

## Task 1

Write a program to implement wordcount using Pig.

Solution:

```
lines = LOAD 'demo.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;
```

```
2018-10-17 20:38:25,789 [main] WARN
2018-10-17 20:38:25,802 [main] INFO
2018-10-17 20:38:25,802 [main] INFO
(.,3)
(my,1)
(data,1)
(file,2)
(Datta,1)
(Rahul,1)
(Shobha,1)
(sample,1)
(Motiram,1)
(Ningole,4)
(.,0)
grunt> █
```

## Task 2

(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)

Solution:

1. emp\_details = LOAD 'employee\_details.txt' Using PigStorage(',') as (EmpID:int,Name:chararray,Salary:int,EmployeeRating:int);
2. emp\_grp = GROUP emp\_details by EmployeeRating;
3. emp\_order = ORDER emp\_grp by group desc;
4.  
data\_top = FOREACH emp\_order {  
    top = TOP(1, 0, emp\_details);  
    GENERATE top;  
}
5.  
getIdName = FOREACH data\_top GENERATE emp\_details.EmpID,emp\_details.Name;

```

grunt> emp_details = LOAD 'employee details.txt' Using PigStorage(',') as (EmpID:int,Name:chararray,Salary:int,EmployeeRating:int);
grunt> emp_grp = GROUP emp_details by EmployeeRating;
grunt> emp_order = ORDER emp_grp by group desc;
grunt> data_top = FOREACH emp_order {
>>   top = TOP(1, 0, emp_details);
>>   GENERATE top;
>> }
grunt> getIdName = FOREACH data_top GENERATE emp_details.EmpID,emp_details.Name;
grunt>

```

```

2018-10-17 20:46:02,941 [main] :
2018-10-17 20:46:02,941 [main] :
({{110}},{{Priyanka}})
({{109}},{{Katrina}})
({{108}},{{Ranbir}})
({{114}},{{Madhuri}})
({{113}},{{Jubeen}})
grunt>

```

B)

(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)

Solution:

SPLIT emp\_details into empwithevenid if  $\text{EmpID} \% 2 == 0$ , empwithevenodd if  $\text{EmpID} \% 2 \neq 0$ ;  
empoddhighestsalary = ORDER empwithevenodd by Salary desc;

empoddhighestsalarytop3 = LIMIT empoddhighestsalary 3;

empoddhighestsalarytop3idname = FOREACH empoddhighestsalarytop3 GENERATE  
EmpID,Name;

```

Applications  Places  Terminal
root@hdpmaster:/home/admin/Documents/pig - Terminal

File Edit View Search Terminal Tabs Help

root@hdpmaster:/home/admin/Documents/pig - Terminal

Output(s):
Successfully stored 7 records in: "file:/tmp/temp-1005967909/tmp-565879807"

Counters:
Total records written : 7
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local167194793_0010

2018-10-17 15:00:54,405 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId=
- already initialized
2018-10-17 15:00:54,406 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId=
- already initialized
2018-10-17 15:00:54,406 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId=
- already initialized
2018-10-17 15:00:54,407 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2018-10-17 15:00:54,408 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-10-17 15:00:54,418 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-10-17 15:00:54,418 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,Amitabh,20000,1)
(103,Akshay,11000,3)
(105,Pawan,2500,5)
(107,Salman,17500,2)
(109,Katrina,1000,4)
(111,Tushar,500,1)
(113,Jubeen,1000,1)
grunt>

```

```

2018-10-17 15:09:44,952 [main
2018-10-17 15:09:44,952 [main
(101,Amitabh)
(107,Salman)
(103,Akshay)

```

C)

(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)

Solution:

```
employee_expenses = LOAD 'employee_expenses.txt' Using PigStorage('\t') as
(EID:int,Expenditure:int);
```

```
emp_details_expenses = JOIN emp_details BY EmpID, employee_expenses BY EID;
```

```
getListOfEmpviaExpenses = ORDER emp_details_expenses by Expenditure
```

```
grpEmpExpenses = GROUP emp_details_expenses by Expenditure;
```

```
orderbyhighestexpenses = ORDER grpEmpExpenses by group desc;
```

```
data_top = FOREACH orderbyhighestexpenses {
    top = TOP(1, 0, emp_details_expenses);
    GENERATE top;
}
```

```
CfinalResult = FOREACH data_top GENERATE
emp_details_expenses.EmpID,emp_details_expenses.Name;
```

```
(101,Amitabh,20000,1,101,100)
(101,Amitabh,20000,1,101,200)
(102,Shahrukh,10000,2,102,400)
(102,Shahrukh,10000,2,102,100)
(104,Anubhav,5000,4,104,300)
(105,Pawan,2500,5,105,100)
(110,Priyanka,2000,5,110,400)
(114,Madhuri,2000,2,114,200)
```

```
grunt> grpEmpExpenses = GROUP emp_details_expenses by Expence;
grunt> █
```

```
2018-10-17 16:32:45,621 [main] INFO
2018-10-17 16:32:45,621 [main] INFO
({(110,Priyanka,2000,5,110,400)})
({(104,Anubhav,5000,4,104,300)})
({(114,Madhuri,2000,2,114,200)})
({(105,Pawan,2500,5,105,100)})
```

```
2018-10-17 16:37:11,495 [ma
({(110)}},{(Priyanka)})
({(104)}},{(Anubhav)})
({(114)}},{(Madhuri)})
({(105)}},{(Pawan)})
```

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

Solution:

```
joinwithemployee_expenses = JOIN emp_details BY EmpID RIGHT, employee_expenses BY EID;
```

```
datainemployee_expenses = FILTER joinwithemployee_expenses;
```

```
getID = FILTER joinwithemployee_expenses by EmpID is not null;
```

```
(101,Amitabh,20000,1,101,100)
(101,Amitabh,20000,1,101,200)
(102,Shahrukh,10000,2,102,400)
(102,Shahrukh,10000,2,102,100)
(104,Anubhav,5000,4,104,300)
(105,Pawan,2500,5,105,100)
(110,Priyanka,2000,5,110,400)
(114,Madhuri,2000,2,114,200)
(,,,119,200)
```

```

2018-10-17 17:09:24,103 [main] INFO org.apache.pig.backend.hadoop.executionengine
(101,Amitabh,20000,1,101,100)
(101,Amitabh,20000,1,101,200)
(102,Shahrukh,10000,2,102,400)
(102,Shahrukh,10000,2,102,100)
(104,Anubhav,5000,4,104,300)
(105,Pawan,2500,5,105,100)
(110,Priyanka,2000,5,110,400)
(114,Madhuri,2000,2,114,200)
grunt> getID = FILTER joinwithemployee_expenses by EmpID is not null;

```

E:

List of employees (employee id and employee name) having no entry in employee\_expenses file.

Solution:

```
joinwithemployee_details = JOIN emp_details BY EmpID LEFT, employee_expenses BY EID;
```

```
no_entry_in_employee_expenses = FILTER joinwithemployee_details by EID is null;
```

```
no_entry_in_employee_expenses_id_name = FOREACH no_entry_in_employee_expenses
GENERATE EmpID,Name;
```

```
dump no_entry_in_employee_expenses_id_name;
```

```

(103,Akshay,11000,3,,)
(106,Aamir,25000,1,,)
(107,Salman,17500,2,,)
(108,Ranbir,14000,3,,)
(109,Katrina,1000,4,,)
(111,Tushar,500,1,,)
(112,Ajay,5000,2,,)
(113,Jubeen,1000,1,,)
grunt> no_entry_in_employee_expenses_id_name = FOREACH no_entry_in_employee_expenses GENERATE EmpID,Name;
grunt> dump no_entry_in_employee_expenses_id_name

```

```

(103,Akshay)
(106,Aamir)
(107,Salman)
(108,Ranbir)
(109,Katrina)
(111,Tushar)
(112,Ajay)
(113,Jubeen)

```