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)**

**Solutions:**

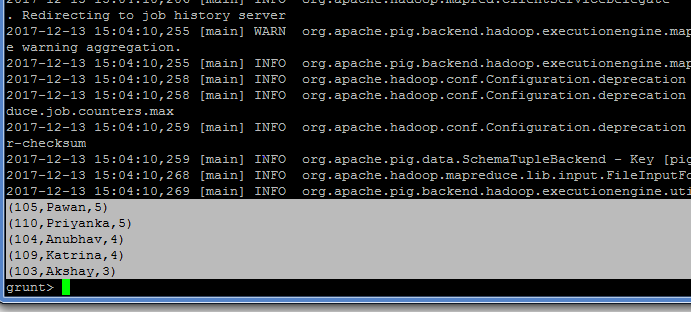
emp\_data= LOAD '/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_details.txt' USING PigStorage(',') AS(empId:int,empName:chararray,Salary:int,rating:int);

A =FOREACH emp\_data GENERATE (int)$0 as employee\_id,(chararray)$1 as employee\_name,(int)$3 as rating;

B= ORDER A BY rating DESC,employee\_name;

C= LIMIT B 5;

dump C;



1. **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)**

**Solutions:**

emp\_data= LOAD '/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_details.txt' USING PigStorage(',') AS(empId:int,empName:chararray,Salary:int,rating:int);

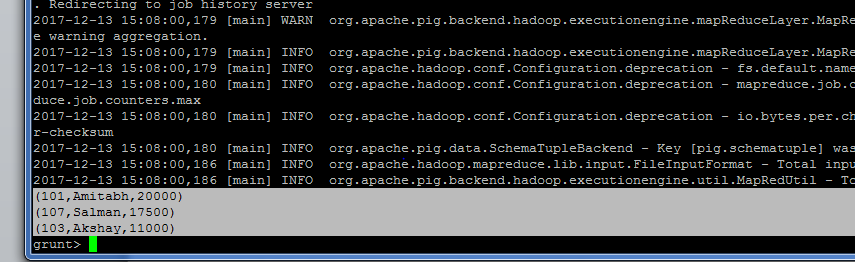
A =FOREACH emp\_data GENERATE (int)$0 as employee\_id,(chararray)$1 as employee\_name, (int)$2 as salary;

B= ORDER A BY salary DESC, employee\_name;

C= FILTER B BY (int)$0%2==1;

D= LIMIT C 3;

dump D;



1. **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)**

**solutions:**

emp\_data= LOAD '/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_details.txt' USING PigStorage(',') AS(empId:int,empName:chararray,Salary:int,rating:int);

emp\_expenses= LOAD '/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_expenses.txt' USING PigStorage('\t') AS(empId:int,empexpenses:int);

join\_data\_expenses= JOIN emp\_data BY empId, emp\_expenses BY empId;

joined\_data = FOREACH join\_data\_expenses GENERATE emp\_data::empId, emp\_data::empName, emp\_expenses::empexpenses;

joined\_data\_group = GROUP joined\_data by (empId,empName);

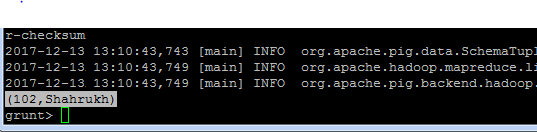
joined\_data\_SUM = FOREACH joined\_data\_group GENERATE group, SUM(joined\_data.emp\_expenses::empexpenses) as sum;

joined\_data\_desc = ORDER joined\_data\_SUM by sum DESC;

joined\_data\_limit = LIMIT joined\_data\_desc 1;

joined\_data\_flatten = FOREACH joined\_data\_limit GENERATE FLATTEN(group);

DUMP joined\_data\_flatten;



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

emp\_data= LOAD'/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_details.txt' USING PigStorage(',') AS(empId:int,empName:chararray,Salary:int,rating:int);

emp\_expenses= LOAD'/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_expenses.txt' USING PigStorage('\t') AS(empId:int,empexpenses:int);

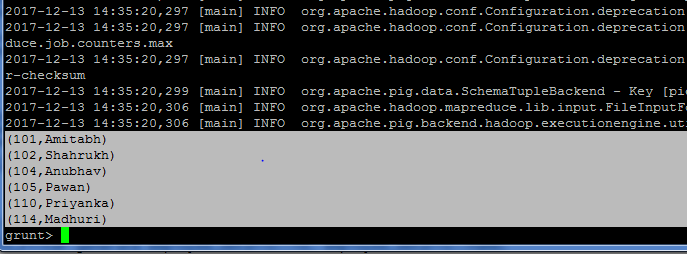
join\_data\_expenses= JOIN emp\_data BY empId, emp\_expenses BY empId;

joined\_data = FOREACH join\_data\_expenses GENERATE emp\_data::empId,

emp\_data::empName;

joined\_data\_distinct = DISTINCT joined\_data;

dump joined\_data\_distinct;



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

emp\_data= LOAD'/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_details.txt' USING PigStorage(',') AS(empId:int,empName:chararray,Salary:int,rating:int);

emp\_expenses= LOAD '/user/acadgild/hadoop/pig\_data/assignment\_input\_data\_5.1/employee\_expenses.txt' USING PigStorage('\t') AS(empId:int,empexpenses:int);

left\_outer\_join\_data\_expenses= JOIN emp\_data BY empId LEFT OUTER, emp\_expenses BY empId;

left\_outer\_join\_filter = FILTER left\_outer\_join\_data\_expenses by emp\_expenses::empexpenses is NULL;

joined\_table = FOREACH left\_outer\_join\_filter GENERATE emp\_data::empId, emp\_data::empName;

dump joined\_table;

