**Assignment 5.1:**

We have employee\_details and employee\_expenses files. Use local mode while running Pig and

write Pig Latin script to get below results:

https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee\_details.t

xt

https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee\_expense

s.txt

(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:**

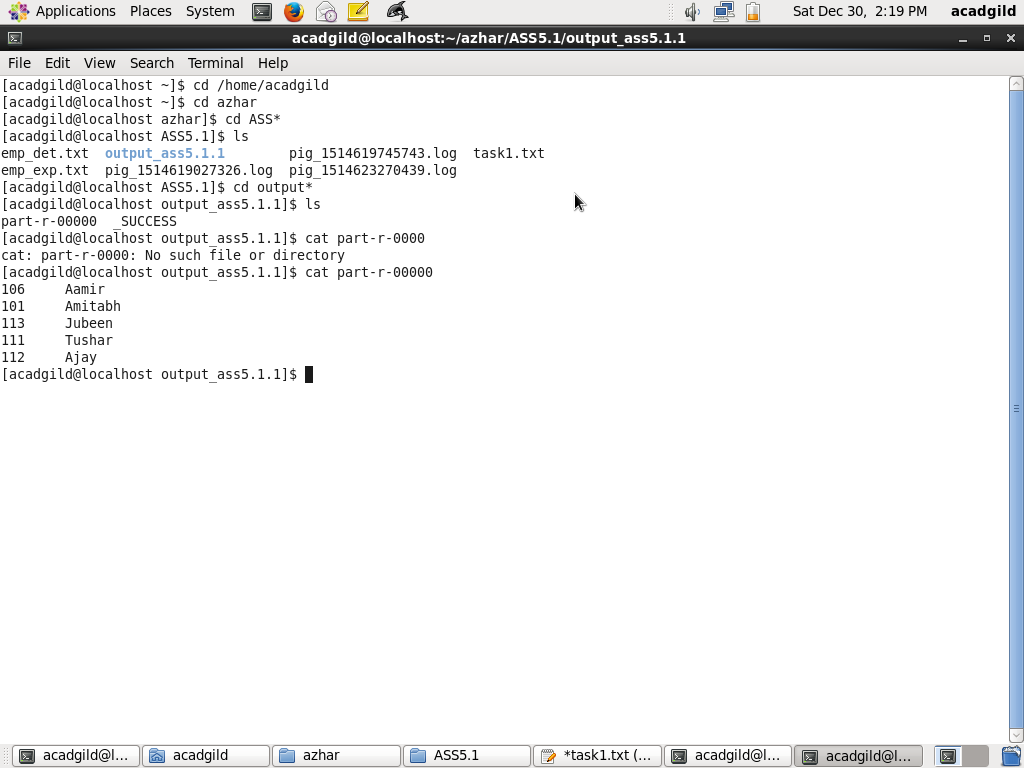
emp1 = load 'emp\_det.txt' USING PigStorage(',') AS (empid:int , empname:chararray, empsal: int, emprat: int);

order\_emp1= order emp1 by emprat,empname;

limit\_emp1 = limit order\_emp1 5;

out\_emp1 = foreach limit\_emp1 generate empid,empname;

store out\_emp1 into 'output\_ass5.1.1';



(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:**

emp1 = load 'emp\_det.txt' USING PigStorage(',') AS (empid:int , empname:chararray, empsal: int, emprat: int);

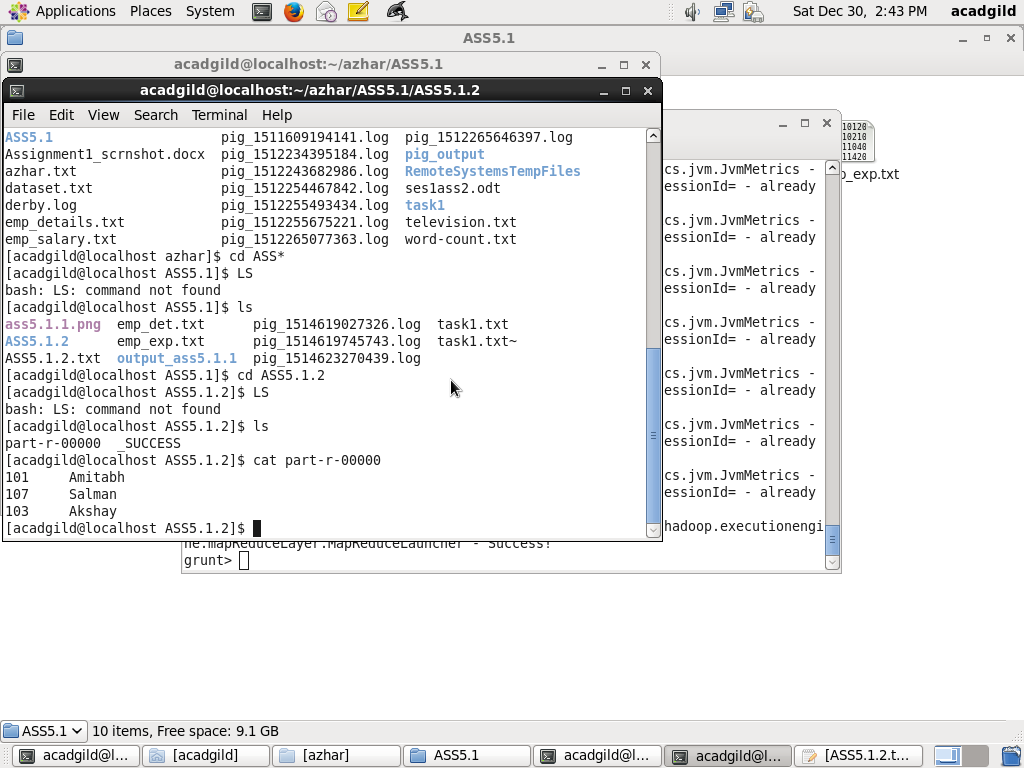
order\_emp1\_sal= order emp1 by empsal desc,empname;

fil\_emp1 = filter order\_emp1\_sal by empid%2 > 0;

lim\_emp1\_sal = limit fil\_emp1 3;

ans\_emp1\_sal = foreach lim\_emp1\_sal generate empid,empname;

store ans\_emp1\_sal into 'ASS5.1.2';



(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)

**Solution:**

emp1 = load 'emp\_det.txt' USING PigStorage(',') AS (empid:int , empname:chararray, empsal: int, emprat: int);

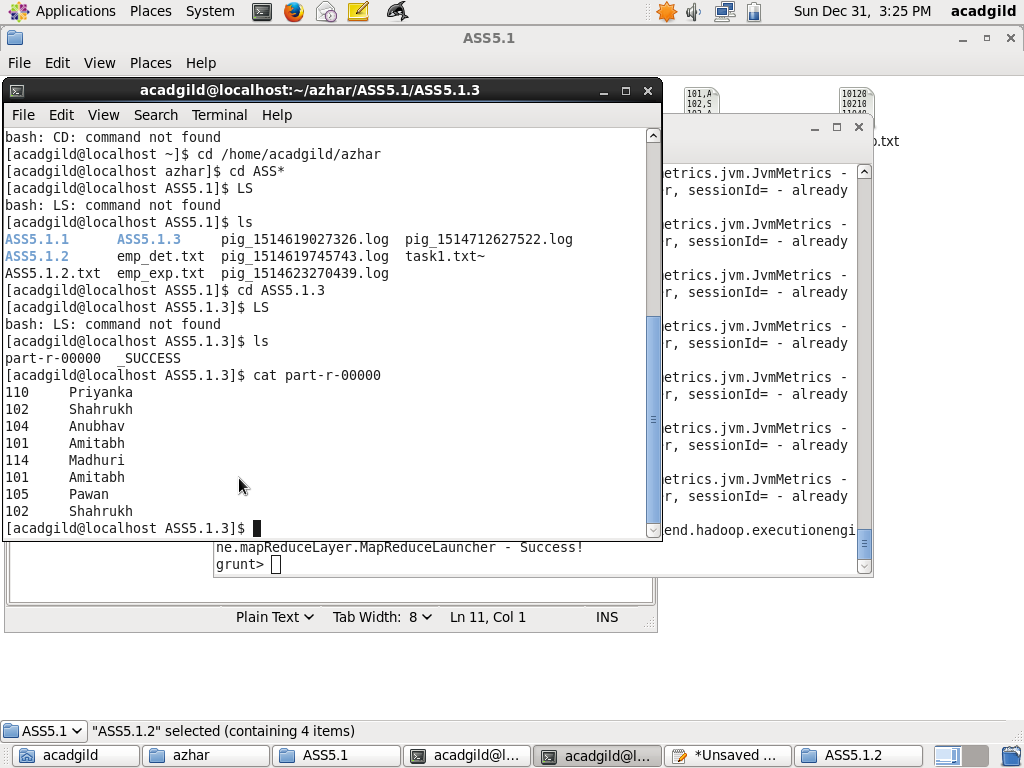
exp = load 'emp\_exp.txt' USING PigStorage() AS (empid:int , empexp: int);

join\_emp\_exp = join emp1 by empid,exp by empid;

ord\_emp\_exp = order join\_emp\_exp by exp::empexp desc,emp1::empname;

res\_emp\_exp = foreach ord\_emp\_exp generate emp1::empid,emp1::empname;

store res\_emp\_exp into 'ASS5.1.3';



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

file.

**Solution:**

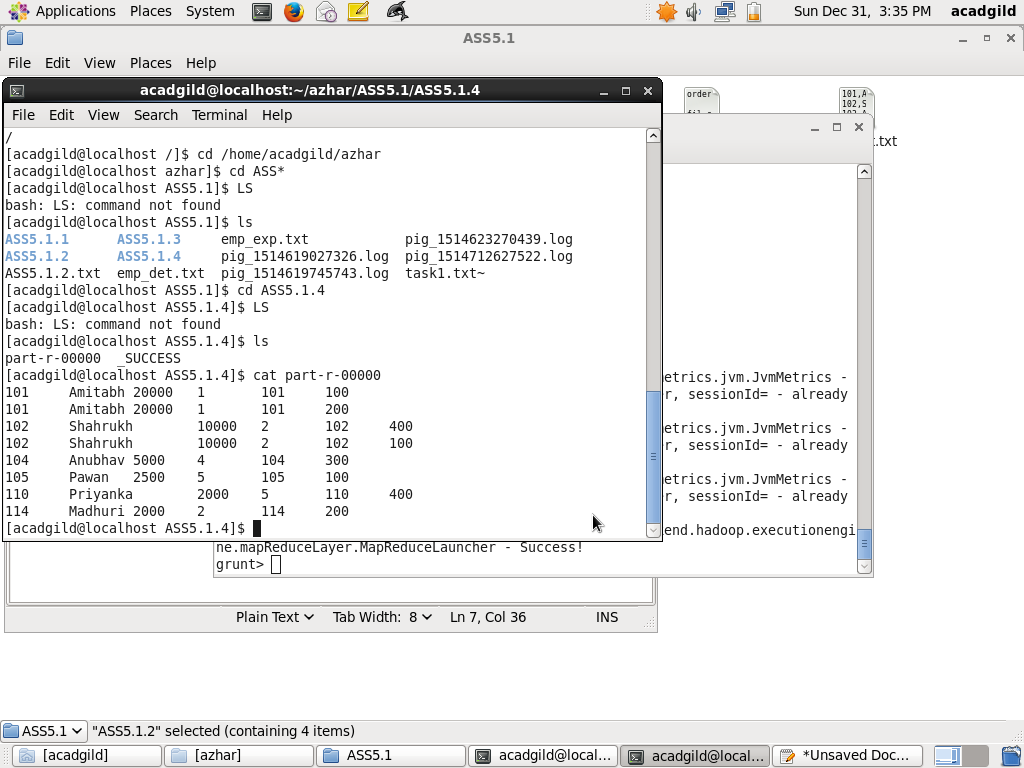
emp1 = load 'emp\_det.txt' USING PigStorage(',') AS (empid:int , empname:chararray, empsal: int, emprat: int);

exp = load 'emp\_exp.txt' USING PigStorage() AS (empid:int , empexp: int);

join\_emp\_exp = join emp1 by empid,exp by empid;

res\_emp\_exp = foreach join\_emp\_exp generate emp1::empid,emp1::empname;

store res\_emp\_exp into 'ASS5.1.4';



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

file.

**Solution:**

emp1 = load 'emp\_det.txt' USING PigStorage(',') AS (empid:int , empname:chararray, empsal: int, emprat: int);

exp = load 'emp\_exp.txt' USING PigStorage() AS (empid:int , empexp: chararray);

leftjoin\_emp\_exp = join emp1 by empid left,exp by empid;

fil\_emp\_exp = filter leftjoin\_emp\_exp by exp::empexp is null;

res\_emp\_exp = foreach join\_emp\_exp generate emp1::empid,emp1::empname;

store res\_emp\_exp into 'ASS5.1.4';

