



Session 5: Exploring Pig

# Assignment 1 Question

**Downloaded data from given link:-**

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

Dataset: -

[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/employee\_details.txt

101,Amitabh,20000,1

102,Shahrukh,10000,2

103,Akshay,11000,3

104,Anubhav,5000,4

105,Pawan,2500,5

106,Aamir,25000,1

107,Salman,17500,2

108,Ranbir,14000,3

109,Katrina,1000,4

110,Priyanka,2000,5

111,Tushar,500,1

112,Ajay,5000,2

113,Jubeen,1000,1

114,Madhuri,2000,2

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

[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/employee\_expenses.txt

101 200

102 100

110 400

114 200

119 200

105 100

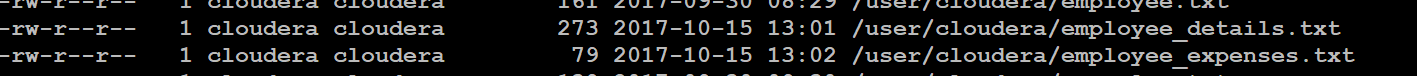
101 100

104 300

102 400

[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Downloads/employee\_details.txt /user/cloudera/

[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Downloads/employee\_expenses.txt /user/cloudera/



*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)*

grunt> emp\_det = LOAD '/user/cloudera/employee\_details.txt' USING PigStorage(',') AS (id:int, name:chararray, salary:int, rating:int);

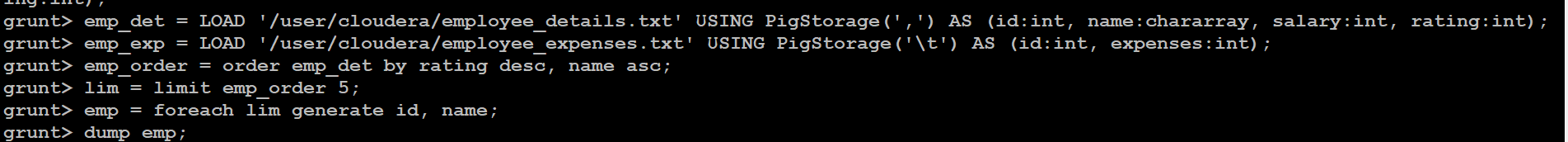
grunt> emp\_exp = LOAD '/user/cloudera/employee\_expenses.txt' USING PigStorage('\t') AS (id:int, expenses:int);

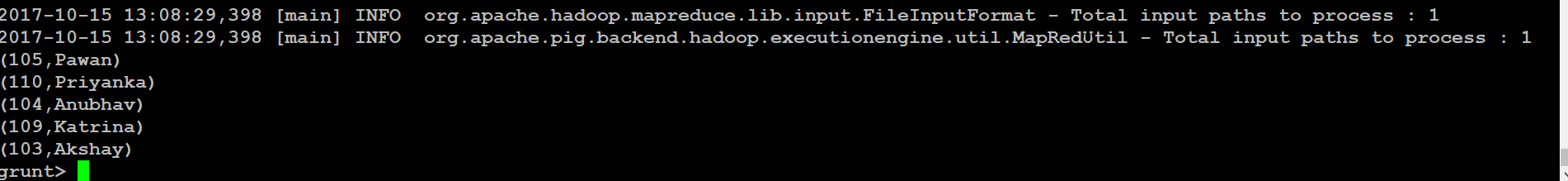
grunt> emp\_order = order emp\_det by rating desc, name asc;

grunt> lim = limit emp\_order 5;

grunt> emp = foreach lim generate id, name;

grunt> dump emp;





Output: -

(105,Pawan)

(110,Priyanka)

(104,Anubhav)

(109,Katrina)

(103,Akshay)

2): - *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)*

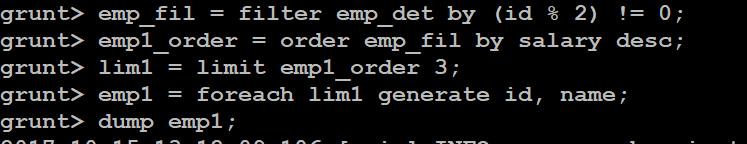
grunt> emp\_fil = filter emp\_det by (id % 2) != 0;

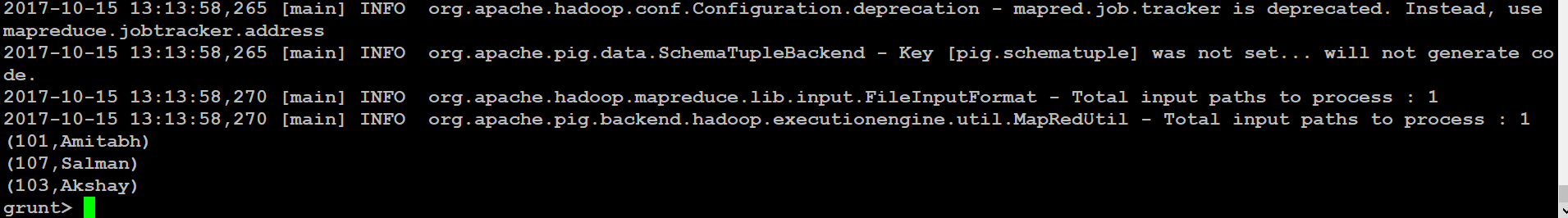
grunt> emp1\_order = order emp\_fil by salary desc;

grunt> lim1 = limit emp1\_order 3;

grunt> emp1 = foreach lim1 generate id, name;

grunt> dump emp1;





Output: -

(101,Amitabh)

(107,Salman)

(103,Akshay)

3): - *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)*

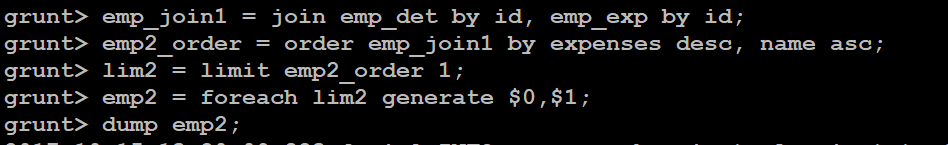
grunt> emp\_join1 = join emp\_det by id, emp\_exp by id;

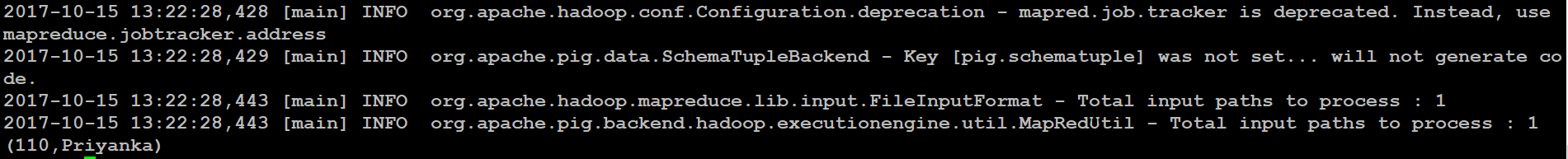
grunt> emp2\_order = order emp\_join1 by expenses desc, name asc;

grunt> lim2 = limit emp2\_order 1;

grunt> emp2 = foreach lim2 generate $0,$1;

grunt> dump emp2;





Output: -

(110,Priyanka)

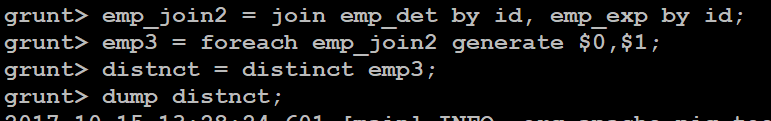
4): - *List of employees (employee id and employee name) having entries in employee\_expenses file.*

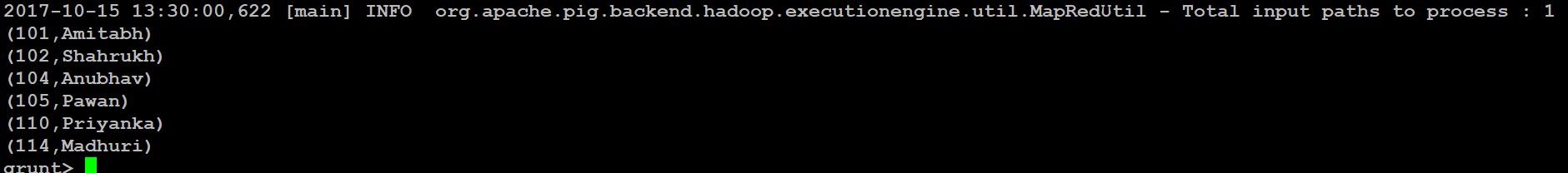
grunt> emp\_join2 = join emp\_det by id, emp\_exp by id;

grunt> emp3 = foreach emp\_join2 generate $0,$1;

grunt> distnct = distinct emp3;

grunt> dump distnct;





Output: -

(101,Amitabh)

(102,Shahrukh)

(104,Anubhav)

(105,Pawan)

(110,Priyanka)

(114,Madhuri)

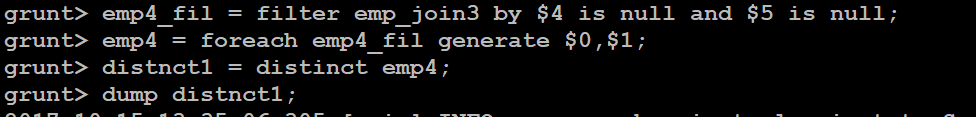
*5):-List of employees (employee id and employee name) having no entry in employee\_expenses file.*

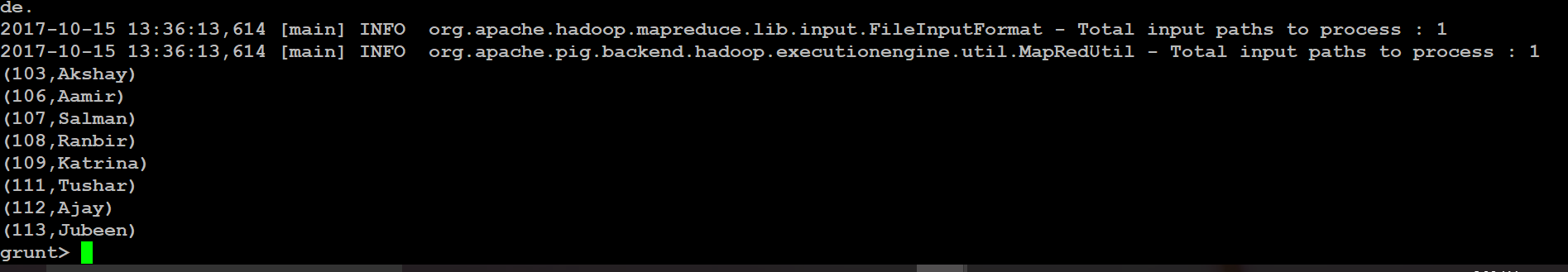
grunt> emp4\_fil = filter emp\_join3 by $4 is null and $5 is null;

grunt> emp4 = foreach emp4\_fil generate $0,$1;

grunt> distnct1 = distinct emp4;

grunt> dump distnct1;





Output: -

(103,Akshay)

(106,Aamir)

(107,Salman)

(108,Ranbir)

(109,Katrina)

(111,Tushar)

(112,Ajay)

(113,Jubeen)