(a) Top 5 employees (employee id and employee name) with highest rating. (In case two

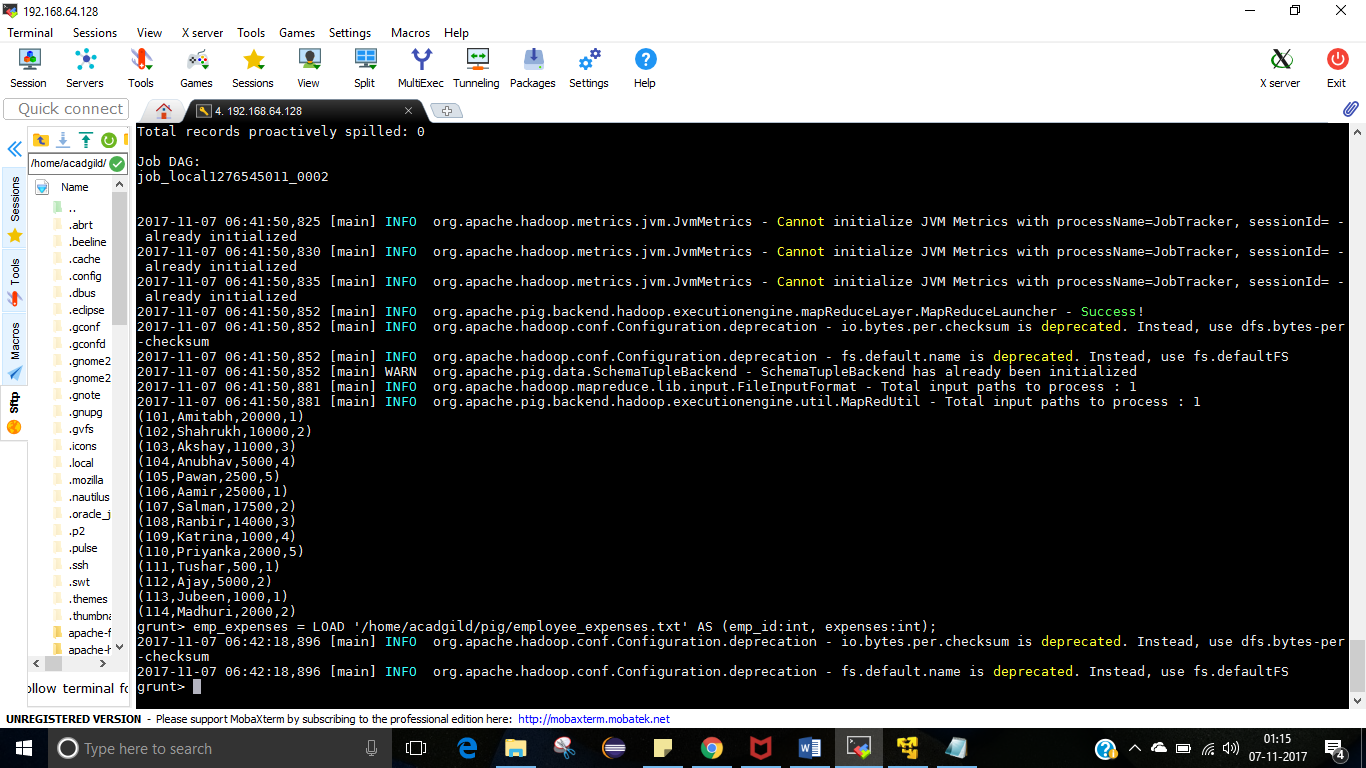
employees have same rating, employee with name coming first in dictionary should get

preference)

Solution:

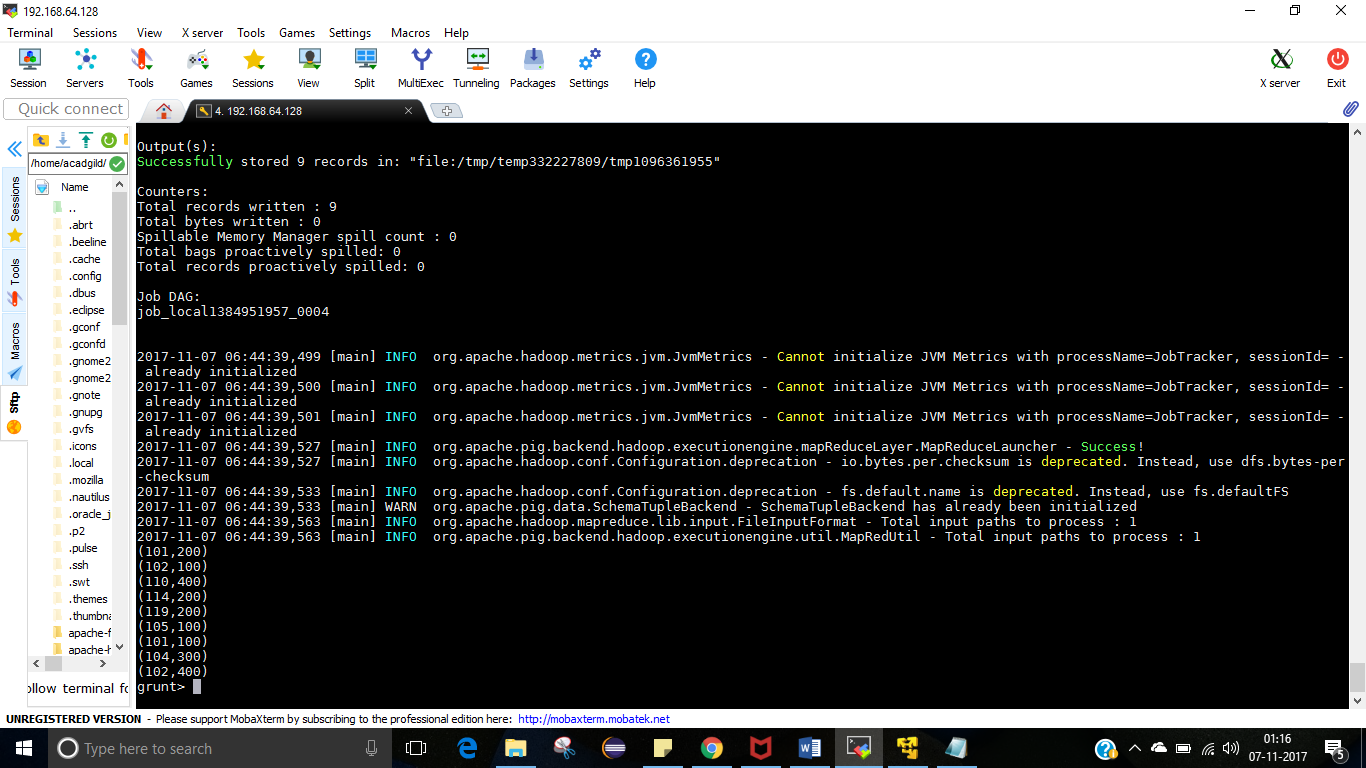
emp = LOAD '/home/acadgild/pig/employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_name:chararray, emp\_salary:int, emp\_rating:int);

DUMP emp;

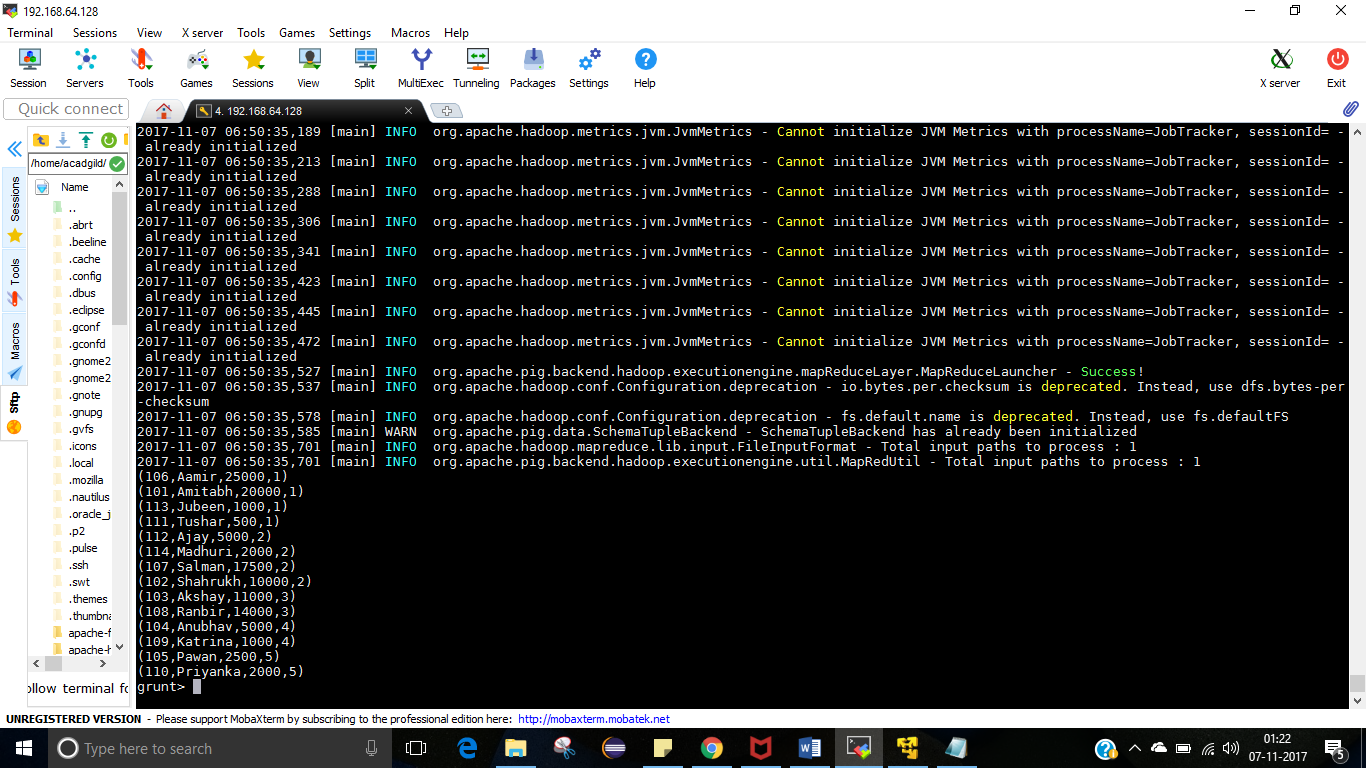


emp\_expenses = LOAD '/home/acadgild/pig/employee\_expenses.txt' AS (emp\_id:int, expenses:int);

DUMP emp\_expenses;



emp\_high\_rating = ORDER emp BY emp\_rating ASC , emp\_name ASC;



five\_emp = limit emp\_high\_rating 5;

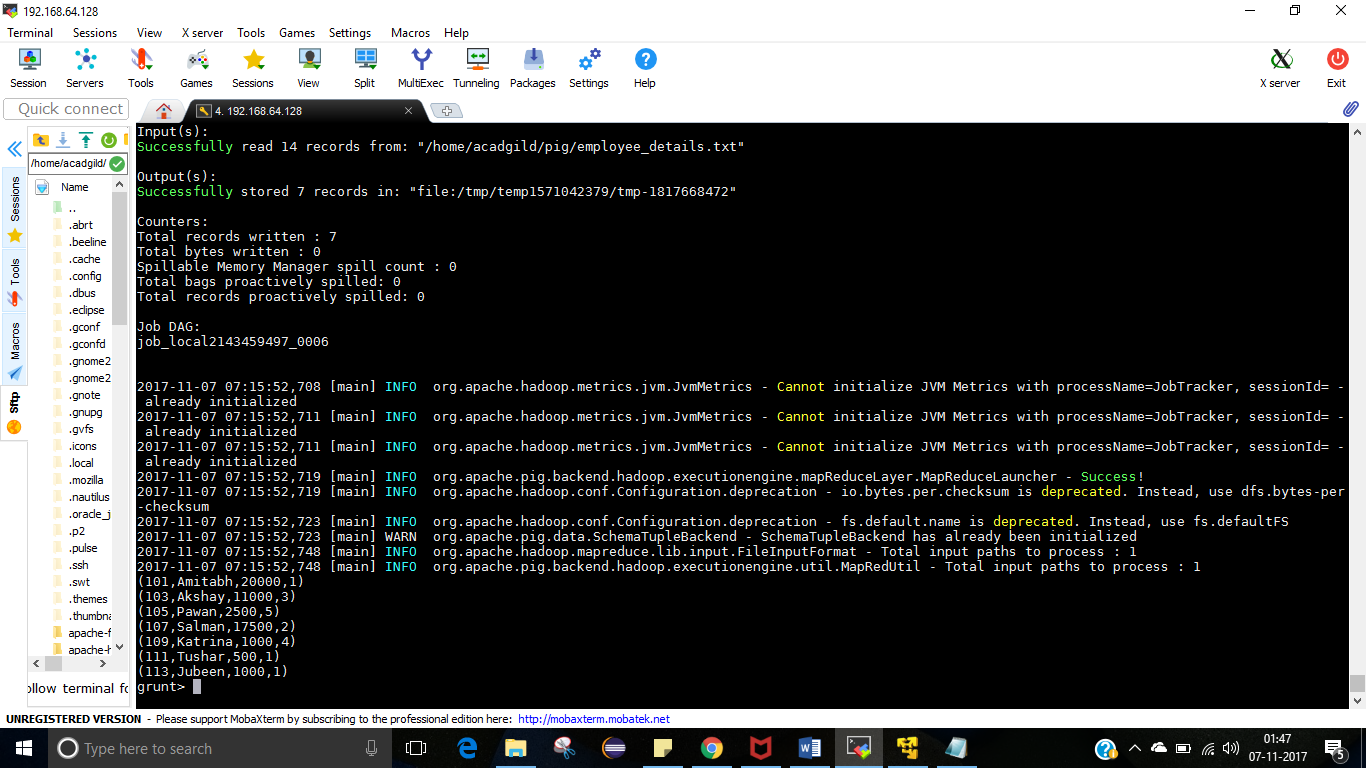
(b) Top 3 employees (employee id and employee name) with highest salary, whose employee id

is an odd number. (In case two employees have same salary, employee with name coming first

in dictionary should get preference)

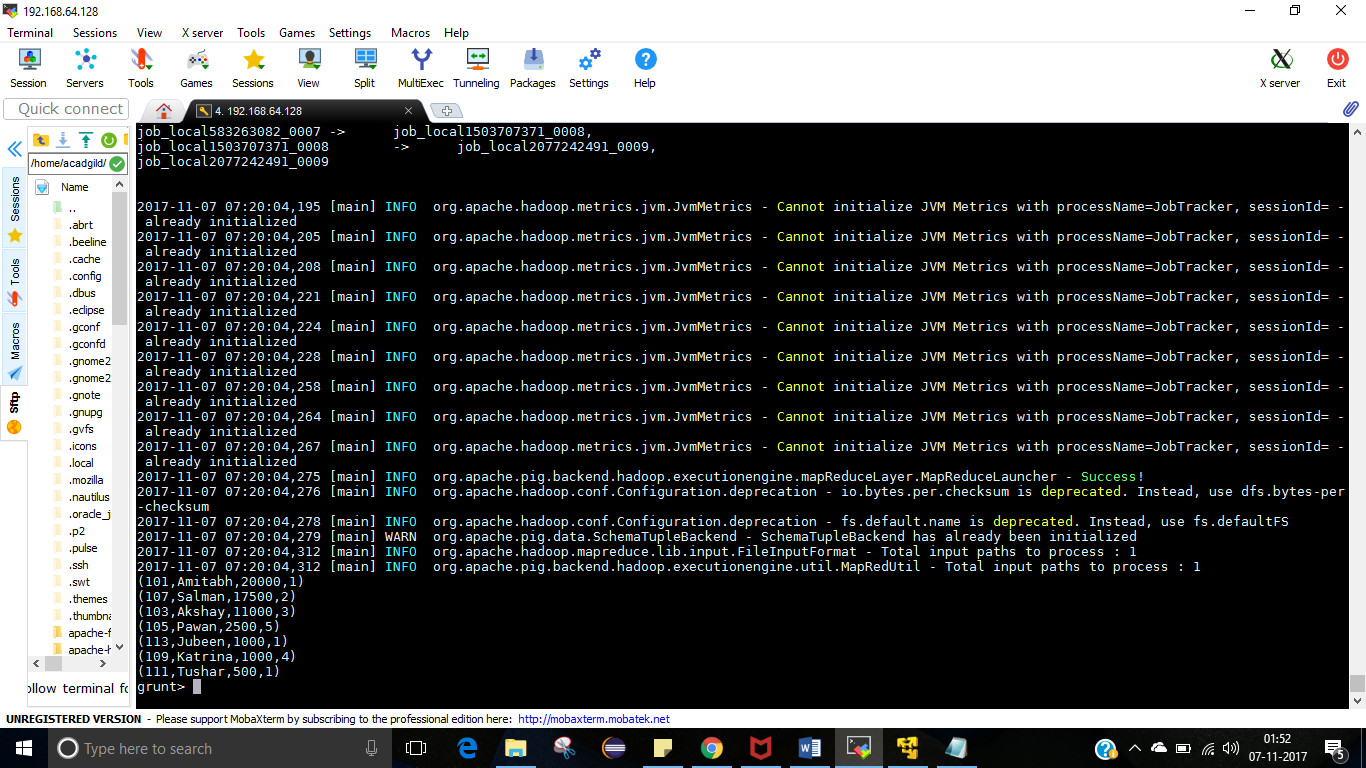
Solution :

filter\_odd\_emp\_id = FILTER emp BY emp\_id%2!=0;



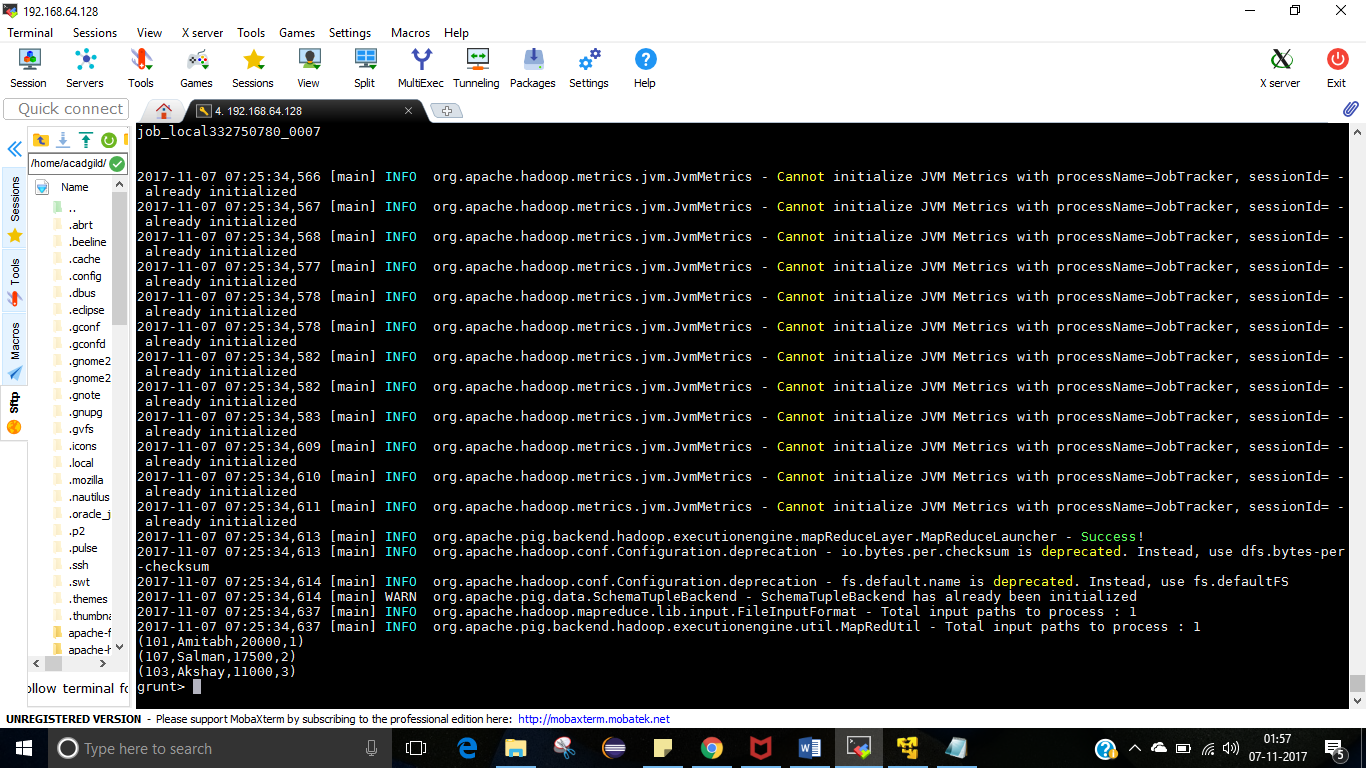
order\_odd\_id\_employee = ORDER filter\_odd\_emp\_id BY emp\_salary DESC ,emp\_name ASC;

DUMP order\_odd\_id\_employee;



three\_emp = limit order\_odd\_id\_employee 3;

DUMP three\_emp;



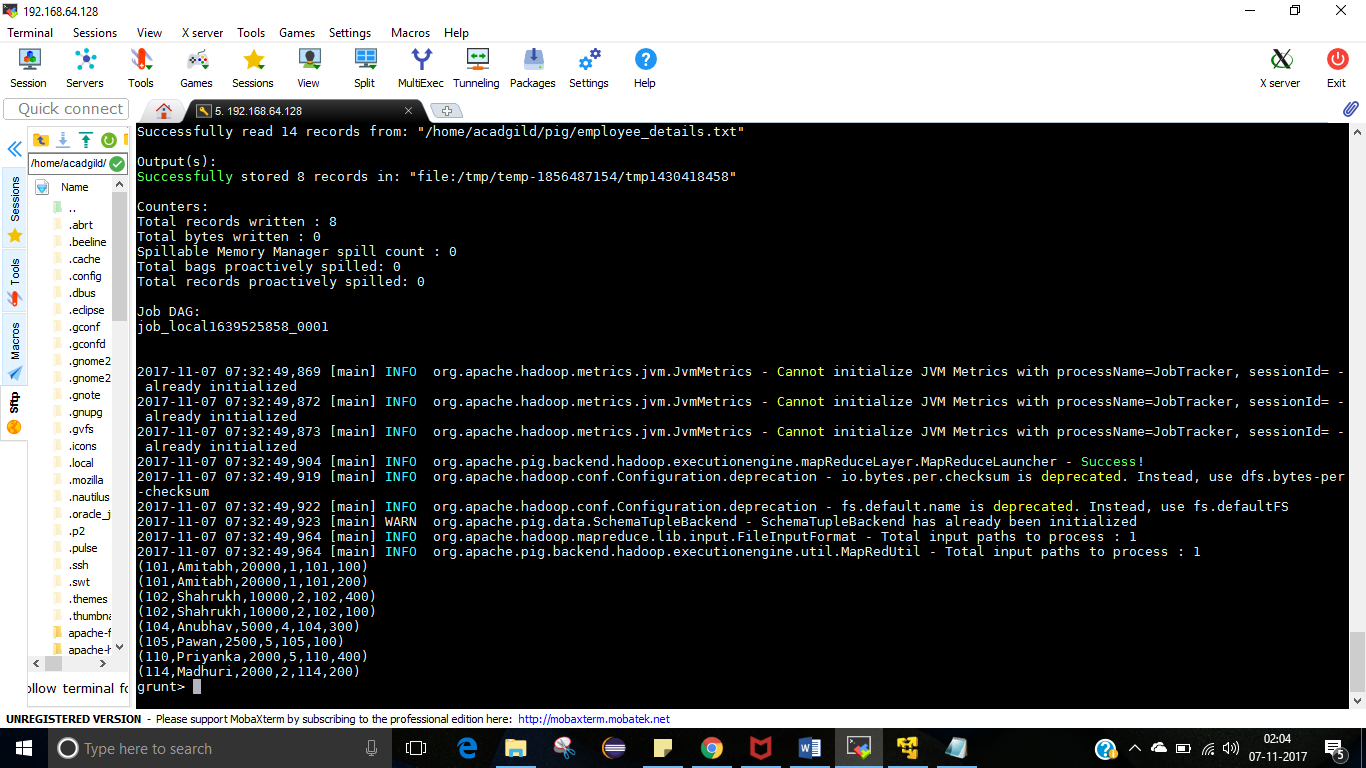
(c) Employee (employee id and employee name) with maximum expense (In case two

employees have same expense, employee with name coming first in dictionary should get

preference)

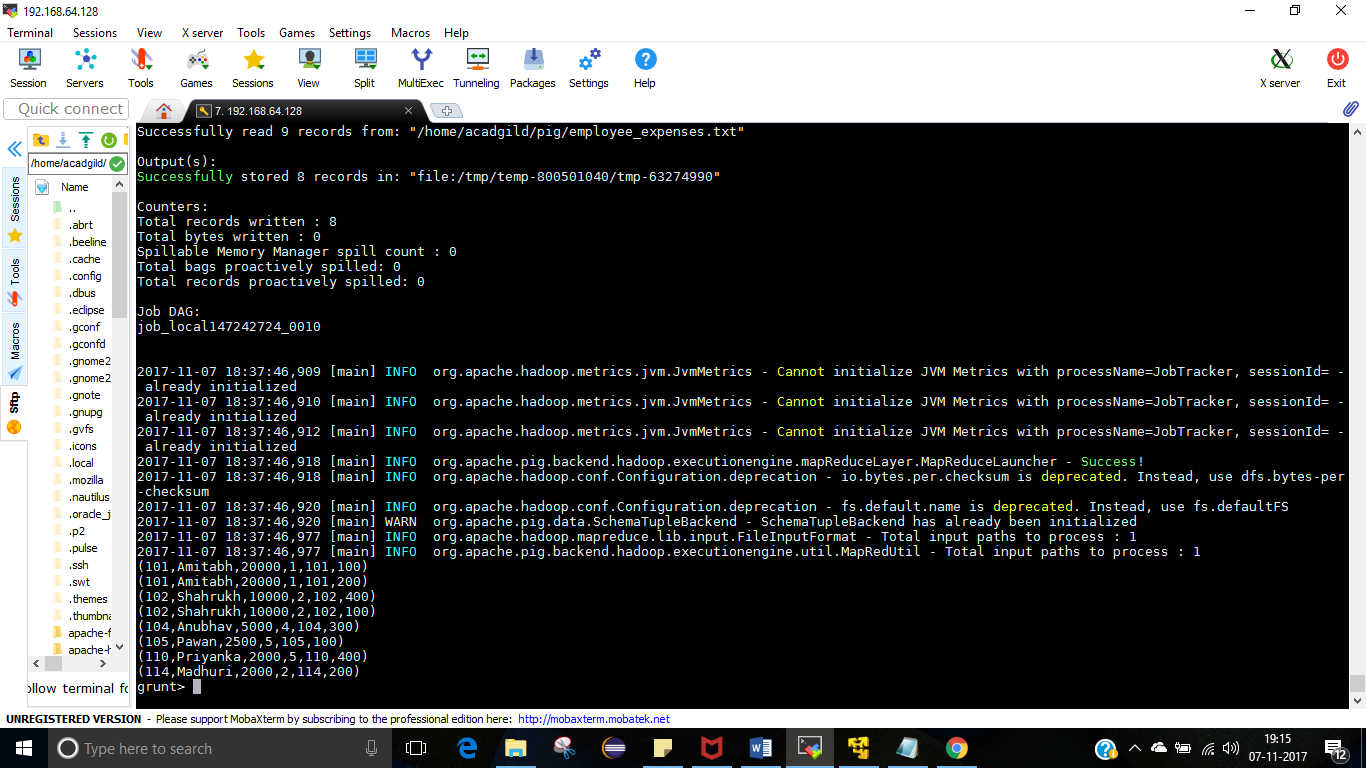
joined\_data = join emp by emp\_id , emp\_expenses by emp\_id;

dump joined\_data;



order\_by\_name\_expense = ORDER joined\_data BY expenses DESC;

DUMP order\_by\_name\_expense;

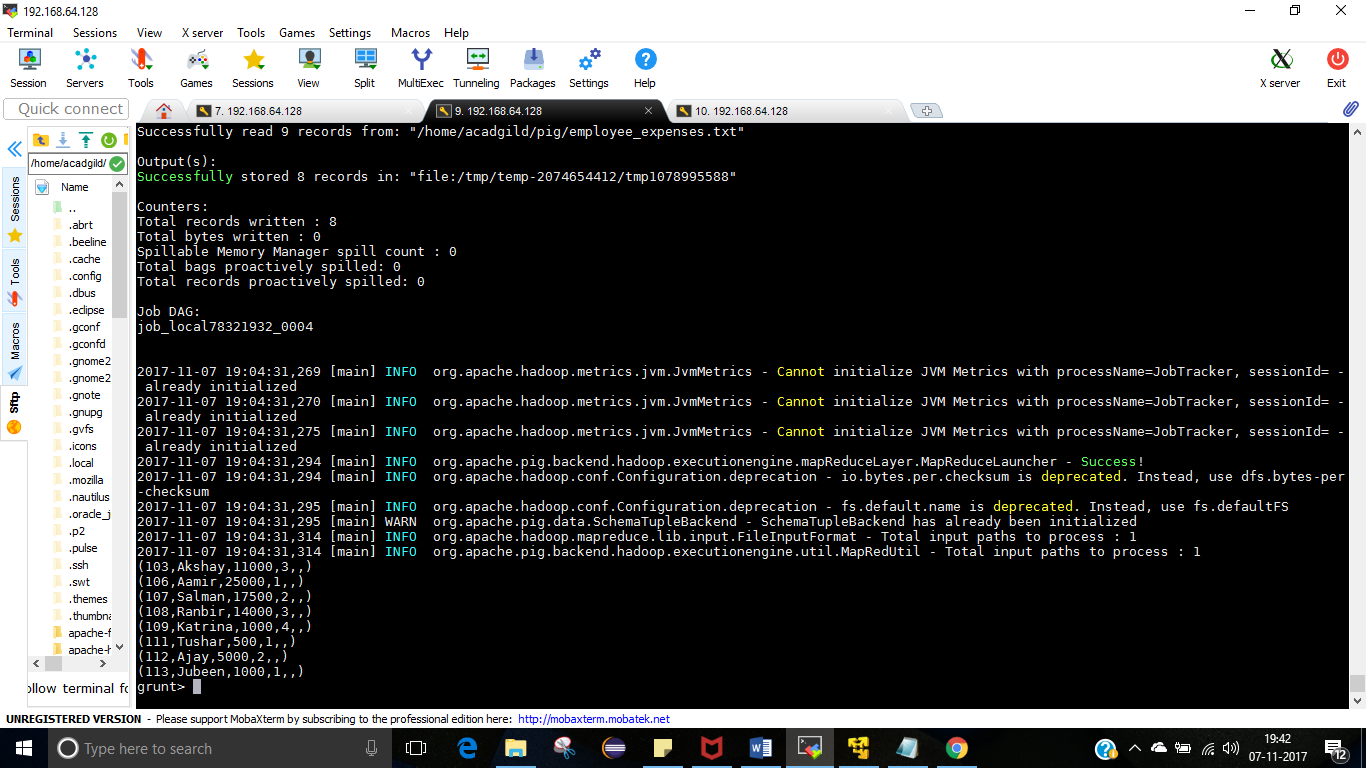


(e) List of employees (employee id and employee name) having no entry in employee\_expenses

file.

filter\_emp\_id = FILTER left\_outer\_joined\_data BY emp\_expenses::emp\_id is NULL;

DUMP filter\_emp\_id;



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

file.

filter\_emp\_id = FILTER left\_outer\_joined\_data BY emp\_expenses::emp\_id is not NULL;

DUMP filter\_emp\_id ;

