Session 5: EXPLORIN PIG

Assignment 5.1

Student Name: Subham Vishal

Course: Big Data Hadoop & Spark Training

Assignment 5.1 – We have **employee_details** and **employee_expenses** files. Use local mode while running Pig and write Pig Latin script to get below results:

Contents

Introduction:

In this assignment we are going to write PIG Latin codes for the below tasks.

Input Data Sets:

employee_details.txt:

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

```
101,Amitabh,20000,1
102,Shahrukh,10000,
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
```

employee_expenses.txt:

https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee_expense s.txt

```
101     200
102     100
110     400
114     200
119     200
105     100
101     100
104     300
102     400
```

Load Data into Pig Storage:

Before going into the tasks we are loading the data sets to Pig Storage.

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

employee_expenses = LOAD '/home/acadgild/hadoop/employee_expenses.txt' USING PigStorage() AS (id:int, expenses:int);

DUMP employee_details;

```
(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>
```

DUMP employee_expenses;

```
(101,200)
(102,100)
(110,400)
(114,200)
(119,200)
(105,100)
(101,100)
(104,300)
(102,400)
grunt>
```

Task a

(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)

Code

We are using the function ORDER to order the rating and the ID by ascending order and we are limiting the employee by top 5 with id and the name.

employee = ORDER employee_details BY rating asc, id asc;

limit_employee = LIMIT employee 5;

top_employee = FOREACH limit_employee GENERATE id,name;

```
(101,Amitabh,20000,1)
(106,Aamir,25000,1)
(111,Tushar,500,1)
(113,Jubeen,1000,1)
(102,Shahrukh,10000,2)
(107,Salman,17500,2)
(112,Ajay,5000,2)
(114,Madhuri,2000,2)
(103,Akshay,11000,3)
(108,Ranbir,14000,3)
(104,Anubhav,5000,4)
(109,Katrina,1000,4)
(105,Pawan,2500,5)
(110,Priyanka,2000,5)
grunt>
```

```
(101,Amitabh,20000,1)
(106,Aamir,25000,1)
(111,Tushar,500,1)
(113,Jubeen,1000,1)
(102,Shahrukh,10000,2)
grunt>
```

Output

```
(101,Amitabh)
(106,Aamir)
(111,Tushar)
(113,Jubeen)
(102,Shahrukh)
grunt>
```

Task 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)

Code

We are using the function ORDER to order the employee_details by arranging the salary in descending order and filtering the even number using the logic **(id%2)!=0** and limiting the employee by 3 and bringing the result according to the employee ID and Name.

1. ***order_employee = ORDER employee_details BY salary DESC;***
2. ***filter_employee = FILTER order_employee BY (id%2)!=0;***
3. ***limit_employee = LIMIT filter_employee 3;***
4. ***top_3employee = FOREACH limit_employee GENERATE id,name;***
5. ***DUMP top_3employee;***
1.

```
(106,Aamir,25000,1)
(101,Amitabh,20000,1)
(107,Salman,17500,2)
(108,Ranbir,14000,3)
(103,Akshay,11000,3)
(102,Shahrukh,10000,2)
(112,Ajay,5000,2)
(104,Anubhav,5000,4)
(105,Pawan,2500,5)
(110,Priyanka,2000,5)
(114,Madhuri,2000,2)
(109,Katrina,1000,4)
(113,Jubeen,1000,1)
(111,Tushar,500,1)
grunt>
grunt>
```

2.
3.
4.

```
(101,Amitabh,20000,1)
(107,Salman,17500,2)
(103,Akshay,11000,3)
(105,Pawan,2500,5)
(113,Jubeen,1000,1)
(109,Katrina,1000,4)
(111,Tushar,500,1)
grunt>
```

5.
6.

```
(101,Amitabh,20000,1)
(107,Salman,17500,2)
(103,Akshay,11000,3)
grunt>
```

7.

Output

```
(101,Amitabh)
(107,Salman)
(103,Akshay)
grunt>
```

Task 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)

Code

In this task, we are joining the tables employee_details and employee_expenses using the JOIN function and sorting the expenses by Descending order and limiting the employee first 2 rows and bringing the id and name of the employee as the output.

1. ***emp_det_exp = JOIN employee_details BY id, emp_expenses by id;***
2. ***order_exp = ORDER emp_det_exp BY expenses DESC;***

3. ***limit_exp = LIMIT order_exp 2;***
4. ***max_exp = FOREACH limit_exp GENERATE $0,$1;***
5. ***DUMP max_exp;***

1.

```
(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>
grunt>
```

2.

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

3.

```
(102,Shahrukh,10000,2,102,400)
(110,Priyanka,2000,5,110,400)
grunt>
```

Output

```
(102,Shahrukh)
(110,Priyanka)
grunt>
grunt>
```

Task D

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

Code

We are joining the table's employee_details and employee_expenses using the JOIN function and removing the redundant tuples from the relation and displaying the ID and name of the employee.

1. ***emp_det_exp = JOIN employee_details BY id, emp_expenses by id;***
2. ***emp_names = FOREACH emp_det_exp GENERATE($0,$1);***
3. ***names = DISTINCT emp_names;***
4. ***DUMP names;***

Output

```
((101,Amitabh))
((101,Amitabh))
((102,Shahrukh))
((102,Shahrukh))
((104,Anubhav))
((105,Pawan))
((110,Priyanka))
((114,Madhuri))
grunt>
```

```
((101,Amitabh))
((102,Shahrukh))
((104,Anubhav))
((105,Pawan))
((110,Priyanka))
((114,Madhuri))
grunt>
```

Task E

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

Code

Using the LEFT OUTER JOIN, we are joining the table's employee_details and employee_expenses and filtering the empty values in the columns ($4: employee_expenses.id and $5: employee_expenses.expenses) and generating the output by ID and the name of the employee.

1. ***emp_det_exp = JOIN employee_details BY id LEFT OUTER, employee_expenses BY id;***
2. ***filter_emp_det_exp = FILTER emp_det_exp BY $4 Is NULL and $5 Is NULL;***
3. ***gen_emp_det_exp = FOREACH filter_emp_det_exp GENERATE $0,$1;***
4. ***DUMP gen_emp_det_exp;***

```
(101,Amitabh,20000,1,101,100)
(101,Amitabh,20000,1,101,200)
(102,Shahrukh,10000,2,102,400)
(102,Shahrukh,10000,2,102,100)
(103,Akshay,11000,3,,)
(104,Anubhav,5000,4,104,300)
(105,Pawan,2500,5,105,100)
(106,Aamir,25000,1,,)
(107,Salman,17500,2,,)
(108,Ranbir,14000,3,,)
(109,Katrina,1000,4,,)
(110,Priyanka,2000,5,110,400)
(111,Tushar,500,1,,)
(112,Ajay,5000,2,,)
(113,Jubeen,1000,1,,)
(114,Madhuri,2000,2,114,200)
grunt>
```

```
(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>
```

Output

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