(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)

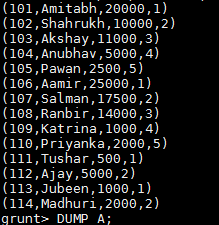
A = LOAD '/home/acadgild/leena/ass/ass-72/employee\_details.txt' USING PigStorage(',') AS (id:int,name:chararray,salary:int,rating:int);

s = ORDER A BY rating DESC, name ASC;

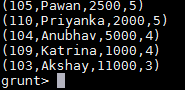
top5 = limit s 5;

resultTop5 = FOREACH top5 GENERATE id, name;

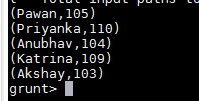
Some sample outputs :



top5 = limit s 5;



resultTop5 = FOREACH top5 GENERATE id, name;

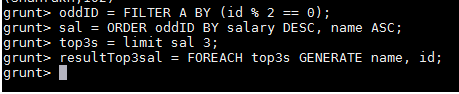


(b) Top 3 employees (employee id and employee name) with highest salary, whose employee id

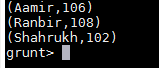
is an odd number.(In case two employees have same salary, employee with name coming first in dictionaryshould get preference)

Using the same relation ‘A’ from above :

A = LOAD '/home/acadgild/leena/ass/ass-72/employee\_details.txt' USING PigStorage(',') AS (id:int,name:chararray,salary:int,rating:int);



DUMP resultTop3sal;



(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)

A = LOAD '/home/acadgild/leena/ass/ass-72/employee\_details.txt' USING PigStorage(',') AS (id:int,name:chararray,salary:int,rating:int);

B = LOAD '/home/acadgild/leena/ass/ass-72/employee\_expenses.txt' AS (id:int, expenses:int);

joinAB = join A by id LEFT OUTER, B by id;

AB = FILTER joinAB BY (B::expenses is not null);

ABgrp\_id = GROUP AB by A::id;

(101,{(101,Amitabh,20000,1,101,200),(101,Amitabh,20000,1,101,100)})

(102,{(102,Shahrukh,10000,2,102,100),(102,Shahrukh,10000,2,102,400)})

(104,{(104,Anubhav,5000,4,104,300)})

(105,{(105,Pawan,2500,5,105,100)})

(110,{(110,Priyanka,2000,5,110,400)})

(114,{(114,Madhuri,2000,2,114,200)})

grunt> DESCRIBE ABgrp\_id;

ABgrp\_id: {group: int,AB: {(A::id: int,A::name: chararray,A::salary: int,A::rating: int,B::id: int,B::expenses: int)}}

ABsum = FOREACH ABgrp\_id GENERATE group, AB.name AS name, SUM(AB.expenses) AS totalexpenses;

(101,{(Amitabh),(Amitabh)},300)

(102,{(Shahrukh),(Shahrukh)},500)

(104,{(Anubhav)},300)

(105,{(Pawan)},100)

(110,{(Priyanka)},400)

(114,{(Madhuri)},200)

grunt> DESCRIBE ABsum;

ABsum: {group: int,name: {(A::name: chararray)},totalexpenses: long}

ABorder = ORDER ABsum BY totalexpenses DESC, name DESC;

(102,{(Shahrukh),(Shahrukh)},500)

(110,{(Priyanka)},400)

(101,{(Amitabh),(Amitabh)},300)

(104,{(Anubhav)},300)

(114,{(Madhuri)},200)

(105,{(Pawan)},100)

max\_expenses = limit ABorder 1;

(102,{(Shahrukh),(Shahrukh)},500)

(d)

A = LOAD '/home/acadgild/leena/ass/ass-72/employee\_details.txt' USING PigStorage(',') AS (id:int,name:chararray,salary:int,rating:int);

B = LOAD '/home/acadgild/leena/ass/ass-72/employee\_expenses.txt' AS (id:int, expenses:int);

joinAB\_right = join A by id RIGHT OUTER, B by id;

withexpenses = FOREACH joinAB\_right GENERATE A::name, A::id;

DUMP withexpenses;

(Amitabh,101)

(Amitabh,101)

(Shahrukh,102)

(Shahrukh,102)

(Anubhav,104)

(Pawan,105)

(Priyanka,110)

(Madhuri,114)

(,)

grunt> DESCRIBE withexpenses;

withexpenses: {A::name: chararray,A::id: int}

grunt>

result\_withexpenses = FILTER withexpenses BY A::name is not null;

(Amitabh,101)

(Amitabh,101)

(Shahrukh,102)

(Shahrukh,102)

(Anubhav,104)

(Pawan,105)

(Priyanka,110)

(Madhuri,114)

grunt> DESCRIBE result\_withexpenses;

result\_withexpenses: {A::name: chararray,A::id: int}

grunt>

e)

A = LOAD '/home/acadgild/leena/ass/ass-72/employee\_details.txt' USING PigStorage(',') AS (id:int,name:chararray,salary:int,rating:int);

B = LOAD '/home/acadgild/leena/ass/ass-72/employee\_expenses.txt' AS (id:int, expenses:int);

joinAB = join A by id LEFT, B by id;

noexpenses = FILTER joinAB BY (B::expenses is null);

(103,Akshay,11000,3,,)

(106,Aamir,25000,1,,)

(107,Salman,17500,2,,)

(108,Ranbir,14000,3,,)

(109,Katrina,1000,4,,)

(111,Tushar,500,1,,)

(112,Ajay,5000,2,,)

(113,Jubeen,1000,1,,)

noexp = FOREACH(Akshay,103)

(Aamir,106)

(Salman,107)

(Ranbir,108)

(Katrina,109)

(Tushar,111)

(Ajay,112)

(Jubeen,113)