**Session 8: Pig**

**Assignment 2**

**Steps of Execution.**

1. Create a sample\_data as shown below.



Here created sample data with ‘,’ separated

1st :First\_Name

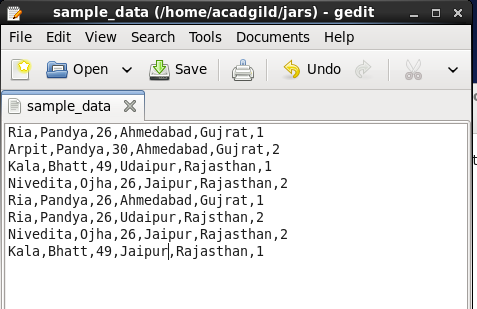
2nd : Last Name

3rd : Age

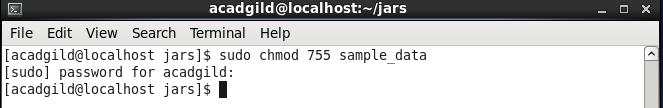
4th : City

5th: State

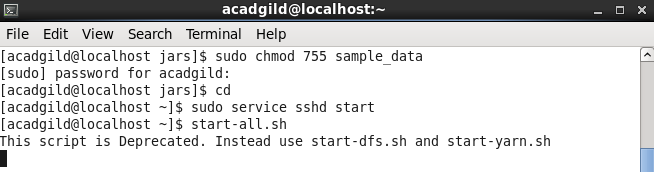
6th: Number\_of\_houses



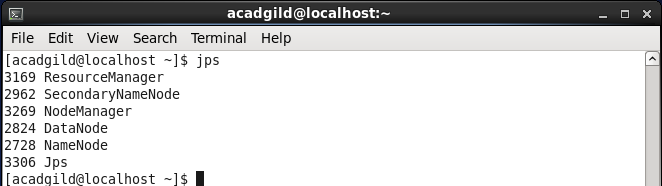
2. Change the permission of file as shown below.



3. Start the service of hdfs as shown below

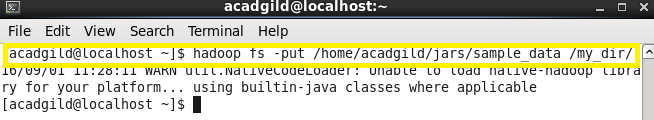


4. Check the health of nodes by typing jps command

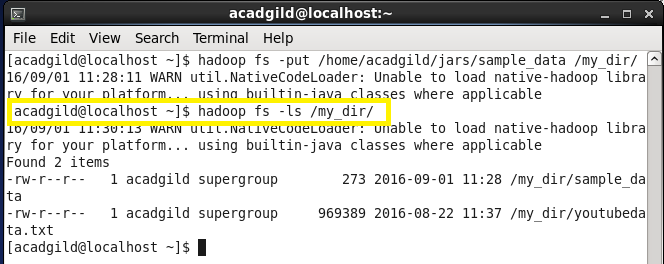


5. Copy the file from local file system to hadoop file system using command put as shown below.

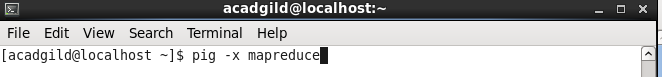
My files are present in jars folder in local file system and I am storing files in my\_dir on hadoopfile system.



Listing of files present in hdfs system in folder name my\_dir can be done using ls command as shown below.



6. Run pig in mapreduce mode as shown below



7. Commands in Pig

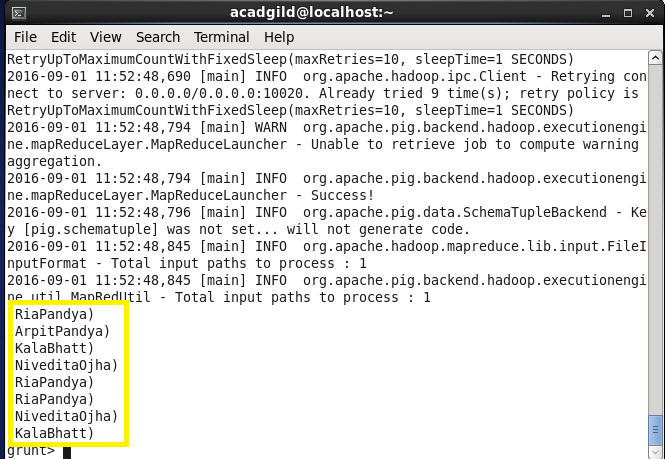
**Load the data**

student\_details = load ‘/my\_dir/sample\_data’ USING PigStorage(‘,’) as (f\_name:chararray,l\_name:chararray,age:int,city:chararray,state:chararray,no\_house:int);

a) **CONCAT Command:** Concatenating first name and last name.

concat\_students = FOREACH student\_details GENERATE CONCAT(f\_name,l\_name);

dump concat\_students;



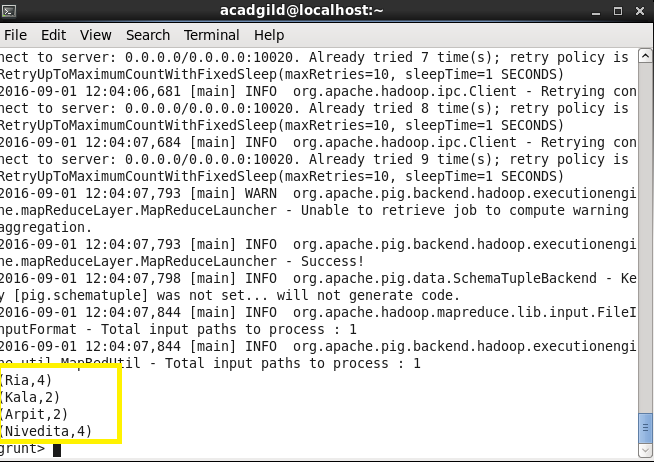
**b) Sum Command:** SUM function to compute the sum of a set of numeric values in a single-column bag.

Here doing sum of no\_houses on basis of first name.

group\_name = GROUP student\_details by f\_name;

sum\_house = FOREACH group\_name GENERATE group , SUM(student\_details.no\_house);

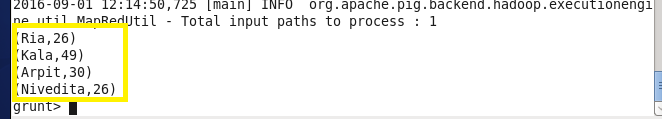
dump sum\_house;



**C) Min Command:** Use the MIN function to compute the minimum of a set of numeric values or chararrays in a single-column bag. MIN requires a preceding GROUP… ALL statement for global minimums and a GROUP … BY statement for group minimums.

min\_age = FOREACH group\_name GENERATE group , MIN(student\_details.age);

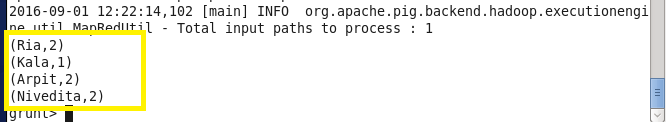
dump min\_age;



**d) MAX Command:** Computes the maximum of the numeric values or chararrays in a single-column bag. MAX requires a preceding GROUP ALL statement for global maximums and a GROUP BY statement for group maximums.

max\_house= FOREACH group\_name GENERATE group,MAX(student\_details.no\_house);

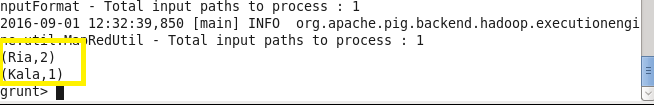
dump max\_house;



**e) LIMIT Command:** Is used to specify the range of data set we need to dump.

limit\_maxHouse = limit max\_house 2;

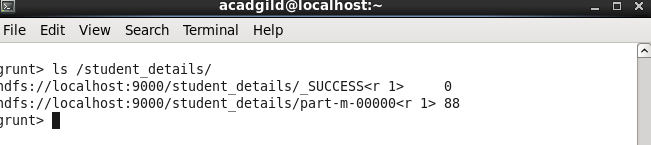
dump limit\_maxHouse;



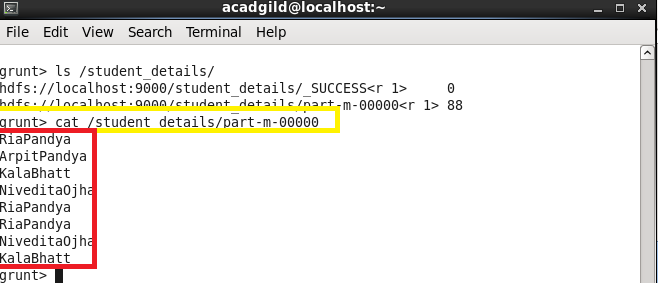
**f) STORE Command:** The DUMP command is only used to display information onto the standard output. If we need to store the data to a file we can use the store command:

store concat\_students into ‘/student\_details/’;

ls /student\_details/



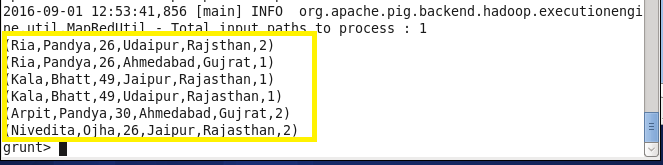
We can see the output using cat command as shown below.



**g) DISTINCT Command:** The DISTINCT statement is used to remove duplicated records. It works only on entire records, not on individual fields.

distinct\_students = DISTINCT student\_details;

dump distinct\_students;



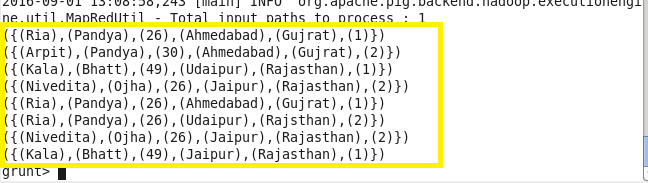
**h) TOKENIZE Command**:- Splits a string and outputs a bag of words.

student\_details1 = LOAD ‘/my\_dir/sample\_data’ as (details:chararray);



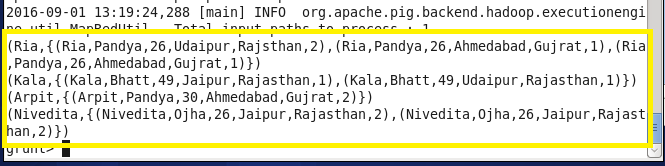
tokenize\_students= FOREACH student\_details1 GENERATE TOKENIZE(details);

dump tokenize\_students;



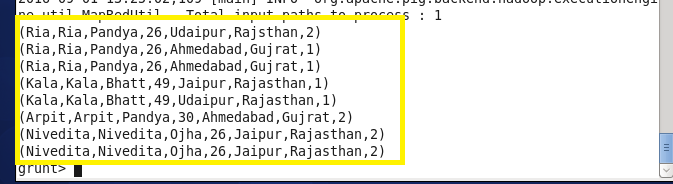
**i)FLATTEN:** Flatten un-nests tuples as well as bags.

dump group\_name;

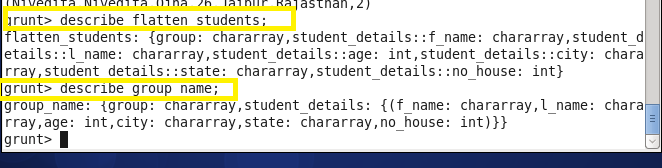


flatten\_student = FOREACH group\_name GENERATE $0 , FLATTEN($1);

dump flatten\_student;



After describing group\_name and flatten\_student



**j) IS EMPTY Command:** Checks if a bag or map is empty.

X = COGROUP student\_details by f\_name , min\_age by group;

isempty\_data = filter X by IsEmpty(student\_details.f\_name);

dump isempty\_data;

