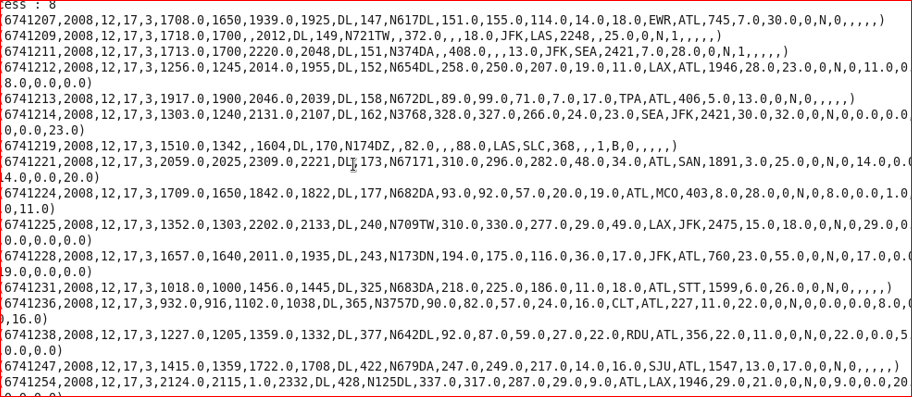
Task:1

Find out the top 5 most visited destination

REGISTER ‘/path/piggybank.jar’;

file = LOAD 'DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

First we will load our DelayedFlights.csv file using load function CSVExcelStorage , This



fetch = FOREACH sourceFile GENERATE $18 as(destination :bytearray) , 1 as (val : int);

Generate the details about dest from relation sourceFile which is at index $18 , we will generate int 1 for each records so as to group with it

dest = GROUP fetch by $0 ;

dest: {group: chararray,fetchDest: {(destination: chararray)}}

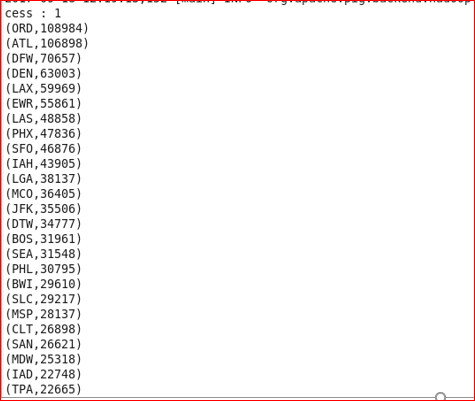
Group the relation fetch by dest

dest = FOREACH dest GENERATE group as (dest : bytearray) , COUNT(fetch) as (destCount : long);

Get the destination and the count the number of destinations

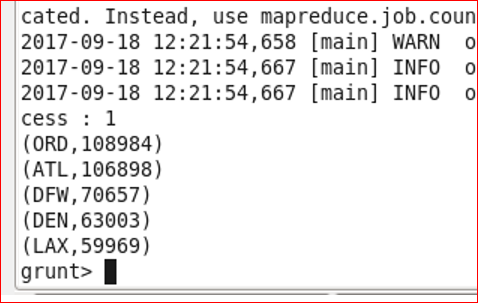
5) orderDest = ORDER countDest by destCount DESC;

Sort the relation dest by descreasing order



Top5Dest = LIMIT 5 orderDest;

Get the top 5 destinations using limit operator



airfile = LOAD 'airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

Load airport data and skip the header

8) information = FOREACH airfile GENERATE $0 as (dest : bytearray) , $2 as (city : chararray ), $3 as (state :chararray) ,$4 as (country : chararray);

-- Fetch the required details from airfile relation

Join Top5Dest with cityInfo:

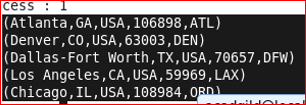
9) join = JOIN information by $0 RIGHT OUTER, Top5Dest by dest ;

join: {cityInfo::dest: bytearray,cityInfo::city: chararray,cityInfo::state: chararray, information::country: chararray,Top5Dest::dest: bytearray,Top5Dest::destCount: long}

Perform right outer join between information and Top5Dest to get the location details

10) Top5Destinations = FOREACH join GENERATE information::city , information::state , information::country , Top5Dest::destCount, Top5Dest::dest ; (Final output)

Fetch the Location details



Task:2

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

file = LOAD 'DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

Cancellation = FILTER file by $22 == 1 AND $23 == 'B';

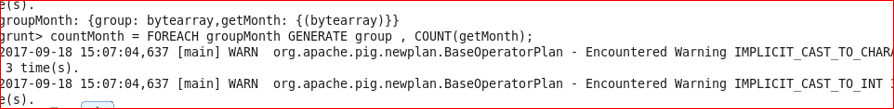
Filter the records where cancellationcode is 1(TRUE) which is at index 22 and CANCELLATION is done due to bad weather (B) which is at index location 23

removeJunkRecords = FILTER Cancellation by $2 is not null

Remove any junk records where null exists

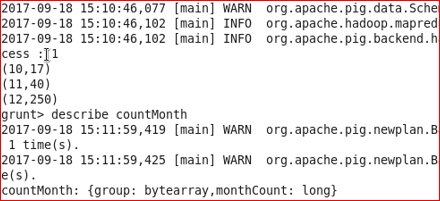
getMonth = FOREACH removeJunkRecords GENERATE $2;

groupMonth = GROUP getMonth by $0;



countMonth = FOREACH groupMonth GENERATE group , COUNT(getMonth) as (monthCount : LONG);

--Count the number of months for each month

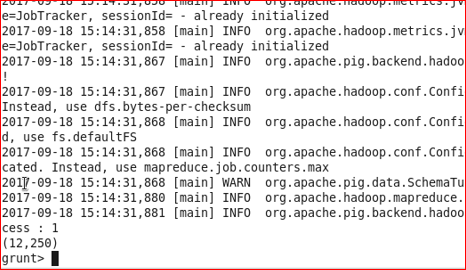


orderMonth = ORDER countMonth by monthCount desc;

sort the relation in descending order of monthCount to get the hishest month where cancellation is done

MaxMonth = LIMIT orderMonth 1; (Final output)

Get the month where highest cancellation is done due to bad weather



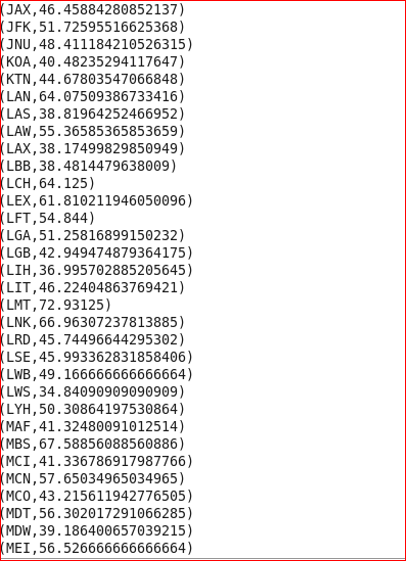
Task:3

Top ten origins with the highest AVG departure delay

file = LOAD 'DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

getOrigin = FOREACH sourceFile GENERATE $16 as (debDelay : bytearray ), $17 as (origin : bytearray);

groupOrigin = GROUP getOrigin by $1;

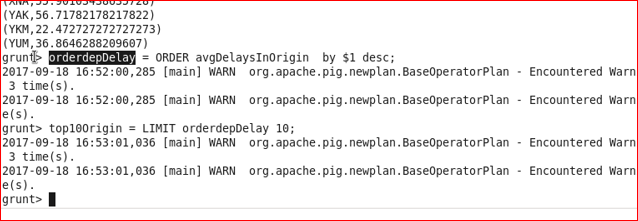


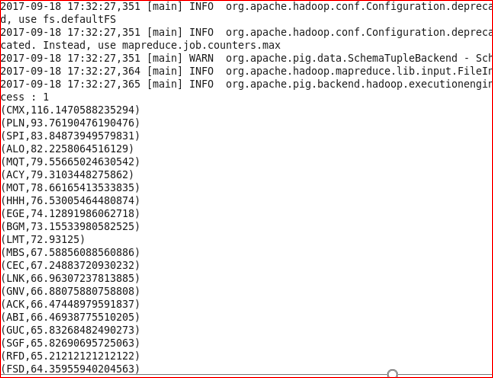
avgDelaysInOrigin = FOREACH groupOrigin GENERATE (chararray)group , AVG(getOrigin.debDelay) as (depDElay : double) ;

--count the average of depDelay for each origin

orderdepDelay = ORDER avgDelaysInOrigin by $1 desc;

--sort the relation avgDelaysInOrigin by descending of avg

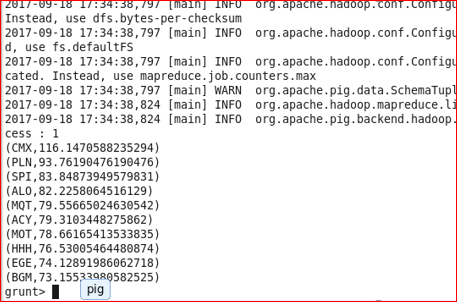




top10Origin = LIMIT orderdepDelay 10; (Final output)

--Fetch the 10 ten origin by using the limit operator

top10Origin: {group: chararray,depDElay: double}



Task:4

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

fileile = LOAD 'DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER');

2) fetch = FOREACH sourceFile GENERATE $17 ,$18, $24 ;

Fetch the origin which is at index $17 . destination which is at $17 and diversion which is at index $24

filter = FILTER fetch by ( $0 is not null )AND ($1 is not null) AND ($2 == 1 ) ;

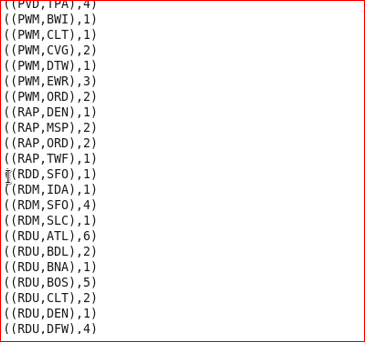
--filter the junk records where origin and dest is null and where diversion is true

routes = GROUP filter by($0,$1);

--group relation with origin and destination

rate = FOREACH routes GENERATE group , COUNT(filterDiverted);

--Generate the number of diversions taken with repsect to each route



orderRouteCount = ORDER rate by $1 DESC;

--order the relation in descending order of route count to get the highest diverions

topDiverted = LIMIT orderRouteCount 1; (Final output)

--fetch the top diverted route

