
SIX WEEKS SUMMER TRAINING REPORT

On

# (BIG DATA)

## Topic: Analysis of Airlines Data

Submitted by

Name: Suthar Lalitkumar Jagdish

Reg. No.: 12103479

Program Name: P132-L::B.Tech. (Computer Science & Engineering) [Lateral Entry]

Under the Guidance of

Allsoft Solution (IBM)

Mr Kuldeep Gupta

**School of Computer Science**

**Lovely Professional University, Phagwara**

**(7/June/2019 to 16/July/2019)**

# DECLARATION

I hereby declare that I have completed my six-week summer training at (AllSoft Solution (IBM)

I hereby undertake that the project undertaken by me is the genuine work of mine and is an authentic record of my own work carried out during our degree under the guidance of Mr.Kuldeep gupta from (07/06/2019) to (16/07/2019.

(Signature of student)

Name of Student: Suthar LalitKumar Jagdish

Registration no: 12103479

Date:11/07/2019

## ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher (**Mr Kuldeep Gupta**) as Well as our Director who gave me the golden chance to do this project on the vibrant field of Data analysis (**Analysis of Airlines Data**), which also helped me in doing a lot of Research about the changing scenario of the Airlines in the world and I came to know about so many new things I am really thankful to them.

- Suthar Lalitkumar Jagdish


Allsoft Solutions
An ISO 9001 : 2008 Certified Company

# PROJECT COMPLETION CERTIFICATE

**In recognition of the commitment to achieve professional excellence this is to certify that Ms./Mr.**

Suthar Lalit Kumar Jagdish

**has successfully completed an Industry-oriented project.**

| | |
|---|---|
| **Project Name** | Airlines Analysis |
| **Technologies Used** | Big Data Analytics |
| **Reference No.** | AIP/CEP2022/IN/41548 |
| **Training Date** | June 2022 – July 2022 |
| **Training Duration** | 6 Weeks |
| **Training Location** | Online Live Mode |

**Program Co-ordinator**
**Industry/Academic Alliance**

Business Partner IBM

**Director**
**Training and Development**
**Allsoft Solutions and Services**

BIG DATA - ANALYTICS | IoT | ORACLE | J2EE | PHP | CLOUD COMPUTING

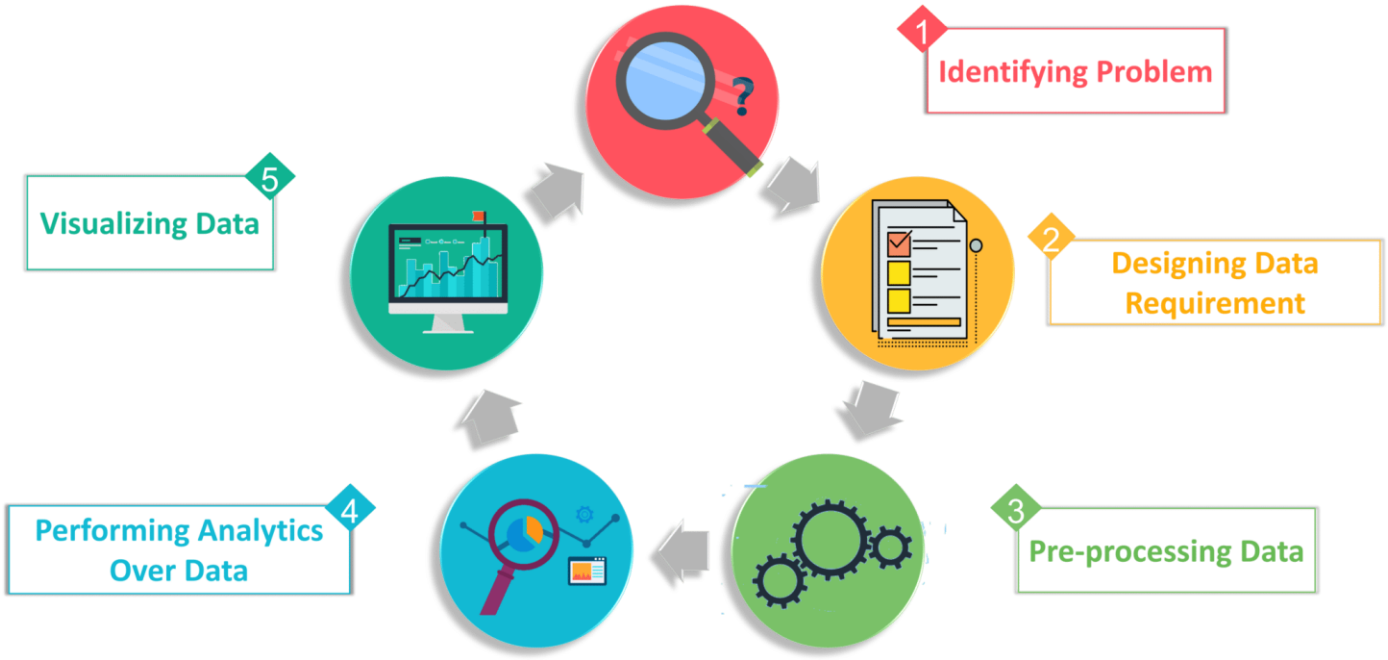
Table of Content:

# INTRODUCTION

## What is Big Data ?

**Big Data** is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

In most enterprise scenarios the volume of **data** is too **big** or it moves too fast or it exceeds current processing capacity




**Why this Big Data?**

• Over 2.5 Exabyte(2.5 billion gigabytes) of data is generated every day. Following are some of the sources of the huge volume of data:

• A typical, large stock exchange captures more than 1 TB of data every day. There are around 5 billion mobile phones (including 1.75 billion smart phones) in the world.

• YouTube users upload more than 48 hours of video every minute.

• Large social networks such as Twitter and Facebook capture more than 10 TB of data daily.

• There are more than 30 million networked sensors in the world.

**4 V's Of Big Data**

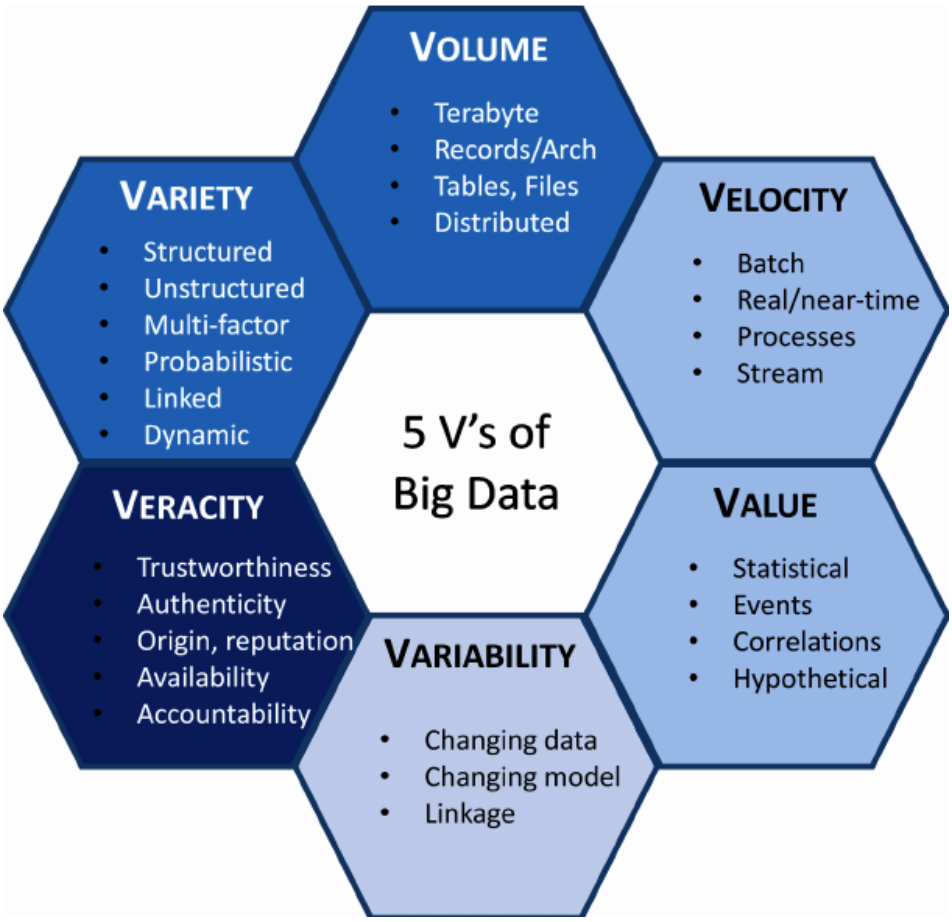
• **Volume**:- Big data is always large in volume. It actually doesn't have to be a certain number of petabytes to qualify. If your store of old data and new incoming data has gotten so large that you are having difficulty handling it, that's big data. Remember that it's going to keep getting bigger.

• **Velocity** :-Velocity or speed refers to how fast the data is coming in, but also to how fast we need to be able to analyze and utilize it. If we have one or more business processes that require real-time data analysis, we have a velocity challenge. Solving this issue might mean expanding our private cloud using a hybrid model that allows bursting for additional compute power as-needed for data analysis.

• **Variety**:- Variety points to the number of sources or incoming vectors leading to databases. That might be embedded sensor data, phone conversations, documents, video uploads or feeds, social media, and much more. Variety in data means variety in databases – we will almost certainly need to add a non-relational database

• **Veracity** :-Veracity is probably the toughest nut to crack. If we can't trust the data itself, the source of the data, or the processes we are using to identify which data points are important, we have a veracity problem. One of the biggest problems with big data is the tendency for errors to snowball. User entry errors, redundancy and corruption all affect the value of data. We must clean our existing data and put processes in place to reduce the accumulation of dirty data going forward.

Some extra V's in Big data




**TYPES OF BIG DATA**

- **Structured data:**

- Data which is represented in a tabular format
- E.g.: Databases

• **Semi-structured data:**

- Data which does not have a formal data model
- E.g.: XML files

• **Unstructured data:**

- Data which does not have a pre-defined data model
- E.g.: Text files

**Structured Data:**

• Structured data refers to kinds of data with a high level of organization, such as information in a relational database.

• When information is highly structured and predictable, search engines can more easily organize and display it in creative ways.

• Structured data markup is a text-based organization of data that is included in a file and served from the web.

**Semi-structured data:**

• It is a form of structured data that does not conform with the formal structure of data models associated with relational databases or other forms of data tables.

• But nonetheless contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data. Therefore, it is also known as self-describing structure.

• In semi-structured data, the entities belonging to the same class may have different attributes even though they are grouped together.

**Unstructured data:**

• It refers to information that either does not have a pre-defined data model or is not organized in a pre-defined manner.

• It is typically text-heavy, but may contain data such as dates, numbers, and facts as well.

• This results in irregularities and ambiguities that make it difficult to understand using traditional programs as compared to data stored in fielded form in databases or annotated (semantically tagged) in documents

**2. TOOLS AND WORKING ENVIRONMENT**:

Project will be made using Cloudera working in VMware.

Working environment will be the Cloudera version 5.13. It includes data manipulation and visualization libraries such as :

**1.HDFS**

**2.Sqoop**

**3.Hive**

**4.Pig**

**2.1 Hadoop Distributed File System**

HDFS is a distributed file system that handles large data sets running on commodity hardware. It is used to scale a single Apache Hadoop cluster to hundreds (and even thousands) of nodes. HDFS is one of the major components of Apache Hadoop, the others being MapReduce and YARN. HDFS should not be confused with or replaced by Apache HBase, which is a column-oriented non-relational database management system that sits on top of HDFS and can better support real-time data needs with its in-memory processing engine.

**THE GOALS OF HDFS**

• Fast recovery from hardware failures: Because one HDFS instance may consist of thousands of servers, failure of at least one server is inevitable. HDFS has been built to detect faults and automatically recover quickly.

• Access to streaming data: HDFS is intended more for batch processing versus interactive use, so the emphasis in the design is for high data throughput rates, which accommodate streaming access to data sets. • Accommodation of large data sets : HDFS accommodates applications that have data sets typically gigabytes to terabytes in size. HDFS provides high aggregate data bandwidth and can scale to hundreds of nodes in a single cluster

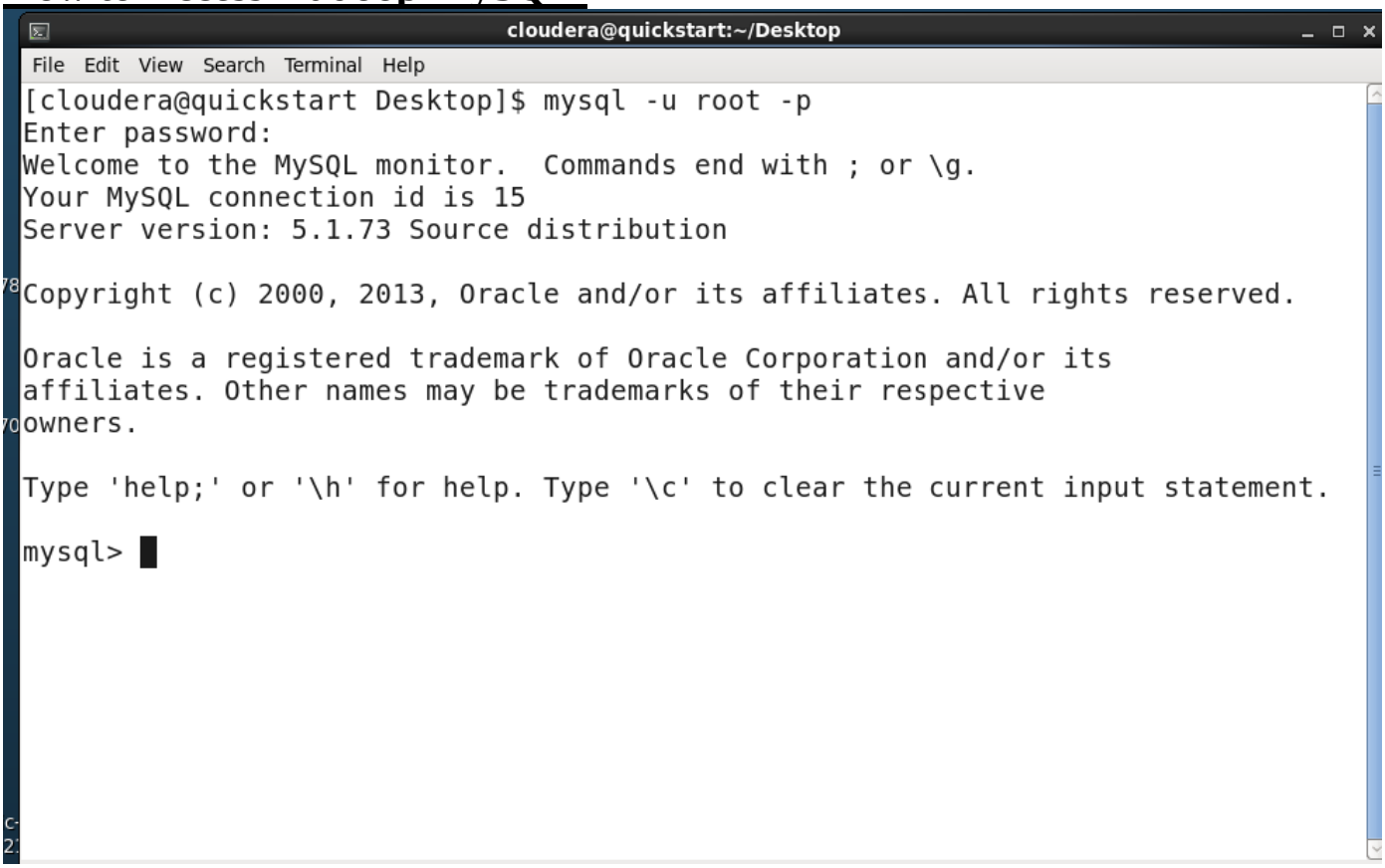
.• Portability: To facilitate adoption, HDFS is designed to be portable across multiple hardware platforms and to be compatible with a variety of underlying operating systems.

## AN EXAMPLE OF HDFS

- Consider a file that includes the phone numbers for everyone in the United States; the numbers for people with a last name starting with A might be stored on server 1,B on server 2, and so on.
- With Hadoop, pieces of this phonebook would be stored across the cluster, and to reconstruct the entire phonebook, your program would need the blocks from every server in the cluster.

- To ensure availability if and when a server fails, HDFS replicates these smaller pieces onto two additional servers by default. (The redundancy can be increased or decreased on a per-file basis or for a whole environment; for example, a development Hadoop cluster typically doesn't need any data redundancy.) This redundancy offers multiple benefits, the most obvious being higher availability.
- The redundancy also allows the Hadoop cluster to break up work into smaller chunks and run those jobs on all the servers in the cluster for better scalability. Finally, you gain the benefit of data locality, which is critical when working with large data sets.

## How to Access Hadoop MySQL




## 2.2 SQOOP

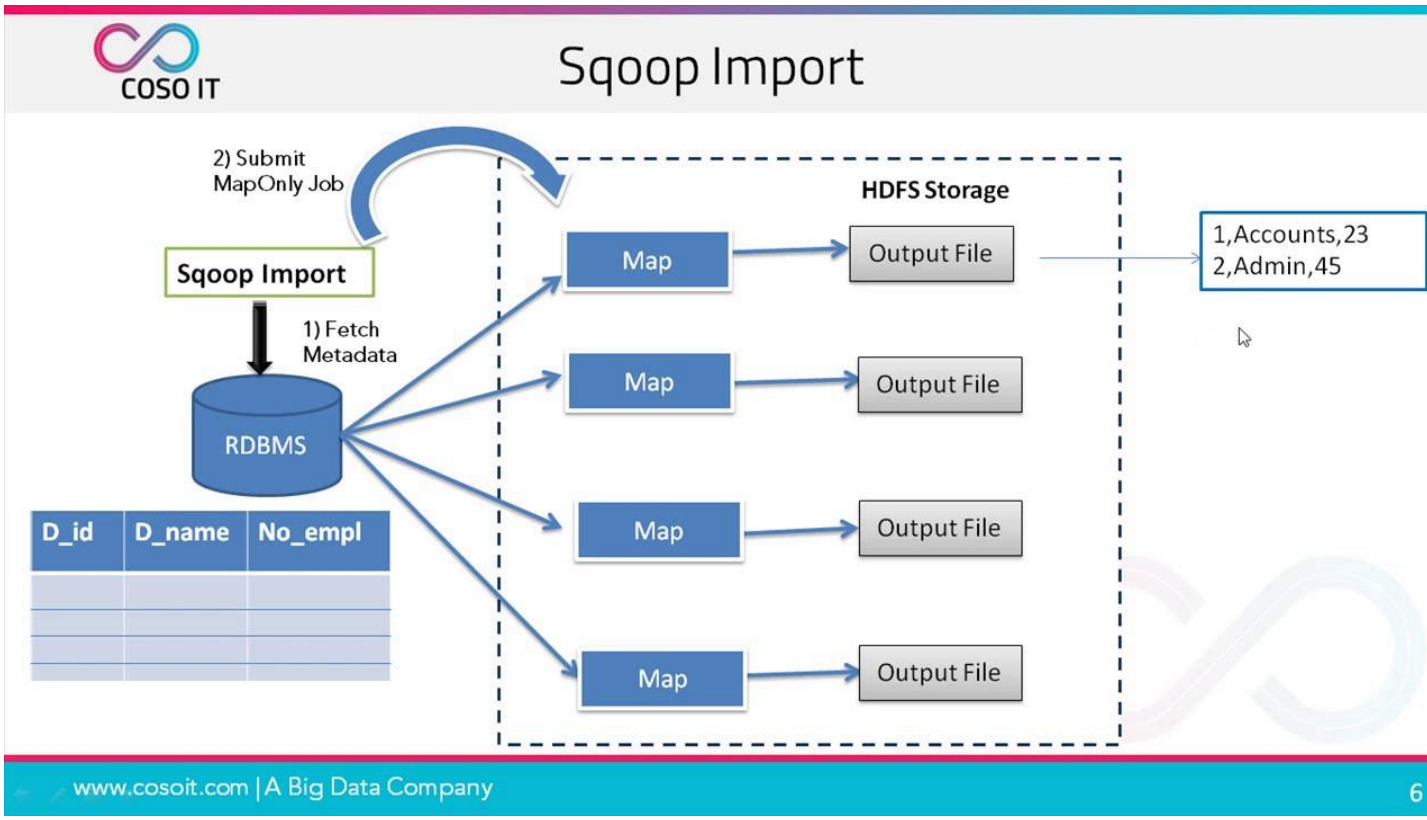
Apache Sqoop is **a big data tool for transferring data between Hadoop and relational database servers**. Sqoop is used to transfer data from RDBMS (relational database management system) like MySQL and Oracle to HDFS (Hadoop Distributed File System)

An example use case of Hadoop Sqoop is an enterprise that runs a nightly Sqoop import toload the day's data from a production transactional RDBMS into a Hive data warehouse for further analysis

## SQOOP ARCHITECTURE

All the existing **Database Management Systems** are designed with SQL standard in mind. However, each DBMS differs with respect to dialect to some extent. So, this difference poses challenges when it comes to data transfers across the systems. Sqoop Connectors are components which help overcome these challenges.

Data transfer between Sqoop Hadoop and external storage system is made possible with the help of Sqoop's connectors.




NoSQL stores (such as Couchbase). However, these connectors do not come with Sqoop bundle; those need to be downloaded separately and can be added easily to an existing Sqoop installation.

## WHY DO WE NEED SQOOP?

Analytical processing using Hadoop requires loading of huge amounts of data from diverse sources into Hadoop clusters. This process of bulk data load into Hadoop, from heterogeneous sources and then processing it, comes with a certain set of challenges. Maintaining and ensuring data consistency and ensuring efficient utilization of resources, are some factors to consider before selecting the right approach for data load.

**MAJOR ISSUES:**

1. **Data load using Scripts** The traditional approach of using scripts to load data is not suitable for bulk data load into Hadoop; this approach is inefficient and very time-consuming.
2. **Direct access to external data** via Map-Reduce application Providing direct access to the data residing at external systems(without loading into Hadoop) for map-reduce applications complicates these applications. So, this approach is not feasible.

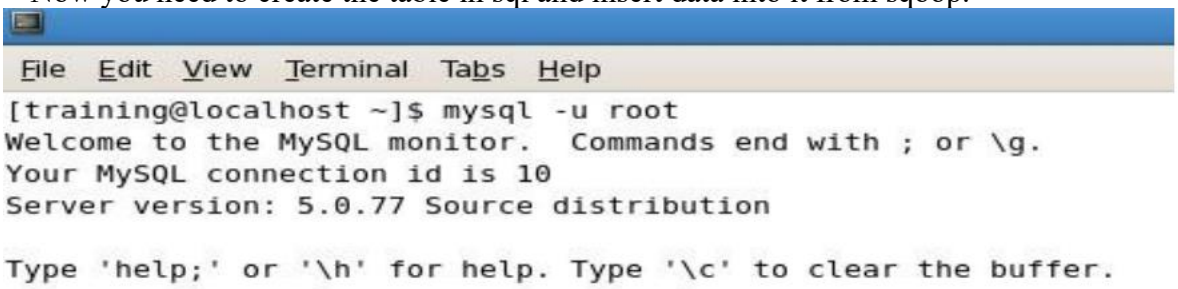
**Sqoop vs Flume vs HDFS in Hadoop**

| Sqoop | Flume | HDFS |
|---|---|---|
| Sqoop is used for importing data from structured data sources such as RDBMS. | Flume is used for moving bulk streaming data into HDFS. | HDFS is a distributed file system used by Hadoop ecosystem to store data. |
| Sqoop has a connector based architecture. Connectors know how to connect to the respective data source and fetch the data. | Flume has an agent-based architecture. Here, a code is written (which is called as 'agent') which takes care of fetching data. | HDFS has a distributed architecture where data is distributed across multiple data nodes. |

| | | |
|---|---|---|
| HDFS is a destination for data import using Sqoop. | Data flows to HDFS through zero or more channels. | HDFS is an ultimate destination for data storage. |
| Sqoop data load is not event-driven. | Flume data load can be driven by an event. | HDFS just stores data provided to it by whatsoever means. |
| In order to import data from structured data sources, one has to use Sqoop commands only, because its connectors know how to interact with structured data sources and fetch data from them. | In order to load streaming data such as tweets generated on Twitter or log files of a web server, Flume should be used. Flume agents are built for fetching streaming data. | HDFS has its own built-in shell commands to store data into it. HDFS cannot import streaming data |

**Load Data into Sqoop environments**.

- Open the terminal.
- Write the sqoop command on the terminal.
- This will open the shell.
- Now you need to create the table in sql and insert data into it from sqoop.




## Sqoop command to list databases :

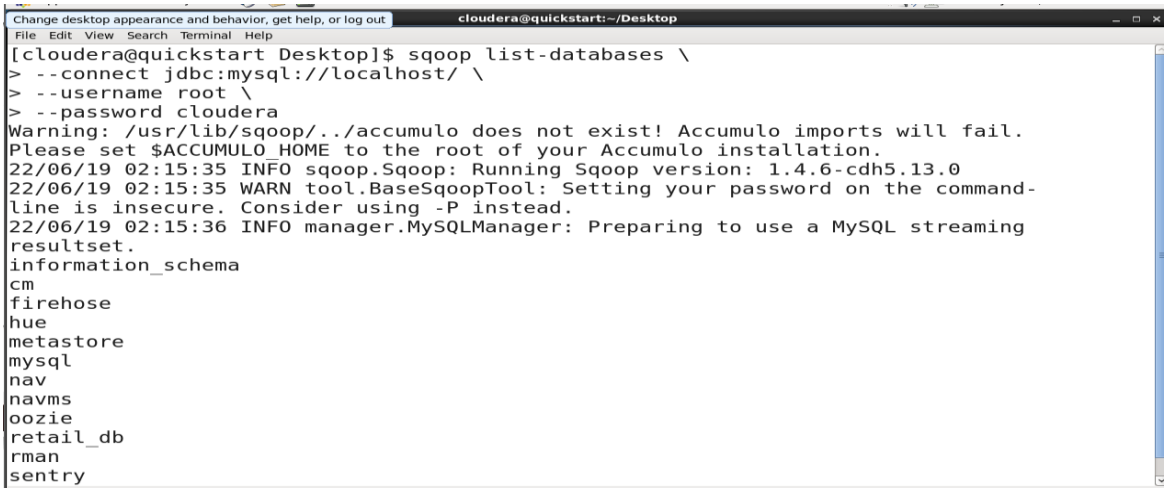
This command is used to list the total no. of databases in the system.

$ sqoop list-databases

--connect jdbc:mysql://localhost/

--username “root”

--password “root”



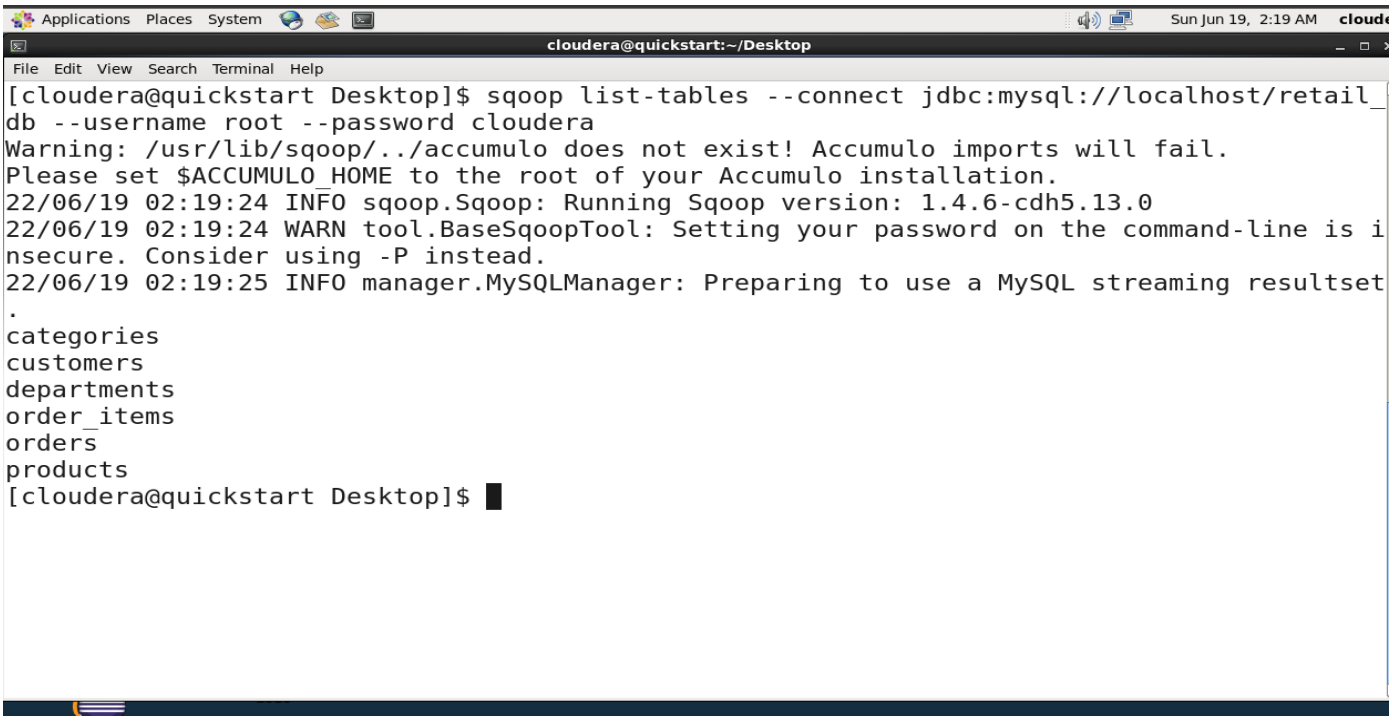

## Sqoop command to list tables :

The following command is used to show the total no. of tables with in a databases.

$ sqoop list-tables \

--connect jdbc:mysql://localhost/movielens \

--username root



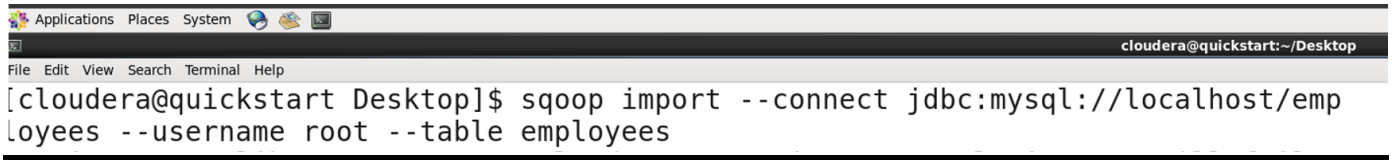

## Sqoop command to import a table:

The following command is used to import a table from a database.

$ sqoop import \ --connect jdbc:mysql://local host/mysql \

--username root \

-- table db




## Sqoop command to import all tables tables :

```
[cloudera@quickstart Desktop]$ sqoop import-all-tables --connect jdbc:mysql://localhost/IBM --username root --password cloudera -m 1
```

### 5) Sqoop command to import specific column from database:

The following command is used to import a specific column the the tables with in a database.

```
$ sqoop import \
--connect jdbc:mysql://local host/cs
-username root  \
--table adi \
--columns name
```

```
[training@localhost ~]$ sqoop import \
> --connect jdbc:mysql://localhost/cs \
> --username root \
> --table adi \
> --columns name
```

```
22/06/21 22:55:27 INFO mapreduce.ImportJobBase: Transferred 11 bytes in 20.7
533 seconds (0.53 bytes/sec)
22/06/21 22:55:27 INFO mapreduce.ImportJobBase: Retrieved 2 records.
```

### 6) Sqoop commnd export table:

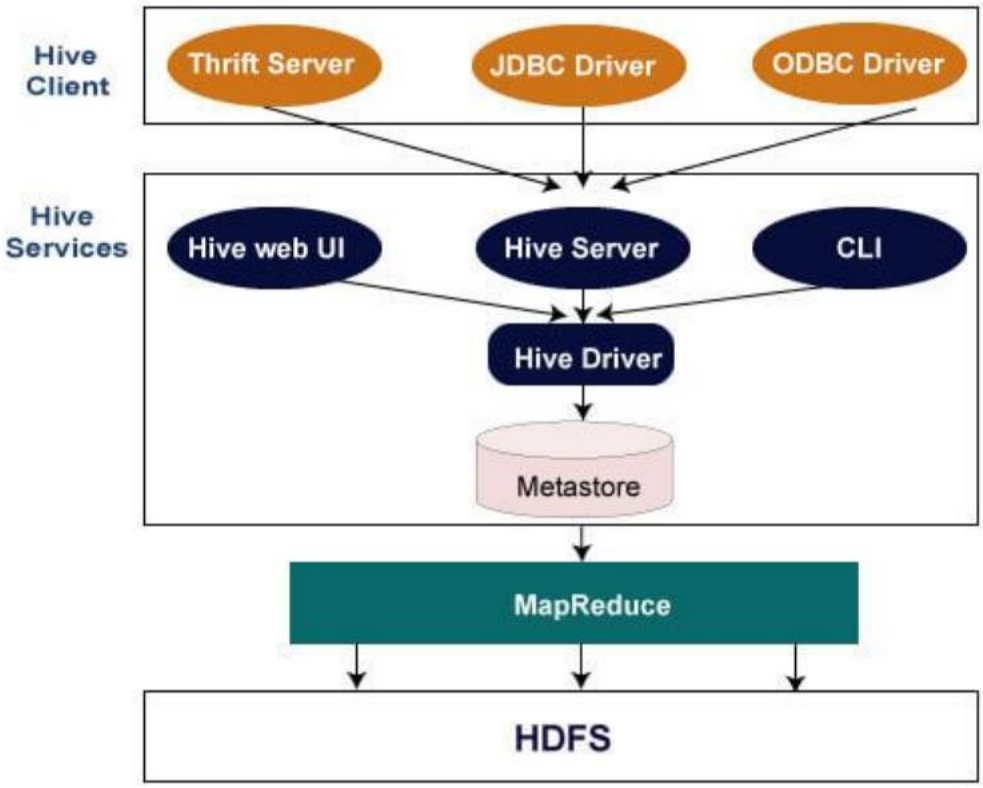
The following command is used to export the desired table from database to hdfs.

```
$ sqoop export \ --connect jdbc:mysql://local host/movielens \
--username root \ -table xyz \
 --fields-terminated-by "\t"
 --export-dir “/user/training/movies
```

**HIVE**

The Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The structure can be projected onto data already in storage." In other words, Hive is an open-source system that processes structured data in Hadoop, residing on top of the latter for summarizing Big Data, as well as facilitating analysis and queries.

**Architecture of Hive**



## Hive chiefly consists of three core parts:

• Hive Clients: Hive offers a variety of drivers designed for communication with different applications. For example, Hive provides Thrift clients for Thrift-based applications. These clients and drivers then communicate with the Hive server, which falls under Hive services.

• Hive Services: Hive services perform client interactions with Hive. For example, if a client wants to perform a query, it must talk with Hive services.

• Hive Storage and Computing: Hive services such as file system, job client, and meta store then communicates with Hive storage and stores things like metadata table information and query results.

## These are Hive's chief characteristics:

• Hive is designed for querying and managing only structured data stored in tables • Hive is scalable, fast, and uses familiar concepts

• Schema gets stored in a database, while processed data goes into a Hadoop Distributed File System (HDFS)

• Tables and databases get created first; then data gets loaded into the proper tables

• Hive supports four file formats: ORC, SEQUENCEFILE, RCFILE (Record Columnar File), and TEXTFILE

• Hive uses an SQL-inspired language, sparing the user from dealing with the complexity of MapReduce programming. It makes learning more accessible by utilizing familiar concepts found in relational databases, such as columns, tables, rows, and schema, etc.

• The most significant difference between the Hive Query Language (HQL) and SQL is that Hive executes queries on Hadoop's infrastructure instead of on a traditional database

• Since Hadoop's programming works on flat files, Hive uses directory structures to "partition" data, improving performance on specific queries

• Hive supports partition and buckets for fast and simple data retrieval • Hive supports custom user-defined functions (UDF) for tasks like data cleansing and filtering. Hive UDFs can be defined according to programmers' requirements

## Load Data into hive environments.

• Open the terminal.

• Write the Hive command on the terminal.

• This will open the hive shell.

• Now you need to create the table in hive and load data into it from HDFS.

## HIVE COMMANDS

### 1) Create an internal table:

The following command is used to create an internal table in hive.

create table internal (name string ,id int); Row format delimited Fields terminated by '\t';

```
hive> create table internal(name string,id int)
    > row format delimited
    > fields terminated by "\t";
OK
Time taken: 0.794 seconds
```

### 2) Create an external table:

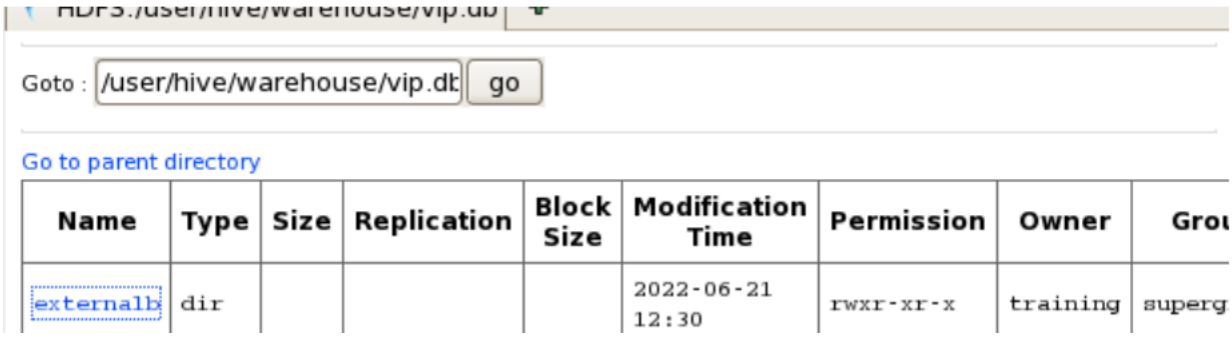
The following command is used to create an external table in hive.

create external table externalb(name string ,id int)

Row format delimited Fields terminated by '\t

' LOCATION '/user/guru99hive/external;

```
Time taken: 4.255 seconds
hive> create external table externalb(name string,id int)
    > row format delimited
    > fields terminated by "\t";
OK
Time taken: 1.045 seconds
```



## 3) Load data in table from local system:

The following command is used to copy data from a file in local system in Hadoop to a table in hive.

load data local inpath ‘dir_path' into table
;

```
Time taken: 0.794 seconds
hive> load data local inpath "/home/training/vips.txt" into table internal;
Copying data from file:/home/training/vips.txt
Copying file: file:/home/training/vips.txt
Loading data to table vip.internal
OK
Time taken: 0.672 seconds
hive> select *from internal;
OK
vpin,1234       NULL
gourav.234      NULL
rahul,345       NULL
Time taken: 3.961 seconds
hive>
```

[sqoopexport.png]

## 4) Load data in table from HDFS:

The following command is used to copy data from a file in hdfs Hadoop to a table in hive.

load data inpath ‘dir_path' into table student;

# Set: - Analysis of Airlines Data

This project is to analyses a data set related **Analysis of Airlines Data** in the country. The proposed method is made by considering following scenario under consideration An Airlines has huge amount of data related to number of flights, data and time of arrival and dispatch, flight routes, No. of Airliness operating in each country, list of active airlines in each country. The problem they faced till now it’s, they have ability to analyze limited data from databases. The Proposed model intension is to develop a model for the airline data to provide platform for new analytics based on the following queries.

### 1.1. Purpose

The purpose of data analytics in aviation is to examine the vast amount of data generated daily and provide useful information to airlines, Airliness and other aviation stakeholders so that they can improve their operational planning and execution, as well as any related products and services..

### 1.2. Project Scope

1. It could be used to anticipate analysis the airline.Airlines and route in different regions of various countries.
2. It is valid for descriptive analysis and sneak peak of predictive analysis.
3. It is valid for particular set of compatible attributes.

**GOALS: -**

a) list of Airliness operating in the country India
b) list of airlines having zero stops
c) list of airlines operating with code share
d) which country has highest Airliness
e) list of active airlines in United State

### 1.3. References

https://www.kaggle.com http://www.data.gov.com

### 1.4 About Data Set

This data set contains information **related to explore detailed analysis on airline data sets such as listing Airliness operating in the India, list of airlines having** in the world. It contains airlineid ,Airlinesid,name,cityid,country, Source Airlines ID etc . This POC contains the Hive and Pig queries to manipulate and analyze the data. The data set contains approx. 59k entries.

### 1.5 The details of the data set are given below:

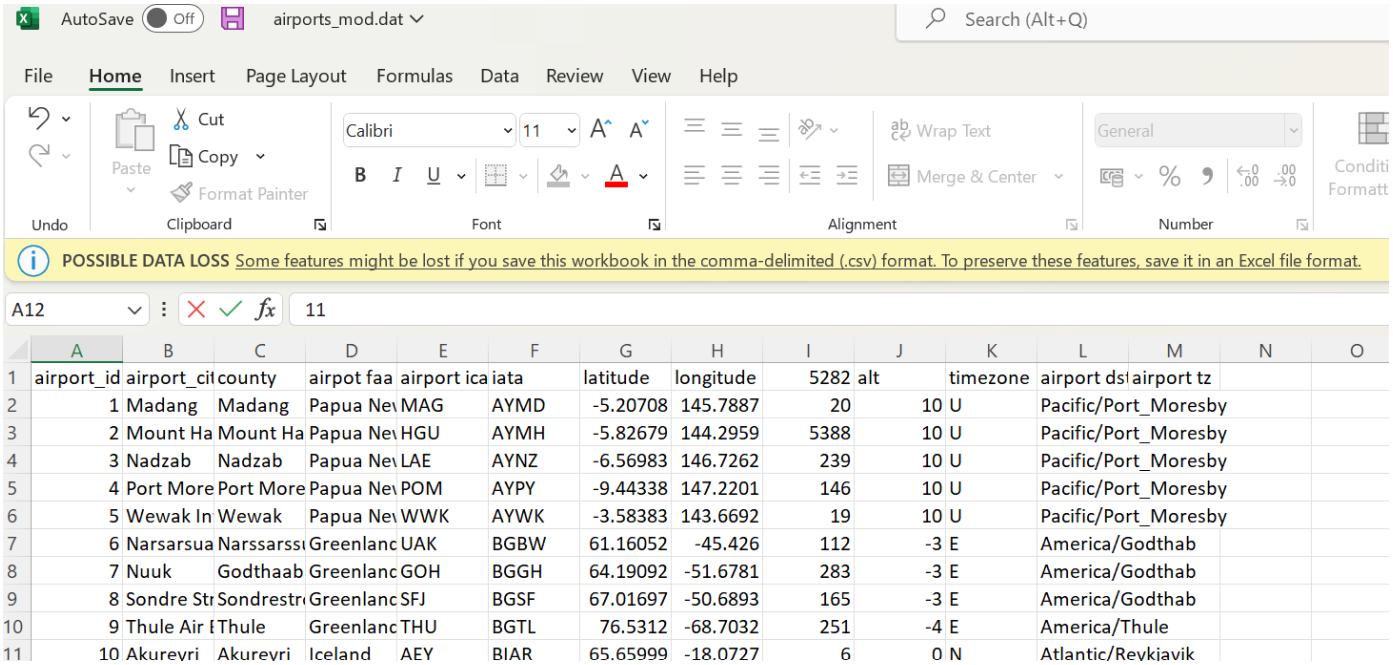
**Format:**

The data set contains the following information about analysis Airlines,airline and route

database , they are divided by ‘,’ in the data file:

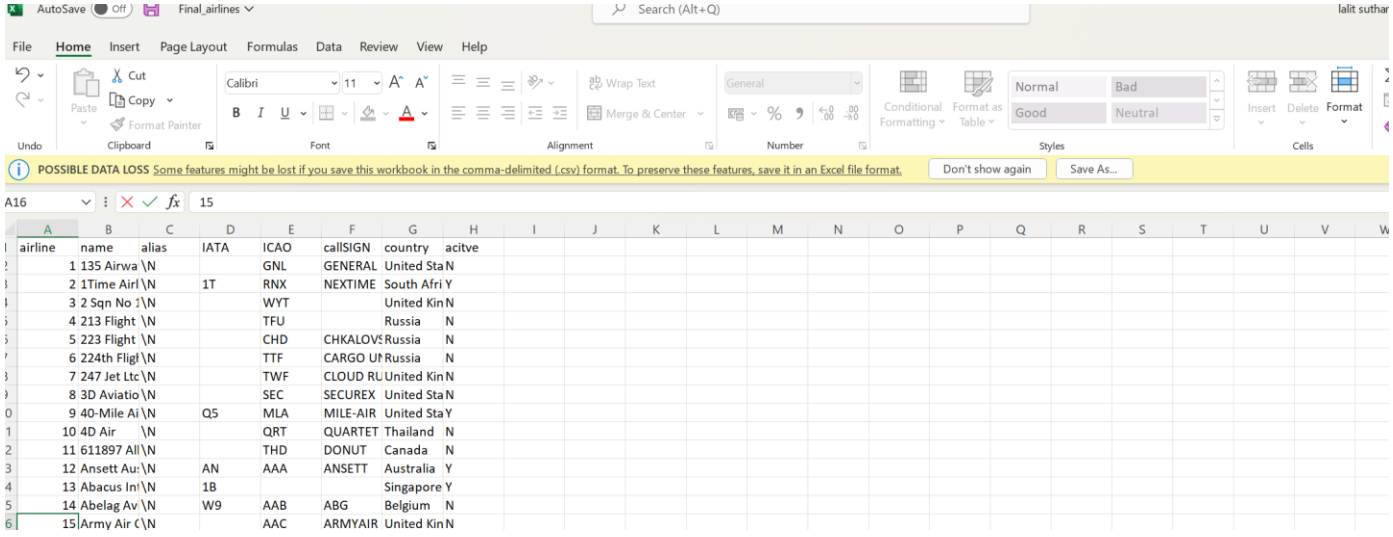
**Screenshot:_**

Airlines**.csv**



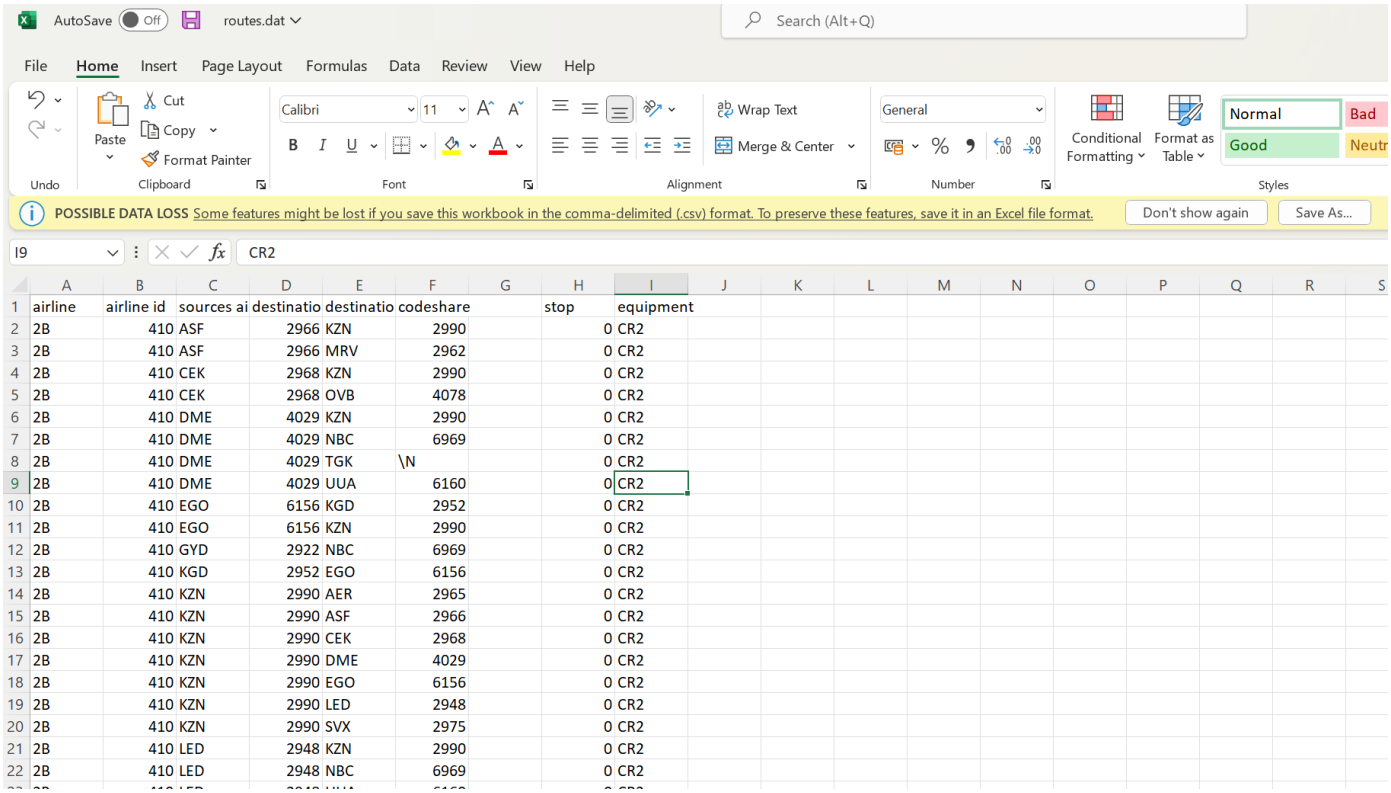

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | airport_id | airport_cit | county | airpot faa | airport ica | iata | latitude | longitude | 5282 | alt | timezone | airport ds | airport tz | | |
| 2 | 1 | Madang | Madang | Papua Nev | MAG | AYMD | -5.20708 | 145.7887 | 20 | 10 | U | Pacific/Port_Moresby | | | |
| 3 | 2 | Mount Ha | Mount Ha | Papua Nev | HGU | AYMH | -5.82679 | 144.2959 | 5388 | 10 | U | Pacific/Port_Moresby | | | |
| 4 | 3 | Nadzab | Nadzab | Papua Nev | LAE | AYNZ | -6.56983 | 146.7262 | 239 | 10 | U | Pacific/Port_Moresby | | | |
| 5 | 4 | Port More | Port More | Papua Nev | POM | AYPY | -9.44338 | 147.2201 | 146 | 10 | U | Pacific/Port_Moresby | | | |
| 6 | 5 | Wewak In | Wewak | Papua Nev | WWK | AYWK | -3.58383 | 143.6692 | 19 | 10 | U | Pacific/Port_Moresby | | | |
| 7 | 6 | Narsarsua | Narssarss | Greenlanc | UAK | BGBW | 61.16052 | -45.426 | 112 | -3 | E | America/Godthab | | | |
| 8 | 7 | Nuuk | Godthaab | Greenlanc | GOH | BGGH | 64.19092 | -51.6781 | 283 | -3 | E | America/Godthab | | | |
| 9 | 8 | Sondre Str | Sondrestr | Greenlanc | SFJ | BGSF | 67.01697 | -50.6893 | 165 | -3 | E | America/Godthab | | | |
| 10 | 9 | Thule Air I | Thule | Greenlanc | THU | BGTL | 76.5312 | -68.7032 | 251 | -4 | E | America/Thule | | | |
| 11 | 10 | Akurevri | Akurevri | Iceland | AEY | BIAR | 65.65999 | -18.0727 | 6 | 0 | N | Atlantic/Revkjavik | | | |

## Airline.csv




| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| | airline | name | alias | IATA | ICAO | callSIGN | country | acitve |
| | 1 | 135 Airwa | \N | | GNL | GENERAL | United Sta | N |
| | 2 | 1Time Airl | \N | 1T | RNX | NEXTIME | South Afri | Y |
| | 3 | 2 Sqn No 1 | \N | | WYT | | United Kin | N |
| | 4 | 213 Flight | \N | | TFU | | Russia | N |
| | 5 | 223 Flight | \N | | CHD | CHKALOVS | Russia | N |
| | 6 | 224th Flig | \N | | TTF | CARGO UI | Russia | N |
| | 7 | 247 Jet Ltc | \N | | TWF | CLOUD RU | United Kin | N |
| | 8 | 3D Aviatio | \N | | SEC | SECUREX | United Sta | N |
| | 9 | 40-Mile Ai | \N | Q5 | MLA | MILE-AIR | United Sta | Y |
| | 10 | 4D Air | \N | | QRT | QUARTET | Thailand | N |
| | 11 | 611897 All | \N | | THD | DONUT | Canada | N |
| | 12 | Ansett Au | \N | AN | AAA | ANSETT | Australia | Y |
| | 13 | Abacus Int | \N | 1B | | | Singapore | Y |
| | 14 | Abelag Av | \N | W9 | AAB | ABG | Belgium | N |
| | 15 | Army Air C | \N | | AAC | ARMYAIR | United Kin | N |

## Route.csv




| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | airline | airline id | sources ai | destinatio | destinatio | codeshare | | stop | equipment | | | | | | | | | | |
| 2 | 2B | 410 | ASF | 2966 | KZN | 2990 | | 0 | CR2 | | | | | | | | | | |
| 3 | 2B | 410 | ASF | 2966 | MRV | 2962 | | 0 | CR2 | | | | | | | | | | |
| 4 | 2B | 410 | CEK | 2968 | KZN | 2990 | | 0 | CR2 | | | | | | | | | | |
| 5 | 2B | 410 | CEK | 2968 | OVB | 4078 | | 0 | CR2 | | | | | | | | | | |
| 6 | 2B | 410 | DME | 4029 | KZN | 2990 | | 0 | CR2 | | | | | | | | | | |
| 7 | 2B | 410 | DME | 4029 | NBC | 6969 | | 0 | CR2 | | | | | | | | | | |
| 8 | 2B | 410 | DME | 4029 | TGK | \N | | 0 | CR2 | | | | | | | | | | |
| 9 | 2B | 410 | DME | 4029 | UUA | 6160 | | 0 | CR2 | | | | | | | | | | |
| 10 | 2B | 410 | EGO | 6156 | KGD | 2952 | | 0 | CR2 | | | | | | | | | | |
| 11 | 2B | 410 | EGO | 6156 | KZN | 2990 | | 0 | CR2 | | | | | | | | | | |
| 12 | 2B | 410 | GYD | 2922 | NBC | 6969 | | 0 | CR2 | | | | | | | | | | |
| 13 | 2B | 410 | KGD | 2952 | EGO | 6156 | | 0 | CR2 | | | | | | | | | | |
| 14 | 2B | 410 | KZN | 2990 | AER | 2965 | | 0 | CR2 | | | | | | | | | | |
| 15 | 2B | 410 | KZN | 2990 | ASF | 2966 | | 0 | CR2 | | | | | | | | | | |
| 16 | 2B | 410 | KZN | 2990 | CEK | 2968 | | 0 | CR2 | | | | | | | | | | |
| 17 | 2B | 410 | KZN | 2990 | DME | 4029 | | 0 | CR2 | | | | | | | | | | |
| 18 | 2B | 410 | KZN | 2990 | EGO | 6156 | | 0 | CR2 | | | | | | | | | | |
| 19 | 2B | 410 | KZN | 2990 | LED | 2948 | | 0 | CR2 | | | | | | | | | | |
| 20 | 2B | 410 | KZN | 2990 | SVX | 2975 | | 0 | CR2 | | | | | | | | | | |
| 21 | 2B | 410 | LED | 2948 | KZN | 2990 | | 0 | CR2 | | | | | | | | | | |
| 22 | 2B | 410 | LED | 2948 | NBC | 6969 | | 0 | CR2 | | | | | | | | | | |

# Attribute Information

**Table 1: Airport Data Set [6]**

| Attribute | Description |
|---|---|
| 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" |
| ICAO | 4-letter ICAO code. |
| 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 | Timezone in "tz" (Olson) format, eg. "America/Los_Angeles". zone |

**Table 2: Airline Data Set [6]**

| Attribute | Description |
|---|---|
| Airline | Unique OpenFlights identifier for this airline. ID |
| 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". |

**Table 3: Route Data Set [6]**

| Attribute | Description |
|---|---|
| Airline | 2-letter (IATA) or 3-letter(ICAO) code of the airline. |
| Airline ID | Unique OpenFlights identifier for airline |
| Source airport | 3-letter (IATA) or 4-letter (ICAO) code of the source airport |
| Source airport ID | Unique OpenFlights identifier for source airport |
| Destination airport | 3-letter (IATA) or 4-letter (ICAO) code of the destination airport. |
| Destination airport ID | Unique OpenFlights identifier for destination 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 |

## Queries

- **Find list of Airliness operating in the Country India;**
- **Find the list of Airlines having zero stops**
- **List of Airlines operating with code share**
- **Which country (or) territory having highest Airliness**
- **Find the list of Active Airlines in United state**

# Analysis of Airlines Data using Hadoop-Hive

**1.creating table Airlines for Airliness_mod.dat**:

create table Airliness (Airlines_id int,Airlines_name string,Airlines_city string,Airlines_country string,Airlines_faa string,Airlines_icao string,Airlines_lat double,Airlines_long double,Airlines_alt double,Airlines_timezone double,Airlines_dst string,Airlines_tz string) row format delimited fields terminated by ',';

```
hive> create table airports(airport_id int,airport_name string,airport_city string,airport_country string,airport_faa string,airport_icao string,airport_lat do
uble,airport_long double,airport_alt double,airport_timezone double,airport_dst string,airport_tz string) row format delimited fields terminated by ',';
OK
Time taken: 0.106 seconds
```

2.**creating table finalairlines for Final_airlines** :

create table final_airlines (airlineID string,airline_name string, airline_alias string, airline_iata string, airline_icao string,callsign string,territory string, active string) row format delimited fields terminated by ',';

```
hive> create table final_airlines (airlineID string,airline_name string, airline_alias string, airline_iata string, airline_icao string,callsign string,territo
ry string, active string) row format delimited fields terminated by ',';
OK
Time taken: 0.042 seconds
```

**3.Creating table route for routes.dat**:

create table routes (route_iata string,route_airid int,route_source_iata string,route_source_airid int,route_des_iata string,route_des_airid int,route_codeshare string,route_stops int,route_equip string) row format delimited fields terminated by ',';

```
hive> create table routes (route_iata string,route_airid int,route_source_iata string,route_source_airid int,route_des_iata string,route_des_airid int,route_co
deshare string,route_stops int,route_equip string) row format delimited fields terminated by ',';
OK
Time taken: 0.042 seconds
```

**4.loading data into Airlines table**

```
hive> load data inpath 'data/Air-datasets/airports_mod.dat' into table airports;
Loading data to table kausikchat.airports
Table kausikchat.airports stats: [numFiles=1, totalSize=739515]
OK
Time taken: 0.441 seconds
```

**5.loading data into final airlines table**

```
load data inpath 'data/Air-datasets/Final_airlines' into table final_airlines;
```

```
hive> load data inpath 'data/Air-datasets/Final_airlines' into table final_airlines;
Loading data to table kausikchat.final_airlines
Table kausikchat.final_airlines stats: [numFiles=1, totalSize=316243]
OK
Time taken: 0.227 seconds
```

**6.loading data into route table**

```
load data inpath 'data/Air-datasets/routes.dat' into table routes
```

```
hive> load data inpath 'data/Air-datasets/routes.dat' into table routes;
Loading data to table kausikchat.routes
Table kausikchat.routes stats: [numFiles=1, totalSize=2375505]
OK
Time taken: 0.208 seconds
```

- **Problem solution**:

**a.Find list of Airliness operating in the Country India**;

create table india_opert_Airlines as  select * from Airliness where Airlines_country LIKE '%India%';

```
hive> create table india_opert_airport as  select * from airports where airport_country LIKE '%India%';
Query ID = edureka_263261_20190103060101_e7d8d66b-2b7f-47ae-a0a2-bfd60324a9db
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1528714825862_73019, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73019/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73019
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-01-03 06:01:19,469 Stage-1 map = 0%,  reduce = 0%
2019-01-03 06:01:24,606 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.88 sec
MapReduce Total cumulative CPU time: 2 seconds 880 msec
Ended Job = job_1528714825862_73019
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/.hive-staging_hive_2019-01-03_06-01-09_321_9099508579175469760-1/-ext-10001
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/india_opert_airport
Table kausikchat.india_opert_airport stats: [numFiles=1, numRows=141, totalSize=11853, rawDataSize=11712]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.88 sec   HDFS Read: 744271 HDFS Write: 11943 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 880 msec
OK
Time taken: 16.588 seconds
```

select * from india_opert_Airlines limit 10; (to show only first 10 values due to large nos of data)

```
hive> select * from india_opert_airport limit 10;
OK
895    Diego Garcia Nsf       Diego Garcia Island    British Indian Ocean Territory        FJDG   -7.313267      72.411089      9.0    6.0    U    I
ndian/Chagos
2994    Ahmedabad      Ahmedabad      India   AMD    VAAH    23.077242      72.63465       189.0  5.5    N      Asia/Calcutta
2995    Akola  Akola   India   AKD    VAAK    20.699006      77.058628      999.0  5.5    N      Asia/Calcutta
2996    Aurangabad     Aurangabad     India   IXU    VAAU    19.862728      75.398114      1911.0 5.5    N      Asia/Calcutta
2997    Chhatrapati Shivaji Intl       Mumbai  India   BOM    VABB    19.088686      72.867919      37.0   5.5    N      Asia/Calcutta
2998    Bilaspur       Bilaspur       India   PAB    VABI    21.9884 82.110983      899.0  5.5    N      Asia/Calcutta
2999    Bhuj   Bhuj    India   BHJ    VABJ    23.287828      69.670147      268.0  5.5    N      Asia/Calcutta
3000    Belgaum Belgaum India   IXG    VABM    15.859286      74.618292      2487.0 5.5    N      Asia/Calcutta
3001    Vadodara       Baroda  India   BDQ    VABO    22.336164      73.226289      129.0  5.5    N      Asia/Calcutta
3002    Bhopal  Bhopal  India   BHO    VABP    23.287467      77.337375      1719.0 5.5    N      Asia/Calcutta
Time taken: 0.053 seconds, Fetched: 10 row(s)
```

## b.Find the list of Airlines having zero stops

create table stop as select * from route where route_stops LIKE '%0';

```
hive> create table stop as select * from routes where route_stops LIKE '%0';
Query ID = edureka_263261_20190103060606_48a8ad75-6dd3-4e89-813c-353e34c0d0e4
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1528714825862_73020, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73020/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73020
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-01-03 06:06:15,657 Stage-1 map = 0%,  reduce = 0%
2019-01-03 06:06:20,770 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.47 sec
MapReduce Total cumulative CPU time: 3 seconds 470 msec
Ended Job = job_1528714825862_73020
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/.hive-staging_hive_2019-01-03_06-06-11_812_961204894951092203-1/-ext-10001
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/stop
Table kausikchat.stop stats: [numFiles=1, numRows=67652, totalSize=2307487, rawDataSize=2239835]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.47 sec   HDFS Read: 2379802 HDFS Write: 2307566 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 470 msec
OK
Time taken: 10.181 seconds
```

select * from stop limit 10;

```
hive> select * from stop limit 10;
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
Time taken: 0.042 seconds, Fetched: 10 row(s)
```

## c. List of Airlines operating with code share

create table codeshare_1 as select * from routes where routes_codeshare LIKE'%Y%';

```
hive> create table codeshare_1 as select * from routes where routes_codeshare LIKE '%Y%';
FAILED: SemanticException [Error 10004]: Line 1:55 Invalid table alias or column reference 'routes_codeshare': (possible column names are: route_iata, route_ai
rid, route_source_iata, route_source_airid, route_des_iata, route_des_airid, route_codeshare, route_stops, route_equip)
hive> create table codeshare_1 as select * from routes where route_codeshare LIKE '%Y%';
Query ID = edureka_263261_20190103060909_5ce15d89-0c66-41c7-9e1d-cc6817b3fac3
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1528714825862_73021, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73021/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73021
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-01-03 06:09:07,396 Stage-1 map = 0%,  reduce = 0%
2019-01-03 06:09:12,494 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.93 sec
MapReduce Total cumulative CPU time: 2 seconds 930 msec
Ended Job = job_1528714825862_73021
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/.hive-staging_hive_2019-01-03_06-09-02_627_2831242062070087860-1/-ext-10001
Moving data to: hdfs://nameservice1/user/hive/warehouse/kausikchat.db/codeshare_1
Table kausikchat.codeshare_1 stats: [numFiles=1, numRows=14597, totalSize=511399, rawDataSize=496802]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.93 sec   HDFS Read: 2379813 HDFS Write: 511484 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 930 msec
OK
Time taken: 12.679 seconds
```

select * from codeshare_1 limit 10;

```
Time taken: 12.079 seconds
hive> select * from codeshare_1 limit 10;
OK
2P      897     GES     6011    MNL     2397    Y       0       320
2P      897     MNL     2397    GES     6011    Y       0       320
4M      3201    DFW     3670    EZE     3988    Y       0       777
4M      3201    EZE     3988    DFW     3670    Y       0       777
4M      3201    EZE     3988    JFK     3797    Y       0       777
4M      3201    JFK     3797    EZE     3988    Y       0       777
5N      503     ARH     4362    CSH     6110    Y       0       AN4
5N      503     ARH     4362    MMK     2949    Y       0       AN4
5N      503     ARH     4362    USK     4369    Y       0       AN4
5N      503     CSH     6110    ARH     4362    Y       0       AN4
Time taken: 0.042 seconds, Fetched: 10 row(s)
```

## d. Which country (or) territory  having highest Airliness

select Airlines_country,count(*) as cnt from Airliness group by Airlines_country ORDER BY cnt DESC;

```
hive> select airport_country,count(*) as cnt from airports group by airport_country ORDER BY cnt DESC;
Query ID = edureka_263261_20190103061212_326df935-c79d-4d2c-bd8a-6458e622c70f
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 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_1528714825862_73022, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73022/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73022
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-01-03 06:12:49,586 Stage-1 map = 0%,  reduce = 0%
2019-01-03 06:12:53,684 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.89 sec
2019-01-03 06:12:58,779 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.61 sec
MapReduce Total cumulative CPU time: 3 seconds 610 msec
Ended Job = job_1528714825862_73022
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_1528714825862_73023, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73023/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73023
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2019-01-03 06:13:04,554 Stage-2 map = 0%,  reduce = 0%
2019-01-03 06:13:08,640 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.52 sec
2019-01-03 06:13:13,730 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.42 sec
MapReduce Total cumulative CPU time: 3 seconds 420 msec
Ended Job = job_1528714825862_73023
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.61 sec   HDFS Read: 746937 HDFS Write: 7092 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.42 sec   HDFS Read: 11786 HDFS Write: 3151 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 30 msec
OK
United States   1697
Canada  435
Germany 321
Australia       263
Russia  249
```

## E. Find the list of Active Airlines in United state

select *  from final_airlines where territory like '%United States%' AND active like '%Y%';

```
hive> select *  from final_airlines where territory like '%United States%' AND active like '%Y%';
Query ID = edureka_263261_20190103061717_e1425815-0b15-4f2e-a68b-995430c6424d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1528714825862_73024, Tracking URL = http://ip-20-0-21-161.ec2.internal:8088/proxy/application_1528714825862_73024/
Kill Command = /opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1.p0.4/lib/hadoop/bin/hadoop job  -kill job_1528714825862_73024
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2019-01-03 06:17:15,036 Stage-1 map = 0%,  reduce = 0%
2019-01-03 06:17:20,135 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.82 sec
MapReduce Total cumulative CPU time: 2 seconds 820 msec
Ended Job = job_1528714825862_73024
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.82 sec   HDFS Read: 320673 HDFS Write: 7886 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 820 msec
OK
10      40-Mile Air     NULL    Q5      MLA     MILE-AIR        United States   Y
22      Aloha Airlines  NULL    AQ      AAH     ALOHA   United States   Y
24      American Airlines       NULL    AA      AAL     AMERICAN        United States   Y
35      Allegiant Air   NULL    G4      AAY     ALLEGIANT       United States   Y
109     Alaska Central Express  NULL    KO      AER     ACE AIR United States   Y
149     Air Cargo Carriers      NULL    2Q      SNC     NIGHT CARGO     United States   Y
210     Airlift International   NULL            AIR     AIRLIFT United States   Y
281     America West Airlines   NULL    HP      AWE     CACTUS  United States   Y
282     Air Wisconsin   NULL    ZW      AWI     AIR WISCONSIN   United States   Y
287     Allegheny Commuter Airlines     NULL            ALO     ALLEGHENY       United States   Y
295     Air Sunshine    NULL            RSI     AIR SUNSHINE    United States   Y
315     ATA Airlines    NULL            AMT     AMTRAN  United States   Y
397     Arrow Air       NULL    JW      APW     BIG A   United States   Y
452     Atlantic Southeast Airlines     NULL    EV      ASQ     ACEY    United States   Y
659     American Eagle Airlines NULL    MQ      EGF     EAGLE FLIGHT    United States   Y
792     Access Air      NULL    ZA      CYD     CYCLONE United States   Y
882     Air Florida     NULL    QH      FLZ     AIR FLORIDA     United States   Y
928     Atlas Air       NULL    5Y      GTI     GIANT   United States   Y
1316    AirTran Airways NULL    FL      TRS     CITRUS  United States   Y
1442    Bemidji Airlines        NULL    CH      BMJ     BEMIDJI United States   Y
1472    Bering Air      NULL    8E      BRG     BERING AIR      United States   Y
1629    Cape Air        NULL    9K      KAP     CAIR    United States   Y
1739    Chautauqua Airlines     NULL    RP      CHQ     CHAUTAUQUA      United States   Y
1814    Coastal Air     NULL    DQ              U.S. Virgin Islands     United States   Y
1821    Colgan Air      NULL    9L      CJC     COLGAN  United States   Y
1828    Comair  NULL    OH      COM     COMAIR  United States   Y
1843    CommutAir       NULL    C5      UCA     COMMUTAIR       United States   Y
1860    Compass Airlines        NULL    CP      CPZ     Compass Rose    United States   Y
1881    Continental Airlines    NULL    CO      COA     CONTINENTAL     United States   Y
```

End Of Report