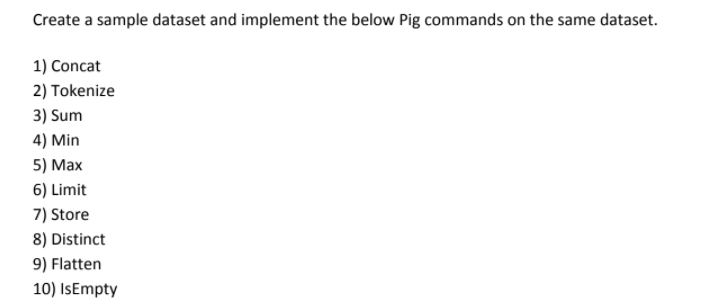


Problem statement



Creating database for the tasks

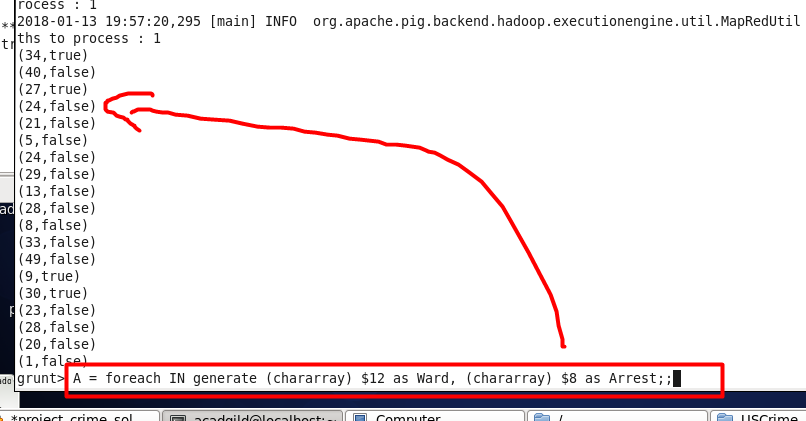
Lets use database that contain a lot of columns in order to be universal for the all cases of the assignment

Task01

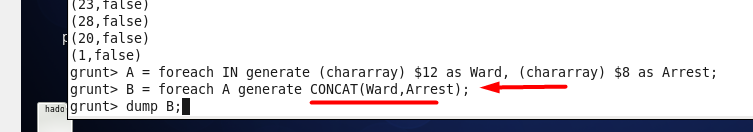


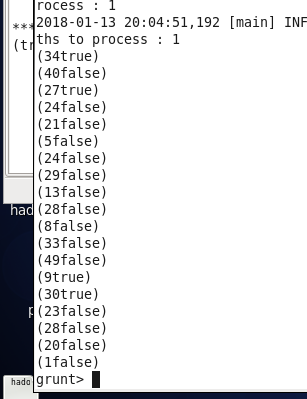
First of all let's load suitable for this function portion of data

Before Concat



After Concat



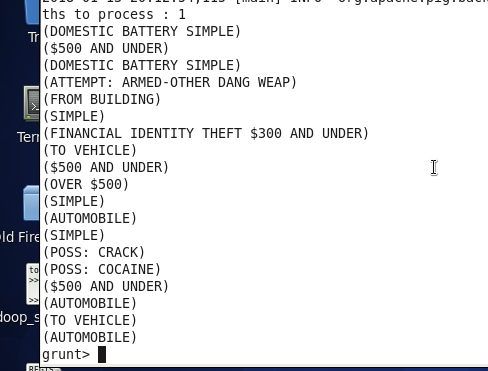


Task02

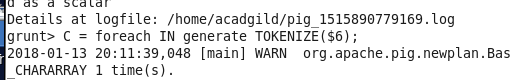
Load suitable portion of data

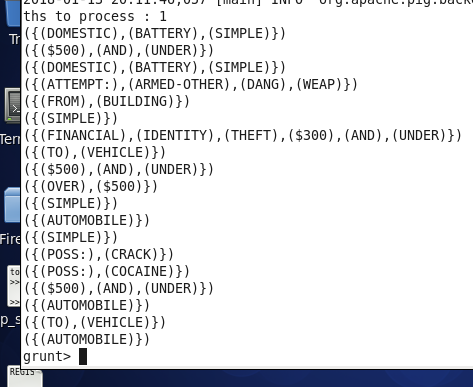


Before function



After function





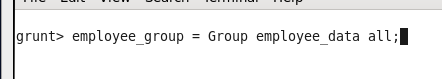
Task03

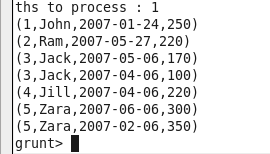


Load suitable portion of data

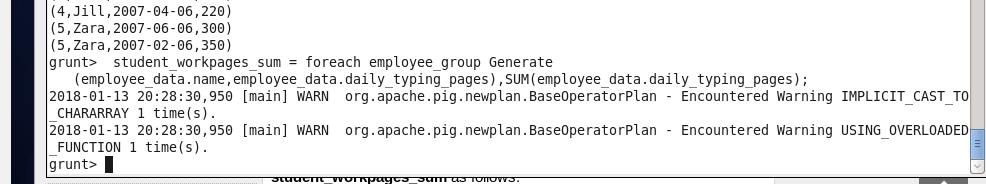


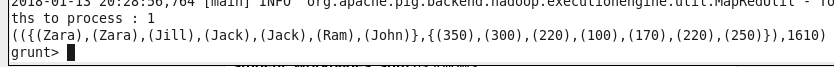
Let’s group





Let sum :





Task04



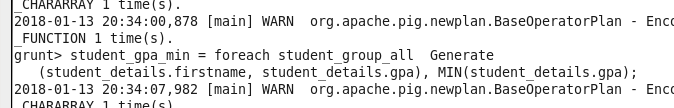
Load suitable portion of data

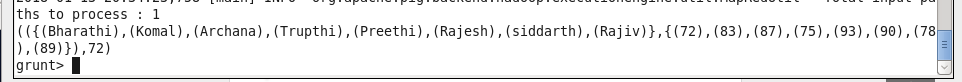


Group



Find min





Task05



The same syntax and data as for min except

grunt> student\_gpa\_max = foreach student\_group\_all Generate  
 (student\_details.firstname, student\_details.gpa), MAX(student\_details.gpa);

So lets dump

grunt> Dump student\_gpa\_max;

The result

(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv) } ,   
 { (72) , (83) , (87) , (75) , (93) , (90) , (78) , (89) }) ,93)

Task06



The same portion of the data as above

grunt> limit\_data = LIMIT student\_details 4;

The result

(1,Rajiv,Reddy,21,9848022337,Hyderabad)   
(2,siddarth,Battacharya,22,9848022338,Kolkata)   
(3,Rajesh,Khanna,22,9848022339,Delhi)   
(4,Preethi,Agarwal,21,9848022330,Pune)

Task07



Load the suitable portion of data

001,Rajiv,Reddy,9848022337,Hyderabad  
002,siddarth,Battacharya,9848022338,Kolkata  
003,Rajesh,Khanna,9848022339,Delhi  
004,Preethi,Agarwal,9848022330,Pune  
005,Trupthi,Mohanthy,9848022336,Bhuwaneshwar  
006,Archana,Mishra,9848022335,Chennai.

grunt> STORE student INTO ' hdfs://localhost:9000/pig\_Output/ ' USING PigStorage (',');

Output will be the file with the same data

Task08



Distinct function return unique row of data if we have some duplicate rows.

Task09



Determine suitable column in our dataset :

(1, {(1,2), (1,3), (1,4)} )

(2, {(2,5), (2,6), (2,7)} )

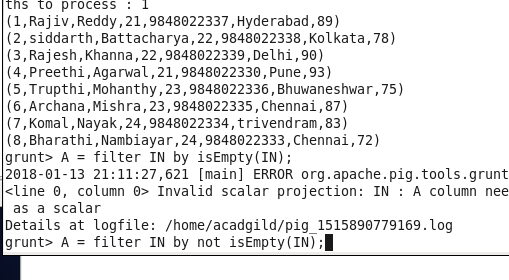
grunt> r4 = foreach r3 { Tmp=foreach $1 generate (a,b); generate FLATTEN(BagToTuple(Tmp)); };

The result

((1,2),(1,3),(1,4)) ((2,5),(2,6),(2,7))

Task10





Result

