**Aviation Data Analysis Using Apache Pig**

**Note: Before running pig scripts, start all Hadoop Daemons with start-all.sh command, and check using jps command whether all daemons are running or not i.e. Namenode, Datanode, SecondaryNamenode, ResourceMangaer, NodeManager. There is no need to start jobhistoryserver because pig is launched in LOCAL mode.**

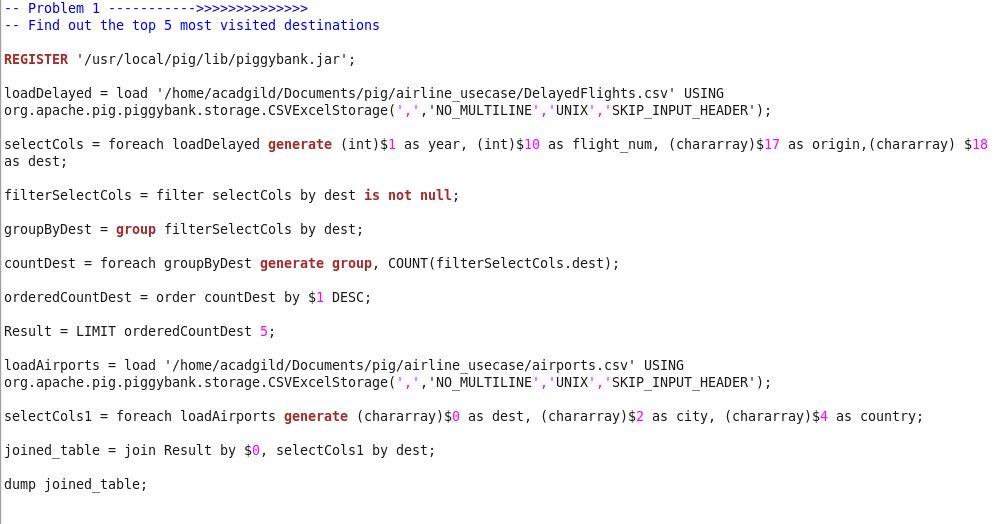
## **Here, we are working on two datasets, which are as follows:**

## **Delayed\_Flights.csv Datasets**

## **Airports.csv Datasets**

## Problem Statement 1

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

** **Source Code: Below source code is saved as “assign5.2\_airline\_Problem1.pig”**

**Explanation of above source code:**

**Line 1**: “piggybank.jar” is registered in order to use the CSVExcelStorage class.

**Line 2:** In relation **loadDelayed**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and headers.

**Line 3:** In relation **selectCols**, we are generating the columns that are required for processing and explicitly typecasting each of them.

**Line 4:** In relation **filterSelectCols**, we are filtering the null values from the “dest” column.

**Line 5:** In relation **groupByDest**, we are grouping relation **filterSelectCols** by “dest.”

**Line 6:** In relation **countDest**, we are generating the grouped column and the count of each.

**Line 7:** Relation **orderedCountDest** stores ordered result of **countDest** relation.

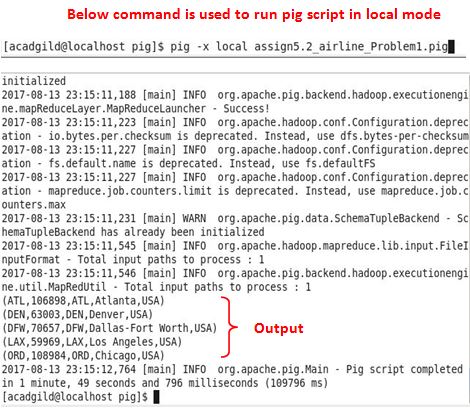
**Line 8: Result relation** stores limited number of tuples (i.e. top 5) of **orderedCountDest** relation.

These are the steps to find the top 5 most visited destinations. However, adding few more steps in this process, we will be using another table to find the city name and country as well.

**Line 9:** In relation **loadAirports**, we are loading another table to which we will look-up and find the city as well as the country.

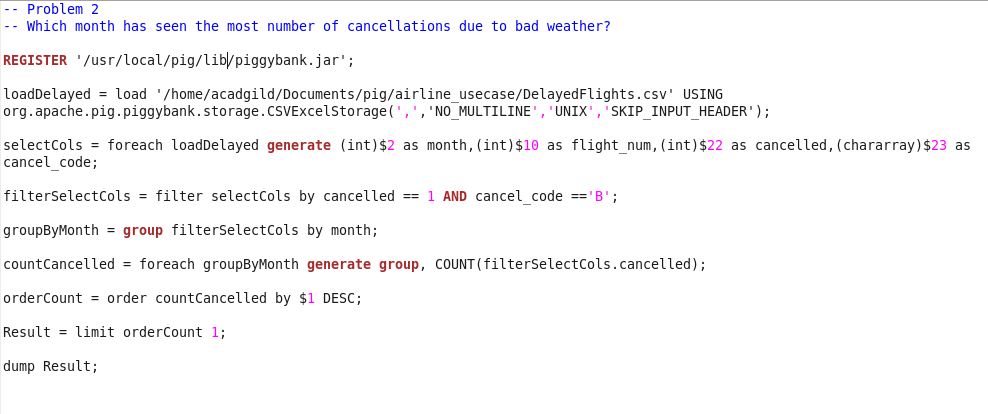
**Line 10:** In relation **selectCols1**, we are generating dest, city, and country from **loadAirports** relation.

**Line 11:** In relation **joined\_table**, we are joining **Result** and **selectCols1** based on a common column, i.e., “dest”

**Line 12:** Finally, using **dump**, we are printing the result of **joined\_table**.

## Problem Statement 2

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

**Source code: Below source code is saved as “assign5.2\_airline\_Problem2.pig”**

**Explanation of above source code:**

**Line 1**: “piggybank.jar” is registered in order to use the CSVExcelStorage class.

**Line 2:** In relation **loadDelayed**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and header.

**Line 3:** In relation **selectCols**, we are generating the columns which are required for processing and explicitly typecasting each of them.

**Line 4:** In relation **filterSelectCols**, we are filtering the data based on cancellation and cancellation code, i.e., cancelled == 1 means flight has been cancelled and cancel\_code == ‘B’ means the reason for cancellation is “weather.” So relation **filterSelectCols** will point to the data which consists of cancelled flights due to bad weather.

**Line 5:** In relation **groupByMonth**, we are grouping the relation **filterSelectCols** based on every month.

**Line 6:** In relation **countCancelled**, we are finding the count of cancelled flights every month.

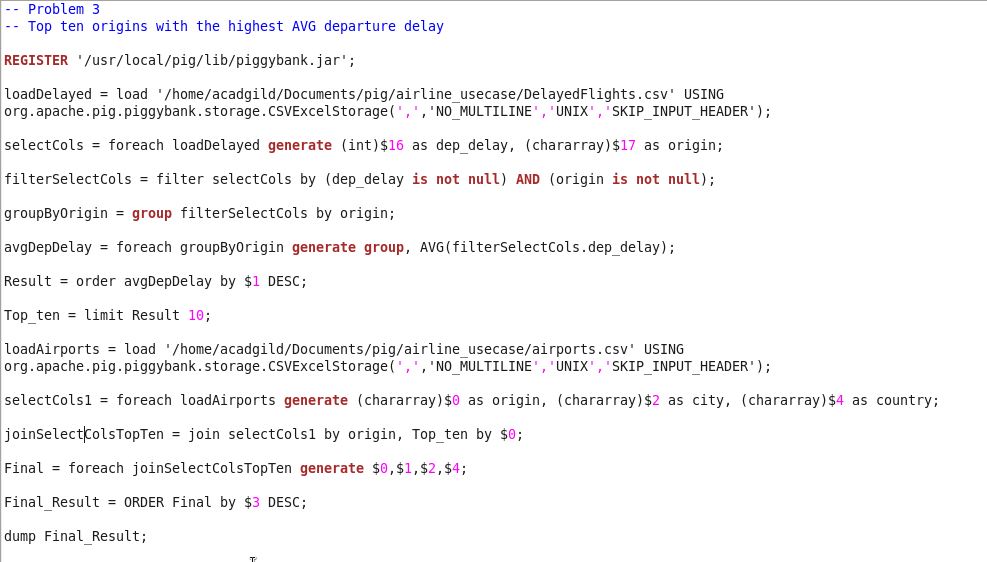
**Line 7:** Relation **orderCount** stored ordered result of **counCancelled** relation

**Line 8: Result relation stores** top month based on cancellation.

**Line 9:** Finally using **dump**, data inside **Result** relation is printed.

## Problem Statement 3

**Top ten origins with the highest AVG departure delay**

**** Source code: Below source code is saved as “assign5.2\_airline\_Problem3.pig”**

**Explanation of above source code:**

**Line 1**: “piggybank.jar” is registered in order to use the CSVExcelStorage class.

**Line 2:** In relation **loadDelayed**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and header.

**Line 3:** In relation **selectCols**, we are generating the columns which are required for processing and explicitly typecasting each of them.

**Line 4:** In relation **filterSelectCols**, we are removing the **null** values fields present if any.

**Line 5:** In relation **groupByOrigin**, we are grouping the data based on column “origin”.

**Line 6:** In relation **avgDepDelay**, we are finding average delay from each unique origin.

**Line 7:** Relation **Result** orders the results in descending order

**Line 8:** Relation **Top\_ten** limits tuples/rows of relation **Result** to top 10.

These steps are good enough to find the top ten origins with the highest average departure delay.

However, rather than generating just the code of origin, we will be following a few more steps to find some more details like country and city.

**Line 9:** In the relation **loadAirports**, we are loading another table to which we will look up and find the city as well as the country.

**Line 10:** In the relation **selectCols1,** we are generating the destination, city, and country from the **loadAirports** relation.

**Line 11:** In the relation **joinSelectColsTopTen**, we are joining relation **Top\_ten** and **selectCols1** based on a common column, i.e., “origin”.

**Line 12:** In the relation **Final,** we are generating required columns from the **joinSelectColsTopTen relation**.

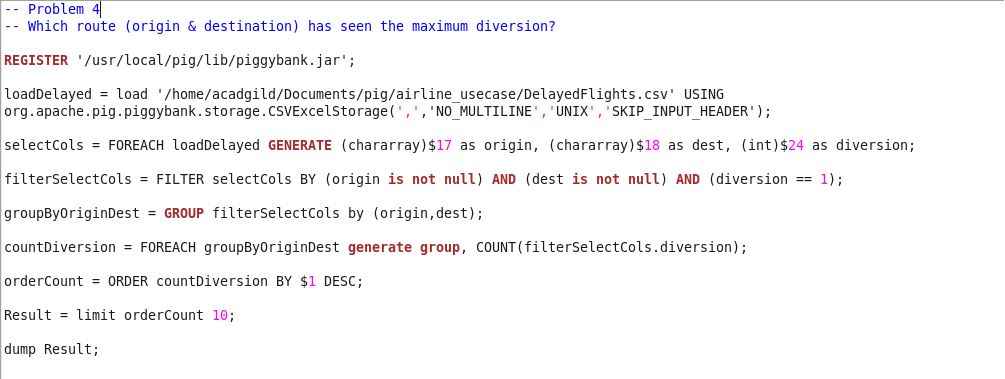
**Line 13:** In relation **Final\_Result,** ordered tuples of **Final** relation are stored

**Line 14:** **dump** commands prints the data inside **Final\_Result.**

****

## Problem Statement 4

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

**Source code: Below source code is saved as “assign5.2\_airline\_Problem4.pig”**

**Explanation of above source code:**

**Line 1**: “piggybank.jar” is registered in order to use the CSVExcelStorage class.

**Line 2:** In relation **loadDelayed**, we are loading the dataset using CSVExcelStorage because of its effective technique to handle double quotes and header.

**Line 3:** In relation **selectCols**, we are generating the columns which are required for processing and explicitly typecasting each of them.

**Line 4:** In relation **filterSelectCols**, we are filtering the data based on **“not null”** condition on **origin,** **dest** and **diversion==1**. This will remove the null records, if any, and give the data corresponding to the diversion taken.

**Line 5:** In relation **groupByOriginDest**, we are grouping the data based on origin and destination.

**Line 6:** Relation **countDiversion** finds the count of diversion taken per unique origin and destination.

**Line 7:** Relations **orderCount** stores the ordered result of countDiversion relation.

**Line 8:** Relation **Result** stores limited (i.e. top 10) tuples/ rows of **orderCount** relation.

**Line 9: dump** displays the data inside **Result** relation.

