ASSIGNMENT 5.2

Aviation Data Analysis Using Apache Pig:

There are 2 different datasets, i.e., Delayed_Flights.csv and Airports.csv.

Delayed_Flights.csv Dataset

There are 29 columns in this dataset. Some of them have been mentioned below:

- Year: 1987 – 2008
- Month: 1 – 12
- FlightNum: Flight number
- Canceled: Was the flight canceled?
- CancelleationCode: The reason for cancellation.

Airports.csv Dataset

- 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.

Source Code:

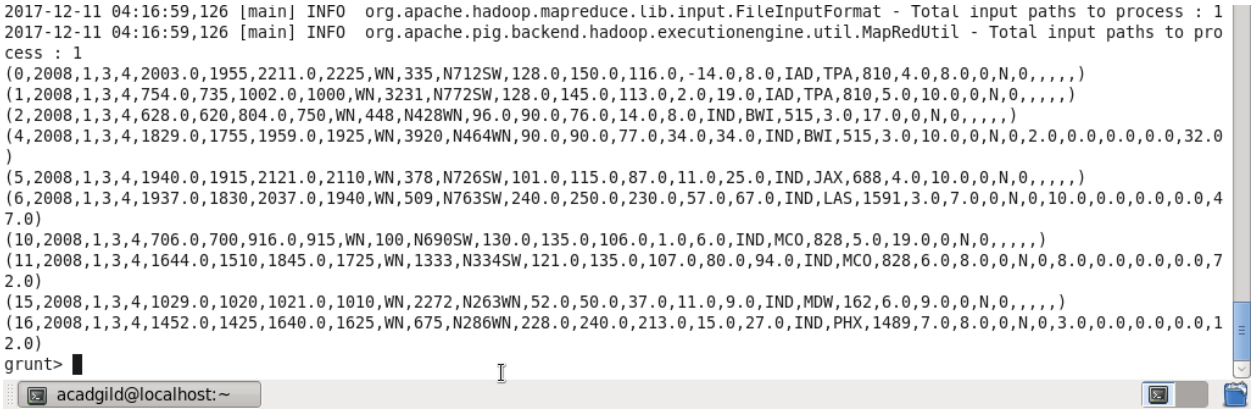
```
REGISTER '/usr/local/pig/lib/piggybank.jar';

A = load '/home/acadgild/airline_usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.

storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

Delayed_flights_limited_records = LIMIT A 10;

DUMP Delayed_flights_limited_records;
```



B = foreach A generate (int)$1 as year, (int)$10 as flight_num, (chararray)$17 as origin,(chararray) $18 as dest;

C = filter B by dest is not null;

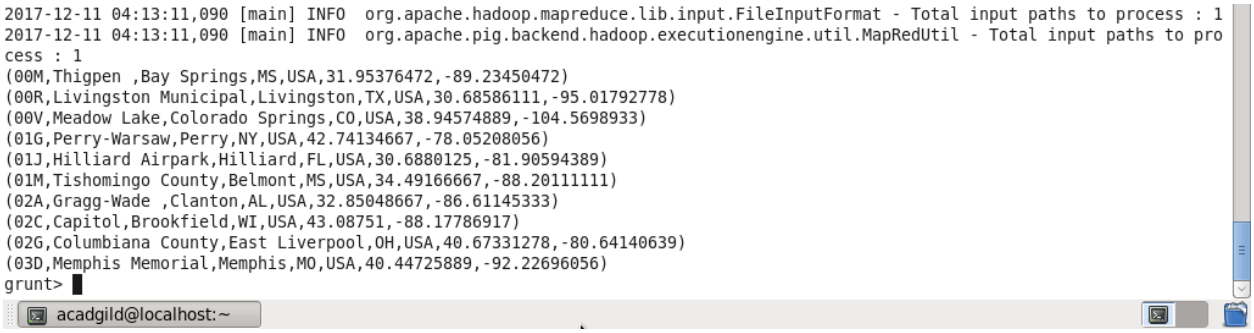
D = group C by dest;

E = foreach D generate group, COUNT(C.dest);

F = order E by $1 DESC;

Result = LIMIT F 5;

A1 = load '/home/acadgild/airline_usecase/airports.csv' USING org.apache.pig.piggybank. storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');



A2 = foreach A1 generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as country;

joined_table = join Result by $0, A2 by dest;

```
grunt> A = load '/home/acadgild/airline usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',
','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');
2017-12-14 16:49:56,532 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is depre
cated. Instead, use mapreduce.job.counters.max
2017-12-14 16:49:56,532 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2017-12-14 16:49:56,532 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
d, use fs.defaultFS
grunt> B = foreach A generate (int)$1 as year, (int)$10 as flight_num, (chararray)$17 as origin,(chararray) $18 as dest;
grunt> C = filter B by dest is not null;
grunt> D = group C by dest;
grunt> E = foreach D generate group, COUNT(C.dest);
grunt> F = order E by $1 DESC;
grunt> Result = LIMIT F 5;
grunt> A1 = load '/home/acadgild/airline usecase/airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO
_MULTILINE','UNIX','SKIP_INPUT_HEADER');
2017-12-14 17:52:48,174 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is depre
cated. Instead, use mapreduce.job.counters.max
2017-12-14 17:52:48,175 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2017-12-14 17:52:48,175 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
d, use fs.defaultFS
grunt> A2 = foreach A1 generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as country;
grunt>
grunt> joined_table = join Result by $0, A2 by dest;
grunt>
```

In Line 1: We are registering the *piggybank* jar in order to use the CSVExcelStorage class.

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

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

In relation **C**, we are filtering the null values from the “dest” column.

In relation **D**, we are grouping relation C by “dest.”

In relation **E**, we are generating the grouped column and the count of each.

Relation **F** and **Result** is used to order and limit the result to top 5.

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.

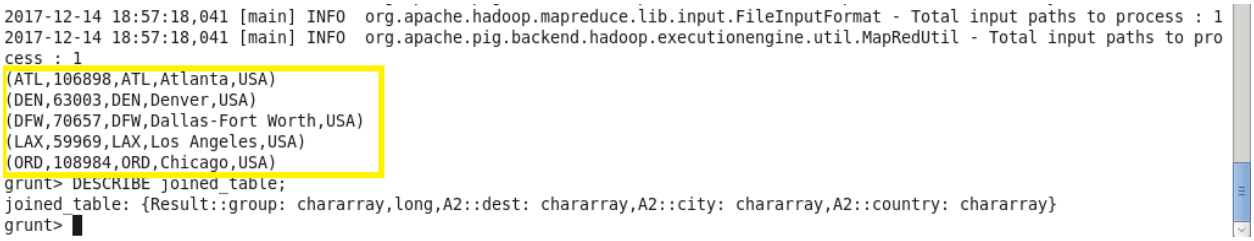
In relation **A1**, we are loading another table to which we will look-up and find the city as well as the country.

In relation **A2**, we are generating dest, city, and country from the previous relation.

In relation **joined_table**, we are joining Result and A2 based on a common column, i.e., “dest”

Finally, using dump, we are printing the result.

DUMP joined_table;



Problem Statement 2:

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

Source code:

REGISTER '/home/acadgild/airline_usecase/piggybank.jar';

A = load '/home/acadgild/airline_usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B = foreach A generate (int)$2 as month,(int)$10 as flight_num,(int)$22 as cancelled,(chararray)$23 as cancel_code;

C = filter B by cancelled == 1 AND cancel_code =='B';

D = group C by month;

E = foreach D generate group, COUNT(C.cancelled);

F= order E by $1 DESC;

Result = limit F 1;

```
2017-12-14 18:59:10,753 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.userlog.retain.hours is deprec
ated. Instead, use mapreduce.job.userlog.retain.hours
2017-12-14 18:59:10,754 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.local.dir.minspacekill is depr
ecated. Instead, use mapreduce.tasktracker.local.dir.minspacekill
grunt> B = foreach A generate (int)$2 as month,(int)$10 as flight_num,(int)$22 as cancelled,(chararray)$23 as cancel_code;
grunt> C = filter B by cancelled == 1 AND cancel_code =='B';
grunt> D = group C by month;
grunt> E = foreach D generate group, COUNT(C.cancelled);
grunt> F= order E by $1 DESC;
grunt> Result = limit F 1;
grunt>
```

In Line 1: We are registering *piggybank* jar in order to use the CSVExcelStorage class.

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

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

In relation **C**, we are filtering the data based on cancellation and cancellation code, i.e., canceled = 1 means flight have been canceled and cancel_code = 'B' means the reason for cancellation is "weather." So relation C will point to the data which consists of canceled flights due to bad weather.

In relation **D**, we are grouping the relation C based on every month.

In relation **E**, we are finding the count of canceled flights every month.

Relation **F** and **Result** is for ordering and finding the top month based on cancellation.

Finally, using dump, we are printing the result.

DUMP Result;

```
2017-12-14 19:01:25,285 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-14 19:01:25,285 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(12,250)
grunt> DESCRIBE Result;
Result: {group: int,long}
grunt>
```

Problem Statement 3:

Top ten origins with the highest AVG departure delay

Source code:

REGISTER '/home/acadgild/airline_usecase/piggybank.jar';

```
A = load '/home/acadgild/airline_usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B1 = foreach A generate (int)$16 as dep_delay, (chararray)$17 as origin;

C1 = filter B1 by (dep_delay is not null) AND (origin is not null);

D1 = group C1 by origin;

E1 = foreach D1 generate group, AVG(C1.dep_delay);

Result = order E1 by $1 DESC;

Top_ten = limit Result 10;

Lookup = load '/home/acadgild/airline_usecase/airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

Lookup1 = foreach Lookup generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;

Joined = join Lookup1 by origin, Top_ten by $0;

Final = foreach Joined generate $0,$1,$2,$4;

Final_Result = ORDER Final by $3 DESC;
```

```
grunt> A = load '/home/acadgild/airline_usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',
','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');
2017-12-14 19:02:41,285 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is depre
cated. Instead, use mapreduce.job.counters.max
2017-12-14 19:02:41,285 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2017-12-14 19:02:41,285 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
d, use fs.defaultFS
grunt> B1 = foreach A generate (int)$16 as dep_delay, (chararray)$17 as origin;
grunt> C1 = filter B1 by (dep_delay is not null) AND (origin is not null);
grunt> D1 = group C1 by origin;
grunt> E1 = foreach D1 generate group, AVG(C1.dep_delay);
grunt> Result = order E1 by $1 DESC;
grunt> Top_ten = limit Result 10;
grunt> Lookup = load '/home/acadgild/airline_usecase/airports.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(','
,'NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');
2017-12-14 19:03:36,681 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is depre
cated. Instead, use mapreduce.job.counters.max
2017-12-14 19:03:36,681 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2017-12-14 19:03:36,681 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
d, use fs.defaultFS
grunt> Lookup1 = foreach Lookup generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;
grunt> Joined = join Lookup1 by origin, Top_ten by $0;
grunt> Final = foreach Joined generate $0,$1,$2,$4;
grunt> Final_Result = ORDER Final by $3 DESC;
grunt>
```

Explanation of first 3 lines are the same as explained in the previous 2 problem statements.

In relation **C1**, we are removing the null values fields present if any.

In relation **D1**, we are grouping the data based on column “origin.”

In relation **E1**, we are finding average delay from each unique origin.

Relations named **Result** and **Top_ten** are ordering the results in descending order and printing the top ten values.

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.

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

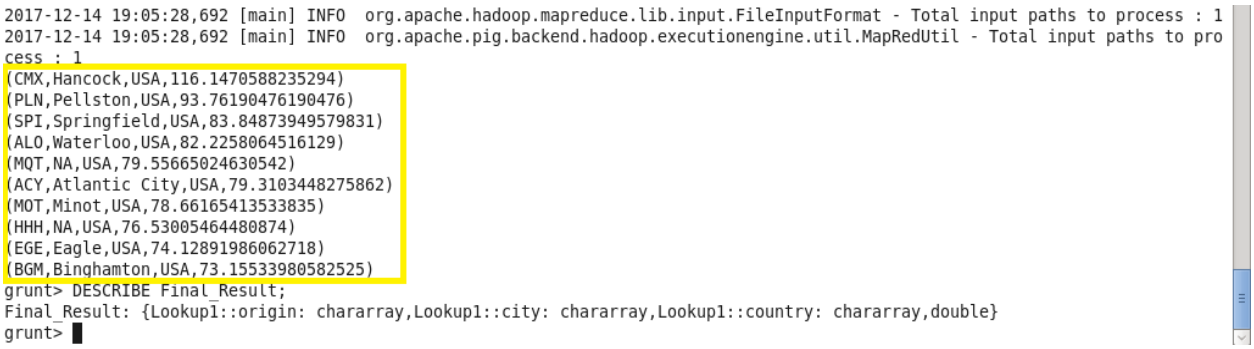
In the relation L**ookup1,** we are generating the destination, city, and country from the previous relation.

In the relation **Joined**, we are joining relation Top_ten and Lookup1 based on common a column, i.e., “origin.”

In the relation **Final,** we are generating required columns from the Joined table.

Finally, using dump, we are printing the result.

DUMP Final_Result;



Problem Statement 4:

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

Source code:

```
REGISTER '/home/acadgild/airline_usecase/piggybank.jar';

A = load '/home/acadgild/airline_usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B = FOREACH A GENERATE (chararray)$17 as origin, (chararray)$18 as dest, (int)$24 as diversion;

C = FILTER B BY (origin is not null) AND (dest is not null) AND (diversion == 1);

D = GROUP C by (origin,dest);

E = FOREACH D generate group, COUNT(C.diversion);

F = ORDER E BY $1 DESC;

Result = limit F 10;
```

```
grunt> A = load '/home/acadgild/airline usecase/DelayedFlights.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',
','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');
2017-12-14 19:06:34,174 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is depre
cated. Instead, use mapreduce.job.counters.max
2017-12-14 19:06:34,175 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2017-12-14 19:06:34,175 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
d, use fs.defaultFS
grunt> B = FOREACH A GENERATE (chararray)$17 as origin, (chararray)$18 as dest, (int)$24 as diversion;
grunt> C = FILTER B BY (origin is not null) AND (dest is not null) AND (diversion == 1);
grunt> D = GROUP C by (origin,dest);
grunt> E = FOREACH D generate group, COUNT(C.diversion);
grunt> F = ORDER E BY $1 DESC;
grunt> Result = limit F 10;
grunt>
```

In Line 1: We are registering *piggybank* jar in order to use CSVExcelStorage class.

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

In relation **B**, we are generating the columns which are required for processing and explicitly type-casting each of them.

In relation **C**, we are filtering the data based on “not null” and diversion =1. This will remove the null records, if any, and give the data corresponding to the diversion taken.

In relation **D**, we are grouping the data based on origin and destination.

Relation **D** finds the count of diversion taken per unique origin and destination.

Relations **F** and **Result** orders the result and produces top 10 results.

Finally, using dump, we are printing the result.

DUMP Result;

```
2017-12-14 19:08:20,758 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-14 19:08:20,758 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
((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)
grunt> DESCRIBE Result;
Result: {group: (origin: chararray,dest: chararray),long}
grunt>
```