Analysing Airlines and Airports Data

edureka!

Version 2.0

Problem statement:

1. Find list of Airports operating in the Country India
2. Find the list of Airlines having zero stops
3. List of Airlines operating with code share
4. Which country (or) territory having highest Airports
5. Find the list of Active Airlines in United state

Important Links:

Link For All codes:

https://edureka.wistia.com/medias/rbvd44d3j3/download?media_file_id=66599282

DataSet:

https://edureka.wistia.com/medias/67vuzsza8j/download?media_file_id=66596539

Data Set Description:

In this use case there are 3 data sets.

Final_airlines

routes.dat

airports_mod.dat

****

Air Ports data set i.e airports_mod.dat

It contains the following fields

Airport ID Unique OpenFlights identifier for this airport.

Name Name of airport. May or may not contain the **City** name.

City Main city served by airport. May be spelled differently from **Name**.

Country Country or territory where airport is located.

IATA/FAA 3-letter FAA code, for airports located in **Country** "United States of America".
3-letter IATA code, for all other airports.
Blank if not assigned.

ICAO 4-letter ICAO code.
Blank if not assigned.

Latitude Decimal degrees, usually to six significant digits. Negative is South, positive is North.

Longitude Decimal degrees, usually to six significant digits. Negative is West, positive is East.

Altitude	In feet.
Timezone	Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5.
DST	Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). *See also: Help: Time*
Tz database time zone	Timezone in "tz" (Olson) format, eg. "America/Los_Angeles".

Air Lines Data set:

It contains the following fields:

Airline ID	Unique OpenFlights identifier for this airline.
Name	Name of the airline.
Alias	Alias of the airline. For example, All Nippon Airways is commonly known as "ANA".
IATA	2-letter IATA code, if available.
ICAO	3-letter ICAO code, if available.
Callsign	Airline callsign.
Country	Country or territory where airline is incorporated.
Active	"Y" if the airline is or has until recently been operational, "N" if it is defunct. This field is *not* reliable: in particular, major airlines that stopped flying long ago, but have not had their IATA code reassigned (eg. Ansett/AN), will incorrectly show as "Y".

Routes Data set i.e routes.dat

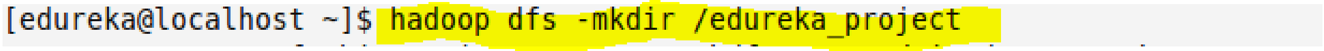
It contains the following fields

Airline	2-letter (IATA) or 3-letter (ICAO) code of the airline.
Airline ID	Unique OpenFlights identifier for airline (see Airline).
Source airport	3-letter (IATA) or 4-letter (ICAO) code of the source airport.
Source airport ID	Unique OpenFlights identifier for source airport (see Airport)
Destination airport	3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
Destination airport ID	Unique OpenFlights identifier for destination airport (see Airport)
Codeshare	"Y" if this flight is a codeshare (that is, not operated by *Airline*, but another carrier), empty otherwise.
Stops	Number of stops on this flight ("0" for direct)
Equipment	3-letter codes for plane type(s) generally used on this flight, separated by spaces

Codes and Explanation:

First we need to create a directory in HDFS.

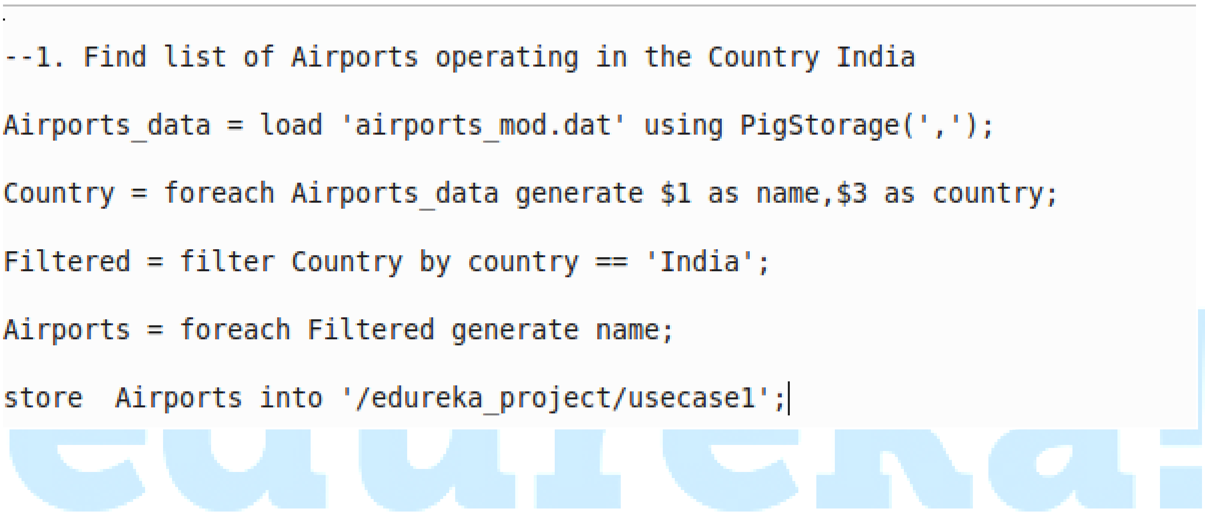
Creating a directory called **edureka_project** in hdfs.



```
[edureka@localhost ~]$ hadoop dfs -mkdir /edureka_project
```

Use case 1:

In this use case we are going to find the list of Airports operating in the country India.



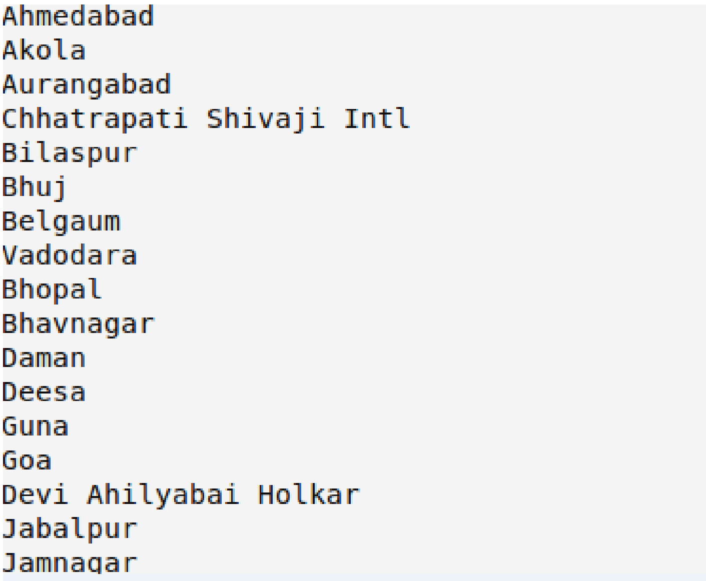
```
--1. Find list of Airports operating in the Country India

Airports_data = load 'airports_mod.dat' using PigStorage(',');

Country = foreach Airports_data generate $1 as name,$3 as country;

Filtered = filter Country by country == 'India';

Airports = foreach Filtered generate name;

store  Airports into '/edureka_project/usecase1';
```

Explanation for usecase1:

- ➔ First we are loading data and applying filter for listing Country India
- ➔ For each generated filter getting the Country name
- ➔ Finally storing the out put into HDFS.

Usecase1 Output:

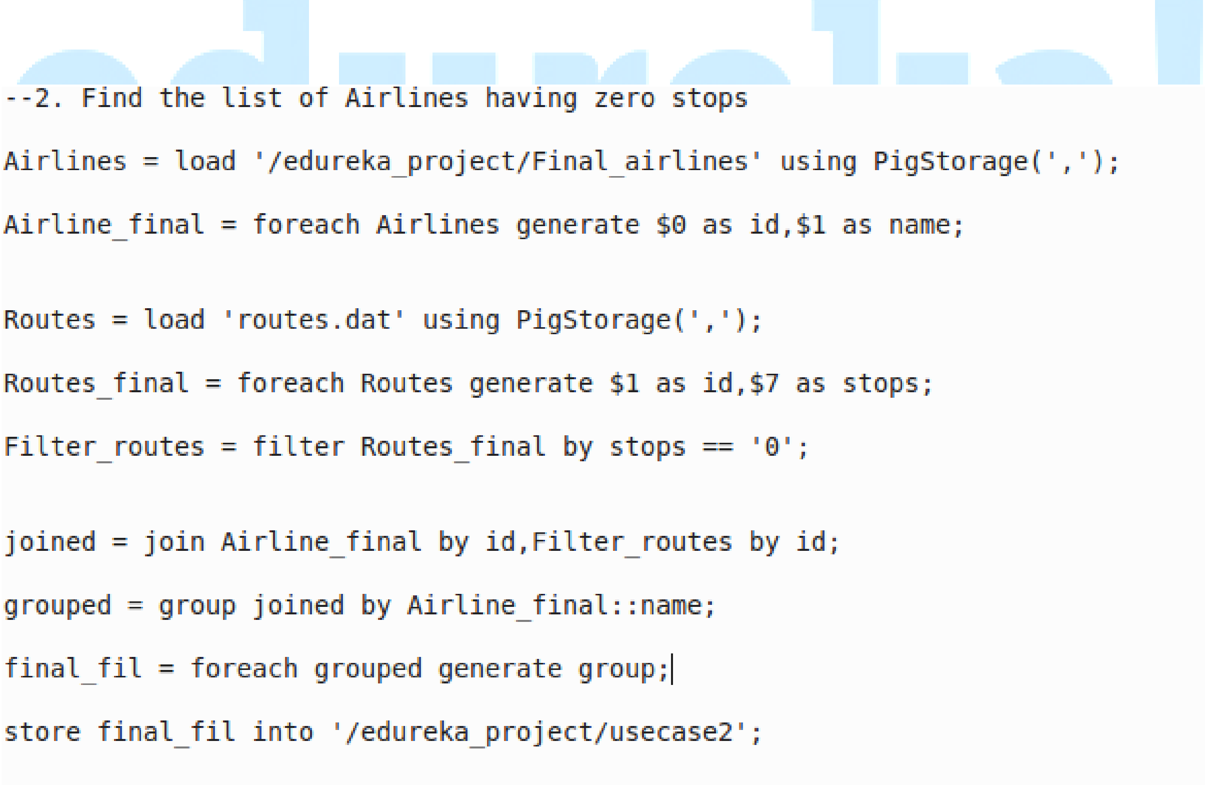
Below is the sample out put screen for usecase1



```
Ahmedabad
Akola
Aurangabad
Chhatrapati Shivaji Intl
Bilaspur
Bhuj
Belgaum
Vadodara
Bhopal
Bhavnagar
Daman
Deesa
Guna
Goa
Devi Ahilyabai Holkar
Jabalpur
Jamnagar
```

Use case 2:

In this use case we are finding the list of Airlines having 0 stops.



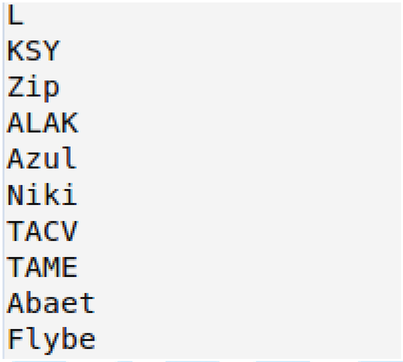
```
--2. Find the list of Airlines having zero stops

Airlines = load '/edureka_project/Final_airlines' using PigStorage(',');

Airline_final = foreach Airlines generate $0 as id,$1 as name;


Routes = load 'routes.dat' using PigStorage(',');

Routes_final = foreach Routes generate $1 as id,$7 as stops;

Filter_routes = filter Routes_final by stops == '0';


joined = join Airline_final by id,Filter_routes by id;

grouped = group joined by Airline_final::name;

final_fil = foreach grouped generate group;

store final_fil into '/edureka_project/usecase2';
```

Explanation for usecase2:

In this use case we are using two data sets. i.e Final_airlines and routes dataset.

- ➔ Loading the two dats sets into two different fields.
- ➔ Finding out the list of airlines having 0 stops.
- ➔ Joining two data sets with field id
- ➔ Grouping the final data with name
- ➔ Final result is storing into HDFS.

Usecase2 Output:



```
L
KSY
Zip
ALAK
Azul
Niki
TACV
TAME
Abaet
Flybe
```

Use case 3:

In this use case we are finding the list of airlines operating with code share.

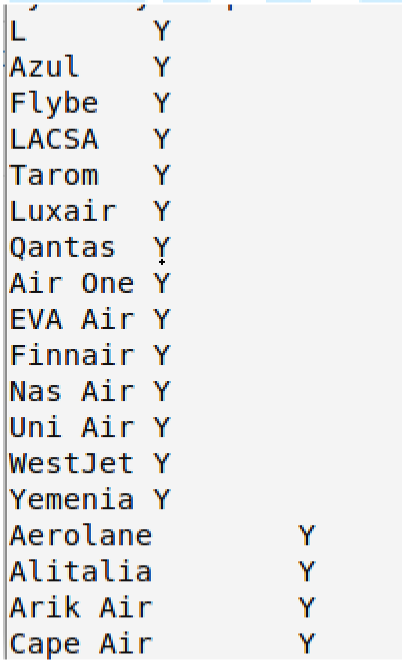
Explanation for usecase3:

For this use case also we are using two datasets. i.e Final_airlines and routes

- ➔ Loading two data sets and finding the code share with option Y
- ➔ Joining the airlines and routes with id
- ➔ Removing the duplicates with DISTINCT
- ➔ Grouping the name and code share
- ➔ Applying FLATTEN to group and saving the final result into HDFS

```
--3. List of Airlines operating with code share

Airlines = load '/edureka_project/Final_airlines' using
PigStorage(',');

Airline_final = foreach Airlines generate $0 as id,$1 as name;

Routes = load '/edureka_project/routes.dat' using PigStorage
(',');

Routes_final = foreach Routes generate $1 as id,$6 as codeshare;

Filter_routes = filter Routes_final by codeshare == 'Y';

joined = join Airline_final by id,Filter_routes by id;

dist = DISTINCT joined;

filt = GROUP dist by (name,codeshare);

grouped = foreach filt GENERATE FLATTEN(group) as (name,
codeshare);

store grouped into '/edureka_project/usecase3';
```

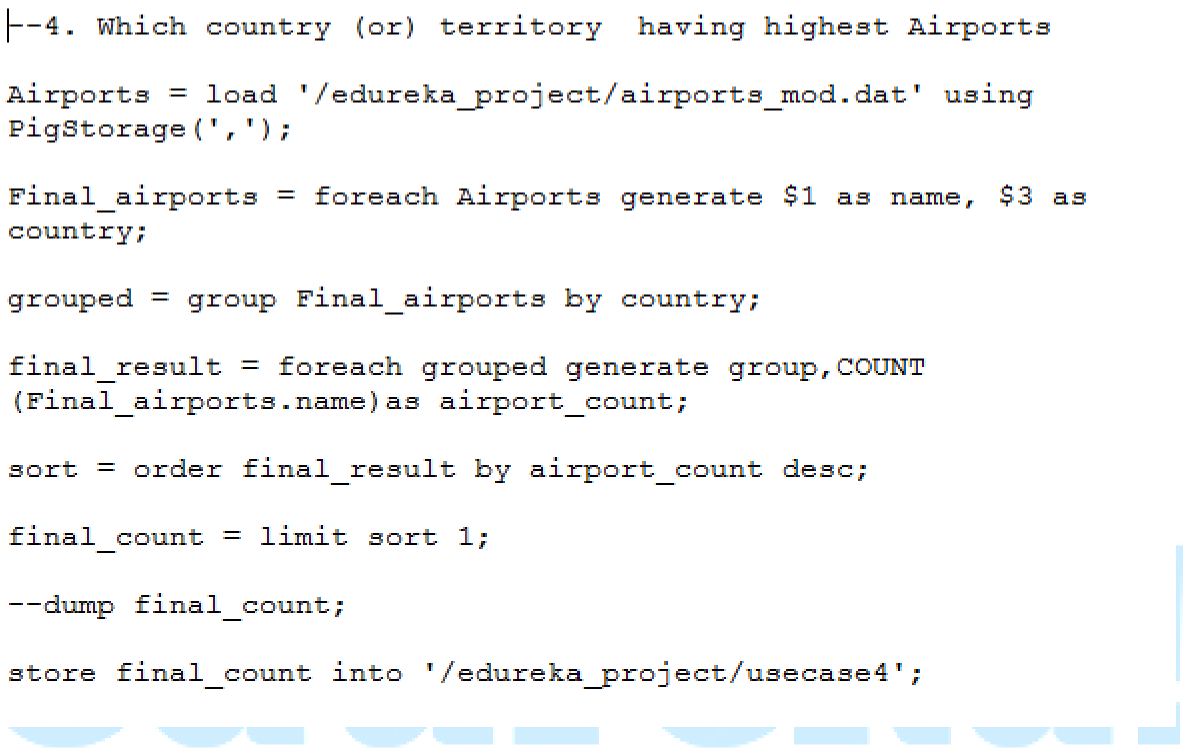
Usecase3 Output:



```
L       Y
Azul    Y
Flybe   Y
LACSA   Y
Tarom   Y
Luxair  Y
Qantas  Y
Air One Y
EVA Air Y
Finnair Y
Nas Air Y
Uni Air Y
WestJet Y
Yemenia Y
Aerolane        Y
Alitalia        Y
Arik Air        Y
Cape Air        Y
```

Use case 4:

In this use case we are finding which country having highest airports.

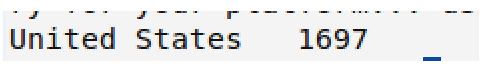


```
--4. Which country (or) territory  having highest Airports

Airports = load '/edureka_project/airports_mod.dat' using
PigStorage(',');

Final_airports = foreach Airports generate $1 as name, $3 as
country;

grouped = group Final_airports by country;

final_result = foreach grouped generate group,COUNT
(Final_airports.name)as airport_count;

sort = order final_result by airport_count desc;

final_count = limit sort 1;

--dump final_count;

store final_count into '/edureka_project/usecase4';
```

Explanation for usecase4:

- ➔ Loading the data and grouping fields by country
- ➔ Finding the count of each generated group by name
- ➔ Order the values by descending order
- ➔ Limit the order to 1 to get the first result and finally storing the output in HDFS.

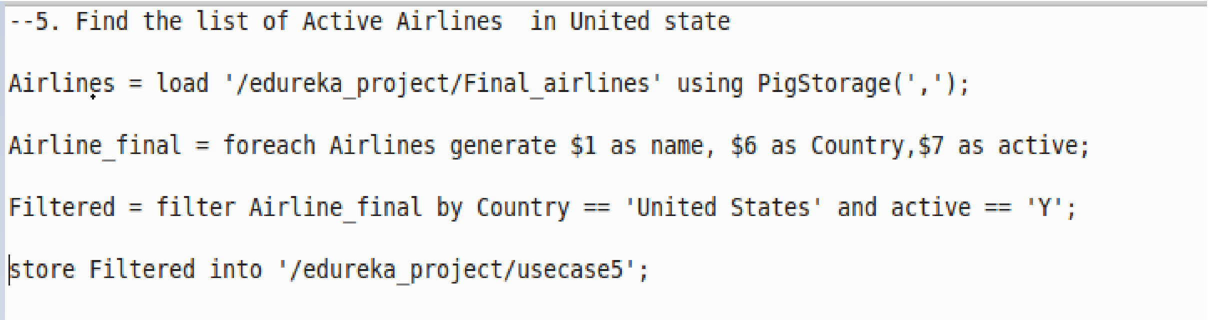
Usecase4 Output:



```
United States   1697
```

Use case 5:

In this use case we are finding the list of active airlines in United States.

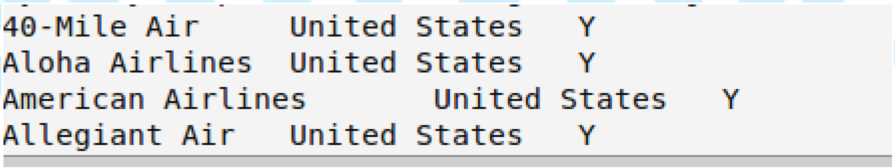


```
--5. Find the list of Active Airlines  in United state

Airlines = load '/edureka_project/Final_airlines' using PigStorage(',');

Airline_final = foreach Airlines generate $1 as name, $6 as Country,$7 as active;

Filtered = filter Airline_final by Country == 'United States' and active == 'Y';

store Filtered into '/edureka_project/usecase5';
```

Explanation for usecase5:

- ➔ Loading the data and finding the airlines with country name and active status
- ➔ Storing the final result into HDFS

Usecase5 Output:



```
40-Mile Air     United States   Y
Aloha Airlines  United States   Y
American Airlines       United States   Y
Allegiant Air   United States   Y
```