

Assignment_7.1

Task:1

Write a program to implement wordcount using Pig.

```
lines =load '/user/word-count.txt' as (line:chararray);  
  
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;  
  
grouped = GROUP words by word;  
  
wordcount = FOREACH grouped GENERATE group, COUNT(words);  
  
DUMP wordcount;
```

Output:-



```
.  
2018-12-04 23:39:45,863 [main] INFO org.apache.hadoop.mapreduce.lib.input  
2018-12-04 23:39:45,872 [main] INFO org.apache.pig.backend.hadoop.execut  
(I,1)  
(My,2)  
(am,1)  
(is,2)  
(and,1)  
(Name,1)  
(Nick,1)  
(name,1)  
(Hadoop,1)  
(Spark.,1)  
(Shanky.,1)  
(learning,1)  
(Shashank.,1)  
grunt>
```

obaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Task 2

We have employee_details and employee_expenses files. Use local mode while running Pig and write Pig Latin script to get below results:

employee_details (EmpID,Name,Salary,EmployeeRating)

https://github.com/prateekATacagdild/DatasetsForCognizant/blob/master/employee_details.txt

employee_expenses(EmpID,Expenct)

https://github.com/prateekATacagdild/DatasetsForCognizant/blob/master/employee_expenses.txt

(a) Top 5 employees (employee id and employee name) with highest rating. (In case two employees have same rating, employee with name coming first in dictionary should get preference)

employee = LOAD '/home/acadgild/employee_details.txt' USING PigStorage(',') as

```

(id:int,name:chararray,salary:int,rating:int);

emp_rating_order = ORDER employee BY rating DESC, name ASC;

emp_rating_limit = LIMIT emp_rating_order 5;

final_output = FOREACH emp_rating_limit GENERATE $0,$1;

dump final_output;

```

```

dfs.bytes-per-checksum
2018-12-12 00:05:29,938 [main] INFO
faultFS
2018-12-12 00:05:29,939 [main] WARN
2018-12-12 00:05:30,023 [main] INFO
2018-12-12 00:05:30,024 [main] INFO
(105,Pawan)
(110,Priyanka)
(104,Anubhav)
(109,Katrina)
(103,Akshay)
grunt>

```

baXterm by subscribing to the professional edition here: [h](#)

(b) Top 3 employees (employee id and employee name) with highest salary, whose employee id is an odd number. (In case two employees have same salary, employee with name coming first in dictionary should get preference)

```

employee = LOAD '/home/acadgild/employee_details.txt' USING PigStorage(',') as
(id:int,name:chararray,salary:int,rating:int);

odd_id = FILTER employee By id%2==1;

emp_high_salary = ORDER odd_id by salary DESC, name ASC;

emp_salary_limit = LIMIT emp_high_salary 3;

output2 = FOREACH emp_salary_limit GENERATE $0,$1;

dump output2;

```

```
, sessionId= - already initialized
2018-12-12 00:13:42,120 [main] INFO or
, sessionId= - already initialized
2018-12-12 00:13:42,141 [main] INFO or
2018-12-12 00:13:42,148 [main] INFO or
dfs.bytes-per-checksum
2018-12-12 00:13:42,149 [main] INFO or
FaultFS
2018-12-12 00:13:42,149 [main] WARN or
2018-12-12 00:13:42,234 [main] INFO or
2018-12-12 00:13:42,235 [main] INFO or
(101,Amitabh)
(107,Salman)
(103,Akshay)
grunt>
```

(c) Employee (employee id and employee name) with maximum expense (In case two employees have same expense, employee with name coming first in dictionary should get preference)

```
details = LOAD '/home/acadgild/employee_details.txt' USING PigStorage(',') as
(id:int,name:chararray,salary:int,rating:int);

expense = LOAD '/home/acadgild/employee_expense.txt' USING PigStorage('\t') as (id:int,expense:int);

result = JOIN details BY id,expense BY id;

top_expense_list = GROUP result BY expense;

B = ORDER top_expense_list BY $0 DESC;

top_expense = LIMIT B 1;

C = FOREACH top_expense GENERATE FLATTEN(top_expense.$1);

output3 = FOREACH C GENERATE $0,$1;

DUMP output3;
```

```
, sessionId= - already initialized
2018-12-12 00:41:03,418 [main] INFO org.apache.hadoop.
, sessionId= - already initialized
2018-12-12 00:41:03,442 [main] INFO org.apache.pig.bac
2018-12-12 00:41:03,474 [main] INFO org.apache.hadoop.
dfs.bytes-per-checksum
2018-12-12 00:41:03,477 [main] INFO org.apache.hadoop.
faultFS
2018-12-12 00:41:03,478 [main] WARN org.apache.pig.dat
2018-12-12 00:41:03,560 [main] INFO org.apache.hadoop.
2018-12-12 00:41:03,561 [main] INFO org.apache.pig.bac
(110,Priyanka)
(102,Shahrukh)
grunt>
```

Learn more about this to the professional edition here: <https://books.google.com/books?id=...>

(d) List of employees (employee id and employee name) having entries in employee_expenses file.

```

details = LOAD '/home/acadgild/employee_details.txt' USING PigStorage(',') as
(id:int,name:chararray,salary:int,rating:int);

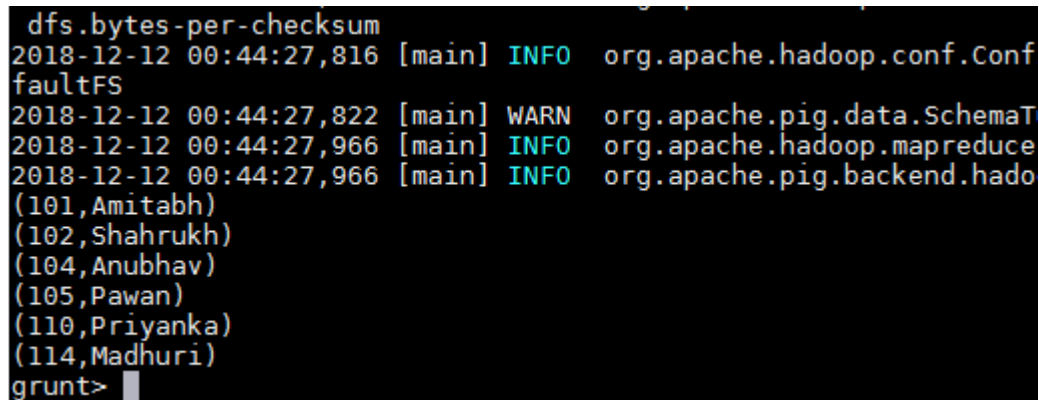
expense = LOAD '/home/acadgild/employee_expense.txt' USING PigStorage('\t') as (id:int,expense:int);

result = JOIN details BY id,expense BY id;

output4 = DISTINCT(Foreach result Generate $0,$1);

DUMP output4;

```



```

dfs.bytes-per-checksum
2018-12-12 00:44:27,816 [main] INFO org.apache.hadoop.conf.Configuration
2018-12-12 00:44:27,822 [main] WARN org.apache.pig.data.SchemaTool
2018-12-12 00:44:27,966 [main] INFO org.apache.hadoop.mapreduce
2018-12-12 00:44:27,966 [main] INFO org.apache.pig.backend.hadoop
(101,Amitabh)
(102,Shahrukh)
(104,Anubhav)
(105,Pawan)
(110,Priyanka)
(114,Madhuri)
grunt>

```

baXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

(e) List of employees (employee id and employee name) having no entry in employee_expenses file.

```

details = LOAD '/home/acadgild/employee_details.txt' USING PigStorage(',') as
(id:int,name:chararray,salary:int,rating:int);

expense = LOAD '/home/acadgild/employee_expense.txt' USING PigStorage('\t') as (id:int,expense:int);

result = JOIN details BY id LEFT OUTER,expense BY id;

emp_without_exp = FILTER result BY expense::id is null;

output5 = FOREACH emp_without_exp Generate $0,$1;

DUMP output5;

```

```

2018-12-12 00:50:13,774 [main] INFO
faultFS
2018-12-12 00:50:13,774 [main] WARN
2018-12-12 00:50:13,888 [main] INFO
2018-12-12 00:50:13,888 [main] INFO
(103,Akshay)
(106,Aamir)
(107,Salman)
(108,Ranbir)
(109,Katrina)
(111,Tushar)
(112,Ajay)
(113,Jubeen)
grunt>

```

baXterm by subscribing to the professional edition here: [ht](#)

Task 3

Implement the use case present in below blog link and share the complete steps along with screenshot(s) from your end.

<https://acadgild.com/blog/aviation-data-analysis-using-apache-pig/>

Problem Statement 1: Find out the top 5 most visited destinations

```

< wordcount_pig.pig > employee_details.txt > employee_expenses.txt > Pig Commands > aviation1.pig >
REGISTER '/home/acadgild/user_acadgild/assignments/Pig/Aviation_Usecase/piggybank-0.15.0.jar';

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

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/user_acadgild/assignments/Pig/Aviation_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;

dump joined_table;

```

The piggybank.jar is registered i.e the methods in the jar file can be used in the program.

The DelayedFlights.CSV file is loaded.

The required columns are fetched and iterated.

To filter out the columns whose destination is null, we use a filter function.

Tuples with the same destination are grouped together and the number of tuples for each destination is counted.

Creating a descending ordered list of tuples and limiting them to 5.

Loading the airports.csv to find out the names of the corresponding 5 destinations by joining them.

Display the final result on the output.

Command :

```
pig -x local /home/acadgild/user_acadgild/assignments/Pig/aviation1.pig
```

```
cess : 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)
```

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

```
word-count.txt  wordcount_pig.pig  Pig Commands  aviation1.pig  aviation2.pig  
REGISTER '/home/acadgild/user_acadgild/assignments/Pig/Aviation_Usecase/piggybank-0.15.0.jar';  
A = load '/home/acadgild/user_acadgild/assignments/Pig/Aviation_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;
```

The piggybank.jar is registered i.e the methods in the jar file can be used in the program.

The DelayedFlights.CSV file is loaded.

The required columns are fetched and iterated.

To filter out the columns where canceled = 1 meaning that it has been canceled and cancel_code = 0 indicating the reason for canceling is due to bad weather.


Tuples are grouped together based on month and the number of tuples for each month is counted.

Creating a descending ordered list of tuples and limiting them to top 1.

Display the final result on the output.

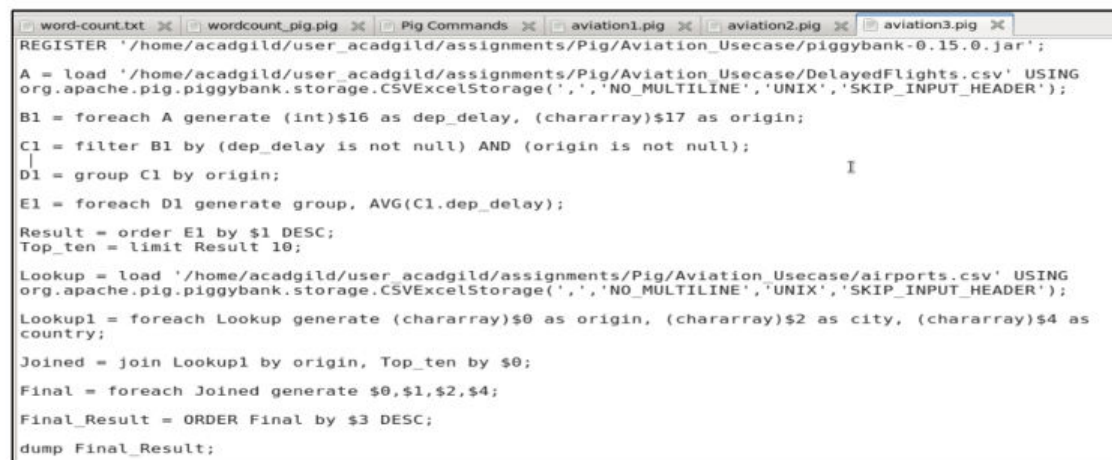
Command :

pig -x local /home/acadgild/user_acadgild/assignments/Pig/aviation2.pig



```
cess : 1
(12,250)
```

Problem Statement 3: Top ten origins with the highest AVG departure delay



```
word-count.txt  wordcount_pig.pig  Pig Commands  aviation1.pig  aviation2.pig  aviation3.pig
REGISTER '/home/acadgild/user_acadgild/assignments/Pig/Aviation_Usecase/piggybank-0.15.0.jar';
A = load '/home/acadgild/user_acadgild/assignments/Pig/Aviation_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/user_acadgild/assignments/Pig/Aviation_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;
```


The piggybank.jar is registered i.e the methods in the jar file can be used in the program.

The DelayedFlights.CSV file is loaded.

The required columns are iterated.

To filter out the columns whose departure delay and origin not is null, we use a filter function.

Tuples with the same origin are grouped together and the average (AVG) departure delay of tuples for each origin is counted.

Creating a descending ordered list of tuples and limiting them to 10.

Loading the airports.csv to find out the names of the corresponding 10 origins' airports by joining them.

Display the final result on the output.

```
cess : 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)
```

Problem Statement 4: Which route (origin & destination) has seen the maximum diversion?

```
< wordcount_pig.pig x Pig Commands x aviation1.pig x aviation2.pig x aviation3.pig x aviation4.pig x
REGISTER '/home/acadgild/user_acadgild/assignments/Pig/Aviation_Usecase/piggybank-0.15.0.jar';
A = load '/home/acadgild/user_acadgild/assignments/Pig/Aviation_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;
dump Result;
```


The piggybank.jar is registered i.e the methods in the jar file can be used in the program.

The DelayedFlights.CSV file is loaded.

The required columns are iterated.

To filter out the columns whose destination and origin is not null also the diversion is 1 indicating that the flight was diverted, we use a filter function.

Tuples are grouped together based on destination and origin and the diversions of are counted.

Creating a descending ordered list of tuples and limiting them to 10.

Display the final result on the output.

```
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)
```

