1. **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)**

* Load file into emp

Grunt> emp = LOAD 'employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating :int);

Grunt > dump emp;



* Ordering the relation on the basis of rating and emp name

Grunt> Emp\_rating\_odering= ORDER emp by emp\_rating,emp\_name;

* Taking top 5 record

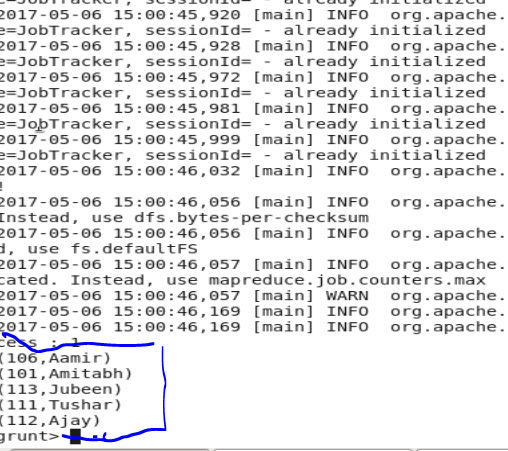
Grunt > Top\_Five\_Emp= limit EMP\_rating\_odering 5;

* Taking only emp\_id and emp\_name

Grunt > Top\_Five\_EmpIDEmpName= FOREACH Top\_Five\_Emp generate emp\_id,emp\_name;

Grunt > dump Top\_Five\_EmpIDEmpName;

Output :



**(b) Top 3 employees (employee id and employee name) with highest salary, whose empl oyee id**

**is an odd number.**

**(In case two employees have same salary, employee with name coming first in dictionary**

**should get preference)**

* Load file into emp

Grunt> emp = LOAD 'employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating :int);

* Emp with odd emp id

Grunt > Emp\_With\_Odd\_ID= FILTER emp by emp\_id % 2 !=0;

* Emp with salary ordering

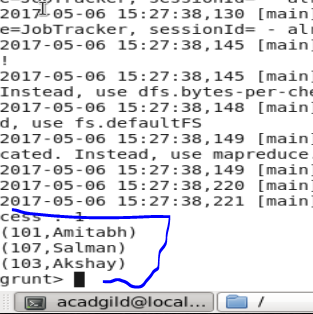
Grunt> Emp\_With\_Salary\_Ordering= ORDER Emp\_With\_Odd\_ID by emp\_salary desc;

* Taking top 3 employee

Grunt > TopThreeEmp = Limit Emp\_With\_Salary\_Ordering 3;

* Taking empid and emp name

Grunt > Result = FOREACH TopThreeEmp generate emp\_id, emp\_name;

****

**(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)**

* Load employee\_details.txt file into emp\_details

Grunt> emp = LOAD 'employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating :int);

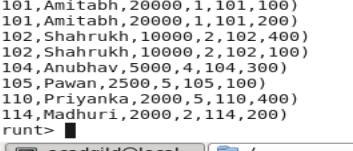
* Load employee\_expenses.txt file into emp\_exp

Grunt> emp\_exp = LOAD 'employee\_expenses.txt' AS (emp\_id:int, exps :int);

* Join both relation on empid basis

Grunt > joinData = join emp by emp\_id , emp\_exp by emp\_id;

Grunt> dump joinData;



* Ordering data on the basis of expenses and emp name

Grunt > orderdData = ORDER joinData by exps desc, emp\_name;

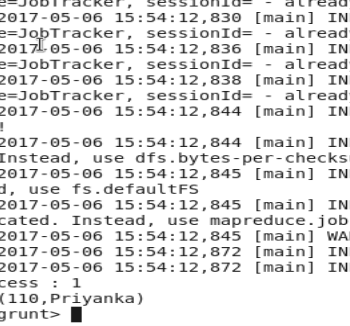
* Get the top 1 record

Grunt > EmpWithMaxExpanse= limit orderedData 1;

* Get emp id and emp name only

Grunt > result= FOREACH EmpWithMaxExpanse generate emp::emp\_id,emp::emp\_name;

Grunt > dump result;



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

**file.**

* Load employee\_details.txt file into emp\_details

Grunt> emp = LOAD 'employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating :int);

* Load employee\_expenses.txt file into emp\_exp

Grunt> emp\_exp = LOAD 'employee\_expenses.txt' AS (emp\_id:int, exps :int);

* Join both relation on empid basis

Grunt > joinData = join emp by emp\_id , emp\_exp by emp\_id;

Join data contain only those recors that are present in both relation.

* Taking empid and emp name

Grunt > result= FOREACH joinData generate emp::emp\_name,emp::emp\_id;

Grunt> dump result;



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

**file.**

* Load employee\_details.txt file into emp\_details

Grunt> emp = LOAD 'employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating :int);

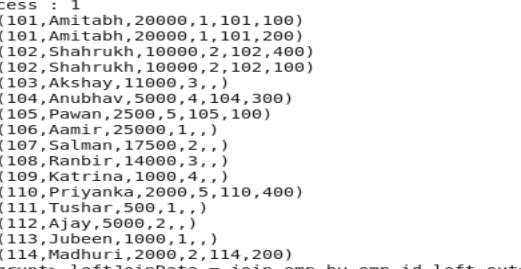
* Load employee\_expenses.txt file into emp\_exp

Grunt> emp\_exp = LOAD 'employee\_expenses.txt' AS (emp\_id:int, exps :int);

* Left outer Join both relation on empid basis

Grunt >leftjoinData = join emp by emp\_id left outer , emp\_exp by emp\_id;

Grunt > dump leftjoinData ;



* Filter record that have null value for column expns

Grunt> FilterNullExpnsData= FILTER leftjoinData by expns is null;

* Take emp name and empid in final result.

Grunt > result= FOREACH FilterNullExpnsData generate emp::emp\_name, emp::emp\_id;

Grunt: dump result;

