Innovative Assignment
Flight Data Analysis using Hive

Date: 2 Nov, 2023
Course Code and Name: 2CS702 Big Data Analysis

BY:
20BCE204
20BCE206

Problem Statement :

The objective of this comprehensive analysis is to glean meaningful insights from aviation datasets, focusing on airports, airlines, and routes. The analysis aims to address specific queries to provide a nuanced understanding of the aviation landscape. The outlined tasks are as follows:

A. Identification of Airports in India:

- Extract and compile a detailed list of airports operating within the sovereign boundaries of India. This information will be crucial for various stakeholders, including government agencies, airlines, and infrastructure planners.

B. Compilation of Zero-Stop Flights:

- Identify and present a comprehensive list of airlines offering zero-stop flights. This data will be instrumental in understanding direct flight options and can be leveraged by travelers and airline companies alike.

C. Analysis of Airlines with Code Share Agreements:

- Compile a detailed list of airlines engaged in code-share agreements. This analysis will shed light on collaborative practices within the aviation industry and aid in understanding the dynamics of airline partnerships.

D. Determination of Country/Territory with Highest Airports:

- Investigate and ascertain the country or territory boasting the highest number of airports. This information will be pivotal for global aviation planning, government policies, and strategic investments in airport infrastructure.

E. Compilation of Active Airlines in the United States:

- Identify and present a comprehensive list of active airlines currently operating in the United States. This data is crucial for market research, competition analysis, and strategic decision-making within the U.S. aviation sector.

This problem statement sets the stage for a thorough exploration of the aviation data landscape, utilizing advanced data processing techniques. The outcomes of this analysis will contribute valuable insights for stakeholders ranging from government bodies to industry participants, fostering a deeper understanding of the intricate dynamics within the aviation domain.

Overview:

The Airlines Analysis Hadoop project aims to extract insights from datasets on airports, airlines, and routes. Using Hadoop, MapReduce, and Hive, the project focuses on tasks such as listing Indian airports, identifying zero-stop flights, listing code-share airlines, finding the country with the most airports, and listing active U.S. airlines. Challenges include handling large datasets and ensuring data quality. The project concludes with summarizing key findings for a better understanding of the aviation landscape.

Introduction

In the dynamic landscape of aviation, understanding the operational intricacies of airports, airlines, and routes is crucial for informed decision-making and strategic planning. The Airlines Analysis Hadoop project embarks on a comprehensive exploration of datasets encompassing airports, airlines, and routes, employing cutting-edge technologies such as Hadoop, MapReduce, and Hive for efficient and distributed data processing.

This report encapsulates the project's objective, dataset description, and analysis tasks, offering a glimpse into the vast realm of the aviation industry. By delving into the details of airports, airlines, and routes, the project aims to uncover valuable insights that can shape the future of air travel. Through the use of Hadoop-based tools, the report outlines the technology stack employed, the expected outputs, and the challenges encountered during the analytical journey.

The subsequent sections provide a detailed overview of the datasets, the analysis tasks at hand, the chosen technology stack, and the anticipated conclusions. As we navigate through the complexities of airport distribution, airline operations, and route characteristics, the report sets the stage for a compelling exploration of the aviation data landscape.

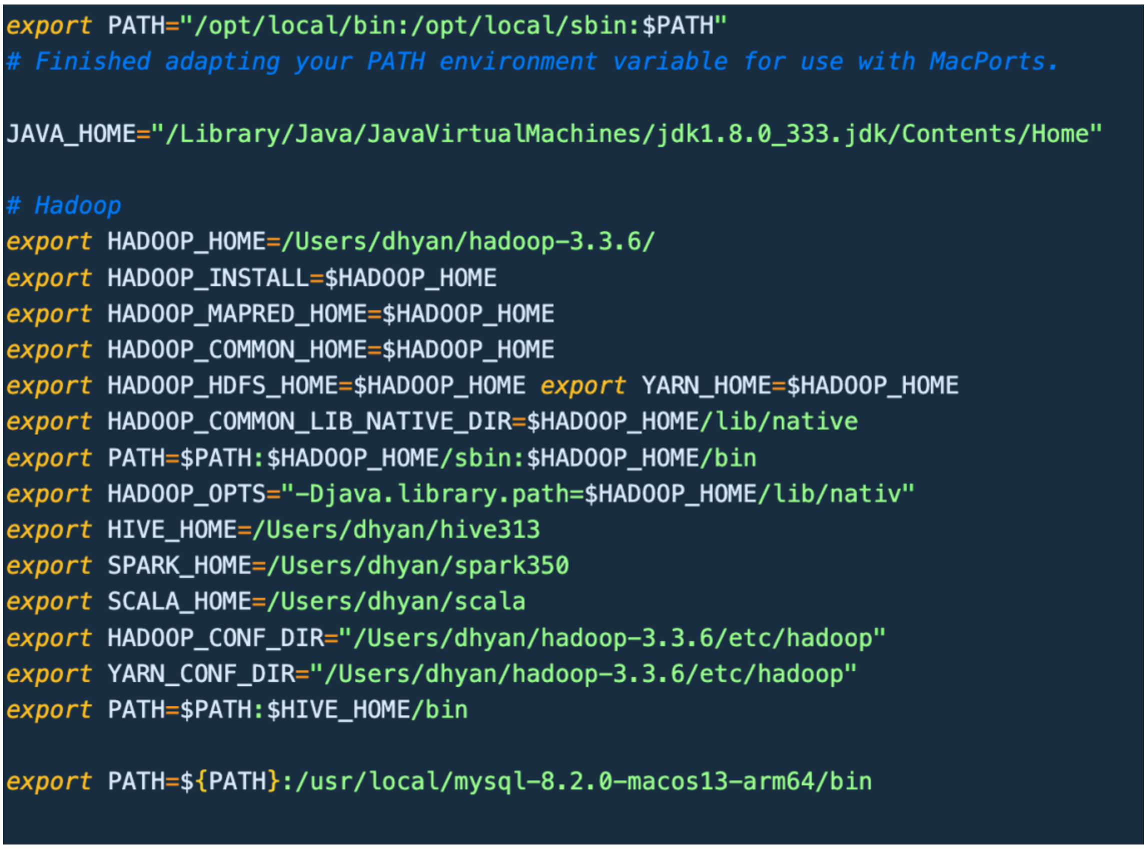
Steps to Install Hive:

1. Download and Untar Hive:

$ wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz

$ tar xzf apache-hive-3.1.2-bin.tar.gz

2. **Configure Hive Environment Variables (zprofile):**

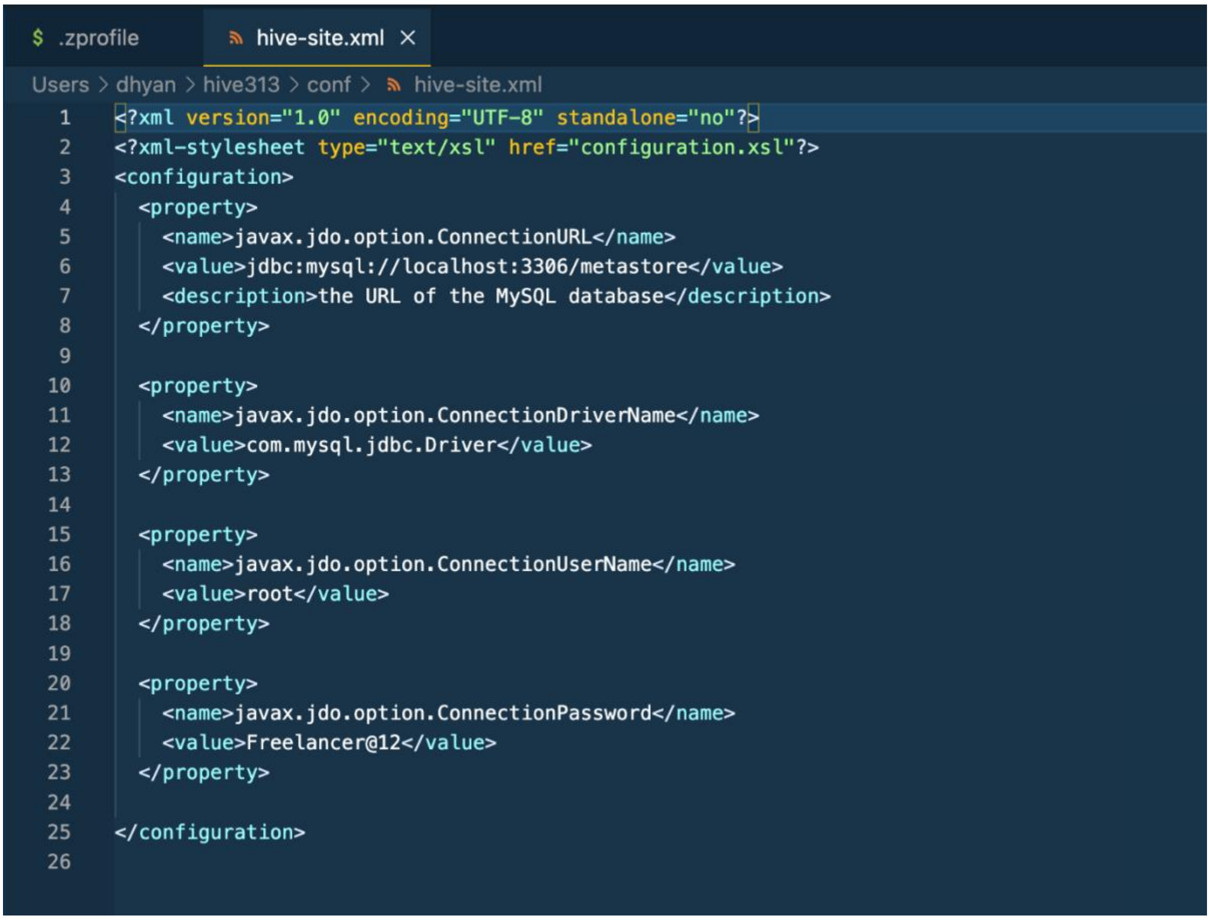


```
export PATH="/opt/local/bin:/opt/local/sbin:$PATH"
# Finished adapting your PATH environment variable for use with MacPorts.

JAVA_HOME="/Library/Java/JavaVirtualMachines/jdk1.8.0_333.jdk/Contents/Home"

# Hadoop
export HADOOP_HOME=/Users/dhyan/hadoop-3.3.6/
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/nativ"
export HIVE_HOME=/Users/dhyan/hive313
export SPARK_HOME=/Users/dhyan/spark350
export SCALA_HOME=/Users/dhyan/scala
export HADOOP_CONF_DIR="/Users/dhyan/hadoop-3.3.6/etc/hadoop"
export YARN_CONF_DIR="/Users/dhyan/hadoop-3.3.6/etc/hadoop"
export PATH=$PATH:$HIVE_HOME/bin

export PATH=${PATH}:/usr/local/mysql-8.2.0-macos13-arm64/bin
```

3. **Connecting MySQL and Setting up JDBC Driver for MySQL**
   (base) dhyan@Dhyans-MacBook-Pro lib % cp /Users/dhyan/Downloads/mysql-connector-j-8.2.0.tar.gz .
   (base) dhyan@Dhyans-MacBook-Pro lib % tar xvf mysql-connector-j-8.2.0.tar.gz

4. **Setting up Hive.xml file**

```
(base) dhyan@Dhyans-MacBook-Pro hive313 % ls
LICENSE                          binary-package-licenses      jdbc
NOTICE                           conf                         lib
RELEASE_NOTES.txt                examples                     scripts
bin                      hcatalog

(base) dhyan@Dhyans-MacBook-Pro hive313 % cd conf
(base) dhyan@Dhyans-MacBook-Pro conf % ls
beeline-log4j2.properties.template     ivysettings.xml
hive-default.xml.template              llap-cli-log4j2.properties.template
hive-env.sh.template                   llap-daemon-log4j2.properties.template

(base) dhyan@Dhyans-MacBook-Pro conf % cp hive-default.xml.template hive-site.xml
(base) dhyan@Dhyans-MacBook-Pro conf % code hive-site.xml
```



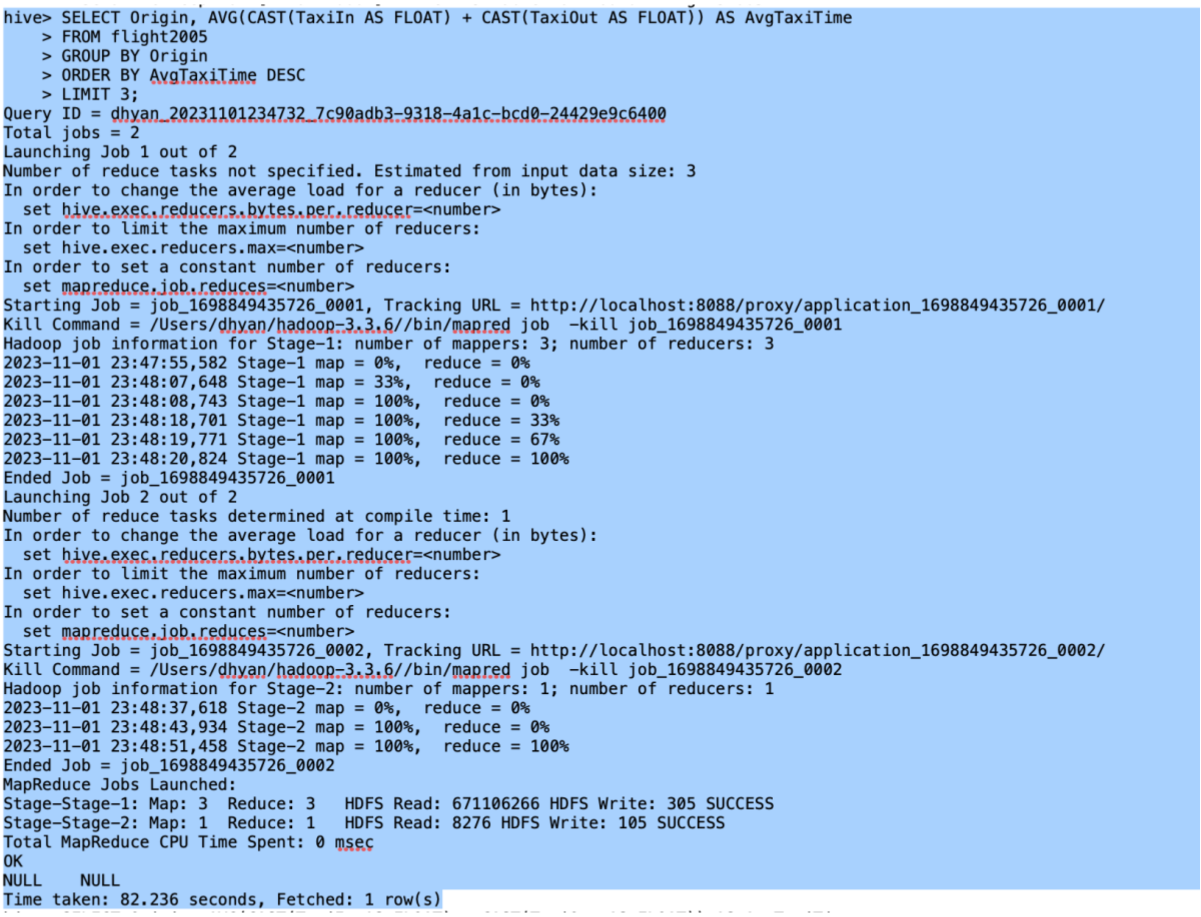
```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/metastore</value>
    <description>the URL of the MySQL database</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>root</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>Freelancer@12</value>
  </property>

</configuration>
```

5. **Starting hive**

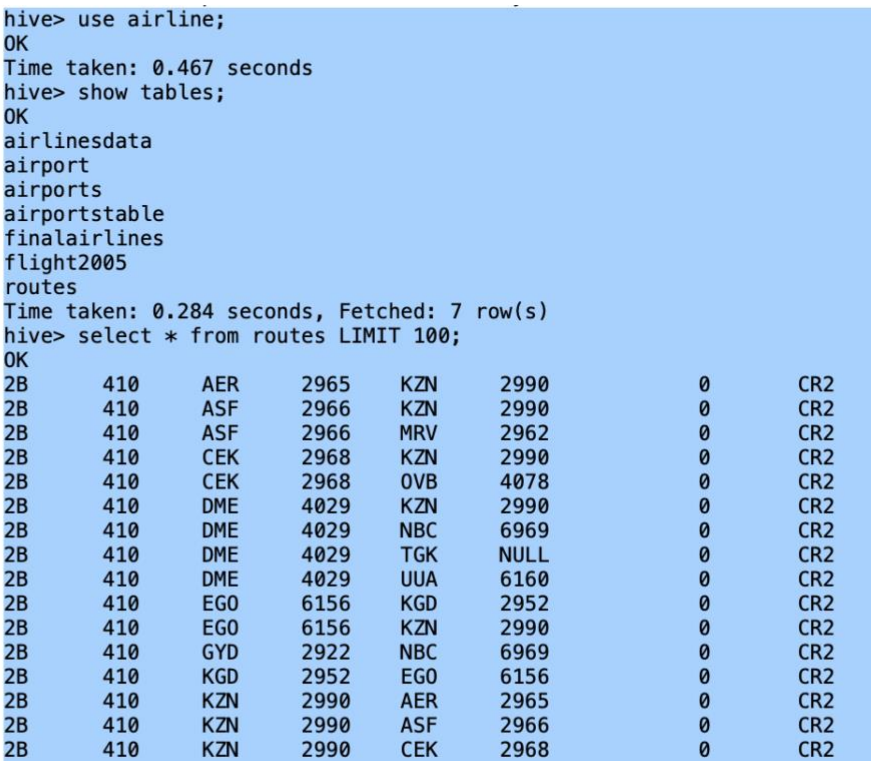
```
(base) dhyan@Dhyans-MacBook-Pro ~ % schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/Users/dhyan/hive313/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

```
SLF4J: Found binding in [jar:file:/Users/dhyan/hadoop-
3.3.6/share/hadoop/common/lib/slf4j-reload4j-
1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:	 jdbc:mysql://localhost:3306/metastore
Metastore Connection Driver :	 com.mysql.jdbc.Driver
Metastore connection User:	 root
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and
manual loading of the driver class is generally unnecessary.
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.mysql.sql



Initialization script completed
schemaTool completed
(base) dhyan@Dhyans-MacBook-Pro ~ % hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/Users/dhyan/hive313/lib/log4j-slf4j-impl-
2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/Users/dhyan/hadoop-
3.3.6/share/hadoop/common/lib/slf4j-reload4j-
1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 96402e96-147e-40f6-bd07-3964e25604dc

Logging initialized using configuration in jar:file:/Users/dhyan/hive313/lib/hive-
common-3.1.3.jar!/hive-log4j2.properties Async: true
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and
manual loading of the driver class is generally unnecessary.
Hive Session ID = 0e94329a-b989-461b-8eec-2027a2fcccdb
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions.
Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X
releases.
hive>
```

Implementation :

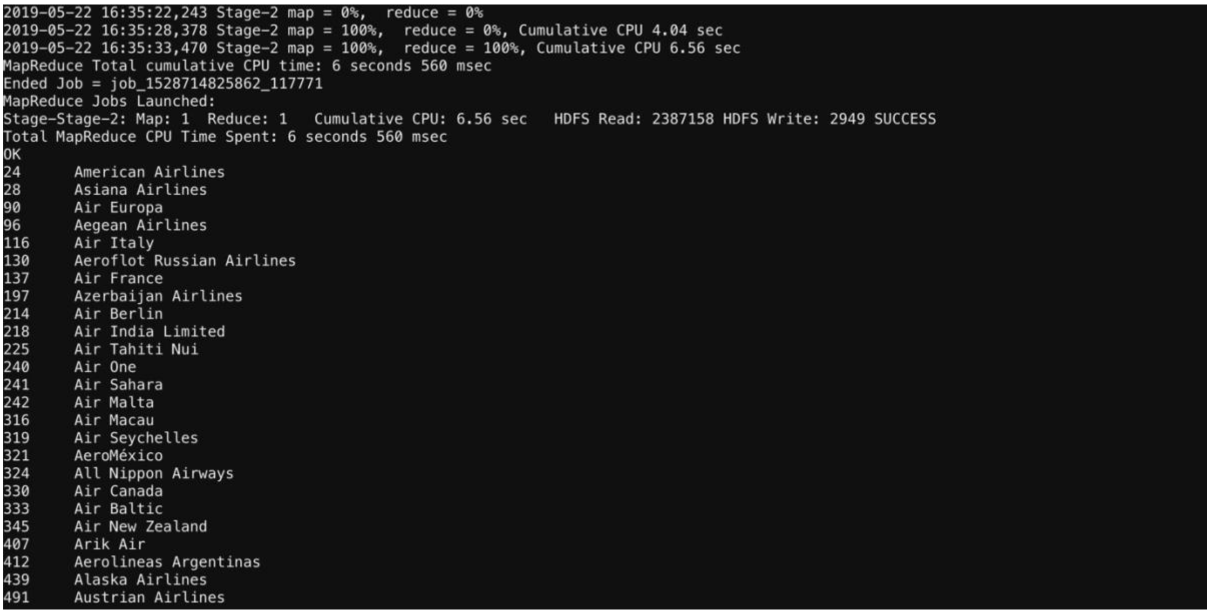


```
hive> SELECT Origin, AVG(CAST(TaxiIn AS FLOAT) + CAST(TaxiOut AS FLOAT)) AS AvgTaxiTime
    > FROM flight2005
    > GROUP BY Origin
    > ORDER BY AvgTaxiTime DESC
    > LIMIT 3;
Query ID = dhyan_20231101234732_7c90adb3-9318-4a1c-bcd0-24429e9c6400
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 3
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_1698849435726_0001, Tracking URL = http://localhost:8088/proxy/application_1698849435726_0001/
Kill Command = /Users/dhyan/hadoop-3.3.6//bin/mapred job  -kill job_1698849435726_0001
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 3
2023-11-01 23:47:55,582 Stage-1 map = 0%,  reduce = 0%
2023-11-01 23:48:07,648 Stage-1 map = 33%,  reduce = 0%
2023-11-01 23:48:08,743 Stage-1 map = 100%,  reduce = 0%
2023-11-01 23:48:18,701 Stage-1 map = 100%,  reduce = 33%
2023-11-01 23:48:19,771 Stage-1 map = 100%,  reduce = 67%
2023-11-01 23:48:20,824 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_1698849435726_0001
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_1698849435726_0002, Tracking URL = http://localhost:8088/proxy/application_1698849435726_0002/
Kill Command = /Users/dhyan/hadoop-3.3.6//bin/mapred job  -kill job_1698849435726_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2023-11-01 23:48:37,618 Stage-2 map = 0%,  reduce = 0%
2023-11-01 23:48:43,934 Stage-2 map = 100%,  reduce = 0%
2023-11-01 23:48:51,458 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_1698849435726_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3  Reduce: 3   HDFS Read: 671106266 HDFS Write: 305 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   HDFS Read: 8276 HDFS Write: 105 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
NULL    NULL
Time taken: 82.236 seconds, Fetched: 1 row(s)
```



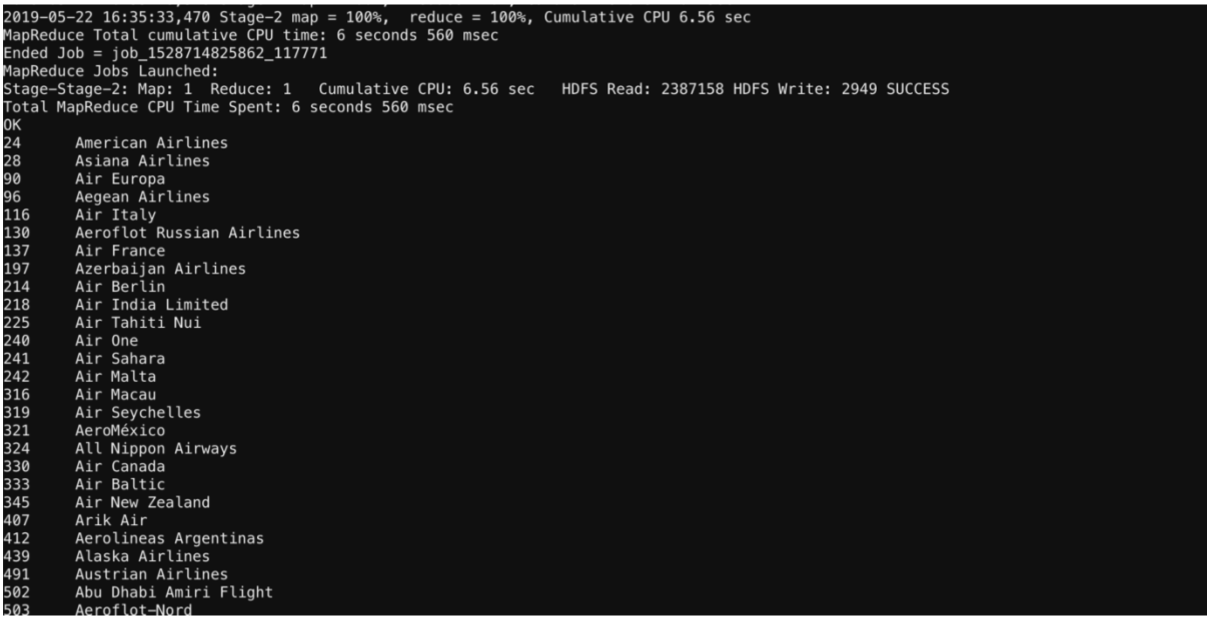
```
hive> use airline;
OK
Time taken: 0.467 seconds
hive> show tables;
OK
airlinesdata
airport
airports
airportstable
finalairlines
flight2005
routes
Time taken: 0.284 seconds, Fetched: 7 row(s)
hive> select * from routes LIMIT 100;
OK
2B      410     AER     2965    KZN     2990            0       CR2
2B      410     ASF     2966    KZN     2990            0       CR2
2B      410     ASF     2966    MRV     2962            0       CR2
2B      410     CEK     2968    KZN     2990            0       CR2
2B      410     CEK     2968    OVB     4078            0       CR2
2B      410     DME     4029    KZN     2990            0       CR2
2B      410     DME     4029    NBC     6969            0       CR2
2B      410     DME     4029    TGK     NULL            0       CR2
2B      410     DME     4029    UUA     6160            0       CR2
2B      410     EGO     6156    KGD     2952            0       CR2
2B      410     EGO     6156    KZN     2990            0       CR2
2B      410     GYD     2922    NBC     6969            0       CR2
2B      410     KGD     2952    EGO     6156            0       CR2
2B      410     KZN     2990    AER     2965            0       CR2
2B      410     KZN     2990    ASF     2966            0       CR2
2B      410     KZN     2990    CEK     2968            0       CR2
```

select distinct(a.airlineid),a.name
from finalairlines a join routes on
a.airlineid=routes.airlineid
where stops=0



```
2019-05-22 16:35:22,243 Stage-2 map = 0%,  reduce = 0%
2019-05-22 16:35:28,378 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.04 sec
2019-05-22 16:35:33,470 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.56 sec
MapReduce Total cumulative CPU time: 6 seconds 560 msec
Ended Job = job_1528714825862_117771
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.56 sec   HDFS Read: 2387158 HDFS Write: 2949 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 560 msec
OK
24      American Airlines
28      Asiana Airlines
90      Air Europa
96      Aegean Airlines
116     Air Italy
130     Aeroflot Russian Airlines
137     Air France
197     Azerbaijan Airlines
214     Air Berlin
218     Air India Limited
225     Air Tahiti Nui
240     Air One
241     Air Sahara
242     Air Malta
316     Air Macau
319     Air Seychelles
321     AeroMéxico
324     All Nippon Airways
330     Air Canada
333     Air Baltic
345     Air New Zealand
407     Arik Air
412     Aerolineas Argentinas
439     Alaska Airlines
491     Austrian Airlines
```

select airlineid, name
from finalairlines
where country="United States" AND active="Y"



```
2019-05-22 16:35:33,470 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.56 sec
MapReduce Total cumulative CPU time: 6 seconds 560 msec
Ended Job = job_1528714825862_117771
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.56 sec   HDFS Read: 2387158 HDFS Write: 2949 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 560 msec
OK
24      American Airlines
28      Asiana Airlines
90      Air Europa
96      Aegean Airlines
116     Air Italy
130     Aeroflot Russian Airlines
137     Air France
197     Azerbaijan Airlines
214     Air Berlin
218     Air India Limited
225     Air Tahiti Nui
240     Air One
241     Air Sahara
242     Air Malta
316     Air Macau
319     Air Seychelles
321     AeroMéxico
324     All Nippon Airways
330     Air Canada
333     Air Baltic
345     Air New Zealand
407     Arik Air
412     Aerolineas Argentinas
439     Alaska Airlines
491     Austrian Airlines
502     Abu Dhabi Amiri Flight
503     Aeroflot-Nord
```

Applications:

1. Airline Operations Optimization:
- The analysis of airports, airlines, and routes can aid in optimizing airline operations. Insights into zero-stop flights and code-share agreements can inform decisions to streamline routes and improve overall efficiency.

2. Market Research for Airlines:
- Understanding the active airlines in the United States and identifying key players provides valuable market research data. Airlines can use this information to assess competition and strategize market expansion.

3. Government Planning and Infrastructure Development:
- Knowledge of the distribution of airports and the country with the highest number of airports is essential for government planning and infrastructure development. It can guide decisions on where to invest in airport facilities and transportation networks.

4. Travel and Tourism Industry Insights:
- The data analysis can offer insights into travel patterns, helping the travel and tourism industry understand popular routes, airlines, and destinations. This information can guide marketing strategies and service offerings.

Conclusion:

In conclusion, the Airlines Analysis Hadoop project provides a comprehensive exploration of aviation data, leveraging the capabilities of Hadoop, MapReduce, and Hive. By addressing specific tasks such as listing airports in India, identifying zero-stop flights, and analyzing active airlines in the United States, the project aims to offer valuable insights for various stakeholders in the aviation industry.

The use of Hadoop technologies ensures efficient processing of large datasets, enabling a deeper understanding of the complexities within the aviation landscape. As the report unfolds, it delves into the challenges faced during the analysis, the technology stack employed, and the expected outputs. Ultimately, the project seeks to contribute meaningful information for informed decision-making, strategic planning, and optimization in the dynamic and competitive field of aviation.