|  |  |
| --- | --- |
| Command | Input File |
| concat | student\_details.txt(Attached with git hub link) |
| tokenize | student\_details.txt |
| sum | student\_details.txt |
| min | student\_details.txt |
| max | student\_details.txt |
| limit | student\_details.txt |
| store | student\_details.txt |
| distinct | Student1\_details1.txt(Attached with git hub link) |
| flatten | student\_details.txt |
| isempty | File1.txt andFile2.txt(Attached with git hub link) |

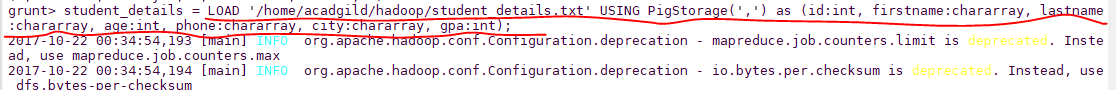
1. Concat Command :

To start pig shell (pig –x local)



-Loading a data(student\_details.txt) to pig

“student\_details = LOAD '/home/acadgild/hadoop/student\_details.txt' USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray, gpa:int);”

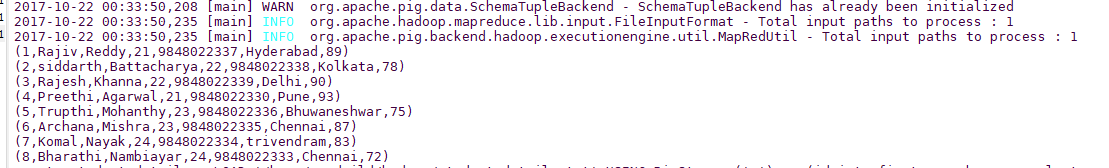


-Dumping a data

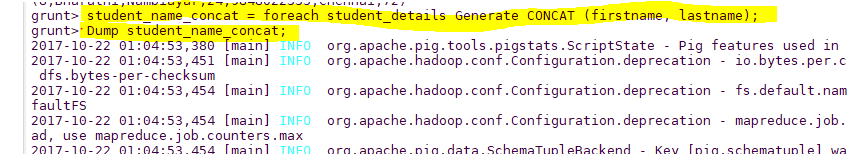
DUMP student\_details;



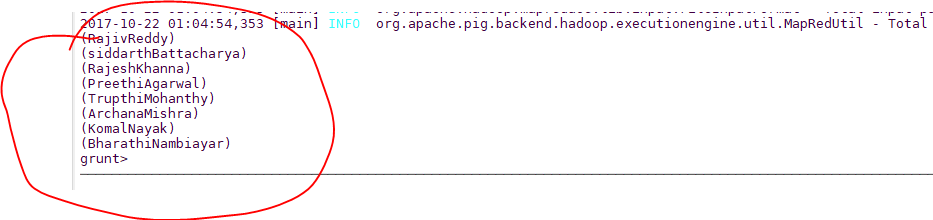
Output of “DUMP student\_details;”



-Running concat command “student\_name\_concat = foreach student\_details Generate CONCAT (firstname, lastname);”



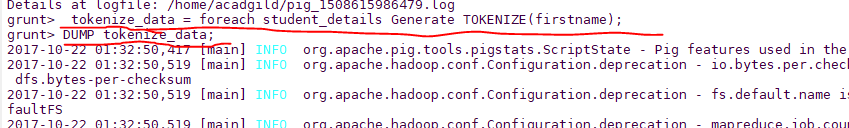
Output of “student\_name\_concat”



2)Tokenize Command

To tokenize the firstname

-tokenize\_data = foreach student\_details Generate TOKENIZE(firstname);



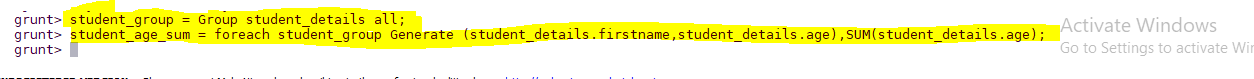
Output of Tokenize



3)sum

-student\_group = Group student\_details all;

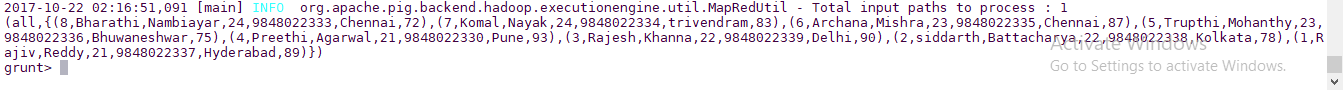
-student\_age\_sum = foreach student\_details Generate (student\_details.firstname,student\_details.age),SUM(student\_details.age);

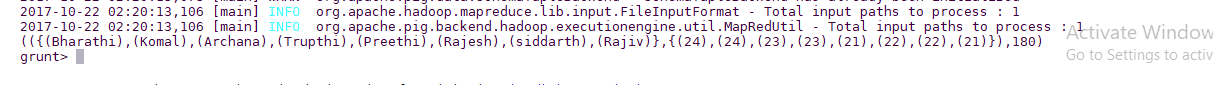


Output

-DUMP student\_group;

-DUMP student\_age\_sum;





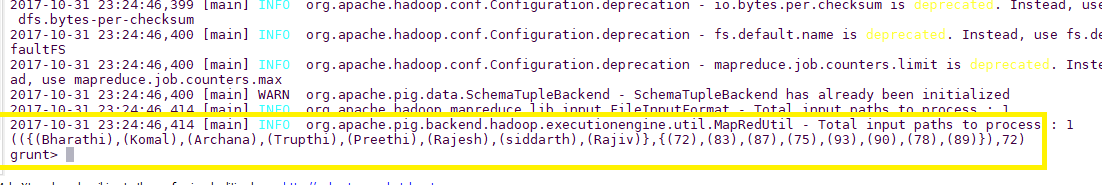
4)Min

-student\_group\_all = Group student\_details All;

-student\_gpa\_min = foreach student\_group\_all Generate (student\_details.firstname, student\_details.gpa), MIN(student\_details.gpa);

-Dump student\_gpa\_min;

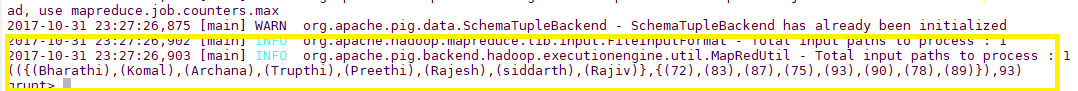
Output



5)max

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

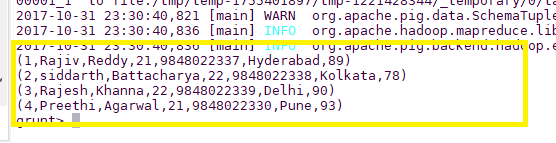
-DUMP student\_gpa\_max;



6)limit

-limit\_data = LIMIT student\_details 4;

-Dump limit\_data;



7)store

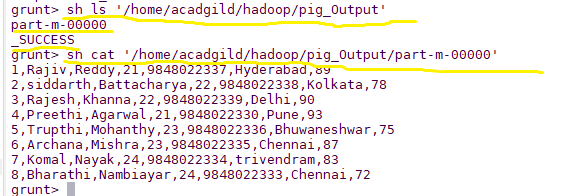
-STORE student INTO ' /home/acadgild/hadoop//pig\_Output/ ' USING PigStorage (',');

-Sh ls ‘/home/acadgild/hadoop/pig\_Output’

-Sh cat ‘/home/acadgild/hadoop/pig\_Output/part-m-00000’

Output

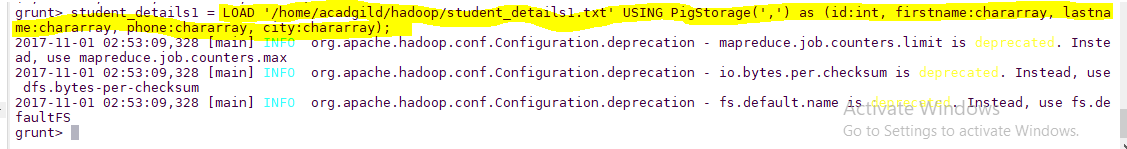


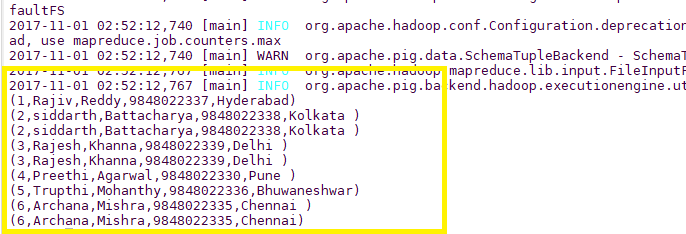


8)Distinct

-student\_details1 = LOAD '/home/acadgild/hadoop/student\_details1.txt' USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, phone:chararray, city:chararray);

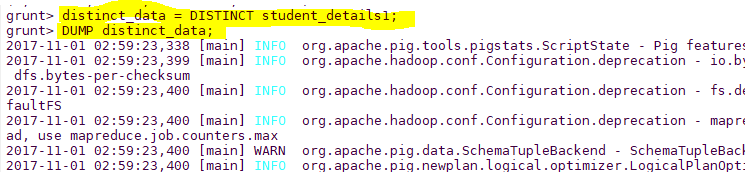
-DUMP student\_details1;

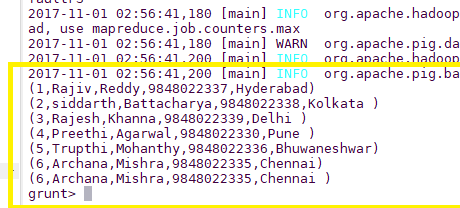




-distinct\_data = DISTINCT student\_details1;

-DUMP distinct\_data;



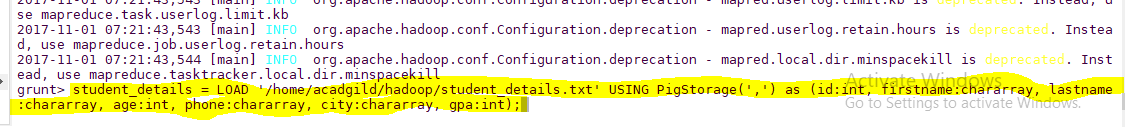


9)Flatten:

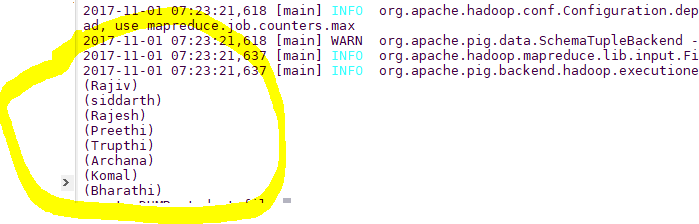
-student\_details = LOAD '/home/acadgild/hadoop/student\_details.txt' USING PigStorage(',') as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray,

-student\_file = FOREACH student\_details GENERATE FLATTEN(TOKENIZE(firstname)) as word;

-DUMP student\_file;







10)IsEmpty

emp\_sales = LOAD '/home/acadgild/hadoop/File1.txt' USING PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

emp\_bonus = LOAD '/home/acadgild/hadoop/File2.txt' USING PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

DUMP emp\_sales;

DUMP emp\_bonus;

cogroup\_data = COGROUP emp\_sales by age, emp\_bonus by age;

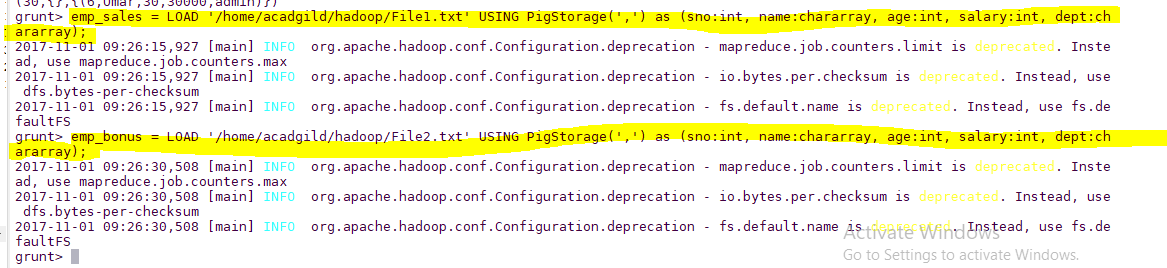
DUMP cogroup\_data;

isempty\_data = filter cogroup\_data by IsEmpty(emp\_sales);

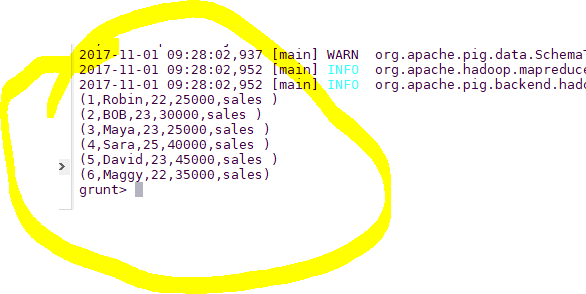
DUMP isempty\_data;

Screenshot

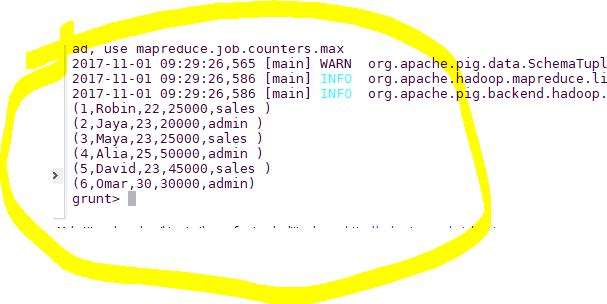
-emp\_sales = LOAD '/home/acadgild/hadoop/File1.txt' USING PigStorage(',') as (sno:int, name:chararray, age:int, salary:int, dept:chararray);



-DUMP emp\_sales;

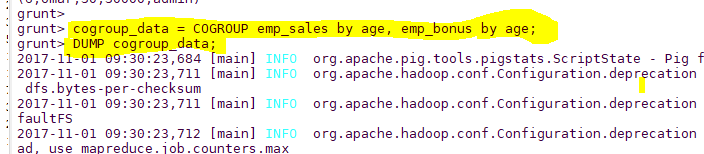


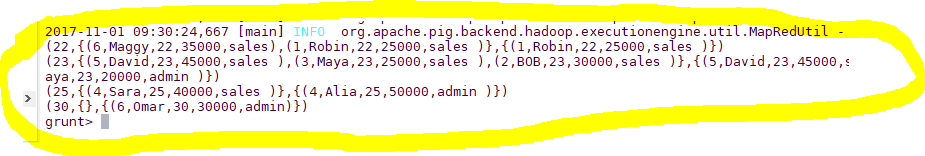
-DUMP emp\_bonus;



-cogroup\_data = COGROUP emp\_sales by age, emp\_bonus by age;

-DUMP cogroup\_data;





-isempty\_data = filter cogroup\_data by IsEmpty(emp\_sales);

-DUMP isempty\_data;

