Session 7: EXPLORING APACHE PIG

Assignment

Task 1

Write a program to implement wordcount using Pig.

```
A = load '/test.txt';
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as word;
C = group B by word;
D = foreach C generate group, COUNT(B);
dump D;
```


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:

(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)

```
(World.Have,1)
grunt> emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',')  AS (emp_id:int, emp_name:chararray, emp_salary:int,emp_rat
ing:int);
2018-08-19 14:52:39,006 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> dump emp;
2018-08-19 14:53:05,012 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2018-08-19 14:53:05,072 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
```

```
2018-08-19 14:53:37,028 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,Amitabh,20000,1)
(102,Shahrukh,10000,2)
(103,Akshay,11000,3)
(104,Anubhav,5000,4)
(105,Pawan,2500,5)
(106,Aamir,25000,1)
(107,Salman,17500,2)
(108,Ranbir,14000,3)
(109,Katrina,1000,4)
(110,Priyanka,2000,5)
(111,Tushar,500,1)
(112,Ajay,5000,2)
(113,Jubeen,1000,1)
(114,Madhuri,2000,2)
grunt>
```

```
Details at logfile: /home/acadgild/pig_1534668789659.log
grunt> emp_expenses= LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int,expenses:int);
2018-08-19 14:57:08,090 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> dump emp_expenses;
2018-08-19 14:57:16,903 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2018-08-19 14:57:16,973 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
```

```
2018-08-19 14:57:43,663 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,200)
(102,100)
(110,400)
(114,200)
(119,200)
(105,100)
(101,100)
(104,300)
(102,400)
grunt>
```

Pig Query:
emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int, emp_name:chararray, emp_salary:int,emp_rating:int);
rating = order emp by emp_rating DESC;
Result = LIMIT rating 5;
Dump Result;

```
(102,400)
grunt> emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int,
>> emp_name:chararray, emp_salary:int,emp_rating:int);
2018-08-19 14:59:25,966 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> rating = order emp by emp_rating DESC;
grunt> Result = LIMIT rating 5;
grunt> Dump Result;
2018-08-19 15:00:02,106 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: ORDER_BY,LIMIT
2018-08-19 15:00:02,253 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
2018-08-19 15:00:02,255 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code
.
2018-08-19 15:00:02,256 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMa
pKeyPrune, ConstantCalculator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, PartitionFilt
erOptimizer, PredicatePushdownOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2018-08-19 15:00:02,275 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation thresho
```

```
2018-08-19 15:02:23,703 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-19 15:02:23,703 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(110,Priyanka,2000,5)
(105,Pawan,2500,5)
(109,Katrina,1000,4)
(104,Anubhav,5000,4)
(108,Ranbir,14000,3)
grunt>
```

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)

Pig Query:
emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int, emp_name:chararray, emp_salary:int,emp_rating:int);
emp_sal_name = order emp by emp_salary desc;
emp_sal_id = FILTER emp_sal_name by emp_id%2==1;
emp_final = FOREACH emp_sal_id generate emp_id,emp_name;
emp_final_limit = LIMIT emp_final 3;


```
Details at logfile: /home/acadgild/pig_1534668789659.log
grunt> emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int,
>> emp_name:chararray, emp_salary:int,emp_rating:int);
2018-08-19 15:08:34,571 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> emp_sal_name = order emp by emp_salary desc;
grunt> emp_sal_id = FILTER emp_sal_name by emp_id%2==1;
grunt> emp_final = FOREACH emp_sal_id generate emp_id,emp_name;
grunt> emp_final_limit = LIMIT emp_final 3;
grunt> dump emp_final_limit;
2018-08-19 15:09:43,283 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: ORDER_BY,FILTER,LIMIT
2018-08-19 15:09:43,323 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
```


```
2018-08-19 15:11:58,908 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-19 15:11:58,908 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(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)

Pig_Query:
emp = LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int, emp_name:chararray, emp_salary:int);
emp_expenses = LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int, expenses:int);
Joinempempexpense = join emp by emp_id, emp_expenses by emp_id;
maxexpense = ORDER Joinempempexpense by emp_expenses::expenses desc;
Limitmaxepnse = LIMIT maxexpense 1;
Limitmaxepensefinal = foreach Limitmaxepnse generate emp::emp_id,emp::emp_name;

dump Limitmaxexpensefinal;

```
grunt> emp = LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int,
>> emp_name:chararray, emp_salary:int);
2018-08-19 15:15:04,595 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> emp_expenses = LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int,
>> expenses:int);
2018-08-19 15:15:23,428 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> Joinempempexpense = join emp by emp_id, emp_expenses by emp_id;
grunt> maxexpense = ORDER Joinempempexpense by emp_expenses::expenses desc;
grunt> Limitmaxepnse = LIMIT maxexpense 1;
grunt> Limitmaxexpensefinal = foreach Limitmaxepnse generate emp::emp_id,emp::emp_name;
grunt> dump Limitmaxexpensefinal;
2018-08-19 15:16:41,911 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN,ORDER_BY,LIMIT
```

```
2018-08-19 15:19:16,076 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-19 15:19:16,076 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(110,Priyanka)
grunt>
```

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

Pig Query:
emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int,
emp_name:chararray, emp_salary:int,emp_rating:int);
emp_expenses = LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int,
expenses:int);
emp_with_exp = JOIN emp BY emp_id, emp_expenses BY emp_id;
emp_with_exp_data = FOREACH emp_with_exp GENERATE emp::emp_id,
emp::emp_name;
emp_with_exp_distinct_data = DISTINCT emp_with_exp_data;
dump emp_with_exp_distinct_data;

```
(110,Priyanka)
grunt> emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int,
>> emp_name:chararray, emp_salary:int,emp_rating:int);
2018-08-19 15:21:43,364 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> emp_expenses = LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int,
>> expenses:int);
2018-08-19 15:21:43,637 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
grunt> emp_with_exp = JOIN emp BY emp_id, emp_expenses BY emp_id;
grunt> emp_with_exp_data = FOREACH emp_with_exp GENERATE emp::emp_id,
>> emp::emp_name;
grunt> emp_with_exp_distinct_data = DISTINCT emp_with_exp_data;
grunt> dump emp_with_exp_distinct_data;
2018-08-19 15:21:44,929 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN,DISTINCT
2018-08-19 15:21:45,004 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.de
faultFS
```

```
2018-08-19 15:23:02,076 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-19 15:23:02,076 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,Amitabh)
(102,Shahrukh)
(104,Anubhav)
(105,Pawan)
(110,Priyanka)
(114,Madhuri)
grunt>
```

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

file.

Pig Query:
emp= LOAD '/hadoopdata/pig/employee_details.txt' USING PigStorage(',') AS (emp_id:int, emp_name:chararray, emp_salary:int,emp_rating:int);
emp_expenses = LOAD '/hadoopdata/pig/employee_expenses.txt' AS (emp_id:int, expenses:int);
emp_without_exp = JOIN emp BY emp_id LEFT OUTER, emp_expenses BY emp_id;
emp_without_exp_filter = FILTER emp_without_exp BY emp_expenses::emp_id is null;
emp_without_exp_filter_data = FOREACH emp_without_exp_filter GENERATE
emp::emp_id, emp::emp_name;
DUMP emp_without_exp_filter_data;

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.
REGISTER '/hadoopdata/pig/piggybank.jar';
A = load '/hadoopdata/pig/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;
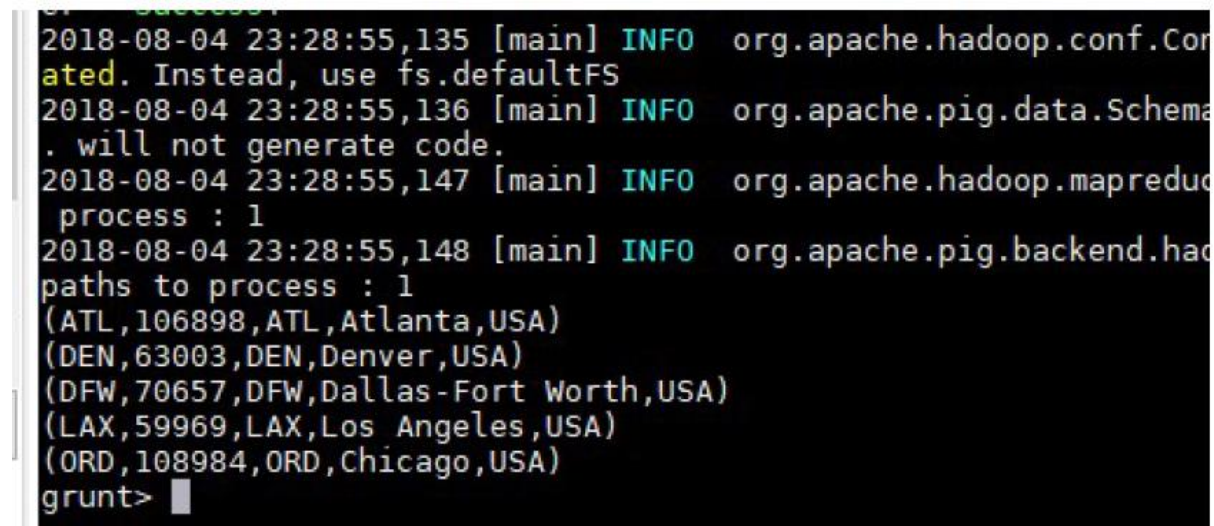C = filter B by dest is not null;

```
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 'hadoopdata/pig/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_I
NPUT_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;
```



```
2018-08-04 23:28:55,135 [main] INFO  org.apache.hadoop.conf.Con
ated. Instead, use fs.defaultFS
2018-08-04 23:28:55,136 [main] INFO  org.apache.pig.data.Schema
. will not generate code.
2018-08-04 23:28:55,147 [main] INFO  org.apache.hadoop.mapredud
 process : 1
2018-08-04 23:28:55,148 [main] INFO  org.apache.pig.backend.had
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)
grunt>
```

Problem Statement 2

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

Source code
REGISTER '/hadoopdata/pig/piggybank.jar';
A = load '/hadoopdata/pig/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_I
NPUT_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;

```
ted. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-08-04 23:35:28,852 [main] INFO  org.apache.pig.backend.hadoop.executi
er - Success!
2018-08-04 23:35:28,852 [main] INFO  org.apache.hadoop.conf.Configuration
ated. Instead, use fs.defaultFS
2018-08-04 23:35:28,856 [main] INFO  org.apache.pig.data.SchemaTupleBacker
. will not generate code.
2018-08-04 23:35:28,862 [main] INFO  org.apache.hadoop.mapreduce.lib.input
 process : 1
2018-08-04 23:35:28,862 [main] INFO  org.apache.pig.backend.hadoop.executi
paths to process : 1
(12,250)
grunt>
```

Problem Statement 3

Top ten origins with the highest AVG departure delay
Source code

```
A = load '/hadoopdata/pig/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_I
NPUT_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 '/hadoopdata/pig/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_I
NPUT_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;
```

```
2018-08-04 23:41:44,624 [main] INFO   org.apache.hadoop.conf.Configuration
ated. Instead, use fs.defaultFS
2018-08-04 23:41:44,624 [main] INFO   org.apache.pig.data.SchemaTupleBacke
. will not generate code.
2018-08-04 23:41:44,632 [main] INFO   org.apache.hadoop.mapreduce.lib.inpu
 process : 1
2018-08-04 23:41:44,632 [main] INFO   org.apache.pig.backend.hadoop.execut
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)
grunt>
```

Problem Statement 4

Which route (origin & destination) has seen the maximum diversion?
Source code
A = load '/hadoopdata/pig/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_I
NPUT_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;

```
er - Success!
2018-08-04 23:45:25,381 [main] INFO   org.apache.hadoop.conf.C
ated. Instead, use fs.defaultFS
2018-08-04 23:45:25,384 [main] INFO   org.apache.pig.data.Sche
. will not generate code.
2018-08-04 23:45:25,387 [main] INFO   org.apache.hadoop.mapred
 process : 1
2018-08-04 23:45:25,387 [main] INFO   org.apache.pig.backend.h
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)
grunt>
```