. I expected and found 116,753,952 rows and also validated the data by querying a few rows and seeing that the stored data looked correct.

2) Overview - System Integration

Used AWS EMR with Hive and DynamoDB.

I launched an EMR cluster with the default, 3 nodes (2 master and 1 slave). My SSH first timed out when I tried SSHing directly into the master node after it was up and running. I resolved this by adding SSH to port 22 in the AWS Security Group.

From there I set up a Hive external table using S3 as the location from which I would import the airline data into DynamoDB. S3 initially imported 116,754,192 records, vs 116,753,952 rows found from Athena before. This is a 240 row difference, for the CSV headers. I then created the DynamoDB tables for Group 2 and Group 3.2. I did spent some time tinkering with the partition and sort keys and learning how they worked for Dy.namoDB, but eventually went back to approaching the problems and optimizing my queries

3) Approaches and Algorithms

I started by testing out my queries in Athena and validating I got the expected solution there. From there, I could copy and paste those queries into the HIVE CLI to run on the EMR Hive cluster when they were ready. From there, it was a simple matter of adopting the query to create an external table and insert the data into DynamoDB. For each question, I created separate DynamoDB tables and ran the HIVE queries to insert data in their respective tables.

For Question 3.1, I ran the Hive query needed for the data points, then added those results to a Hive table which was subsequent exported to a file in S3. I then converted that file from S3 to a CSV file. From there, I created the distribution graphs needed to answer 3.1 For 3.2, it took some time to figure out how I wanted to do this. I first approached this from the perspective where I would have just one table and do one massive query with JOIN/UNION/WHERE/GROUPBY/etc. However, that would be time and performance intensive. Thus, I decided to break the problem up by using three tables: one for the first leg, one for the second leg, and one for the complete flight that selected the necessary fields from the first two tables.

4) Results

Queries and results are included here. Please be aware this section is quite lengthy, about 22 pages, since it contains all the queries and image results of all the questions' queries.

Skip to page 24 if you'd like to go onto the Optimizations section.

Group 1

*Includes HiveQL Query and screenshot of the result

1.1) Rank the top 10 most popular airports by numbers of flights to/from the airport.

- SELECT o.origin as airport, o.flight_nr + d.flight_nr as total from (SELECT origin, count(origin) as flight_nr from airline_ontime_cleaned group by origin) as o, (SELECT dest, count(dest) as flight_nr from airline_ontime_cleaned group by dest) as d where o.origin = d.dest order by total DESC LIMIT 10;

```
hive> select o.origin as airport, o.flight_nr + d.flight_nr as total from (select origin, count(origin) as flight_nr from airline_ontime_cleaned group by origin) as o, (sel
ect dest, count(dest) as flight_nr from airline_ontime_cleaned group by dest) as d where o.origin = d.dest order by total desc limit 10;
Query ID = hadoop_20200626185515_dc371850-fc0b-454c-a16c-b604d71ab95f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0013)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Map 4 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 5 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 85.76 s
--------------------------------------------------------------------------------
OK
ORD     12449354
ATL     11540422
DFW     10799303
LAX     7723596
PHX     6585534
DEN     6273787
DTW     5636622
IAH     5480734
MSP     5199213
SFO     5171023
Time taken: 89.226 seconds, Fetched: 10 row(s)
```

1.2) Rank the top 10 airlines by on-time arrival performance.
Airline: Average delay in minutes

- SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC LIMIT 10;

```
hive> SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626163253_ae702d76-15df-4aac-b0bf-f4676f41ee0c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 55.38 s
--------------------------------------------------------------------------------
OK
HA      -1.01180434574519
AQ      1.1569234424812056
PS      1.4506385127822803
ML (1)  4.747609195734892
PA (1)  5.3224309999287875
F9      5.465881148819851
NW      5.557783392671835
WN      5.5607742598815735
OO      5.736312463662878
9E      5.8671846616957595
Time taken: 57.355 seconds, Fetched: 10 row(s)
hive>
```

1.3) Weekday: Average delay in minutes

- SELECT dayofweek, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned GROUP BY dayofweek ORDER BY averagedeparturedelay ASC;

```
hive> SELECT dayofweek, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned GROUP BY dayofweek ORDER BY averagedeparturedelay ASC;
Query ID = hadoop_20200626163529_4729e09d-e649-4ee4-98eb-0c82be64ac34
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 53.60 s
--------------------------------------------------------------------------------
OK
6       4.301669926076596
2       5.990458841319885
7       6.613280292442754
1       6.716102802585582
3       7.203656394670348
4       9.094441008336657
5       9.721032337585571
Time taken: 54.094 seconds, Fetched: 7 row(s)
hive>
```

For Group 2 + 3.2 Results Below, I have “overall” queries along with Hive queries to save the results into DynamoDB, and screenshots of individual queries and their results required for submission.

Group 2

2.1) For each airport X, rank the top-10 carriers in decreasing order of on-time departure performance from X.

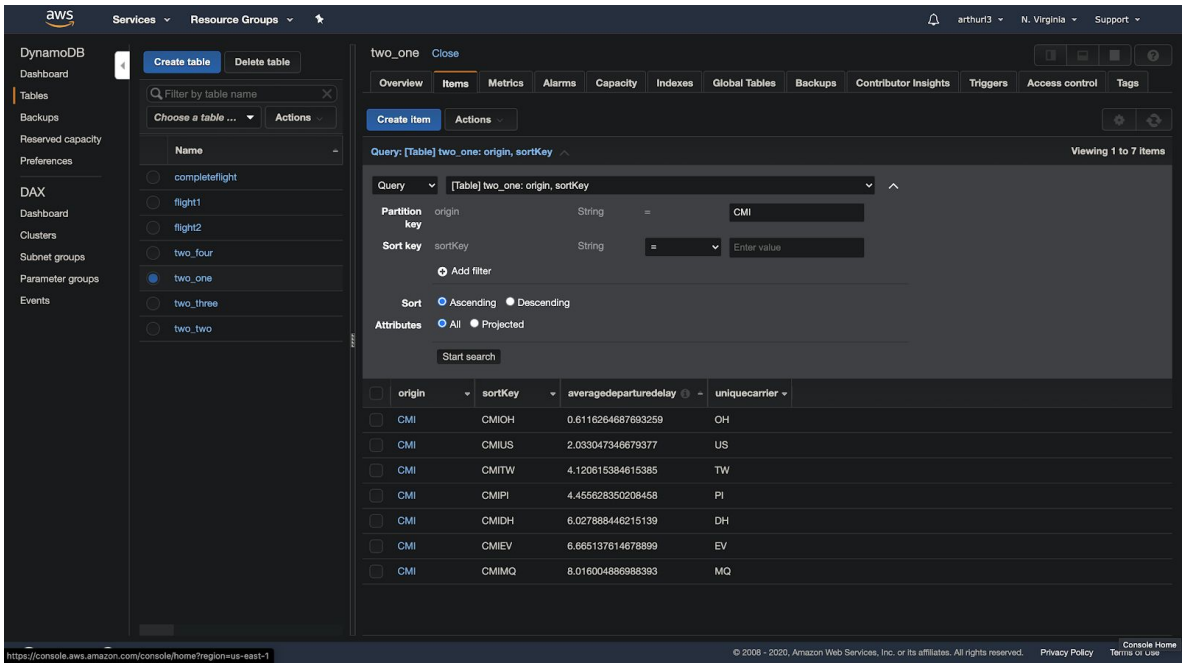
General Hive Query + Adding Data to DDB (Query Result Shown in Video)

- SELECT origin, uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned GROUP BY origin, uniquecarrier ORDER BY origin, averagedeparturedelay;
- CREATE EXTERNAL TABLE two_one(origin STRING, uniquecarrier STRING, averagedeparturedelay DOUBLE, sortKey STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "two_one", "dynamodb.column.mapping" = "origin:origin,uniquecarrier:uniquecarrier,averagedeparturedelay:averagedeparturedelay,sortKey:sortKey");
- INSERT OVERWRITE TABLE two_one SELECT origin, uniquecarrier, AVG(depdelay) as averagedeparturedelay, concat(origin, uniquecarrier) as sortKey from airline_ontime_cleaned GROUP BY origin, uniquecarrier ORDER BY origin, averagedeparturedelay;

Hive Query to Import from S3 to DynamoDB

```
hive> INSERT OVERWRITE TABLE two_one SELECT origin, uniquecarrier, AVG(depdelay) as averagedeparturedelay, concat(origin, uniquecarrier) as sortKey from airline_ont
ime_cleaned GROUP BY origin, uniquecarrier ORDER BY origin, averagedeparturedelay;
Query ID = hadoop_20200627021138_e281cacd-fb04-41e8-b86a-dc273489c845
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1593216829342_0018)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     19         19        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 3012.98 s
--------------------------------------------------------------------------------
OK
Time taken: 3018.606 seconds
```

Sample DynamoDB Query - CMI



Targeted Queries (HIVE)

- CMI (University of Illinois Willard Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'CMI' GROUP BY uniquecarrier ORDER BY averagedepartureddelay ASC LIMIT 10;

```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'CMI' GROUP BY uniquecarrier ORDER BY averagedepartureddelay AS
C LIMIT 10;
Query ID = hadoop_20200626163635_ad876533-fb5f-4607-bde6-56564bf3f092
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 46.36 s
--------------------------------------------------------------------------------
OK
OH      0.6116264687693259
US      2.033047346679377
TW      4.120615384615385
PI      4.455628350208458
DH      6.027888446215139
EV      6.665137614678899
MQ      8.016004886988393
Time taken: 46.928 seconds, Fetched: 7 row(s)
hive>
```

- BWI (Baltimore-Washington International Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'BWI' GROUP BY uniquecarrier ORDER BY averagedepartureddelay ASC LIMIT 10;

```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'BWI' GROUP BY uniquecarrier ORDER BY averagedeparturedelay AS
C LIMIT 10;
Query ID = hadoop_20200626163751_9aac506e-319b-4356-aa77-49f15111c9fd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 44.89 s
--------------------------------------------------------------------------------
OK
F9      0.7562437562437563
PA (1)  4.761904761904762
CO      5.179340976854271
YV      5.496503496503497
NW      5.705573031597727
AA      6.002851840115884
9E      7.239805825242718
US      7.494395794023255
DL      7.676822368501101
UA      7.737921397819683
Time taken: 45.353 seconds, Fetched: 10 row(s)
```

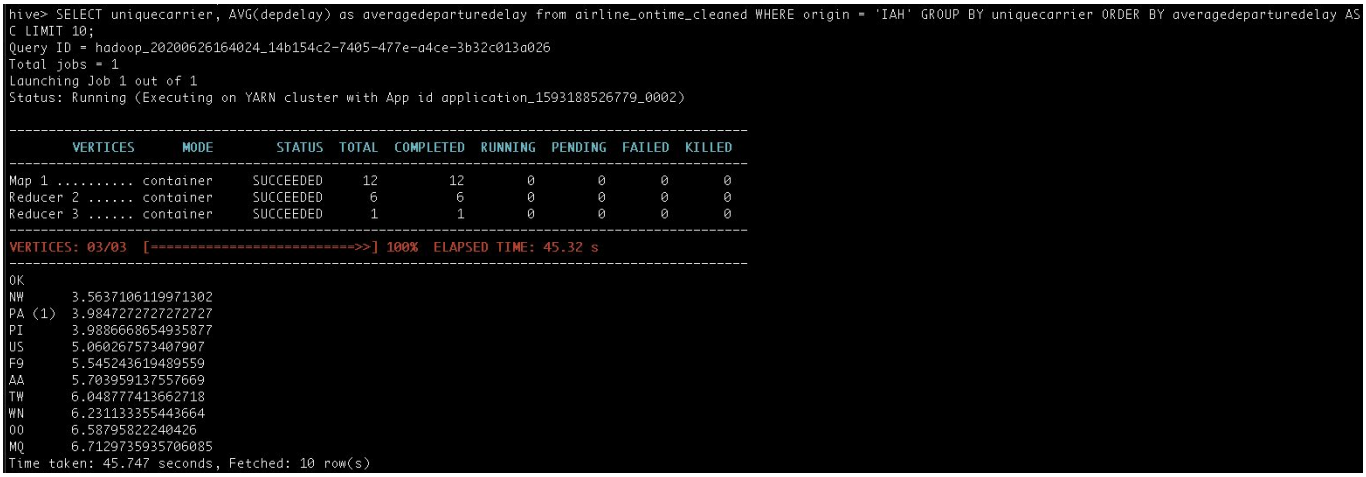
- MIA (Miami International Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'MIA' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC LIMIT 10;

```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'MIA' GROUP BY uniquecarrier ORDER BY averagedeparturedelay AS
C LIMIT 10;
Query ID = hadoop_20200626163843_c292139a-12d8-482f-a9b8-5badae499d15
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 36.41 s
--------------------------------------------------------------------------------
OK
9E      -3.0
EV      1.2026431718061674
TZ      1.782243551289742
XE      1.8731909028256375
PA (1)  4.20000428045544
NW      4.501665523660233
US      6.090665809518826
UA      6.869731753577851
ML (1)  7.504550050556118
FL      8.565107458912768
Time taken: 36.863 seconds, Fetched: 10 row(s)
```

- LAX (Los Angeles International Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC LIMIT 10;

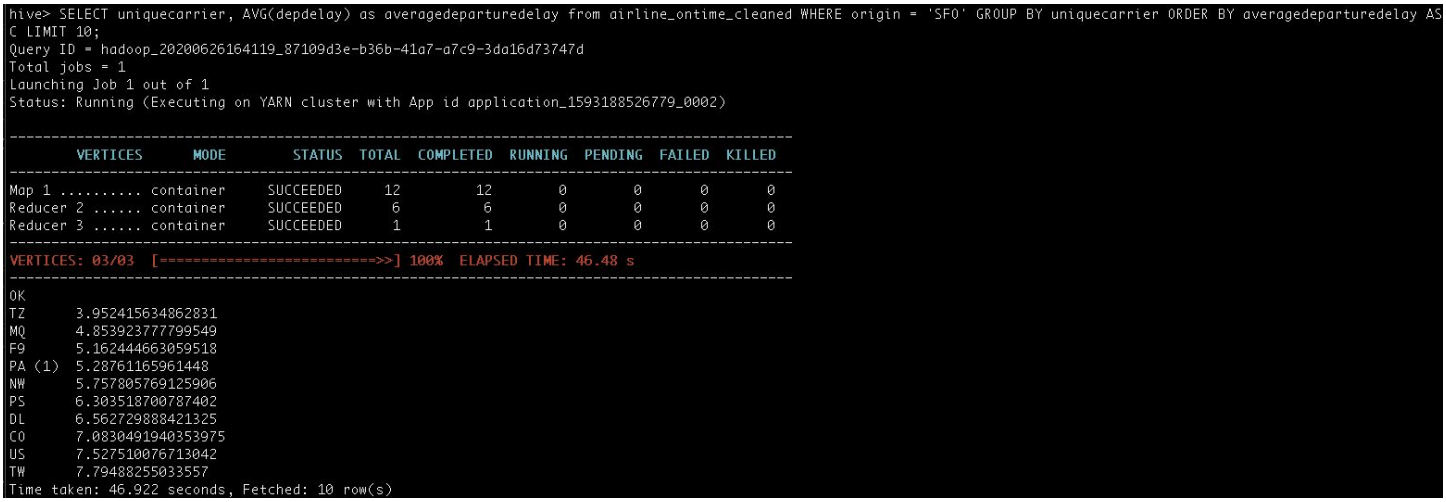
```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' GROUP BY uniquecarrier ORDER BY averagedeparturedelay AS
C LIMIT 10;
Query ID = hadoop_20200626163927_bd64dad7-0146-41ee-a95a-c4b91fef7843
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 37.25 s
--------------------------------------------------------------------------------
OK
MQ      2.407221858260434
OO      4.2219592877139975
FL      4.725127379994636
TZ      4.763940985246312
PS      4.860337041524397
NW      5.11955065127997
F9      5.729155372438469
HA      5.813645621181263
YV      6.024156085475379
US      6.746395368371022
Time taken: 37.712 seconds, Fetched: 10 row(s)
```

- IAH (George Bush Intercontinental Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'IAH' GROUP BY uniquecarrier ORDER BY averagedepartureddelay ASC LIMIT 10;



```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'IAH' GROUP BY uniquecarrier ORDER BY averagedepartureddelay AS
C LIMIT 10;
Query ID = hadoop_20200626164024_14b154c2-7405-477e-a4ce-3b32c013a026
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 45.32 s
--------------------------------------------------------------------------------
OK
NW      3.5637106119971302
PA (1)  3.9847272727272727
PI      3.9886668654935877
US      5.060267573407907
F9      5.545243619489559
AA      5.703959137557669
TW      6.048777413662718
WN      6.231133355443664
00      6.58795822240426
MQ      6.7129735935706085
Time taken: 45.747 seconds, Fetched: 10 row(s)
```

- SFO (San Francisco International Airport)
    - SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'SFO' GROUP BY uniquecarrier ORDER BY averagedepartureddelay ASC LIMIT 10;



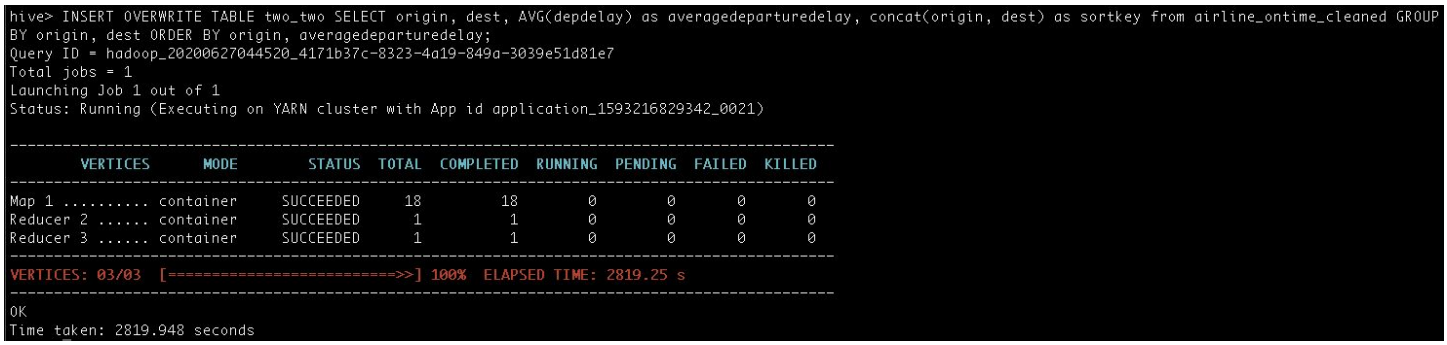
```
hive> SELECT uniquecarrier, AVG(depdelay) as averagedepartureddelay from airline_ontime_cleaned WHERE origin = 'SFO' GROUP BY uniquecarrier ORDER BY averagedepartureddelay AS
C LIMIT 10;
Query ID = hadoop_20200626164119_87109d3e-b36b-41a7-a7c9-3da16d73747d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 46.48 s
--------------------------------------------------------------------------------
OK
TZ      3.952415634862831
MQ      4.853923777799549
F9      5.162444663059518
PA (1)  5.28761165961448
NW      5.757805769125906
PS      6.303518700787402
DL      6.562729888421325
CO      7.0830491940353975
US      7.527510076713042
TW      7.79488255033557
Time taken: 46.922 seconds, Fetched: 10 row(s)
```

2.2) For each source airport X, rank the top-10 destination airports in decreasing order of on-time departure performance from X.
General Hive Query + Adding Data to DDB

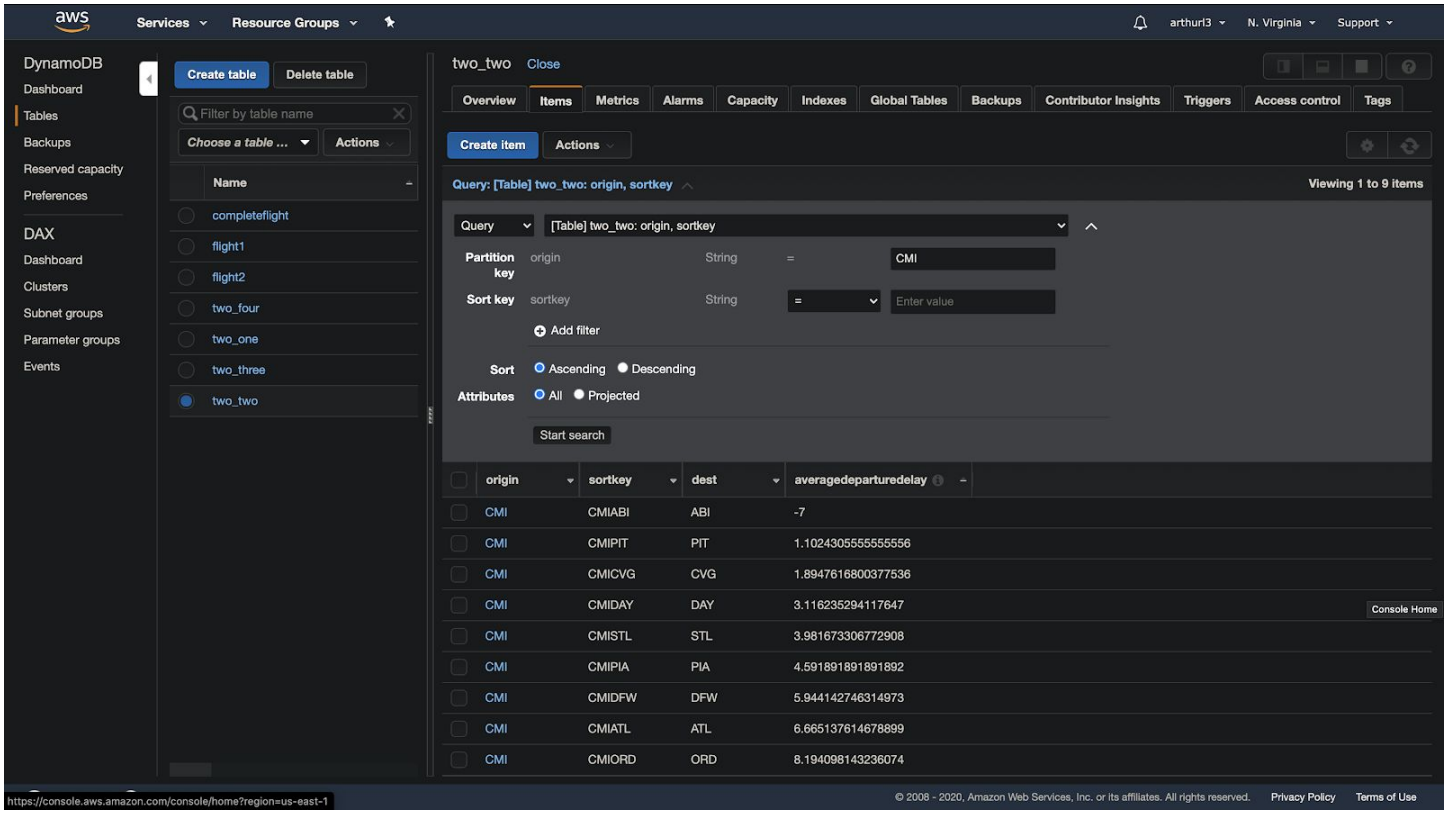
- SELECT origin, dest, AVG(depdelay) as averagedepartureddelay, concat(origin, dest) as sortkey from airline_ontime_cleaned GROUP BY origin, dest ORDER BY origin, averagedepartureddelay;
- CREATE EXTERNAL TABLE two_two(origin STRING, dest STRING, averagedepartureddelay DOUBLE, sortkey STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "two_two", "dynamodb.column.mapping" = "origin:origin,dest:dest,averagedepartureddelay:averagedepartureddelay,sortkey:sortkey");

- INSERT OVERWRITE TABLE two_two SELECT origin, dest, AVG(depdelay) as averagedeparturedelay, concat(origin, dest) as sortkey from airline_ontime_cleaned GROUP BY origin, dest ORDER BY origin, averagedeparturedelay;

Hive Query to Import from S3 to DynamoDB



Sample DynamoDB Query - CMI



Targeted Queries (HIVE)

- CMI (University of Illinois Willard Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'CMI' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;

```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'CMI' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164355_6bc68032-bb52-44cc-8771-c262ff5d6796
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 45.34 s
--------------------------------------------------------------------------------
OK
ABI     -7.0
PIT     1.1024305555555556
CVG     1.8947616800377536
DAY     3.116235294117647
STL     3.981673306772908
PIA     4.591891891891892
DFW     5.944142746314973
ATL     6.665137614678899
ORD     8.194098143236074
Time taken: 45.757 seconds, Fetched: 9 row(s)
```

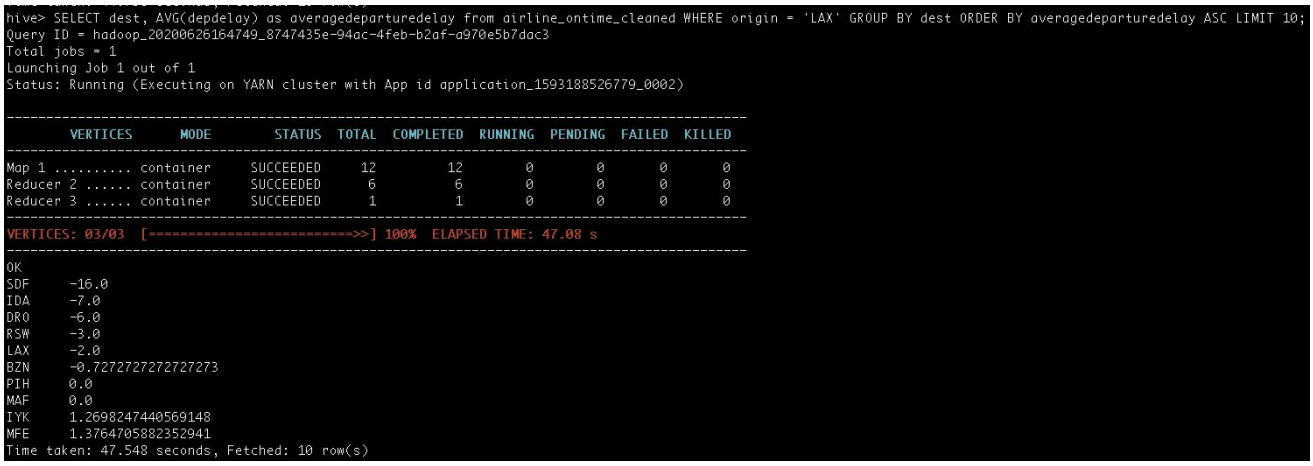
- BWI (Baltimore-Washington International Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'BWI' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;

```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'BWI' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164458_430b2176-37cc-44e5-bf96-59d45d9dada7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 46.36 s
--------------------------------------------------------------------------------
OK
SAV     -7.0
MLB     1.155367231638418
DAB     1.4695945945945945
SRQ     1.5884838880084522
IAD     1.7909407665505226
UCA     3.6541698546289214
CHO     3.744927536231884
GSP     4.197686645636172
SJU     4.44465842286641
OAJ     4.471111111111111
Time taken: 46.763 seconds, Fetched: 10 row(s)
```

- MIA (Miami International Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'MIA' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;

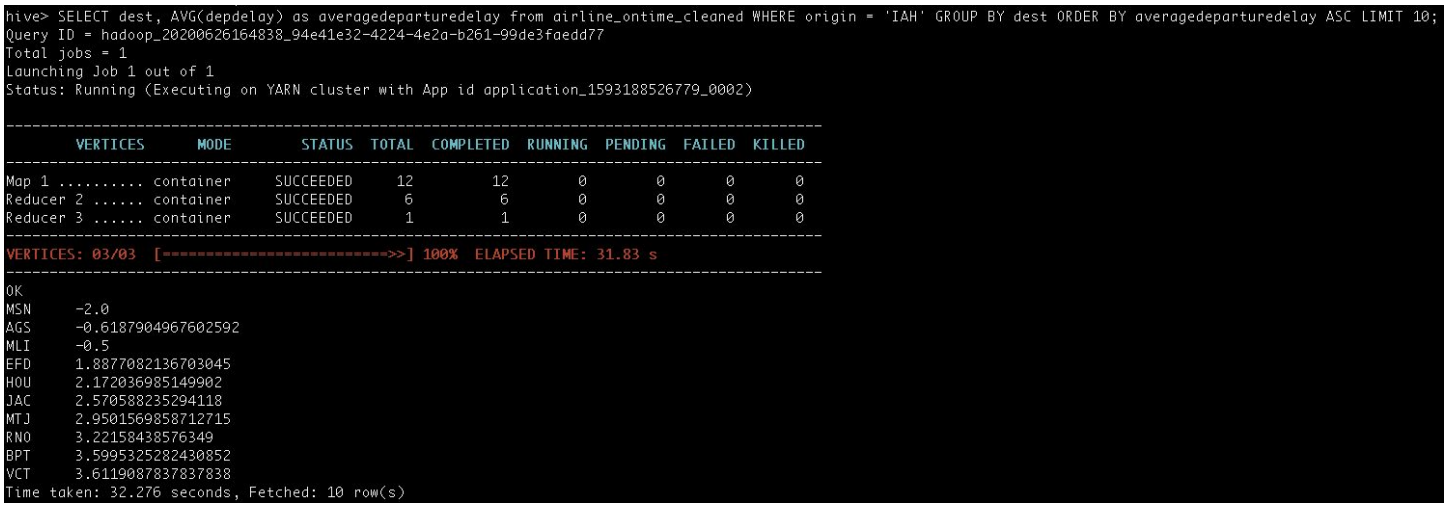
```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'MIA' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164557_4496b50e-f510-4d72-bfb6-d410bc6ef594
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 44.36 s
--------------------------------------------------------------------------------
OK
SHV     0.0
BUF     1.0
SAN     1.710382513661202
SLC     2.5371900826446283
HOU     2.912199124726477
ISP     3.647398843930636
MEM     3.7451066224751424
PSE     3.975845410628019
TLH     4.2614844746916205
MCI     4.612244897959184
Time taken: 44.798 seconds, Fetched: 10 row(s)
```

- LAX (Los Angeles International Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;



```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164749_8747435e-94ac-4feb-b2af-a970e5b7dac3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 47.08 s
--------------------------------------------------------------------------------
OK
SDF     -16.0
IDA     -7.0
DRO     -6.0
RSW     -3.0
LAX     -2.0
BZN     -0.7272727272727273
PIH     0.0
MAF     0.0
IYK     1.2698247440569148
MFE     1.3764705882352941
Time taken: 47.548 seconds, Fetched: 10 row(s)
```

- IAH (George Bush Intercontinental Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'IAH' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;



```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'IAH' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164838_94e41e32-4224-4e2a-b261-99de3faedd77
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 31.83 s
--------------------------------------------------------------------------------
OK
MSN     -2.0
AGS     -0.6187904967602592
MLI     -0.5
EFD     1.8877082136703045
HOU     2.172036985149902
JAC     2.570588235294118
MTJ     2.9501569858712715
RNO     3.22158438576349
BPT     3.5995325282430852
VCT     3.6119087837837838
Time taken: 32.276 seconds, Fetched: 10 row(s)
```

- SFO (San Francisco International Airport)
    - SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'SFO' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;

```
hive> SELECT dest, AVG(depdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'SFO' GROUP BY dest ORDER BY averagedeparturedelay ASC LIMIT 10;
Query ID = hadoop_20200626164927_36f25a0d-f6b2-4bfd-8e8d-817db4908d1a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 47.21 s
--------------------------------------------------------------------------------
OK
SDF     -10.0
MSO     -4.0
PIH     -3.0
LGA     -1.7575757575757576
PIE     -1.3410404624277457
OAK     -0.813200498132005
FAR     0.0
BNA     2.425966447848286
MEM     3.302482299752623
SCK     4.0
Time taken: 47.624 seconds, Fetched: 10 row(s)
```

2.3) For each source-destination pair X-Y, rank the top-10 carriers in decreasing order of on-time arrival performance at Y from X

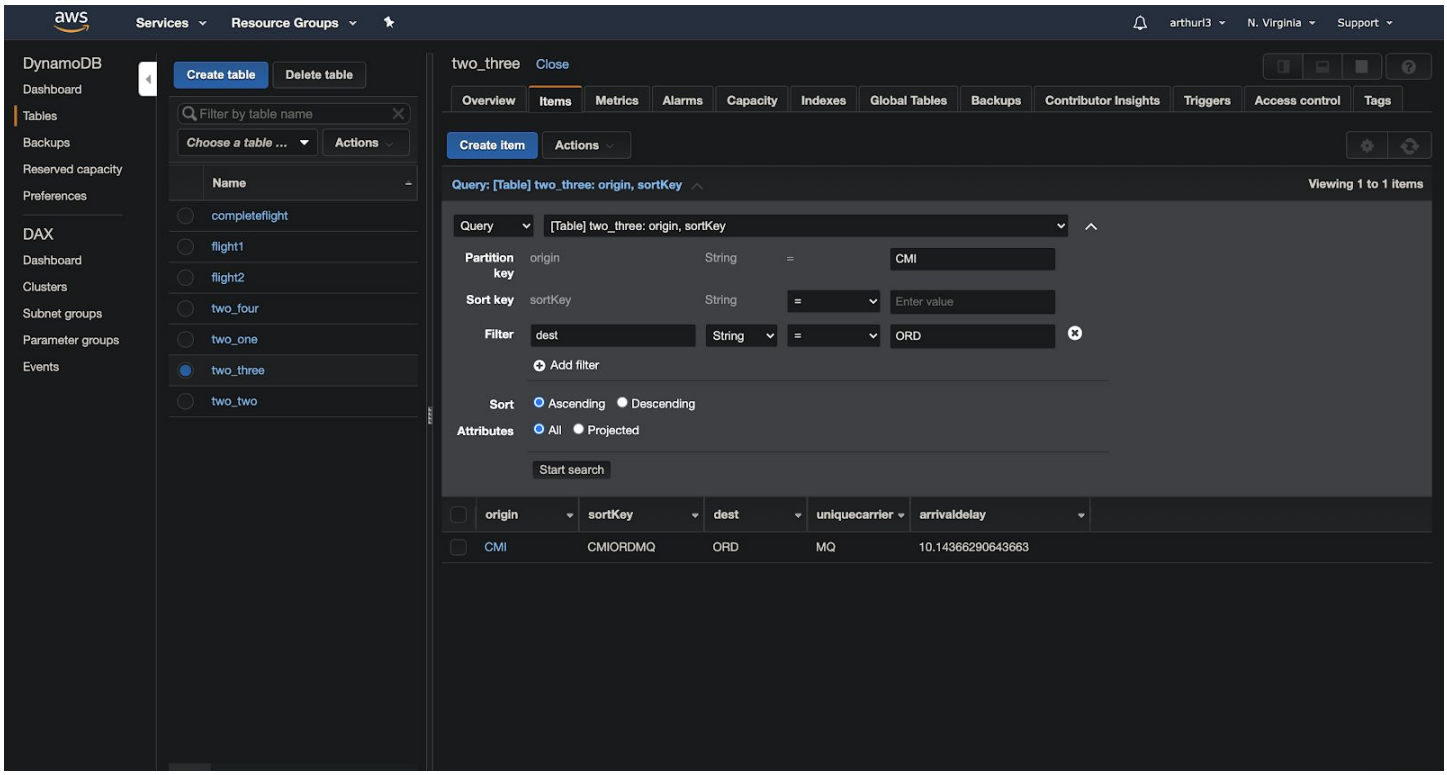
General Hive Query + Adding Data to DDB

- SELECT origin, dest, uniquecarrier, AVG(arrdelay) as arrivaldelay from airline_ontime_cleaned GROUP BY origin, dest, uniquecarrier ORDER BY origin, dest, arrivaldelay;
- CREATE EXTERNAL TABLE two_three(origin STRING, dest STRING, uniquecarrier STRING, arrivaldelay DOUBLE, sortKey STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "two_three", "dynamodb.column.mapping" = "origin:origin,dest:dest,uniquecarrier:uniquecarrier,arrivaldelay:arrivaldelay,sortKey:sortKey");
- INSERT OVERWRITE TABLE two_three SELECT origin, dest, uniquecarrier, AVG(arrdelay) as arrivaldelay, concat(origin, dest, uniquecarrier) as sortKey FROM airline_ontime_cleaned GROUP BY origin, dest, uniquecarrier ORDER BY origin,dest,arrivaldelay;

Hive Query to Import from S3 to DynamoDB

```
hive> INSERT OVERWRITE TABLE two_three SELECT origin, dest, uniquecarrier, AVG(arrdelay) as arrivaldelay, concat(origin, dest, uniquecarrier) as sortKey FROM airlin
e_ontime_cleaned GROUP BY origin, dest, uniquecarrier ORDER BY origin,dest,arrivaldelay;
Query ID = hadoop_20200627013349_b7561b52-507d-4d50-a96d-bdde02e66765
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593216829342_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     25         25        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 6922.11 s
--------------------------------------------------------------------------------
OK
Time taken: 6922.693 seconds
```

Sample DynamoDB Query - CMI



Targeted Queries (HIVE)

- CMI → ORD
    - SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'CMI' AND dest = 'ORD' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;

```
hive> SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'CMI' AND dest = 'ORD' GROUP BY uniquecarrier ORDER BY average
departuredelay ASC limit 10;
Query ID = hadoop_20200626165850_9d175fb4-6fa8-4d13-a73c-40688b86c6e1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 44.54 s
----------------------------------------------------------------------------------------------
OK
MQ      10.14366290643663
Time taken: 44.963 seconds, Fetched: 1 row(s)
```

- IND → CMH
    - SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'IND' AND dest = 'CMH' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;

```
hive> SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'IND' AND dest = 'CMH' GROUP BY uniquecarrier ORDER BY average
departuredelay ASC limit 10;
Query ID = hadoop_20200626165942_d9fa04bb-0f5b-4589-a735-145e1d7fe2b2
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 41.69 s
--------------------------------------------------------------------------------
OK
CO      -2.54585456229736
AA      5.5
HP      5.697254901960784
NW      5.761538461538461
US      6.878469415251954
DL      10.6875
EA      10.813084112149532
Time taken: 42.08 seconds, Fetched: 7 row(s)
```

- DFW → IAH
    - SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'DFW' AND dest = 'IAH' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;

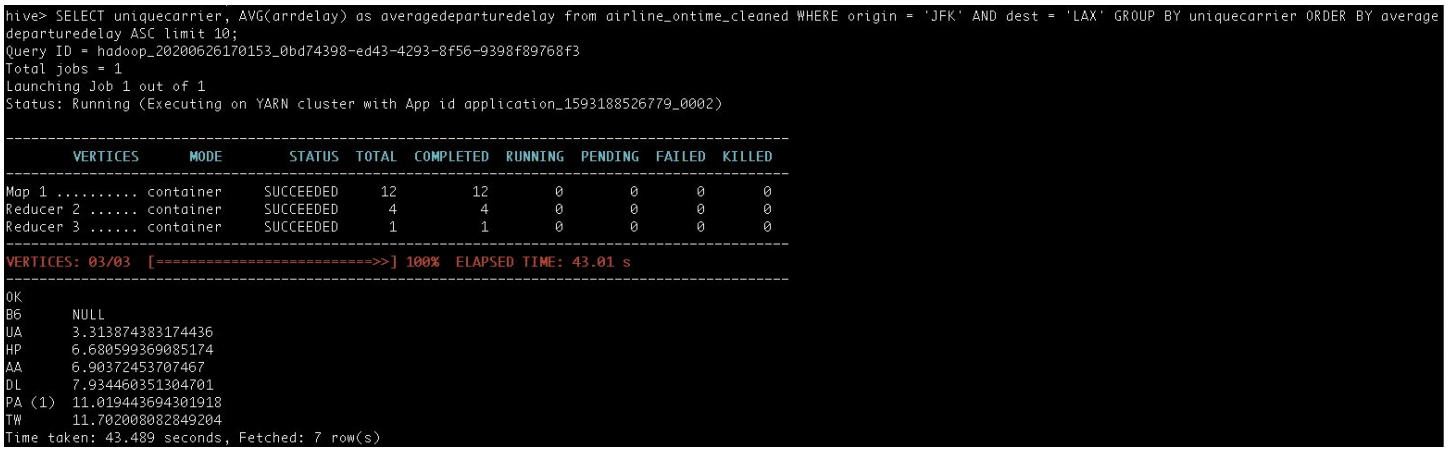
```
hive> SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'DFW' AND dest = 'IAH' GROUP BY uniquecarrier ORDER BY average
departuredelay ASC limit 10;
Query ID = hadoop_20200626170026_8c38a427-7ed1-4c3f-8a6f-274aed5d5045
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 30.72 s
--------------------------------------------------------------------------------
OK
PA (1)  -1.5964912280701755
EV      5.0925133689839575
UA      5.414201183431953
CO      6.493731644930054
OO      7.56400742115027 8
XE      8.094294547498595
AA      8.38122832433381 7
DL      8.59850905218317 3
MQ      9.103211009174313
Time taken: 31.116 seconds, Fetched: 9 row(s)
```

- LAX → SFO
    - SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' AND dest = 'SFO' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;

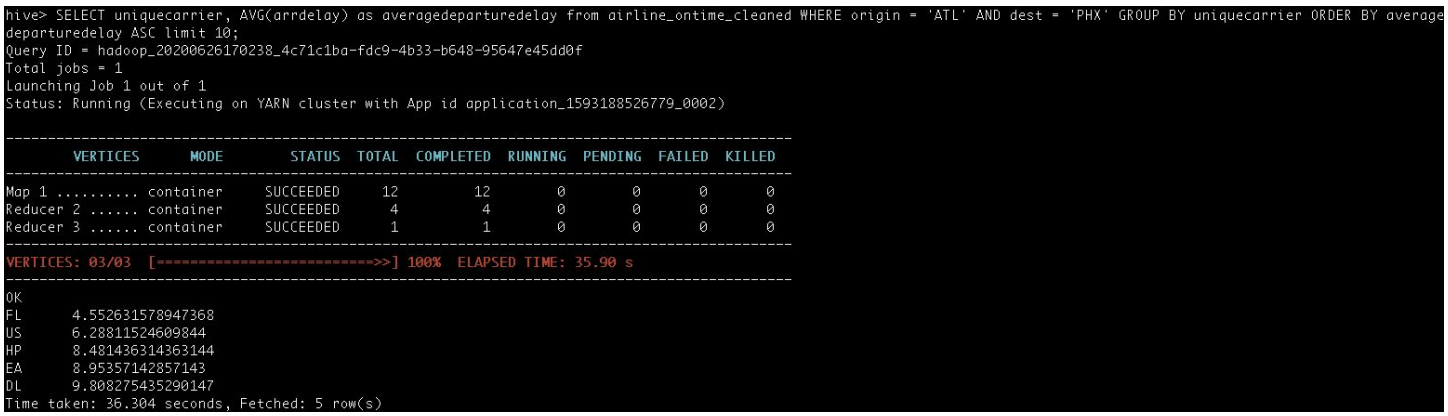
```
hive> SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'LAX' AND dest = 'SFO' GROUP BY uniquecarrier ORDER BY average
departuredelay ASC limit 10;
Query ID = hadoop_20200626170103_b80805d9-6760-4eca-a477-c7592e252257
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 38.10 s
--------------------------------------------------------------------------------
OK
TZ      -7.619047619047619
PS      -2.1463414634146343
F9      -2.028685790527018
EV      6.964630225080386
AA      7.386793490213328
MQ      7.8077634011090575
US      7.964721980345814
WN      8.79205149734117
CO      9.354782608695652
NW      9.84878587196468
Time taken: 38.485 seconds, Fetched: 10 row(s)
```

- JFK → LAX

- SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'JFK' AND dest = 'LAX' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;



- ATL → PHX
  - SELECT uniquecarrier, AVG(arrdelay) as averagedeparturedelay from airline_ontime_cleaned WHERE origin = 'ATL' AND dest = 'PHX' GROUP BY uniquecarrier ORDER BY averagedeparturedelay ASC limit 10;



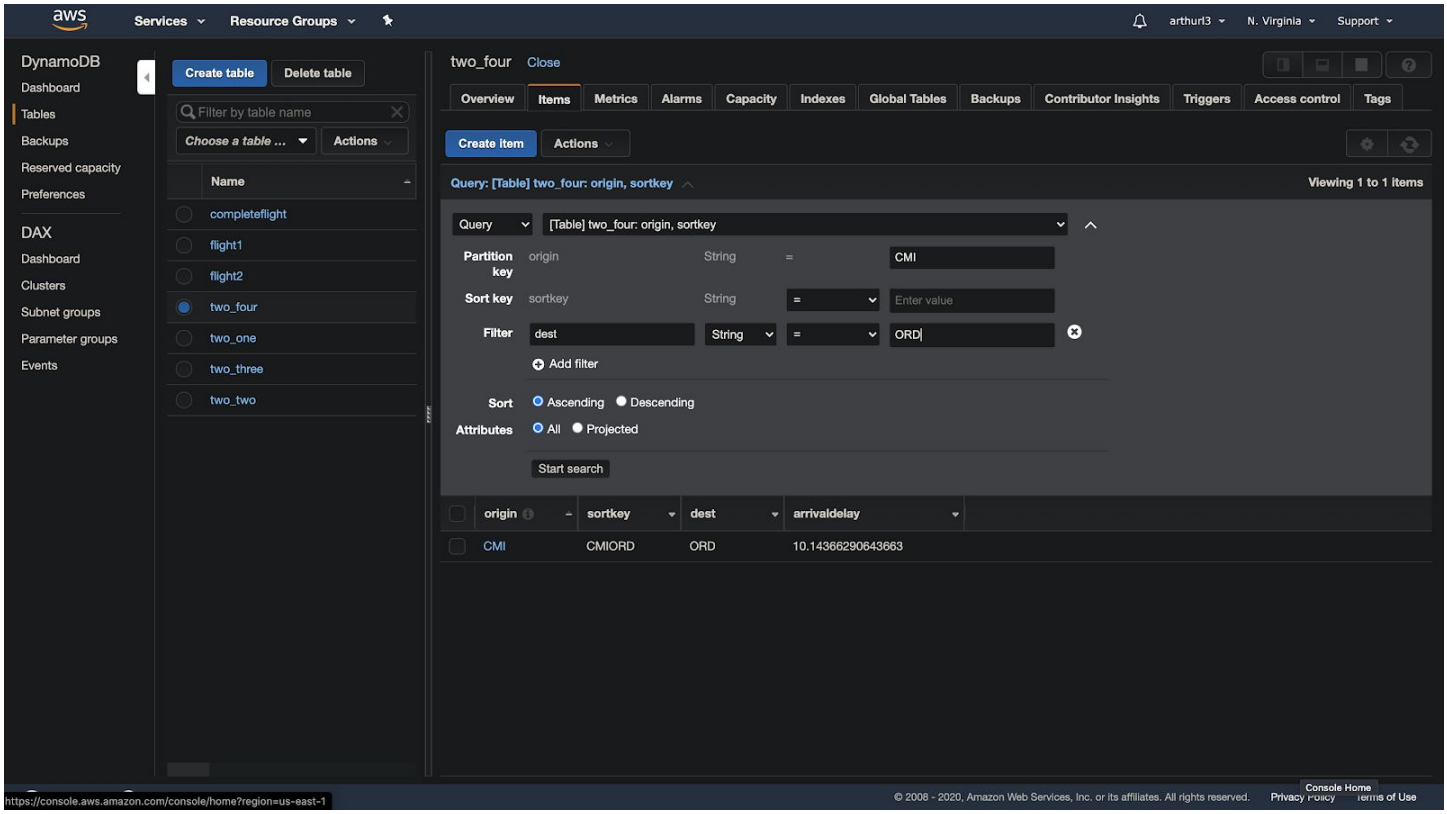
2.4) For each source-destination pair X-Y, determine the mean arrival delay (in minutes) for a flight from X to Y.
General Hive Query + Adding Data to DDB

- SELECT origin, dest, AVG(arrdelay) as arrivaldelay from airline_ontime_cleaned GROUP BY origin, dest ORDER BY origin, dest;
- CREATE EXTERNAL TABLE two_four(origin STRING, dest STRING, arrivaldelay DOUBLE, sortkey STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "two_four", "dynamodb.column.mapping" = "origin:origin,dest:dest,arrivaldelay:arrivaldelay,sortkey:sortkey");
- INSERT OVERWRITE TABLE two_four SELECT origin, dest, AVG(arrdelay) as arrivaldelay, concat(origin, dest) as sortkey from airline_ontime_cleaned GROUP BY origin, dest ORDER BY origin, dest;

Hive Query to Import from S3 to DynamoDB

```
hive> INSERT OVERWRITE TABLE two_four SELECT origin, dest, AVG(arrdelay) as arrivaldelay,  concat(origin, dest) as sortkey from airline_ontime_cleaned GROUP BY orig
in, dest ORDER BY origin, dest;
Query ID = hadoop_20200627044817_0b5aefea-8351-4ff9-8c72-51b5dc787770
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593216829342_0022)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 2550.01 s
--------------------------------------------------------------------------------
OK
Time taken: 2552.656 seconds
```

Sample DynamoDB Query - CMI



Targeted Queries (HIVE)

- CMI → ORD: 10.14
    - SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'CMI' AND dest = 'ORD';

```
hive> SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'CMI' AND dest = 'ORD';
Query ID = hadoop_20200626170454_811f4a8a-cb9e-4424-98a8-25e50236603e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 44.43 s
--------------------------------------------------------------------------------
OK
10.14366290643663
Time taken: 44.841 seconds, Fetched: 1 row(s)
```

- IND → CMH: 2.90
    - SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'IND' AND dest = 'CMH';

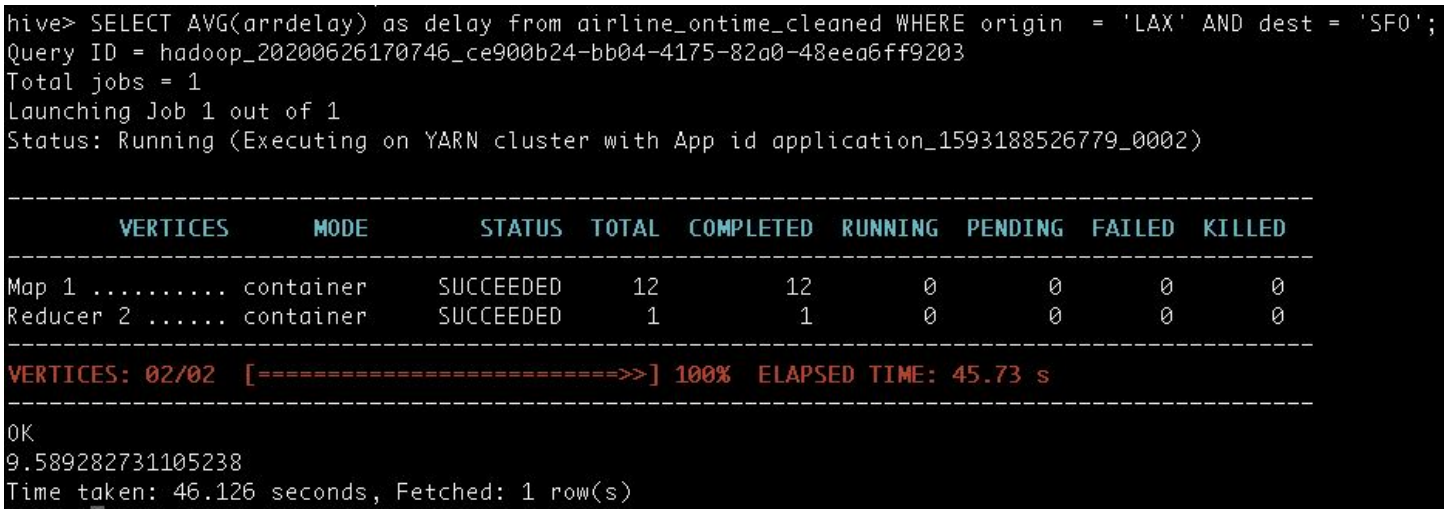
```
hive> SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'IND' AND dest = 'CMH';
Query ID = hadoop_20200626170611_82ebcfc6-795a-49e9-a724-2abaa9ce0b86
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 44.33 s
--------------------------------------------------------------------------------
OK
2.899990366088632
Time taken: 44.777 seconds, Fetched: 1 row(s)
```

- DFW → IAH: 7.65
    - SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'DFW' AND dest = 'IAH';

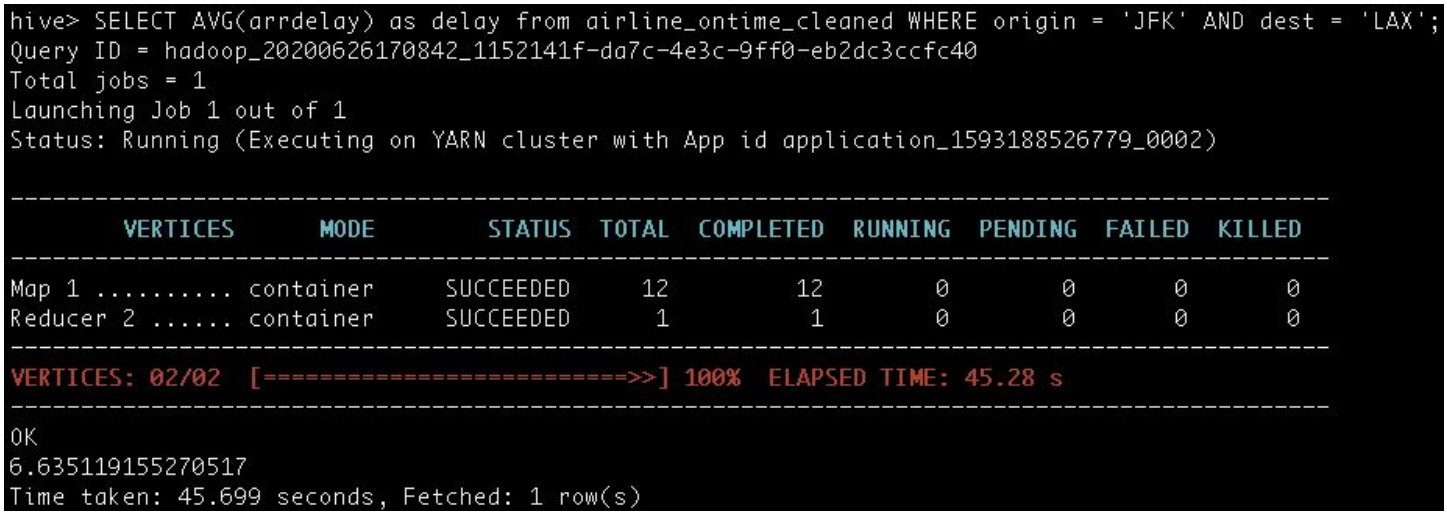
```
hive> SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'DFW' AND dest = 'IAH';
Query ID = hadoop_20200626170701_4d5a93fd-f515-44bd-b972-a398393c340c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 40.06 s
--------------------------------------------------------------------------------
OK
7.654442525768608
Time taken: 40.442 seconds, Fetched: 1 row(s)
```

- LAX → SFO: 9.59

- SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'LAX' AND dest = 'SFO';



- JFK → LAX: 6.64
  - SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'JFK' AND dest = 'LAX';



- ATL → PHX: 9.02
  - SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'ATL' AND dest = 'PHX';

```
hive> SELECT AVG(arrdelay) as delay from airline_ontime_cleaned WHERE origin = 'ATL' AND dest = 'PHX';
Query ID = hadoop_20200626170929_bb368998-cd22-4545-8c25-cfecad9a7d58
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1593188526779_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 35.24 s
----------------------------------------------------------------------------------------------
OK
9.021341881513989
Time taken: 35.616 seconds, Fetched: 1 row(s)
```

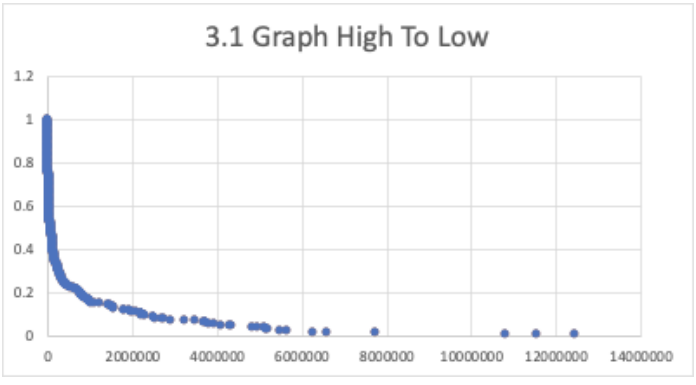
Group 3

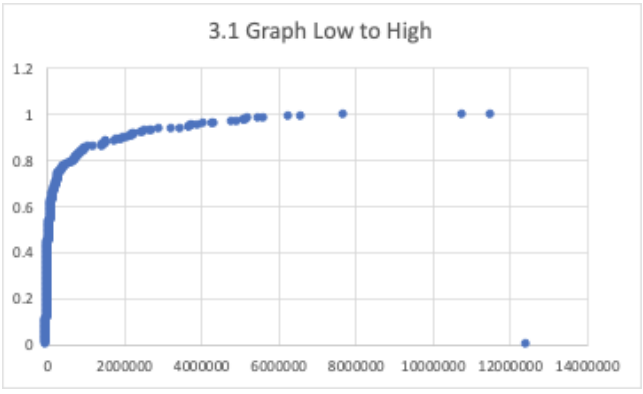
3.1) Does the popularity distribution of airports follow a Zipf distribution? If not, what distribution does it follow?

Export to S3 as a CSV File

- CREATE EXTERNAL TABLE threeCSV(airport STRING,popularity BIGINT) row format delimited fields terminated by ',' lines terminated by '\n' STORED AS TEXTFILE LOCATION 's3n://cs598-testing/export/';
- INSERT OVERWRITE TABLE threeCSV SELECT o.origin as airport, o.flightnum + d.flightnum as popularity FROM (SELECT origin, count(origin) as flightnum FROM airline_ontime_cleaned group by origin) as o, (select dest, count(dest) as flightnum FROM airline_ontime_cleaned group by dest) as d WHERE o.origin = d.dest ORDER BY popularity DESC;

No, when plotting the result of the above query, we can see the overall shape follows a log-normal distribution, and not more of a straight line, which a power-law (Zipfian) distribution would be more similar to. In the attached images, we have the frequency distribution on the y-axis and the popularity on the x-axis. One was plotted with popularity from highest to lowest while the other is plotted from lowest to highest. The chart is similar to the provided example solution in the curvature and distribution, but different in how the scatter plot was created.





3.2) Queries provided first. Results, images of results, and analysis provided after.
Hive Table Setup

- SELECT origin, dest,flightnum,flightdate, deptime, arrdelay + depdelay as delay, uniquecarrier from airline_ontime_cleaned WHERE deptime < "1200" and flightdate like '2008-%';
    - CREATE EXTERNAL TABLE temp_export(origin STRING, dest STRING,flightnum BIGINT, flightdate STRING, deptime STRING, delay DOUBLE, uniquecarrier STRING, sortkey STRING);
    - INSERT OVERWRITE TABLE temp_export SELECT origin, dest,flightnum,flightdate, deptime, arrdelay + depdelay as delay, uniquecarrier, concat(origin, "_", dest, "_", flightdate, "_", uniquecarrier, "_", flightnum) as sortkey from airline_ontime_cleaned WHERE deptime < "1200" and flightdate like '2008-%';
- SELECT origin, dest,flightnum,flightdate, deptime, arrdelay + depdelay as delay, uniquecarrier from airline_ontime_cleaned WHERE deptime < "1200" and flightdate like '2008-%';
    - CREATE EXTERNAL TABLE temp_export2(origin STRING, dest STRING,flightnum BIGINT, flightdate STRING, deptime STRING, delay DOUBLE, uniquecarrier STRING, sortkey STRING);
    - INSERT OVERWRITE TABLE temp_export2 SELECT origin, dest,flightnum,flightdate, deptime, arrdelay + depdelay as delay, uniquecarrier, concat(origin, "_", dest, "_", flightdate, "_", uniquecarrier, "_", flightnum) as sortkey from airline_ontime_cleaned WHERE deptime > "1200" and flightdate like '2008-%';
- SELECT concat(flight1.origin, "_", flight1.dest, "_", flight2.dest) as route, flight1.flightdate as depdate, concat(flight1.uniquecarrier, flight1.flightnum) as firstflight, concat(flight2.uniquecarrier, flight2.flightnum) as secondflight, flight1.delay + flight2.delay as delay, ROW_NUMBER() over (partition by flight1.origin, flight1.dest, flight2.dest, flight1.flightdate order by flight1.delay + flight2.delay asc) as rank FROM flight1, flight2 WHERE flight1.dest = flight2.origin and flight2.flightdate = date_add(flight1.flightdate, 2);
    - CREATE EXTERNAL TABLE temp_complete(route STRING, origin STRING, layover STRING, dest STRING, depdate STRING, deptime STRING, firstflight STRING, second_depdate STRING, second_deptime STRING, secondflight STRING, first_delay STRING, second_delay STRING, total_delay DOUBLE, rank DOUBLE);
    - INSERT OVERWRITE TABLE temp_complete SELECT concat(temp_export.origin, "_", temp_export.dest, "_", temp_export2.dest) as route, temp_export.origin as origin, temp_export.dest as layover, temp_export2.dest as dest, temp_export.flightdate as depdate, temp_export.deptime as deptime, concat(temp_export.uniquecarrier, temp_export.flightnum) as firstflight, concat(temp_export2.uniquecarrier, temp_export2.flightnum) as secondflight, temp_export2.flightdate as second_depdate, temp_export2.deptime as second_deptime, temp_export.delay as first_delay, temp_export2.delay as second_delay, temp_export.delay + temp_export2.delay as total_delay, ROW_NUMBER() over (partition by temp_export.origin, temp_export.dest,temp_export2.dest, temp_export.flightdate order by temp_export.delay + temp_export.delay asc) as rank FROM temp_export, temp_export2 WHERE temp_export.dest = temp_export2.origin and temp_export2.flightdate = date_add(temp_export.flightdate, 2);

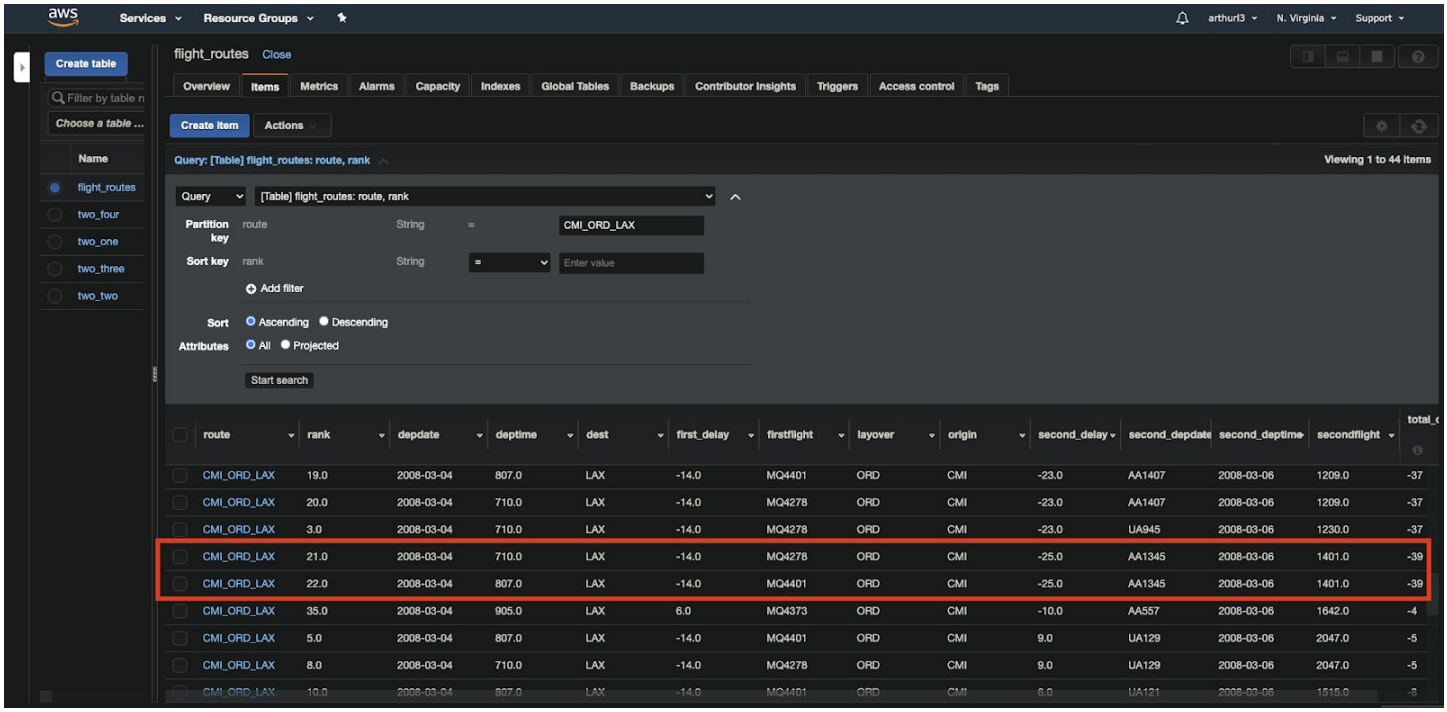
    - Note:; MRJ finished in 40m
- Generate Hive Tables to store results
    - CREATE EXTERNAL TABLE temp_complete(route STRING, origin STRING, layover STRING, dest STRING, depdate STRING, deptime STRING, firstflight STRING, second_depdate STRING, second_deptime STRING, secondflight STRING, first_delay STRING, second_delay STRING, total_delay DOUBLE, rank DOUBLE);
        - For all rows
    - CREATE EXTERNAL TABLE small_temp_complete(route STRING, origin STRING, layover STRING, dest STRING, depdate STRING, deptime STRING, firstflight STRING, second_depdate STRING, second_deptime STRING, secondflight STRING, first_delay STRING, second_delay STRING, total_delay DOUBLE, rank DOUBLE);
        - For rows to be imported into DynamoDB
- Queries for Hive Queries + Insert into DynamoDB
    - SELECT * FROM temp_complete WHERE origin = "CMI" AND layover = 'ORD' AND dest = 'LAX' AND depdate like '2008-03-04';
        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "CMI" AND layover = 'ORD' AND dest = 'LAX' AND depdate like '2008-03-04';
        - Route: CMI_ORD_LAX
    - SELECT * FROM temp_complete WHERE origin = "JAX" AND layover = 'DFW' AND dest = 'CRP' AND depdate like '2008-09-09';
        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "JAX" AND layover = 'DFW' AND dest = 'CRP' AND depdate like '2008-09-09';
        - JAX_DFE_CRP
    - SELECT * FROM temp_complete WHERE origin = "SLC" AND layover = 'BFL' AND dest = 'LAX' AND depdate like '2008-04-01';
        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "SLC" AND layover = 'BFL' AND dest = 'LAX' AND depdate like '2008-04-01';
        - SLC_BFL_LAX
    - SELECT * FROM temp_complete WHERE origin = "LAX" AND layover = 'SFO' AND dest = 'PHX' AND depdate like '2008-07-12';
        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "LAX" AND layover = 'SFO' AND dest = 'PHX' AND depdate like '2008-07-12';
        - LAX_SFO_PHX
    - SELECT * FROM temp_complete WHERE origin = "DFW" AND layover = 'ORD' AND dest = 'DFW' AND depdate like '2008-06-10';
        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "DFW" AND layover = 'ORD' AND dest = 'DFW' AND depdate like '2008-06-10';
        - DFW_ORD_DFW
    - SELECT * FROM temp_complete WHERE origin = "LAX" AND layover = 'ORD' AND dest = 'JFK' AND depdate like '2008-01-01';

        - INSERT INTO TABLE small_temp_complete SELECT * FROM temp_complete WHERE origin = "LAX" AND layover = 'ORD' AND dest = 'JFK' AND depdate like '2008-01-01';
        - LAX_ORD_JFK
- CREATE EXTERNAL TABLE flight_routes(route STRING, origin STRING, layover STRING, dest STRING, depdate STRING, deptime STRING, firstflight STRING, second_depdate STRING, second_deptime STRING, secondflight STRING, first_delay STRING, second_delay STRING, total_delay DOUBLE, rank STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "flight_routes", "dynamodb.column.mapping" = "route:route,origin:origin,layover:layover,dest:dest,depdate:depdate,deptime:deptime,firstflight:firstflight,second_depdate:second_depdate,second_deptime:second_deptime,secondflight:secondflight,first_delay:first_delay,second_delay:second_delay,total_delay:total_delay,rank:rank");
- INSERT INTO TABLE flight_routes SELECT * from small_temp_complete;
    - DDB - Imports records for the 6 queries
    - INSERT OVERWRITE TABLE flight_routes SELECT * from small_temp_complete;

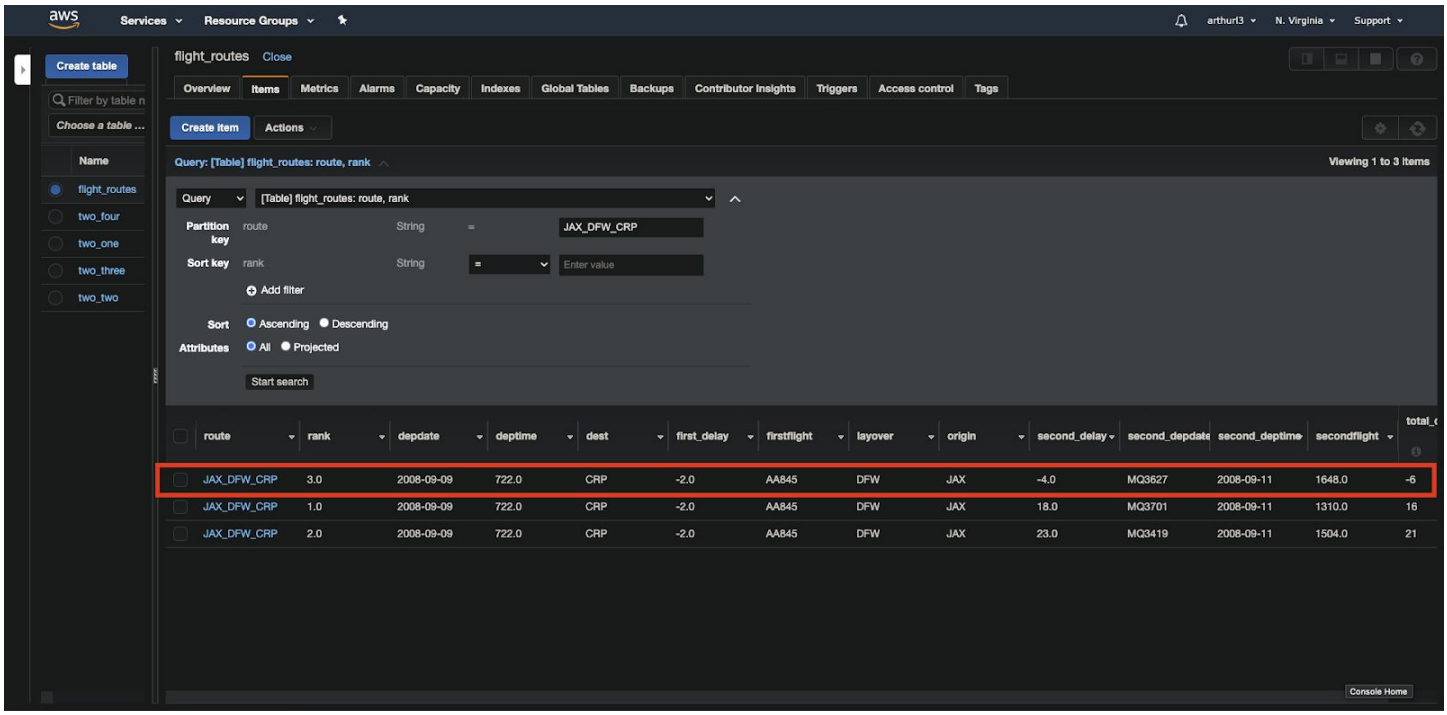
3.2 Results.

* I factored in departure and arrival delay, not just arrival delay. So in the case of query 5, the optimal flight found was not the same as the one provided in the query solutions, although they were very close in total delay. Optimal flights are boxed in red; total_delay is the name of the last column (it's cut off in the images).

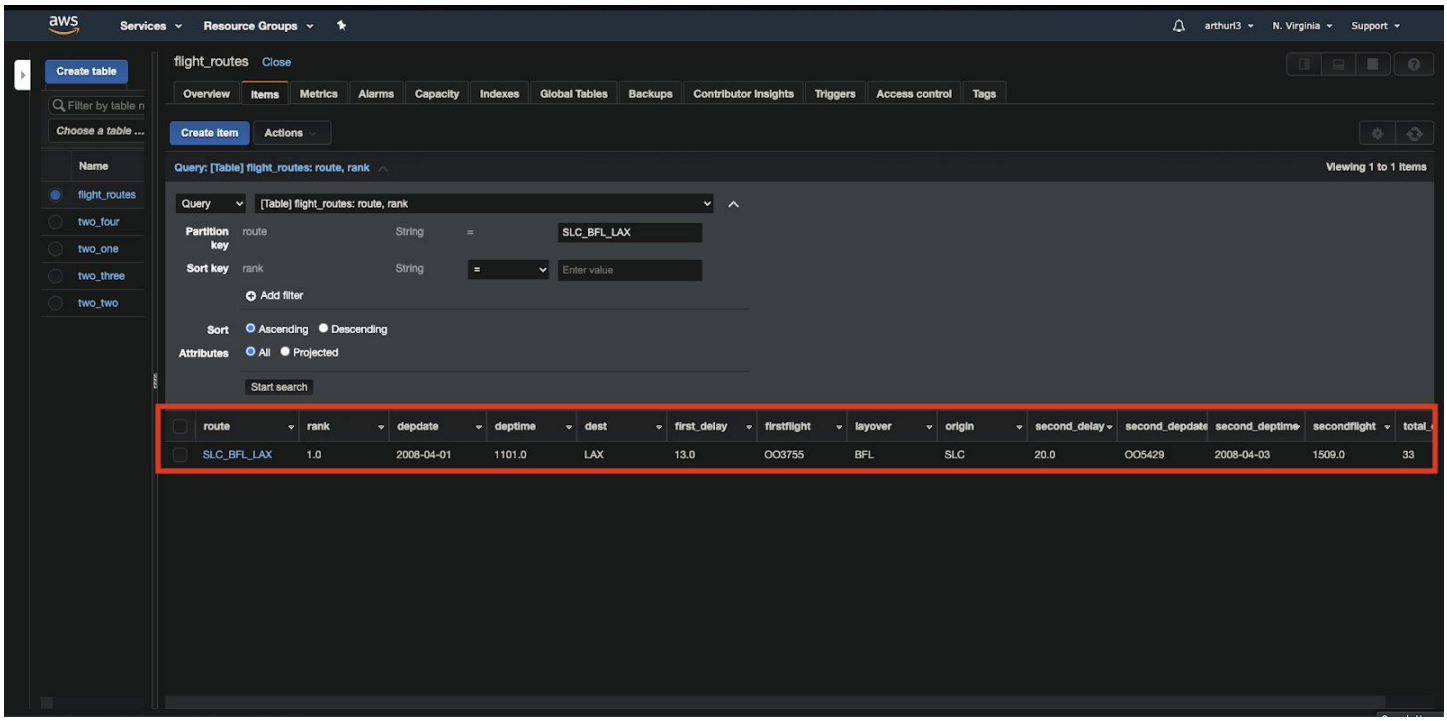
1. CMI → ORD → LAX, 04/03/2008



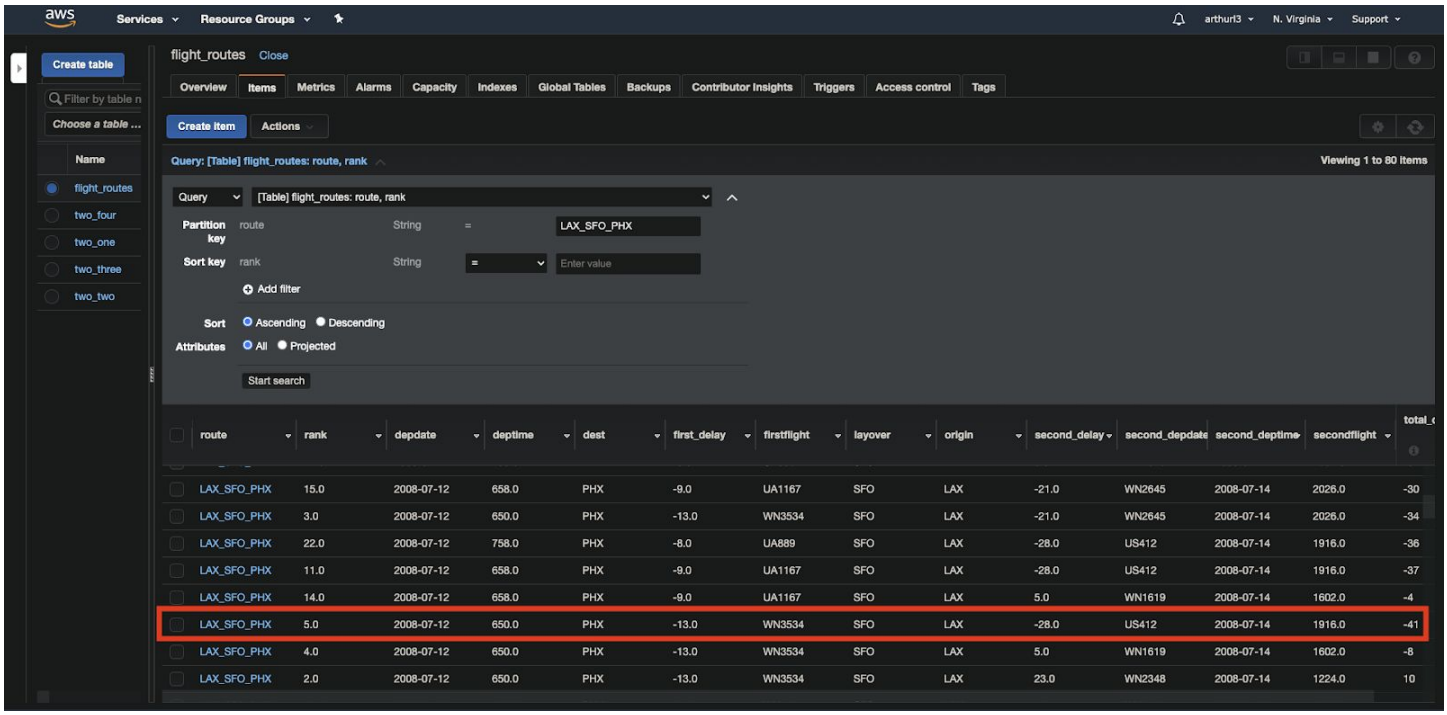
2. JAX → DFW → CRP, 09/09/2008



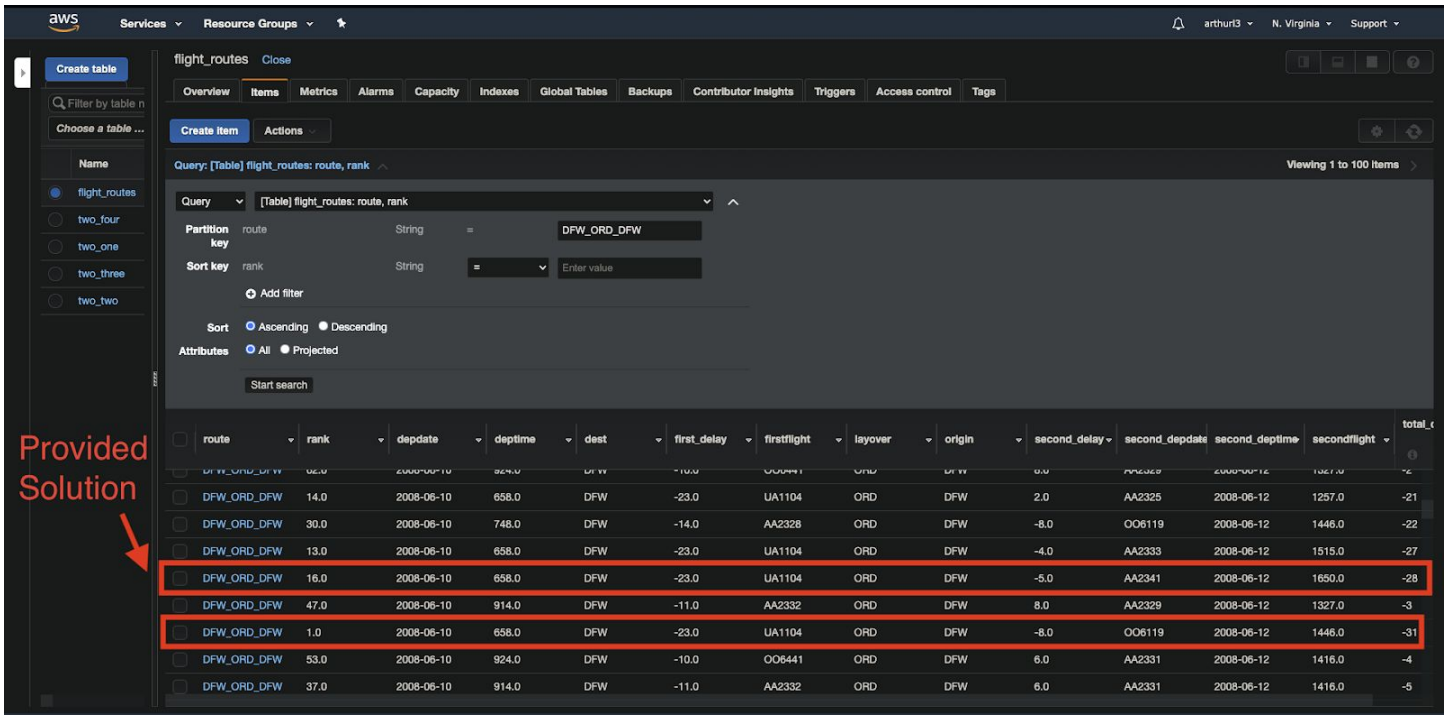
3. SLC → BFL → LAX, 01/04/2008



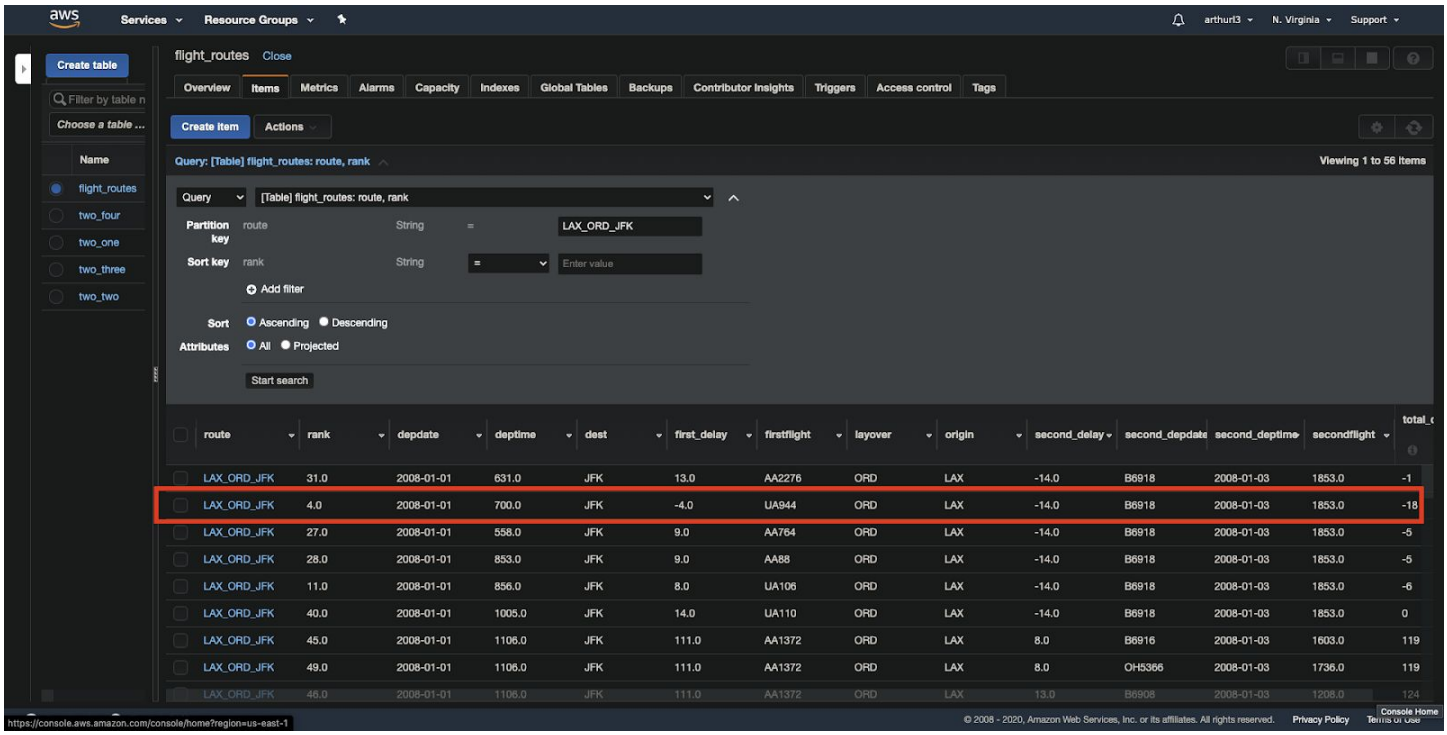
4. LAX → SFO → PHX, 12/07/2008



5. DFW → ORD → DFW, 10/06/2008



LAX → ORD → JFK, 01/01/2008



5) Optimizations

- Cleaning: Removed unused columns in data and cleaned via Python/Jupyter. This, along with gzipping, lowers the amount to be transferred, storage needed, and storage costs.
- DynamoDB: Maintained ordered and sorted results when inserting into DynamoDB using partition and sort keys. When I did not use a sort key, data ingestion took about an hour longer and was incomplete.
    - By adding a sortkey, this optimizes the data integration process for DynamoDB to process the dataset faster and allow for sorting.
    - Partition: origin
    - Sort: averagedepartureddelay
- Data Architecture Optimization. Prior to the updated requirement to not require all the rows for Group 3.2, I was optimizing the data integration process by increasing the write throughput and increasing the EMR cluster size to speed up the data ingestion process.
    - SET dynamodb.throughput.write.percent=1.5;
    - SET mapreduce.job.maps = 20;
    - EMR Resize to 5 instead of default 3.

6) Opinion and Notes

- This was a great project, frustrating and challenging at times, having spent numerous hours figuring out how to do this and put it all together for the first time. Other technical opinions are in line with their response in their respective sections/queries/results.
- While the total is 29 pages in length, the report itself without the results and addendum (everything after this section) is under 5 pages (pages 1-2, 24).
- I added a rubric/expected grading in the next section below for your reference.
- For the Youtube video, it is recorded at 2x speed to fit in the original sub-5 minute requirement. You can use Youtube's playback speed to watch it at a slower pace.

7) Rubric/Expected Grading

Adding PDF, Video Prompts, and Rubric here & adding notes to indicate how they are fulfilled.

PDF Report Prompt

You must submit your report in PDF format. Your report should be no longer than 4-5 pages, 11 point font. Your report should include the following:

- Give a brief overview of how you extracted and cleaned the data.
- Give a brief overview of how you integrated each system.
- What approaches and algorithms did you use to answer each question?
- What are the results of each question? Use only the provided subset for questions from Group 2 and Question 3.2.
- What system- or application-level optimizations (if any) did you employ?

Video

- Ingesting and analyzing data for each question
- Displaying/querying the results for each question

Document Length: 22 pages

- Length of Report: 2-3
    - No more than 4-5 pages excluding the results: Fulfilled, assuming you will not include the "Resources + Addendum" section after the results and do not consider this Rubric/Expected Grading page as part of the report.
    - Source: https://piazza.com/class/ka8oxw9bygm2e9?cid=78
- Addendum
    - Rubric/Expected Grading: Page 3
    - Results: Pages 4-18
    - Queries & Quick Commands: Pages 19+
- 11 Point Font: Fulfilled

Rubric:

- Project Report: 10 points
    - Fulfilled; 1) See Section 1, 2) See Section 3, 3) see Results section 4.
- Project Video: 10 points
    - Fulfilled; 1) Shows data ingestion + data analysis 2) results are queried
    - Youtube Link: https://youtu.be/wqOQvRAVE7M
    - Watch at HD quality since SD quality is too poor and everything is blurry.
- Speed/Efficiency: 10 points
    - Optimization 1: Improve data cleaning before and during ingestion
    - Optimization 2: Maintained ordered and sorted records in DynamoDB, including partition and sort keys.
- System Integration: 10 points
    - Fulfilled - Uses 1) Hadoop or Spark, and 2) Cassandra or DynamoDB.
        - See section 2) for more details
- Quality of Results: 10 points
    - Fulfilled: See results addendum + video. Also did all queries, not just minimum.
- Total: 10+10+10+10+10 = 50/50

Quick Commands

Quick Commands

- scp -i cs598.pem ec2-user@ec2-TBD.compute-1.amazonaws.com:~/newvolume/aviation/airline_ontime /Users/arthurliou/SCP/
- lsblk
- sudo mkdir /home/ec2-user/data
- sudo mount /dev/xvdf /home/ec2-user/data
- sudo umount /home/ec2-user/data
- sudo mount -a
- scp -rp -i cs598.pem ec2-user@ec2-TBD.compute-1.amazonaws.com:~/data/aviation/airline_ontime /Users/arthurliou/SCP/
- sudo scp -r -i cs598.pem /Users/arthurliou/SCP/solution.zip ec2-user@ec2-TBD.compute-1.amazonaws.com:~/
    - Testing Uploaded Successfully
- Organization + Moving.
    - Expected 240 files
        - Ignore 2008_11 and 2008_12 zips to to unzipping issues
    - find . -name "*.zip" -exec unzip {} \;
    - Move all CSV to one directory
        - find . -name '*.csv' -exec mv {} ~/cs/uiuc/cs598/airline_ontime/move/ \;
        - find . -name '*.csv' -exec cp {} ~/cs/uiuc/cs598/airline_ontime/csv/ \;
    - Convert to utf-8
        - find . -type f -exec bash -c 'iconv -f iso-8859-1 -t utf-8 "{}" > ~/cs/uiuc/cs598/airline_ontime/converted/"{}"' \;
    - Compressed each csv into a .gz
        - gzip -r converted
    - Upsert .gz to S3
- AWS Athena Validation Query
    - s3://arthurl3-cs598/airline_ontime_raw_data/
    - s3://arthurl3-cs598/airline_ontime_cleaned/
    - select count(*) from airline_ontime_cleaned;
    - select * from airline_ontime_cleaned limit 10;
- EMR Setup
    - ssh -i cs598-ddb.pem hadoop@ec2-TBD.compute-1.amazonaws.com
    - ssh -i cs598-educate
- Validate Hive External Table Row Count
    - show tblproperties airline_ontime_cleaned;
    - describe extended airline_ontime_cleaned;
    - SELECT count(*) FROM airline_ontime_cleaned;
        - //116754192 with headers
    - SELECT count(*) FROM airline_ontime_cleaned WHERE flightnum IS NOT NULL;
    - SELECT * FROM airline_ontime_cleaned limit 10;

- Setups for Schema, Hive External Tables, DynamoDB Table (see below)

1) year smallint 2) month smallint 3) dayofmonth tinyint 4) dayofweek tinyint 5) flightdate string 6) uniquecarrier string 7) airlineid int 8) carrier string 9) flightnum smallint 10) origin string 11) dest string 12) crsdeptime double 13) deptime double 14) depdelay int 15) depdelayminutes int 16) crsarrtime double 17) arrtime double 18) arrdelay int 19) arrdelayminutes int	Notes • 116754192 - 116753952 = 240 • year, month, dayofmonth, dayofweek, flightdate, uniquecarrier, airlineid, carrier, flightnum, origin, dest,crsdeptime, deptime, depdelay, depdelayminutes, crsarrtime, arrtime, arrdelay, arrdelayminutes, concat(origin, '_', dest, '_', uniquecarrier) as sortKey **Import** INSERT OVERWRITE TABLE airlineTimes SELECT * FROM airline_ontime_cleaned;
External Table for S3 Import CREATE EXTERNAL TABLE airline_ontime_cleaned(year BIGINT, month BIGINT, dayofmonth BIGINT, dayofweek BIGINT, flightdate STRING, uniquecarrier STRING, airlineid BIGINT, carrier STRING, flightnum BIGINT, origin STRING, dest STRING, crsdeptime DOUBLE, deptime DOUBLE, depdelay DOUBLE, depdelayminutes DOUBLE, crsarrtime DOUBLE, arrtime DOUBLE, arrdelay DOUBLE, arrdelayminutes DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 's3://arthurl3-cs598/airline_ontime_cleaned/' tblproperties ('skip.header.line.count'='1');	**Create Hive-DDB Mapping** CREATE EXTERNAL TABLE airlineTimes(year BIGINT, month BIGINT, dayofmonth BIGINT, dayofweek BIGINT, flightdate STRING, uniquecarrier STRING, airlineid BIGINT, carrier STRING, flightnum BIGINT, origin STRING, dest STRING, crsdeptime DOUBLE, deptime DOUBLE, depdelay DOUBLE, depdelayminutes DOUBLE, crsarrtime DOUBLE, arrtime DOUBLE, arrdelay DOUBLE, arrdelayminutes DOUBLE, sortKey STRING) STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler' TBLPROPERTIES ("dynamodb.table.name" = "airlineTimes", "dynamodb.column.mapping" = "year:year,month:month,dayofmonth:dayofmonth,dayofweek:dayofweek,flightdate:flightdate,uniquecarrier:uniquecarrier,airlineid:airlineid,carrier:carrier,flightnum:flightnum,origin:origin,dest:dest,crsdeptime:crsdeptime,deptime:deptime,depdelay:depdelay,depdelayminutes:depdelayminutes,crsarrtime:crsarrtime,arrtime:arrtime,arrdelay:arrdelay,arrdelayminutes:arrdelayminutes,sortKey:sortKey");

Resources

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-copy-snapshot.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html
- https://devopscube.com/mount-ebs-volume-ec2-instance/
- https://docs.aws.amazon.com/AmazonS3/latest/user-guide/upload-objects.html
- https://aws.amazon.com/blogs/big-data/build-a-data-lake-foundation-with-aws-glue-and-amazon-s3/
- https://docs.aws.amazon.com/glue/latest/dg/populate-data-catalog.html
- https://hevodata.com/blog/dynamodb-to-s3-using-aws-glue/
- https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EMRforDynamoDB.Tutorial.html
- https://docs.aws.amazon.com/efs/latest/ug/accessing-fs-create-security-groups.html
- https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EMRforDynamoDB.ExternalTableForDDB.html
- Check Mapper to Maximize Throughput
    - https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop-task-config.html
    - https://stackoverflow.com/questions/41454796/aws-emr-parallel-mappers
    - 12288/3072 = 4. 3x Cluster Size = 12 mappers
    - So number of write capacity units should be greater than 12
    - However, I'm unable to change the Provisioned Capacity setting, so created as is, with 5 Write Capacity Units
- https://aws.amazon.com/getting-started/hands-on/optimize-amazon-emr-clusters-with-ec2-spot/
- https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EMRforDynamoDB.html
- https://docs.aws.amazon.com/emr/latest/ReleaseGuide/EMR_Interactive_Hive.html
- https://docs.aws.amazon.com/emr/latest/ReleaseGuide/EMR_Hive_Commands.html