**Assignment 7.2**

Associated Data Files

https://drive.google.com/file/d/0Bxr27gVaXO5sc3RFZHVvbXNDcUU/view?usp=sharing

https://drive.google.com/file/d/0Bxr27gVaXO5sOGFseVFXQ0hzdlU/view?usp=sharing

. Problem Statement

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)**

employee = load '/home/ec2-user/pigdata/employee\_details.txt' using PigStorage(',') as (id:INT,name:chararray,salary:INT,rating:INT);

order\_with\_rating = order employee BY rating desc,name asc;

top\_five = limit order\_with\_rating 5;

top\_five\_id\_name = foreach top\_five generate id, name;

dump top\_five\_id\_name

(105,Pawan)

(110,Priyanka)

(104,Anubhav)

(109,Katrina)

(103,Akshay)

**(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)**

filtered\_emp = filter employee by id%2!=0;

order\_by\_sal\_name = ORDER filtered\_emp by salary desc, name asc;

top\_three = limit order\_by\_sal\_name 3;

top\_3\_name\_id = foreach top\_three generate id, name;

dump top\_3\_name\_id

(101,Amitabh)

(107,Salman)

(103,Akshay)

**(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)**

expense = load '/home/ec2-user/pigdata/employee\_expenses.txt' using PigStorage() as (id:INT,exp:INT);

grp\_exp= GROUP expense by exp;

max\_exp = ORDER grp\_exp by group desc;

emp\_max = limit max\_exp 1;

exp\_max\_emps = foreach emp\_max generate FLATTEN(expense);

X = join exp\_max\_emps by expense::id LEFT, employee by id;

X\_ordered\_by\_name = ORDER X by employee::name;

top\_spender = limit X\_ordered\_by\_name 1;

dump top\_spender;

(110,400,110,Priyanka,2000,5)

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

**file.**

emps\_with\_expense = JOIN expense by id,employee by id;

emp\_id\_name = foreach emps\_with\_expense generate employee::id as id ,employee::name as name;

emp\_id\_name\_dist = DISTINCT emp\_id\_name;

(101,Amitabh)

(102,Shahrukh)

(104,Anubhav)

(105,Pawan)

(110,Priyanka)

(114,Madhuri)

**(e) List of employees (employee id and employee name) having no entry in employee\_expenses**

**file.**

USING COGROUP

cogroup\_emp\_exp = COGROUP employee by id, expense by id;

employees\_with\_no\_exp = FILTER cogroup\_emp\_exp by IsEmpty(expense);

**id\_name = foreach employees\_with\_no\_exp generate employee.id, employee.name;**

({(103)},{(Akshay)})

({(106)},{(Aamir)})

({(107)},{(Salman)})

({(108)},{(Ranbir)})

({(109)},{(Katrina)})

({(111)},{(Tushar)})

({(112)},{(Ajay)})

({(113)},{(Jubeen)})

USING JOIN

emp\_with\_no\_expense = JOIN employee By id LEFT OUTER,emp\_id\_name\_dist By id;

emp\_with\_no\_expense\_filter = FILTER emp\_with\_no\_expense By emp\_id\_name\_dist::name is null;

**emp\_id\_name\_without\_expense = foreach emp\_with\_no\_expense\_filter generate employee::id,employee::name;**

(103,Akshay)

(106,Aamir)

(107,Salman)

(108,Ranbir)

(109,Katrina)

(111,Tushar)

(112,Ajay)

(113,Jubeen)