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.

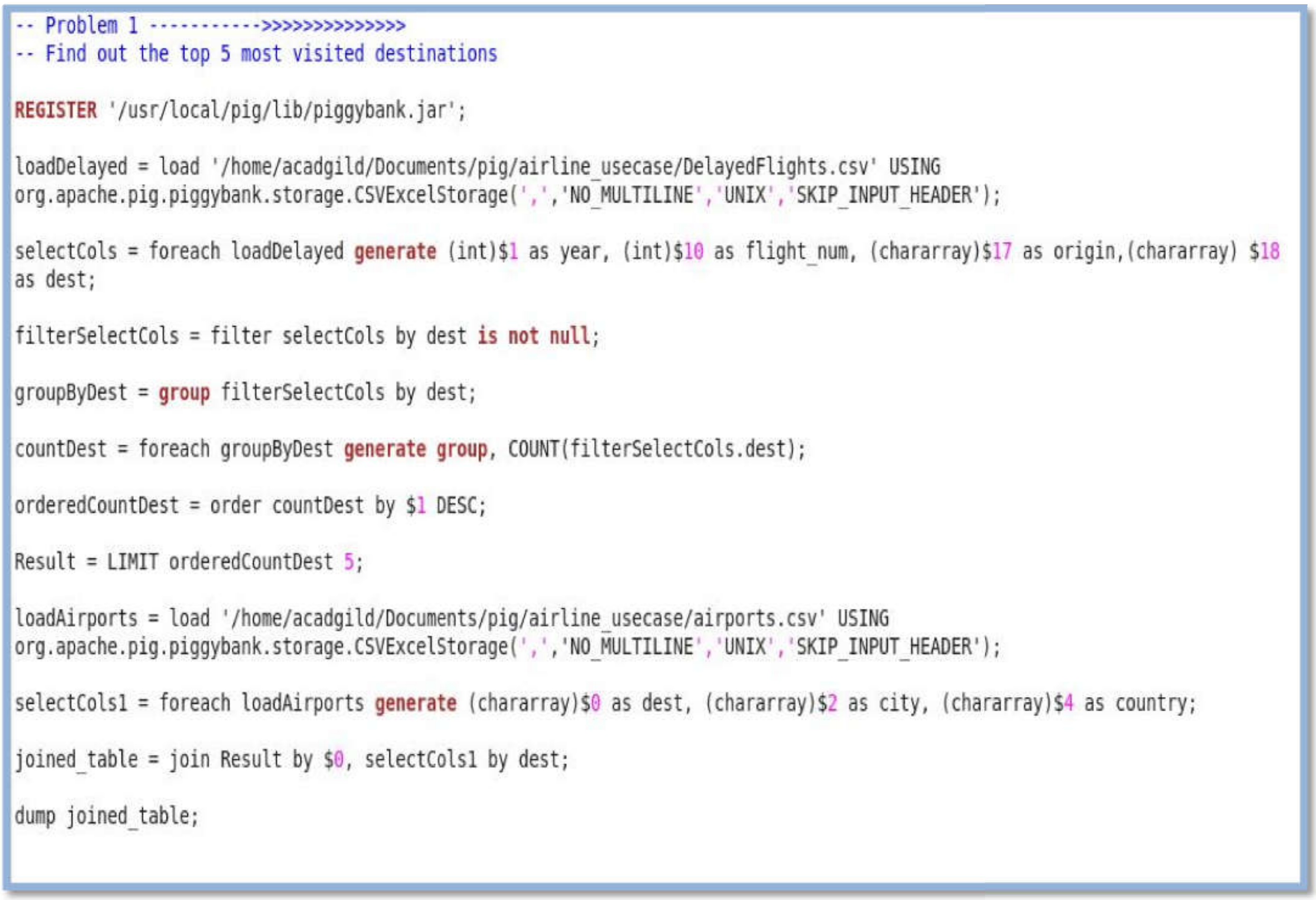
Here, we are working on two datasets, which are as follows:

- 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"



```
-- Problem 1 ----------->>>>>>>>>>>>>>>
-- Find out the top 5 most visited destinations

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

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

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

filterSelectCols = filter selectCols by dest is not null;

groupByDest = group filterSelectCols by dest;

countDest = foreach groupByDest generate group, COUNT(filterSelectCols.dest);

orderedCountDest = order countDest by $1 DESC;

Result = LIMIT orderedCountDest 5;

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

selectCols1 = foreach loadAirports generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as country;

joined_table = join Result by $0, selectCols1 by dest;

dump joined_table;
```

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.

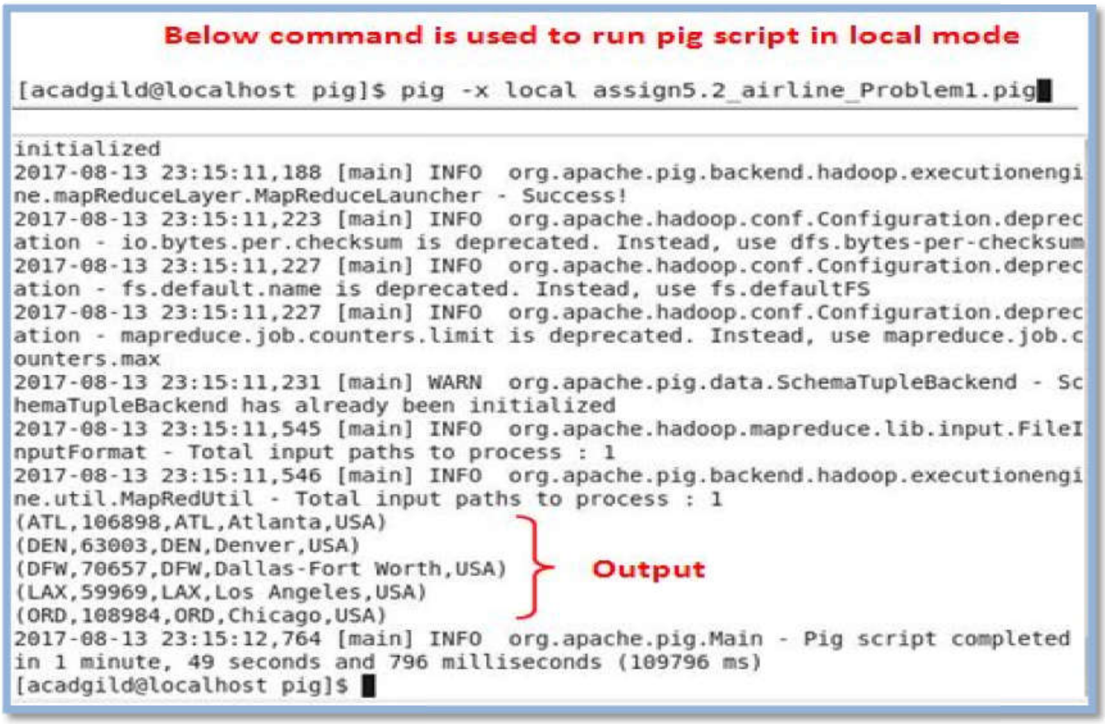
These are the steps to find the top 5 most visited destinations. However, adding few more steps in this process, we will be using another table to find the city name and country as well.

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**.



Below command is used to run pig script in local mode

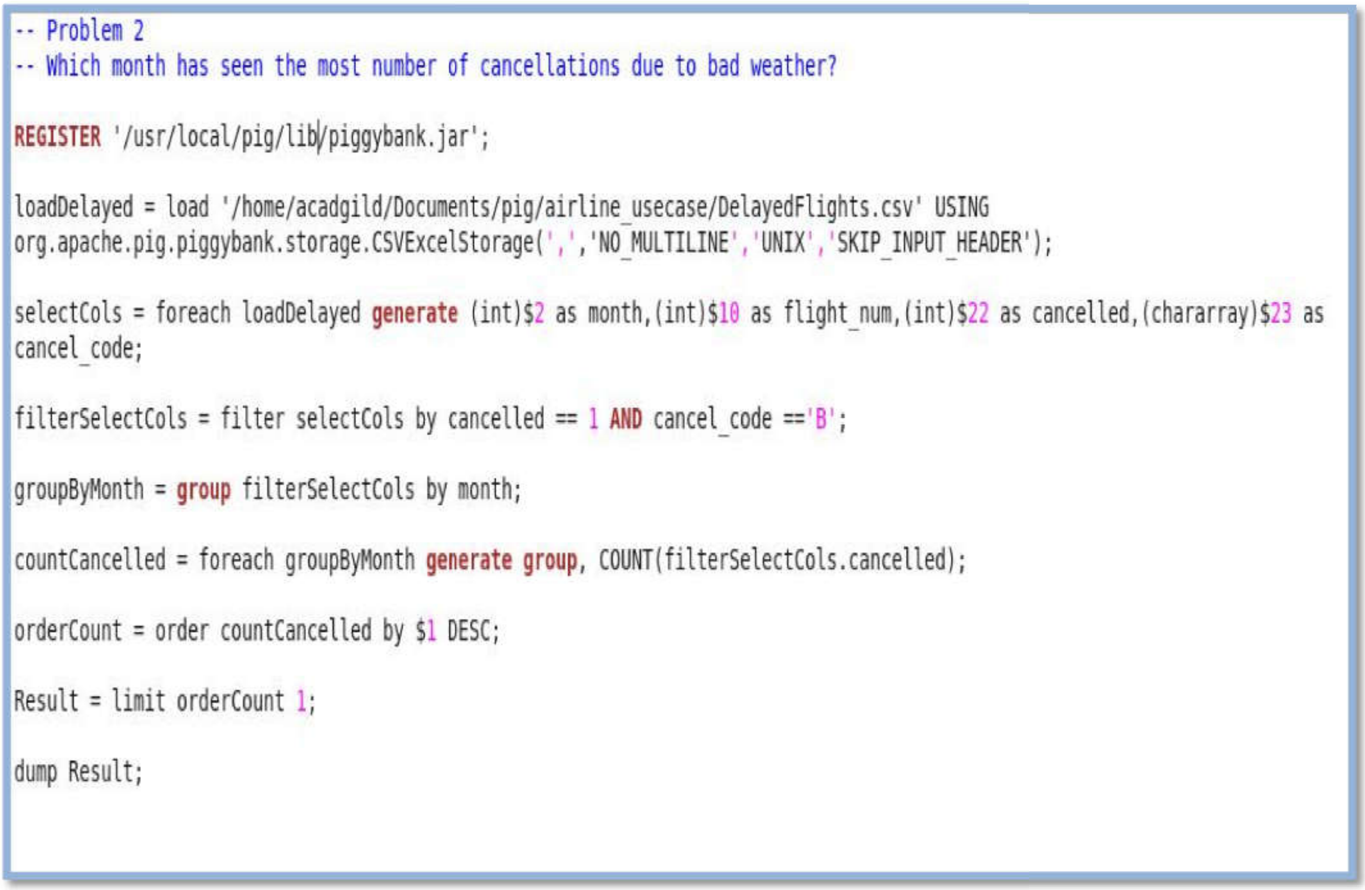
```
[acadgild@localhost pig]$ pig -x local assign5.2_airline_Problem1.pig
```

```
initialized
2017-08-13 23:15:11,188 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2017-08-13 23:15:11,223 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-08-13 23:15:11,227 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-08-13 23:15:11,227 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-08-13 23:15:11,231 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2017-08-13 23:15:11,545 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2017-08-13 23:15:11,546 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(ATL,106898,ATL,Atlanta,USA)
(DEN,63003,DEN,Denver,USA)
(DFW,70657,DFW,Dallas-Fort Worth,USA)
(LAX,59969,LAX,Los Angeles,USA)
(ORD,108984,ORD,Chicago,USA)
2017-08-13 23:15:12,764 [main] INFO  org.apache.pig.Main - Pig script completed
in 1 minute, 49 seconds and 796 milliseconds (109796 ms)
[acadgild@localhost pig]$
```

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”



```
-- Problem 2
-- Which month has seen the most number of cancellations due to bad weather?

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

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

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

filterSelectCols = filter selectCols by cancelled == 1 AND cancel_code =='B';

groupByMonth = group filterSelectCols by month;

countCancelled = foreach groupByMonth generate group, COUNT(filterSelectCols.cancelled);

orderCount = order countCancelled by $1 DESC;

Result = limit orderCount 1;

dump Result;
```

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.

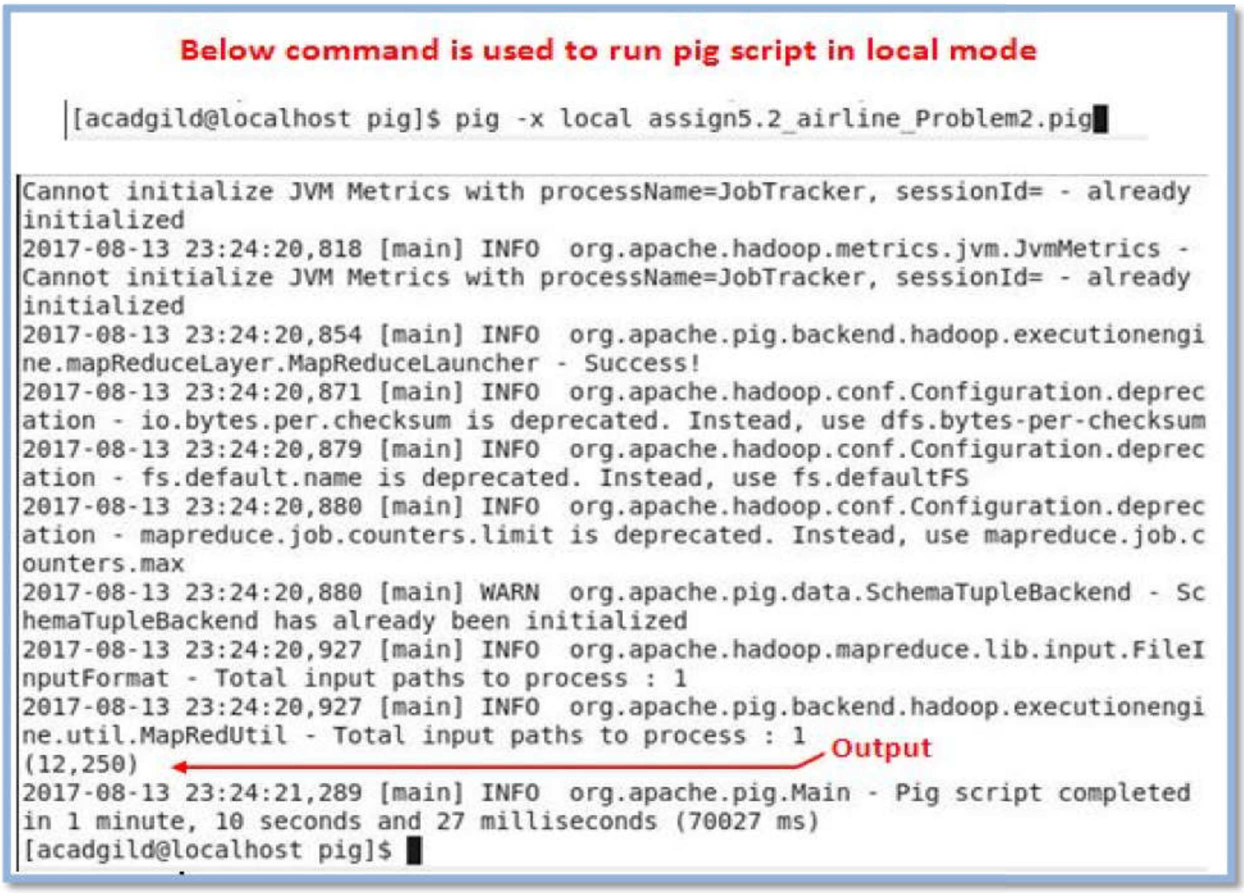
Line 5: In relation **groupByMonth**, we are grouping the relation **filterSelectCols** based on every month.

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.



Below command is used to run pig script in local mode

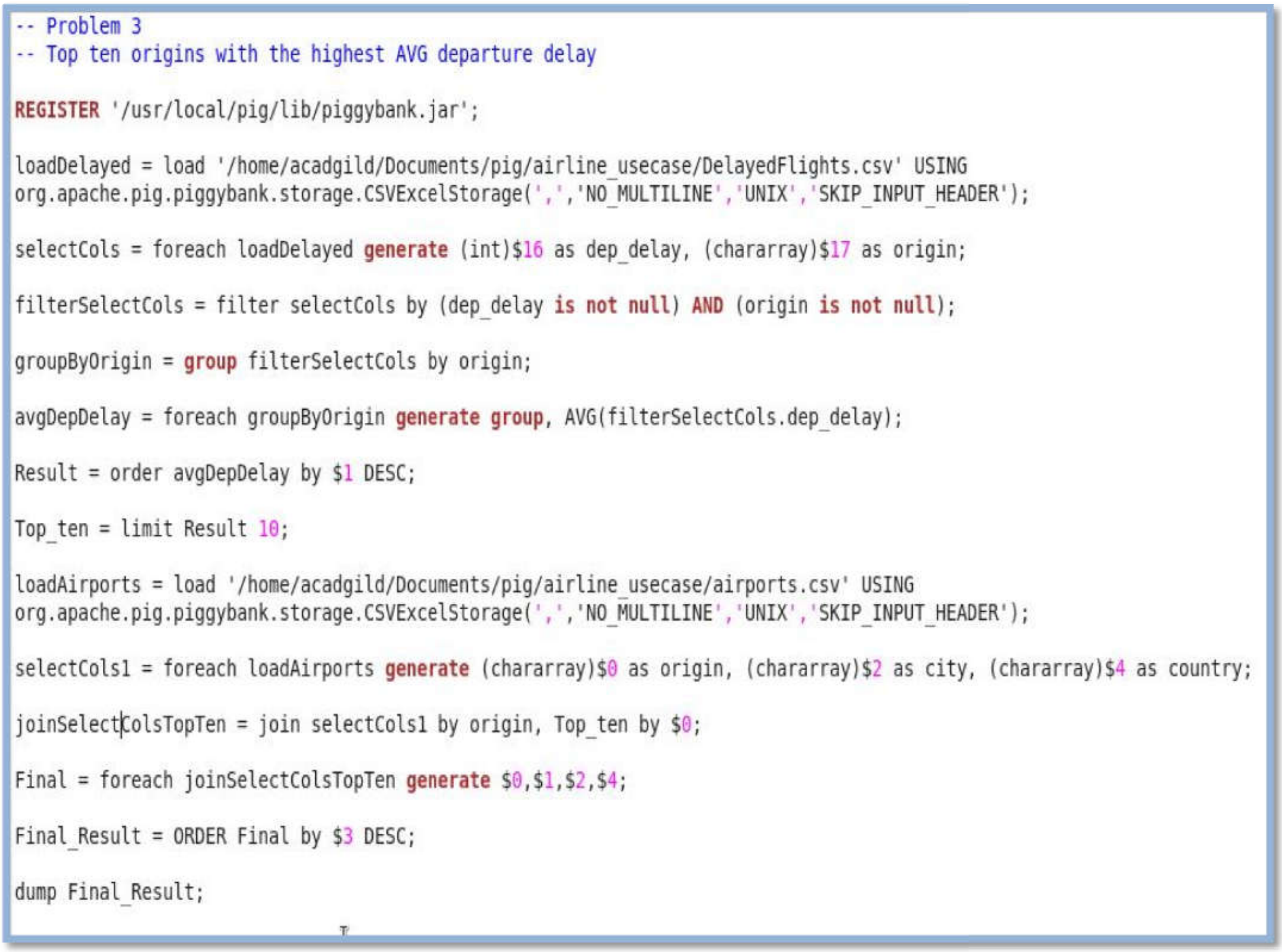
```
[acadgild@localhost pig]$ pig -x local assign5.2_airline_Problem2.pig
```

```
Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already
initialized
2017-08-13 23:24:20,818 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics -
Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already
initialized
2017-08-13 23:24:20,854 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2017-08-13 23:24:20,871 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-08-13 23:24:20,879 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-08-13 23:24:20,880 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-08-13 23:24:20,880 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2017-08-13 23:24:20,927 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2017-08-13 23:24:20,927 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(12,250)
2017-08-13 23:24:21,289 [main] INFO  org.apache.pig.Main - Pig script completed
in 1 minute, 10 seconds and 27 milliseconds (70027 ms)
[acadgild@localhost pig]$
```

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"



```
-- Problem 3
-- Top ten origins with the highest AVG departure delay

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

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

selectCols = foreach loadDelayed generate (int)$16 as dep_delay, (chararray)$17 as origin;

filterSelectCols = filter selectCols by (dep_delay is not null) AND (origin is not null);

groupByOrigin = group filterSelectCols by origin;

avgDepDelay = foreach groupByOrigin generate group, AVG(filterSelectCols.dep_delay);

Result = order avgDepDelay by $1 DESC;

Top_ten = limit Result 10;

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

selectCols1 = foreach loadAirports generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;

joinSelectColsTopTen = join selectCols1 by origin, Top_ten by $0;

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

Final_Result = ORDER Final by $3 DESC;

dump Final_Result;
```

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.

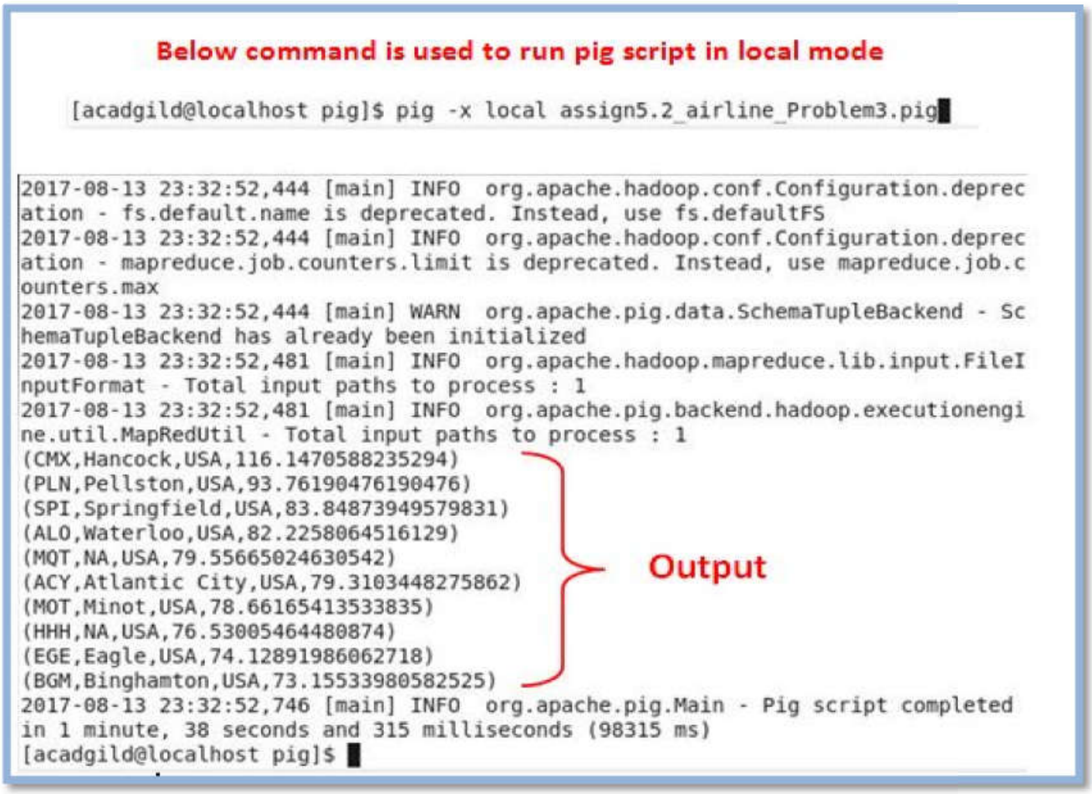
Line 10: In the relation **selectCols1,** we are generating the destination, city, and country from the **loadAirports** relation.

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.**



Below command is used to run pig script in local mode

```
[acadgild@localhost pig]$ pig -x local assign5.2_airline_Problem3.pig
```

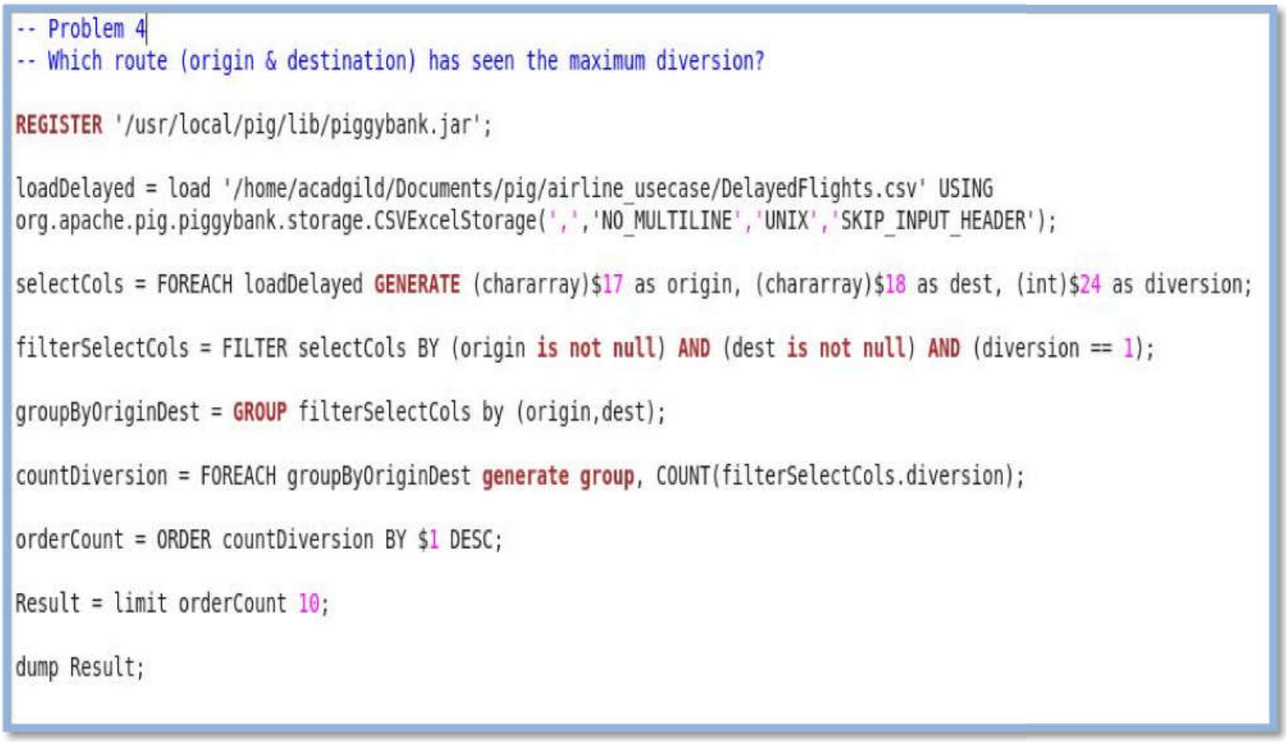
```
2017-08-13 23:32:52,444 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-08-13 23:32:52,444 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-08-13 23:32:52,444 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2017-08-13 23:32:52,481 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2017-08-13 23:32:52,481 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(CMX,Hancock,USA,116.1470588235294)
(PLN,Pellston,USA,93.76190476190476)
(SPI,Springfield,USA,83.84873949579831)
(ALO,Waterloo,USA,82.2258064516129)
(MQT,NA,USA,79.55665024630542)
(ACY,Atlantic City,USA,79.3103448275862)
(MOT,Minot,USA,78.66165413533835)
(HHH,NA,USA,76.53005464480874)
(EGE,Eagle,USA,74.12891986062718)
(BGM,Binghamton,USA,73.15533980582525)
2017-08-13 23:32:52,746 [main] INFO  org.apache.pig.Main - Pig script completed
in 1 minute, 38 seconds and 315 milliseconds (98315 ms)
[acadgild@localhost pig]$
```

Output

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"



```
-- Problem 4
-- Which route (origin & destination) has seen the maximum diversion?

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

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

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

filterSelectCols = FILTER selectCols BY (origin is not null) AND (dest is not null) AND (diversion == 1);

groupByOriginDest = GROUP filterSelectCols by (origin,dest);

countDiversion = FOREACH groupByOriginDest generate group, COUNT(filterSelectCols.diversion);

orderCount = ORDER countDiversion BY $1 DESC;

Result = limit orderCount 10;

dump Result;
```

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.

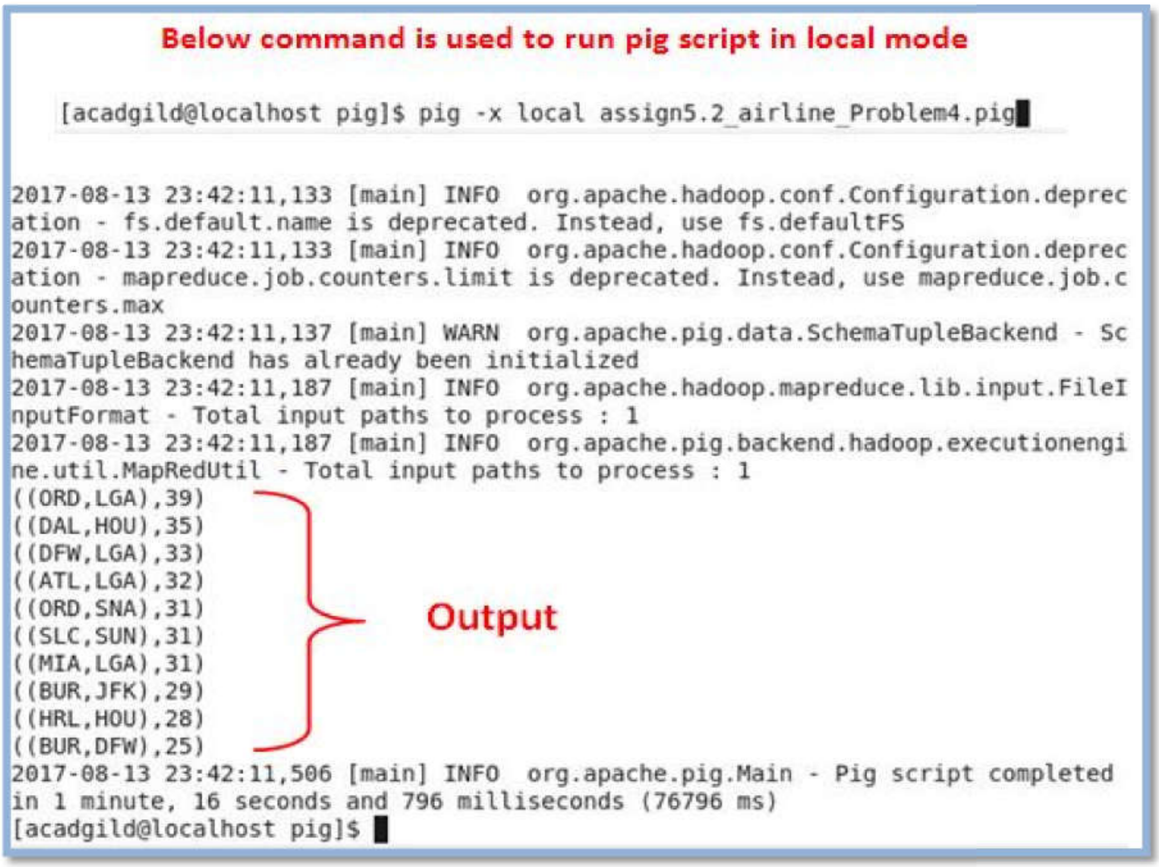
Line 5: In relation **groupByOriginDest**, we are grouping the data based on origin and destination.

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.



Below command is used to run pig script in local mode

```
[acadgild@localhost pig]$ pig -x local assign5.2_airline_Problem4.pig
```

```
2017-08-13 23:42:11,133 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-08-13 23:42:11,133 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-08-13 23:42:11,137 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2017-08-13 23:42:11,187 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2017-08-13 23:42:11,187 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 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)
2017-08-13 23:42:11,506 [main] INFO  org.apache.pig.Main - Pig script completed
in 1 minute, 16 seconds and 796 milliseconds (76796 ms)
[acadgild@localhost pig]$
```