**(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)**

emp = load 'employee\_details' using PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating:int);

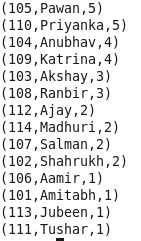
dump emp;

emp\_with\_rating = foreach emp generate emp\_id, emp\_name, emp\_rating;

dump emp\_with\_rating;

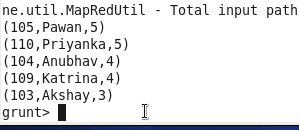


sorted\_emp\_with\_rating = order emp\_with\_rating by emp\_rating desc, emp\_name asc;



top\_five\_emp\_with\_rating = limit sorted\_emp\_with\_rating 5;

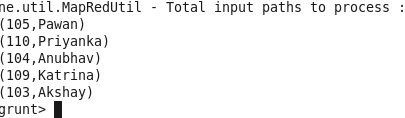
dump top\_five\_emp\_with\_rating;



describe top\_five\_emp\_with\_rating;

top\_five\_emp = foreach top\_five\_emp\_with\_rating generate emp\_id, emp\_name;

dump top\_five\_emp;



**b) Top 3 employees (employee id and employee name) with highest salary, whose employee id is an odd number.**

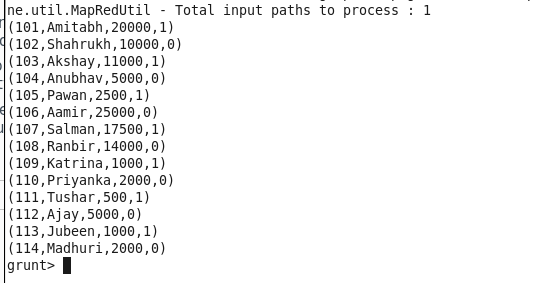
**(In case two employees have same salary, employee with name coming first in dictionary should get preference)**

emp = load 'employee\_details' using PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating:int);

dump emp;

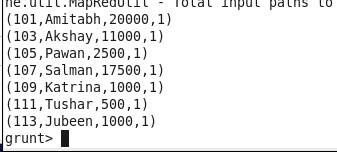
odd\_emp\_id = foreach emp generate emp\_id, emp\_name, emp\_salary, (emp\_id % 2);

dump odd\_emp\_id;



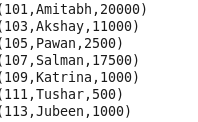
result = filter odd\_emp\_id by (emp\_id%2) > 0;

dump result;



result\_with\_salary = foreach result generate emp\_id , emp\_name, emp\_salary;

dump result\_with\_salary;



sorted\_ result\_with\_salary = order result\_with\_salary by emp\_salary desc, emp\_name asc;

dump sorted\_ result\_with\_salary;

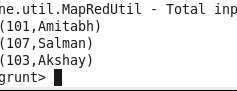
final\_result = foreach sorted\_ result\_with\_salary generate emp\_id, emp\_name;

dump final\_result;



top\_three\_emp = limit final\_result 3;

dump top\_three\_emp;



**(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)**

emp = load 'employee\_details' using PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating:int);

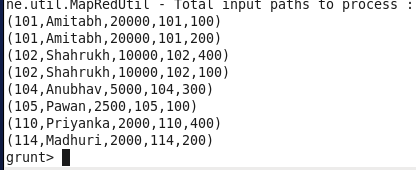
dump emp;

emp\_expenses = load 'employee\_expenses' using PigStorage(',') AS (emp\_id:int, emp\_expenses:int);

dump emp\_expenses;

emp\_details = join emp by emp\_id, emp\_expenses by emp\_id;

dump emp\_details;



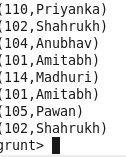
sorted\_emp\_details = order emp\_details by emp\_expenses::expenses desc, emp::emp\_name asc;

dump sorted\_emp\_details;



emp\_details\_final = foreach sorted\_emp\_details generate emp\_id, emp\_name;

dump emp\_details\_final;



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

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

emp\_expenses = LOAD 'employee\_expenses' AS (emp\_id:int, expenses:int);

right\_outer\_joined\_data = join emp by emp\_id left outer, emp\_expenses by emp\_id;

dump right\_outer\_joined\_data;

describe right\_outer\_joined\_data;

filtered\_roj\_data = filter right\_outer\_joined\_data by emp::emp\_id IS NOT NULL;

roj\_result = foreach filtered\_loj\_data generate emp::emp\_id, emp::emp\_name;

dump loj\_result;

****

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

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

emp\_expenses = LOAD 'employee\_expenses' AS (emp\_id:int, expenses:int);

left\_outer\_joined\_data = join emp by emp\_id left outer, emp\_expenses by emp\_id;

filtered\_loj\_data = filter left\_outer\_joined\_data by emp\_expenses::emp\_id IS NULL;

loj\_result = foreach filtered\_loj\_data generate emp::emp\_id, emp::emp\_name;

dump loj\_result;

