

Pig Assignment 5.2

Airline usage test case :

Problem 1 :

Top 5 most visited stations:

Commands:

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

Custom Jar which is meant for easy handling of special characters in the input files is registered first via above command

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

The first input file is loaded from local FS.

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

The required attributes are chosen while assigning datatypes to them based on the problem statement.

C = filter B by dest is not null;

Null entries are filtered out

D = group C by dest;

The valid entries are group based on the destination and their count is also arrived at via the below command

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

The schema of E will be E : {group : chararray, long}

F = order E by \$1 DESC;

The value of relation E is sorted in descending order so the top 5 can be limited via below command

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');

The second input file containing airport details are loaded so the result obtained in the previous step can be presented with the relevant information in terms of descriptions and not codes

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

Only the relevant attributes are chosen using above command and the joined table is created based on the common attribute which is destination.

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

dump joined_table;

Screenshot and output :

```

[acadgild@localhost ~]$ chmod -R 777 airline_usecase/
[acadgild@localhost ~]$ cd airline_usecase/
[acadgild@localhost airline_usecase]$ ls -ltrh
total 238M
-rwxrwxrwx. 1 acadgild acadgild 377K Dec 17 22:50 piggybank.jar
-rwxrwxrwx. 1 acadgild acadgild 237M Dec 17 22:52 DelayedFlights.csv
-rwxrwxrwx. 1 acadgild acadgild 239K Dec 17 22:52 airports.csv
[acadgild@localhost airline_usecase]$ pig -x local
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-12-17 22:55:11,078 INFO [main] pig.ExecTypeProvider: Trying ExecType : LOCAL
2017-12-17 22:55:11,078 INFO [main] pig.ExecTypeProvider: Picked LOCAL as the ExecType
2017-12-17 22:55:11,136 [main] INFO org.apache.pig.Main - Apache Pig version 0.14.0 (r1640057) compiled Nov 16 2014, 18:02:05
2017-12-17 22:55:11,137 [main] INFO org.apache.pig.Main - Logging error messages to: /home/acadgild/airline_usecase/pig_1513531
2017-12-17 22:55:11,195 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/acadgild/.pigbootstrap not found
2017-12-17 22:55:11,477 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-12-17 22:55:11,477 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-12-17 22:55:11,478 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system: file:///
2017-12-17 22:55:11,485 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.used.genericoptionsparser is deprecated. Instead, use mapreduce.client.genericoptionsparser.used
2017-12-17 22:55:11,655 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> REGISTER '/home/acadgild/airline_usecase/piggybank.jar';
2017-12-17 22:55:33,259 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.persist.jobstatus.hours is deprecated. Instead, use mapreduce.jobtracker.persist.jobstatus.hours
2017-12-17 22:55:33,259 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.heartbeat.in.second is deprecated. Instead, use mapreduce.jobtracker.heartbeat.in.second
2017-12-17 22:55:33,259 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - jobclient.completion.poll.interval is deprecated. Instead, use mapreduce.jobclient.completion.poll.interval

```

```

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(',', 'UNIX', 'SKIP_INPUT_HEADER');
2017-12-17 22:59:19,780 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-12-17 22:59:19,784 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-12-17 22:59:19,784 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> A2 = foreach A1 generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as country;
grunt> joined_table = join Result by $0, A2 by dest;
grunt> dump joined_table;

```

```

ad, use mapreduce.job.counters.max
2017-12-17 23:00:09,765 [main] WARN org.apache.
2017-12-17 23:00:09,782 [main] INFO org.apache.
2017-12-17 23:00:09,782 [main] INFO org.apache.
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)
grunt>

```

Problem 2 :

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;

dump Result;

Output :

```

(LAX,59969)
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> dump Result;

```

```

2017-12-17 23:27:16
(12,250)
grunt>

```

Problem 3 :

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;

dump Final_Result;

```
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');
2017-12-17 23:50:46,698 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.max
ad, use mapreduce.job.counters.max
2017-12-17 23:50:46,698 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is
dfs.bytes-per-checksum
2017-12-17 23:50:46,698 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated
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> dump Final_Result;
```

```
(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)
grunt> █
```

Problem 4 :

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','SK
IP_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;

dump Result;

Execution & Output :

```
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> dump Result;
```

```
((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> █
```