**#Making a new folder DelayedFlights and loading the dataset in the hdfs.**

[cloudera@quickstart ~]$ hadoop fs -mkdir /user/pig/DelayedFlights

[cloudera@quickstart ~]$ hadoop fs -put DelayedFlights.csv /user/pig/DelayedFlights

**#Loading pig in cloudera**

[cloudera@quickstart ~]$ pig

**#Loading the data in pig**

grunt> DelayedFlights = LOAD '/user/pig/DelayedFlights/DelayedFlights.csv' USING PigStorage(',')as

>> ( ID:int, Year:int, Month:int, DayofMonth:int, DayOfWeek:int, DepTime:float,

>> CRSDepTime:int, ArrTime:float, CRSArrTime:int, UniqueCarrier:chararray,

>> FlightNum:int, TailNum:chararray, ActualElapsedTime:float,

>> CRSElapsedTime:float, AirTime:float, ArrDelay:float, DepDelay:float,

>> Origin:chararray, Dest:chararray, Distance:int, TaxiIn:float, TaxiOut:float,

>> Cancelled:int, CancellationCode:chararray, Diverted:int, CarrierDelay:float,

>> WeatherDelay:float, NASDelay:float, SecurityDelay:float,

>> LateAircraftDelay:float );

**#Storing the data in pig**

grunt> STORE DelayedFlights INTO '/user/pig/DelayedFlights/DelayedFlights\_Output/' USING PigStorage (',');

**#To view the data**

grunt> dump DelayedFlights

**Pig Queries**

**Q1.) What are the top 5 destinations that have been visited the most in the month of December?**

**Code:**

grunt> records = foreach DelayedFlights generate Dest, Month;

grunt> filtered = filter records by Month == 12;

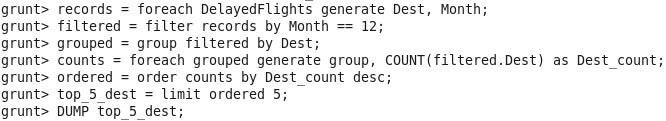
grunt> grouped = group filtered by Dest;

grunt> counts = foreach grouped generate group, COUNT(filtered.Dest) as Dest\_count;

grunt> ordered = order counts by Dest\_count desc;

grunt> top\_5\_dest = limit ordered 5;

grunt> DUMP top\_5\_dest;



**Output:**

(ATL,12437)

(ORD,10016)

(DEN,7509)

(DFW,6692)

(LAX,6069)



**Q2.) What is the least number of destinations that have been visited?**

**Code:**

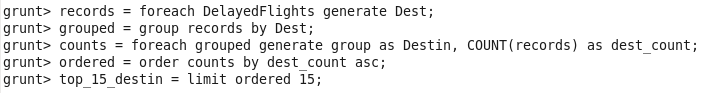
grunt> records = foreach DelayedFlights generate Dest;

grunt> grouped = group records by Dest;

grunt> counts = foreach grouped generate group as Destin, COUNT(records) as dest\_count;

grunt> ordered = order counts by dest\_count asc;

grunt> top\_15\_destin = limit ordered 15;



**Output:**

(OGD,1)

(TUP,1)

(CYS,1)

(PIR,3)

(INL,9)

(BJI,12)

(BLI,13)

(ITH,16)

(SUX,28)

(WYS,31)

(ADK,31)

(TEX,32)

(HTS,33)

(LWB,35)

(ACY,38)



**Q3.) Which Unique Carriers that have the most Carrier Delays?**

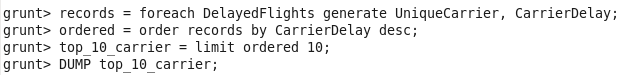
**Code:**

grunt> records = foreach DelayedFlights generate UniqueCarrier, CarrierDelay;

grunt> ordered = order records by CarrierDelay desc;

grunt> top\_10\_carrier = limit ordered 10;

grunt> DUMP top\_10\_carrier;



**Output:**

(NW,2436.0)

(NW,1951.0)

(MQ,1707.0)

(NW,1552.0)

(NW,1542.0)

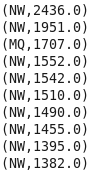
(NW,1510.0)

(NW,1490.0)

(NW,1455.0)

(NW,1395.0)

(NW,1382.0)



**Q4.) What airports have the most NAS Delays?**

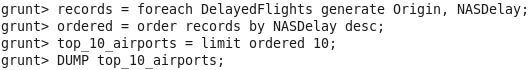
**Code:**

grunt> records = foreach DelayedFlights generate Origin, NASDelay;

grunt> ordered = order records by NASDelay desc;

grunt> top\_10\_airports = limit ordered 10;

grunt> DUMP top\_10\_airports;



**Output:**

(DEN,1357.0)

(TPA,1337.0)

(LGA,1289.0)

(MKE,1207.0)

(EGE,1195.0)

(JFK,1137.0)

(MSN,1030.0)

(SNA,1024.0)

(PHL,1009.0)

(FSM,992.0)



**Q5.) Which TailNumber has the most LateAircraftDelay delay?**

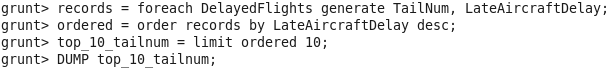
**Code:**

grunt> records = foreach DelayedFlights generate TailNum, LateAircraftDelay;

grunt> ordered = order records by LateAircraftDelay desc;

grunt> top\_10\_tailnum = limit ordered 10;

grunt> DUMP top\_10\_tailnum;



**Output:**

(N392AA,1316.0)

(N548UA,1303.0)

(N597UA,1254.0)

(N848MQ,1236.0)

(N593NW,1184.0)

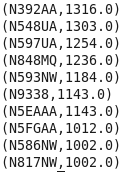
(N9338,1143.0)

(N5EAAA,1143.0)

(N5FGAA,1012.0)

(N586NW,1002.0)

(N817NW,1002.0)



**Q6.) Which route (origin & destination) has seen the maximum diversion?**

**Code:**

grunt> records = foreach DelayedFlights generate Origin, Dest, Diverted;

grunt> filtered = filter records by (Origin is not null) and (Dest is not null) and (Diverted == 1);

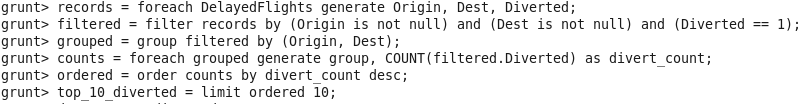
grunt> grouped = group filtered by (Origin, Dest);

grunt> counts = foreach grouped generate group, COUNT(filtered.Diverted) as divert\_count;

grunt> ordered = order counts by divert\_count desc;

grunt> top\_10\_diverted = limit ordered 10;

grunt> dump top\_10\_diverted;



**Output:**

((ORD,LGA),39)

((DAL,HOU),35)

((DFW,LGA),33)

((ATL,LGA),32)

((ORD,SNA),31)

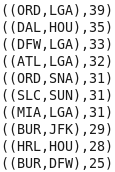
((SLC,SUN),31)

((MIA,LGA),31)

((BUR,JFK),29)

((HRL,HOU),28)

((BUR,DFW),25)



**Q7.) How many flights were delayed caused by weather per month?**

**Code:**

grunt> records = foreach DelayedFlights generate Month, WeatherDelay;

grunt> grouped = GROUP records by Month;

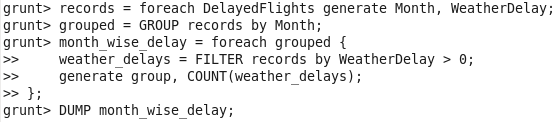
grunt> month\_wise\_delay = foreach grouped {

>> weather\_delays = FILTER records by WeatherDelay > 0;

>> generate group, COUNT(weather\_delays);

>> };

grunt> DUMP month\_wise\_delay;



**Output:**

(1,9658)

(2,12032)

(3,10114)

(4,5434)

(5,5645)

(6,11894)

(7,9991)

(8,8571)

(9,3060)

(10,2428)

(11,3618)

(12,16473)

