**Flight Dataset**

Some insights on the U.S. Airline data using Apache Pig  
  
**Understanding Data**   
The U.S. Department of Transportation’s (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, cancelled, and diverted flights appears in DOT’s monthly Air Travel Consumer Report, published about 30 days after the month’s end   
  
The data format is comma separated values These are 2 different datasets, i.e., Flight\_details.csv and Airports.csv. Let us understand one at a time.   
  
1. Flight\_details.csv   
There are 29 columns in flight\_details.csv dataset. Please check flight\_description file for details about schema/fields.   
  
2. Airports.csv  
• iata: the international airport abbreviation code   
• name of the airport   
• city and country in which airport is located.   
• lat and long: the latitude and longitude of the airport

**Problem Statement 1**

* **Find out the top 5 most visited destinations.**

1) LOAD the flight data set.

grunt> load\_flightdata = LOAD 'file:///home/cloudera/khasimbabu/PIG\_Excercise/CaseStudy3/flights\_details.csv' USING PigStorage(',');  
  
2) Extract the selective columns data  
grunt> flight\_details = FOREACH load\_flightdata GENERATE (int)$1 as year, (int)$10 as flightnu, (chararray)$17 as origin, (chararray)$18 as dest;  
grunt> describe flight\_details;

flight\_details: {year: int,flightnu: int,origin: chararray,dest: chararray}  
  
3) Extract Destination not null data  
grunt> flight\_dest\_notnull = FILTER flight\_details by dest is not null;  
  
4) Group by destination data  
grunt> grp\_flight\_dest = GROUP flight\_dest\_notnull by dest;  
grunt> describe grp\_flight\_dest;

grp\_flight\_dest: {group: chararray,flight\_dest\_notnull: {(year: int,flightnu: int,origin: chararray,dest: chararray)}}  
  
5) Count the flights by destination and arrange them in desending order.  
grunt> grp\_flightcount\_dest = FOREACH grp\_flight\_dest GENERATE group, COUNT(flight\_dest\_notnull.flightnu);  
 grunt> order\_flightcount\_dest = ORDER grp\_flightcount\_dest by $1 DESC;

6) limit – Top 5 records  
grunt> limit\_flightcount\_dest = LIMIT order\_flightcount\_dest 5;  
  
7) LOAD Airport data  
grunt> load\_airportdata = LOAD 'file:///home/cloudera/khasimbabu/PIG\_Excercise/CaseStudy3/airports.csv' USING PigStorage(',');  
  
8) Airport selective columns  
grunt> airport\_details = FOREACH load\_airportdata GENERATE (chararray)$0 as dest,(chararray)$2 as city, (chararray)$4 as country;

grunt> describe airport\_details;

airport\_details: {dest: chararray,city: chararray,country: chararray}  
  
9) Join flight, airport  
grunt> join\_flight\_airport = JOIN limit\_flightcount\_dest by $0, airport\_details by dest;

grunt> describe join\_flight\_airport;

join\_flight\_airport: {limt\_flightcount\_dest::group: chararray,long,airport\_details::dest: chararray,airport\_details::city: chararray,airport\_details::country: chararray}

grunt> dump join\_flight\_airport;  
  
**Problem Statement 2**

* **Which month has seen the most number of cancellations due to bad weather?\**

1) Flight cancel details  
grunt> flight\_cancel\_details = FOREACH load\_flightdata GENERATE (int)$2 as month, (int)$10 as flightnum, (chararray)$22 as cancelled, (chararray)$23 as cancelcode;  
  
2) List the cancelled flights due to bad weather  
grunt> flight\_cancelfilter = FILTER flight\_cancel\_details by cancelled=='1' AND cancelcode=='B';

grunt> describe flight\_cancelfilter;

flight\_cancelfilter: {month: int,flightnum: int,cancelled: chararray,cancelcode: chararray}

3) Group the cancelled flights by month  
grunt> grp\_cancelfilter = GROUP flight\_cancelfilter by month;

grunt> describe grp\_cancelfilter;

grp\_cancelfilter: {group: int,flight\_cancelfilter: {(month: int,flightnum: int,cancelled: chararray,cancelcode: chararray)}}  
  
4) Count, Order and LIMIT the cancelled flights  
grunt> count\_cancelflight = FOREACH grp\_cancelfilter GENERATE group, COUNT(flight\_cancelfilter.cancelcode);

grunt> order\_cancelflight = ORDER count\_cancelflight by $1 DESC;

grunt> limit\_cancelflight = LIMIT order\_cancelflight 1;  
grunt> dump limit\_cancelflight;  
  
**Problem Statement 3**

• **Top ten origins with the highest AVG departure delay**  
  
1) flight\_origindata  
grunt> flight\_origindata = FOREACH load\_flightdata GENERATE (int)$16 as dep\_delay,(chararray)$17 as origin;  
  
2) flight orgin not null  
grunt> grp\_flightbyorigin\_notnull = FILTER flight\_origindata by dep\_delay is not null AND origin is not null;   
  
3) group by origin data  
grunt> grp\_flightbyorigin = GROUP grp\_flightbyorigin\_notnull by origin;   
grunt> describe grp\_flightbyorigin;

grp\_flightbyorigin: {group: chararray,grp\_flightbyorigin\_notnull: {(dep\_delay: int,origin: chararray)}}

4) AVG of depdelay  
grunt> flight\_depdelay\_avg = foreach grp\_flightbyorigin generate group, AVG(grp\_flightbyorigin\_notnull. dep\_delay);

5) Order Origin data  
grunt> flight\_depdelay\_order = ORDER flight\_depdelay\_avg by $1 DESC;

grunt> flight\_depdelay\_limit = LIMIT flight\_depdelay\_order 10;   
  
6) Load Airport data  
grunt> load\_airportdata = LOAD 'file:///home/cloudera/khasimbabu/PIG\_Excercise/CaseStudy3/airports.csv' USING PigStorage(',');  
  
7) Airport origin details  
grunt> airport\_origindata = foreach load\_airportdata generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;

grunt> describe airport\_origindata;

airport\_origindata: {origin: chararray,city: chararray,country: chararray}  
  
8) join\_flight\_airportorigin  
grunt> join\_flight\_airportorigin = JOIN flight\_depdelay\_limit by $0 , airport\_origindata by origin;

grunt> describe join\_flight\_airportorigin;

join\_flight\_airportorigin: {flight\_depdelay\_limit::group: chararray,{(dep\_delay: int)},airport\_origindata::origin: chararray,airport\_origindata::city: chararray,airport\_origindata::country: chararray}  
  
9) Order and Limit the origin  
grunt> flight\_origin\_final = FOREACH join\_flight\_airportorigin GENERATE $0,$1,$2,$4;

grunt> order\_flight\_originfinal = ORDER flight\_origin\_final by $3 DESC;

grunt> dump order\_flight\_originfinal;  
  
**Problem Statement 4**

• **Which route (origin & destination) has seen the maximum diversion?**  
  
1) flight\_diversiondata  
grunt> flight\_diversiondata = FOREACH load\_flightdata GENERATE (chararray)$17 as origin,(chararray)$18 as dest,(int)$24 as diversion;  
  
2) flight\_diversion\_notnull  
grunt> flight\_diversion\_notnull = FILTER flight\_diversiondata by (origin is not null) AND (dest is not null) AND (diversion==1);  
  
3) grp\_flight\_diversion  
grunt> grp\_flight\_diversion = GROUP flight\_diversion\_notnull by (origin,dest);

grunt> describe grp\_flight\_diversion;

grp\_flight\_diversion: {group: (origin: chararray,dest: chararray),flight\_diversion\_notnull: {(origin: chararray,dest: chararray,diversion: int)}}   
  
4) Count, Order, Limit flight diversion  
grunt> count\_flight\_diversion = FOREACH grp\_flight\_diversion GENERATE group,COUNT(flight\_diversion\_notnull.diversion);

grunt> order\_flight\_diversion = ORDER count\_flight\_diversion by $1 DESC;

grunt> describe count\_flight\_diversion;

count\_flight\_diversion: {group: (origin: chararray,dest: chararray),long}

grunt> limit\_flight\_diversion = LIMIT order\_flight\_diversion 10;

grunt> dump limit\_flight\_diversion;