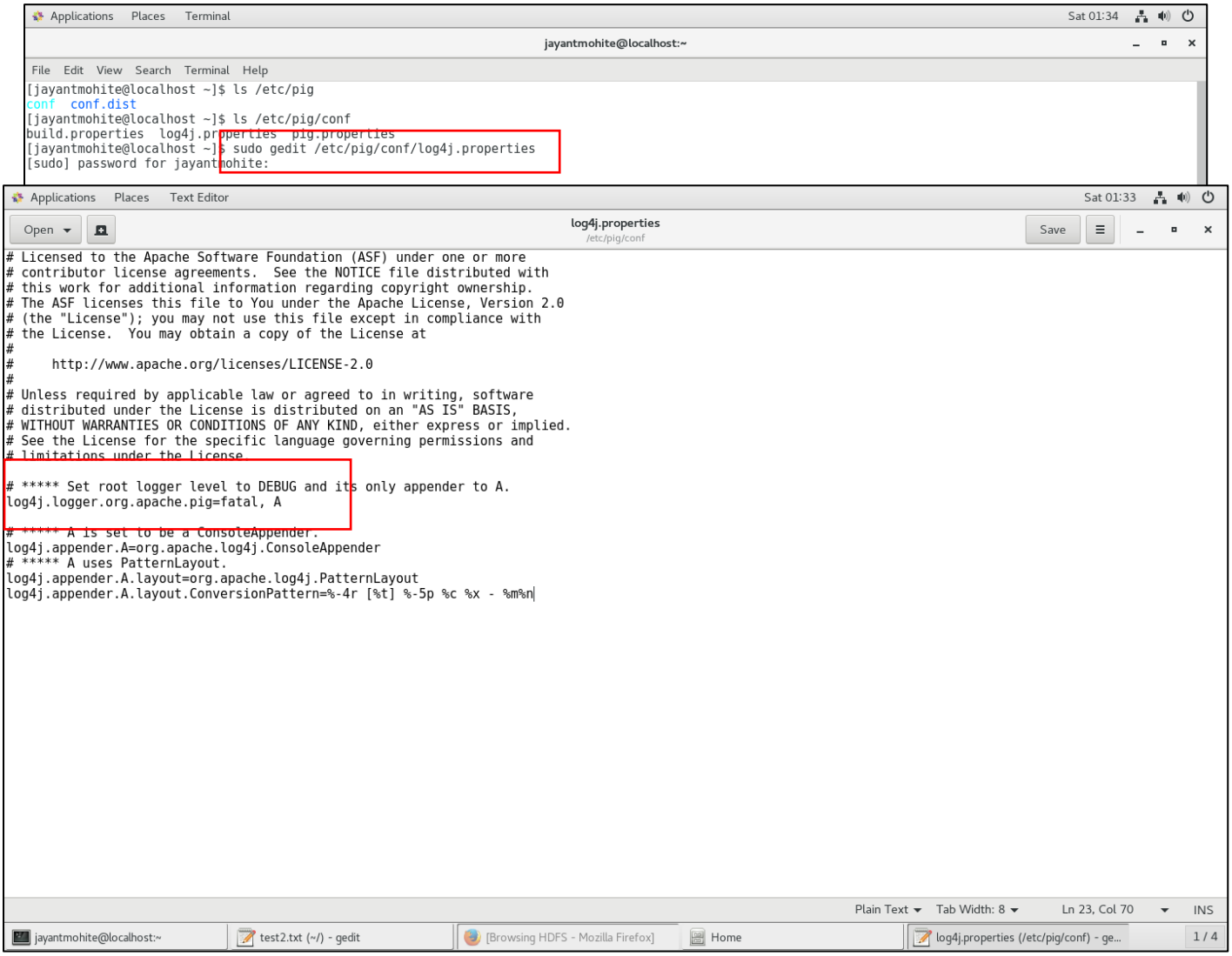
Apache Pig Tutorial

Pig Configuration to avoid displaying logs

We will modify the logging properties of Pig and set it to a mode where only fatal error are displayed. For that we will have to make changes in log4j.properties file available in the configuration directory of Pig.



Steps:

Open the pig log properties by using the command

sudo gedit /etc/pig/conf/log4j.properties

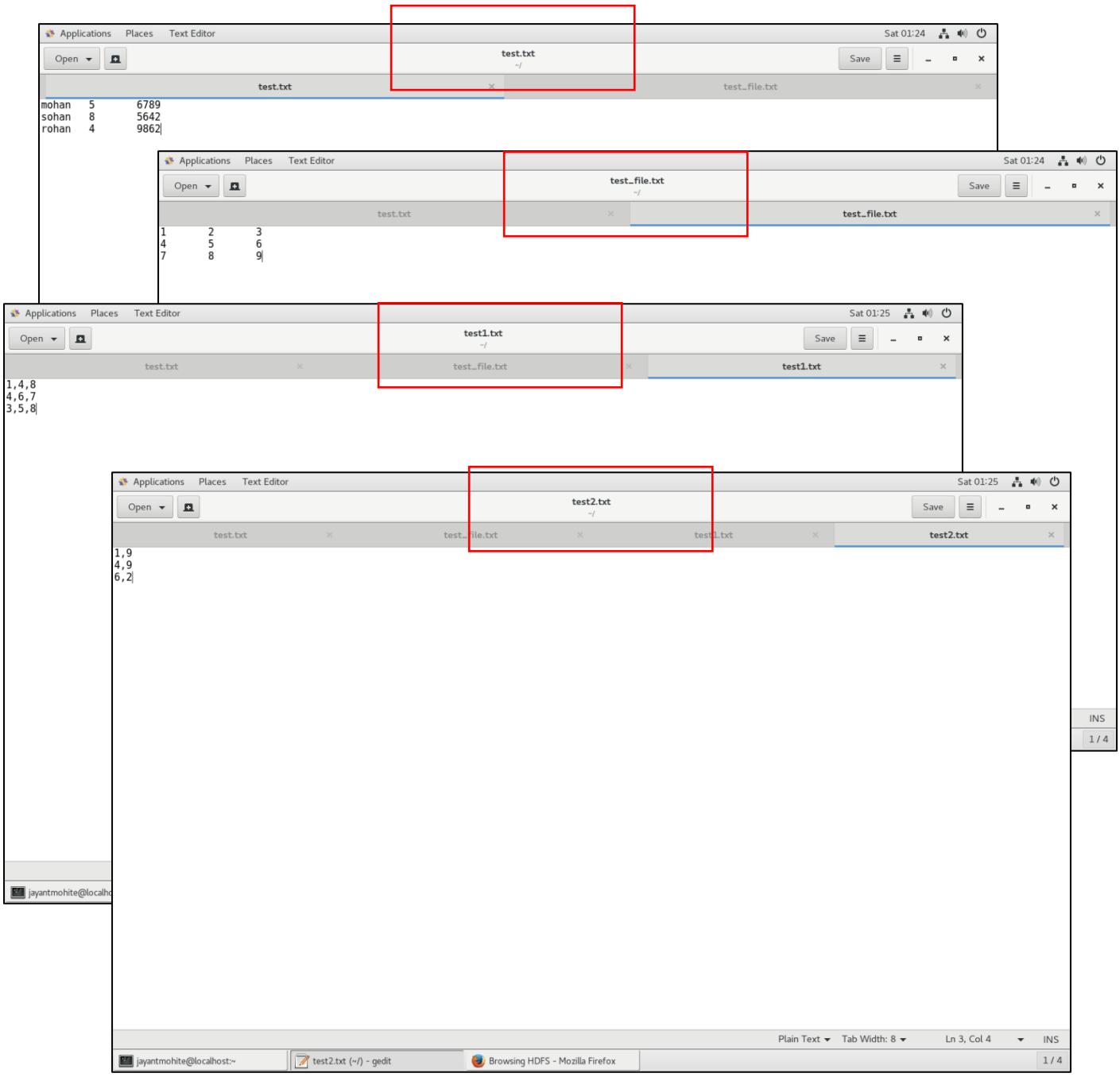
and the to it a line which states

log4j.logger.org.pache.pig=fatal, A

Author: Jayant Mohite

For this lab session we will be using Pig in local mode.

Sample files used:



Author: Jayant Mohite

Step Description:

In this step we will load the input file test.txt located at /home/jayantmohite/ with a schema of 3 columns namely name, emp_num and salary.

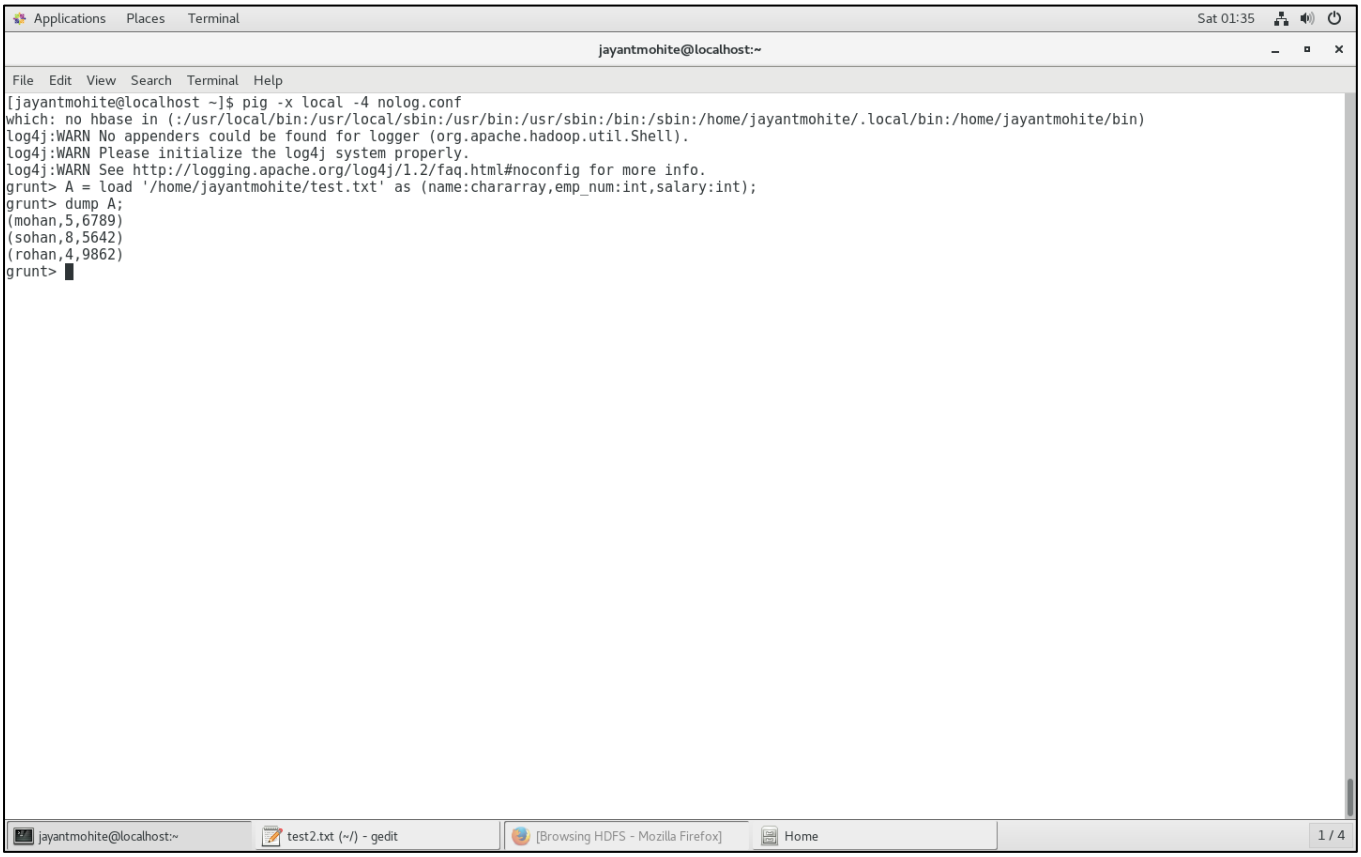
In this case we are not specifying any delimiter as the delimiter used in the file test.txt is tab and in Pig tab (\t) is default delimiter.

Commands:

grunt> A = load '/home/jayantmohite/test.txt' as (name:chararray, emp_num:int, salary:int);
(In this command, A is the bag that we are creating)

grunt> dump A;

Step Visualization:



Do note that all labs in this chapter are performed in the Pig Local Mode.

Step Description

In this step we will create a bag B couple of time with the same input file name test_file.txt. The difference in both these execution will be that first execution, we will not specify any schema but in second execution we will specify the schema.

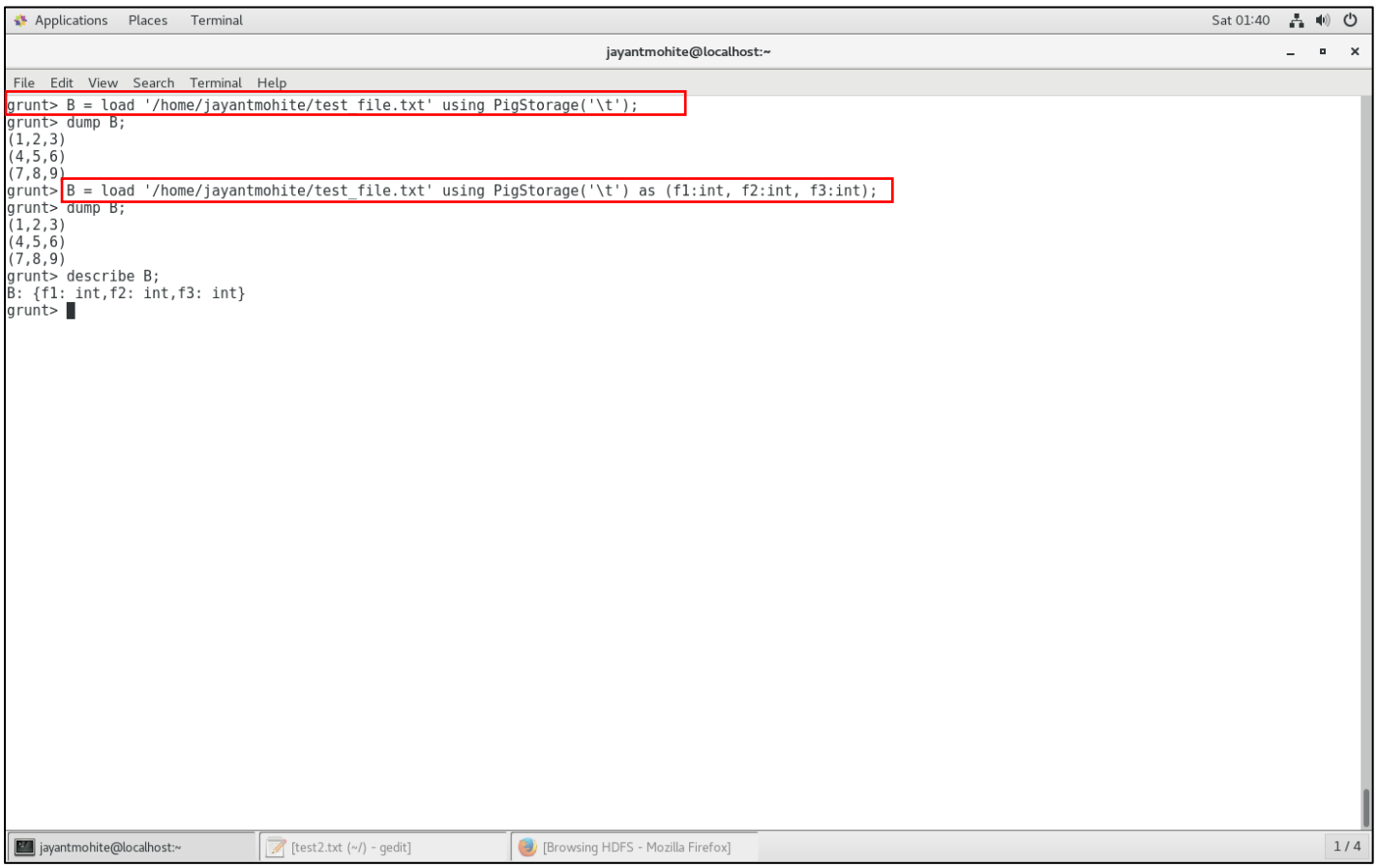
In case where we don't specify the schema, we can reference columns by their positional parameters like column 1 is referenced as $0; column 2 is referenced as $1 and so on.

Commands:

grunt> B = load '/home/jayantmohite/test_file.txt' using PigStorage('\t');
(In this command, though not required but we have specified the delimiter used in the file by using the function PigStorage. We are not specifying any schema in this command.)

grunt> B = load '/home/jayantmohite/test_file.txt' using PigStorage('\t') as (f1:int, f2:int, f3:int);
(In this command, we are specifying the schema for the bag which can be checked by the Describe command.)

Step Visualization:



Author: Jayant Mohite

Step Description:

In this step we will have a look at some operators in Pig like filter, foreach and generate.

Commands:

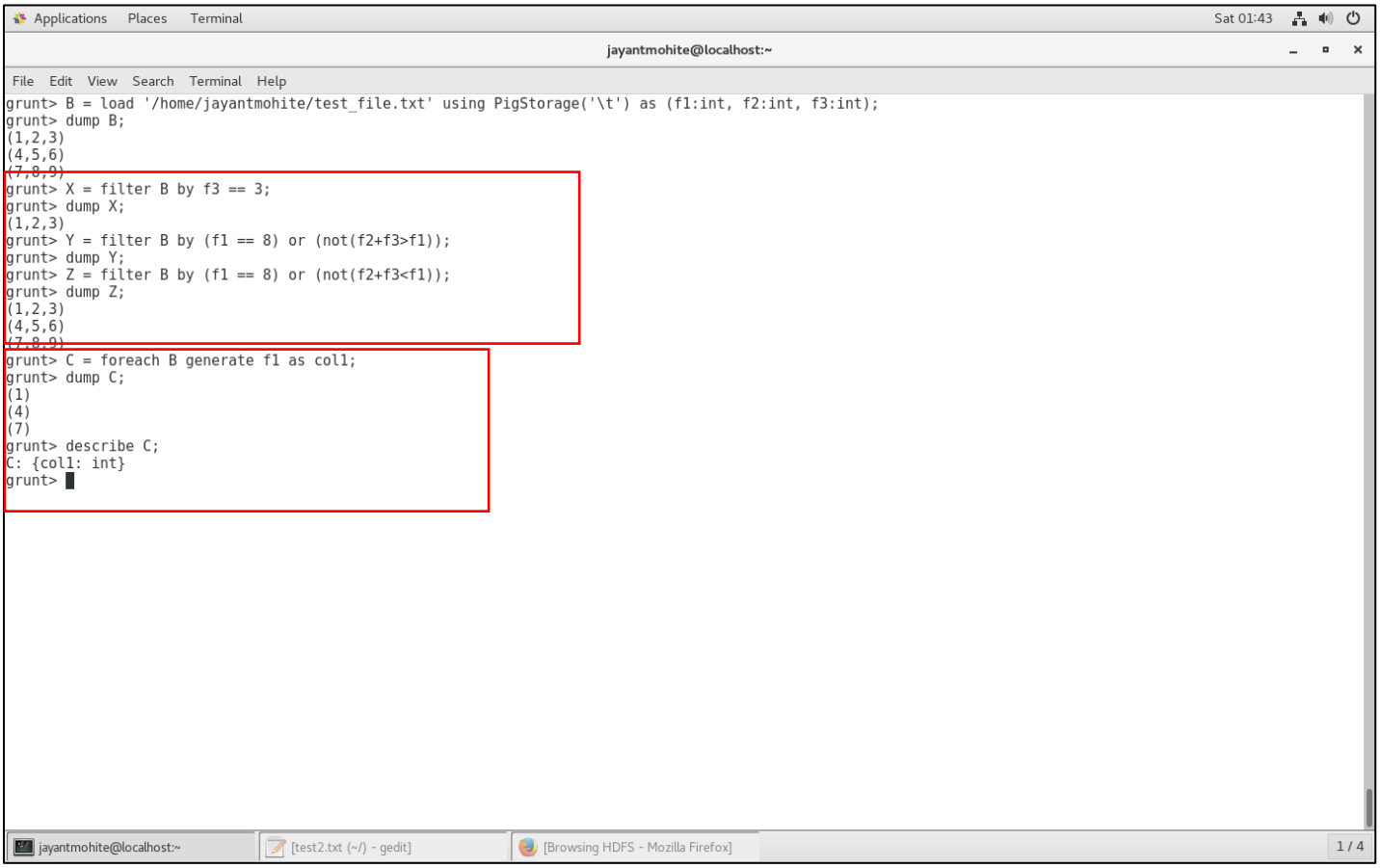
grunt> B = load '/home/jayantmohite/test_file.txt' using PigStorage('\t') as (f1:int, f2:int, f3:int);

grunt> X = filter B by f3 == 3;
(In this command we are creating a bag by name X which will contains records from the bag B where the column f3 has a value equal to 3)

grunt> Z = filter B by (f1 == 8) or (not(f2 + f3 < f1));

grunt> C = foreach B generate f1 as col1;
(In this command we are creating a bag by name C, which will have data of all records only from column f1 of bag B. This new bag C will have a schema where the only column it has will have a name as col1)

Step Visualization:



Author: Jayant Mohite

Step Description:

In this step we will see how Group By works in Pig

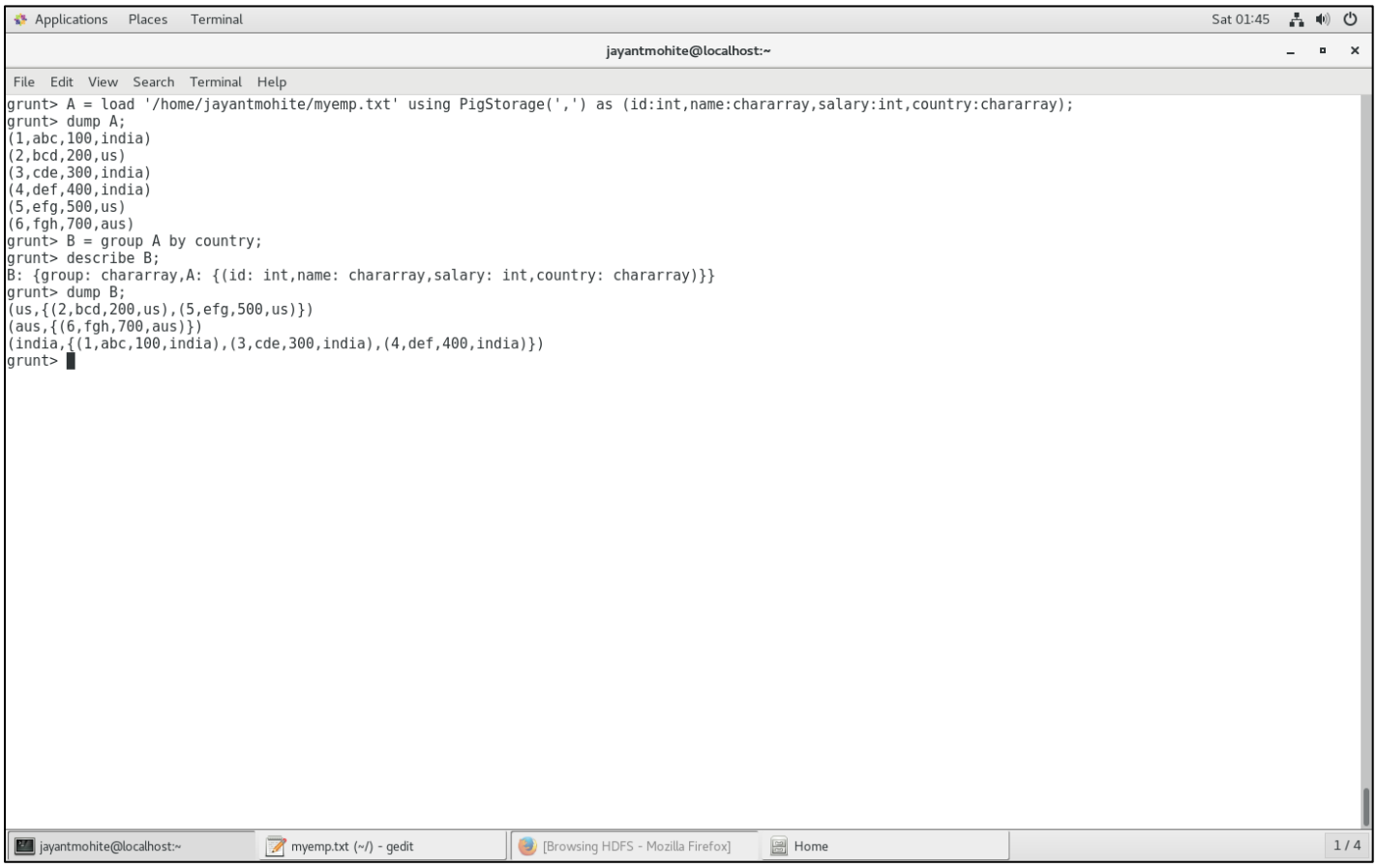
Commands:

grunt> A = load '/home/jayantmohite/myemp.txt' using PigStorage (',') as (id:int, name:chararray, salary:int, country:chararray);
(for this you can use the same input file that we have used in Hive Partitions)

grunt> B = group A by country;

grunt> describe B;
(this command will show the schema of the bag B which contains records from bag A grouped by country. In this bag, every tuple will have 2 atoms. The 1st atom will be the key group and the 2nd atom will be the elements of bag A that belong to this group.)

Step Visualization:



Step Description:

In this step we will have a demonstration of the join and union commands of pig wherein we will be working with two bags in the same command.

Command:

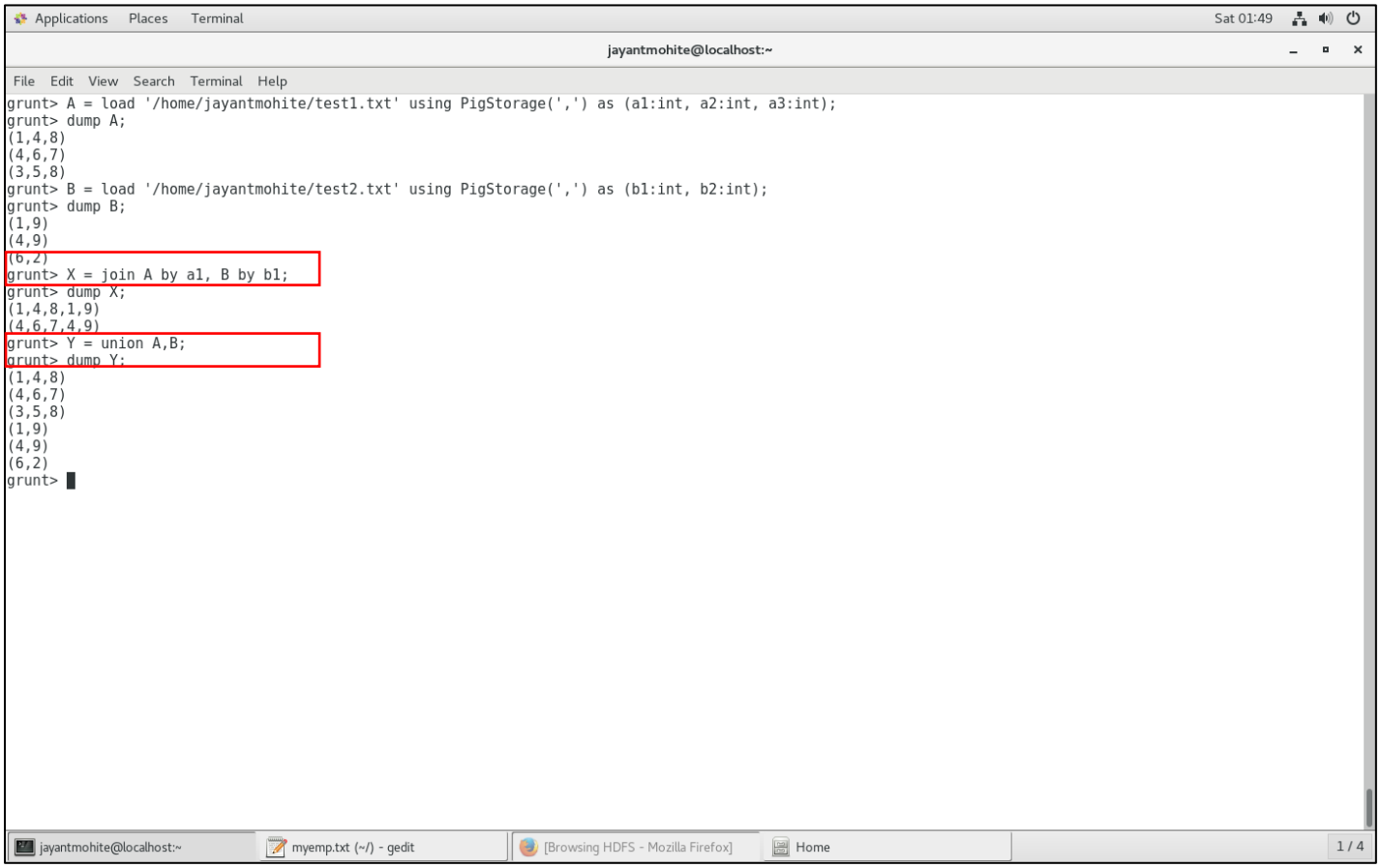
grunt> A = load '/home/jayantmohite/test1.txt' using PigStorage(',') as (a1:int, a2:int, a3:int);

grunt> B = load '/home/jayantmohite/test2.txt' using PigStorage(',') as (b1:int, b2:int);

grunt> X = join A by a1, B by b1;
(the data that we are loading in bag X can be visualized by the SQL query as
select * from A,B where A.a1 = B.b1)

grunt> Y = union A, B;

Step Visualization:



Step Description:

In this step we will have a demonstration of the Sample command. Lets say we have an input data with 10 million records and we need to run some pig analytics on to it. And we are not sure if our program will yield the correct result. In order to verify this, it would be great if we could test our program on a subset of the actual data.

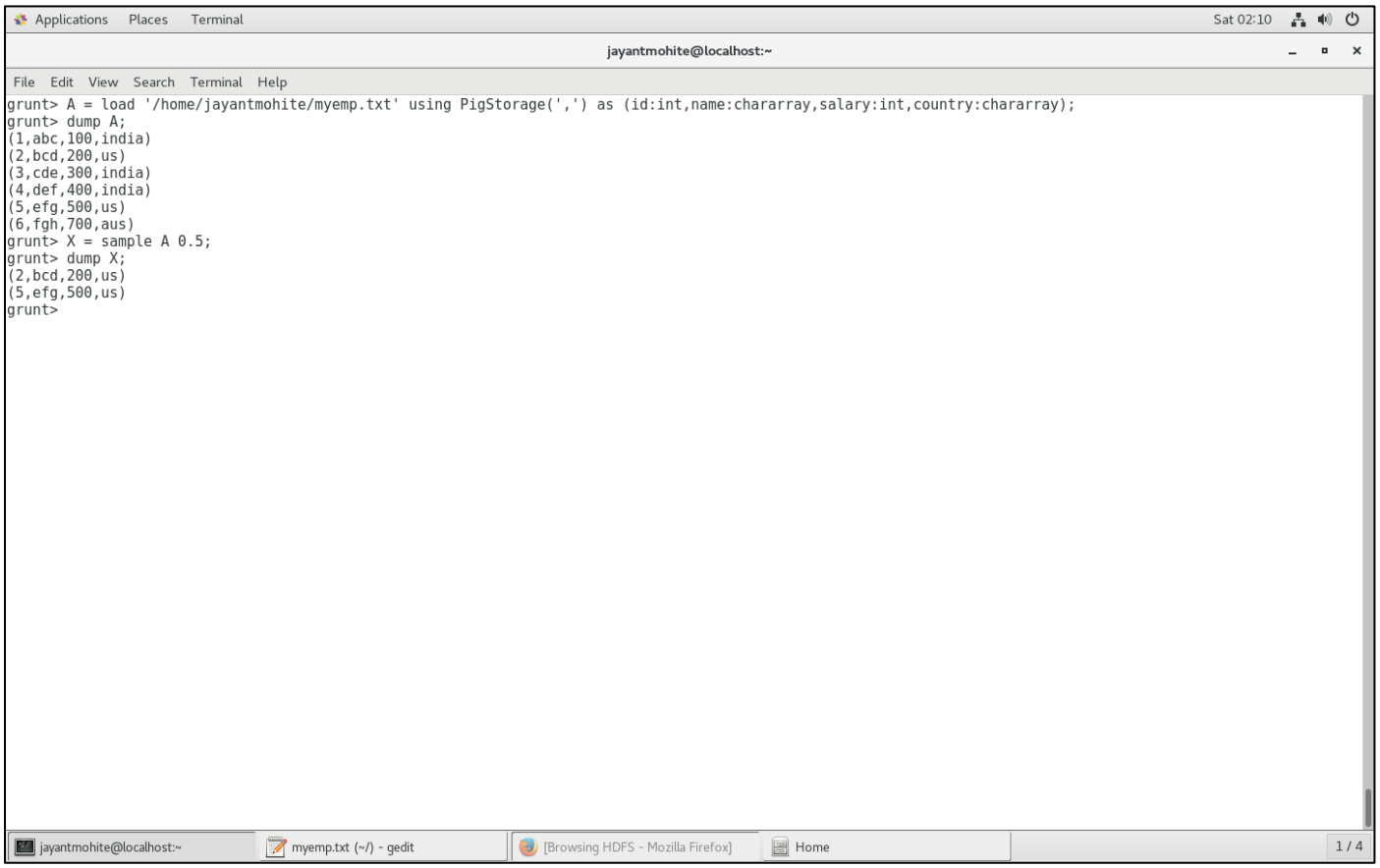
This is exactly what sample command does.

Command:

grunt> A = load '/home/jayantmohite/myemp.txt' using PigStorage (',') as (id:int, name:chararray, salary:int, country: chararray);

grunt> X = sample A 0.5;
(in here o.5 is like 50% of the total input data)

Step Visualization:



Step Description:

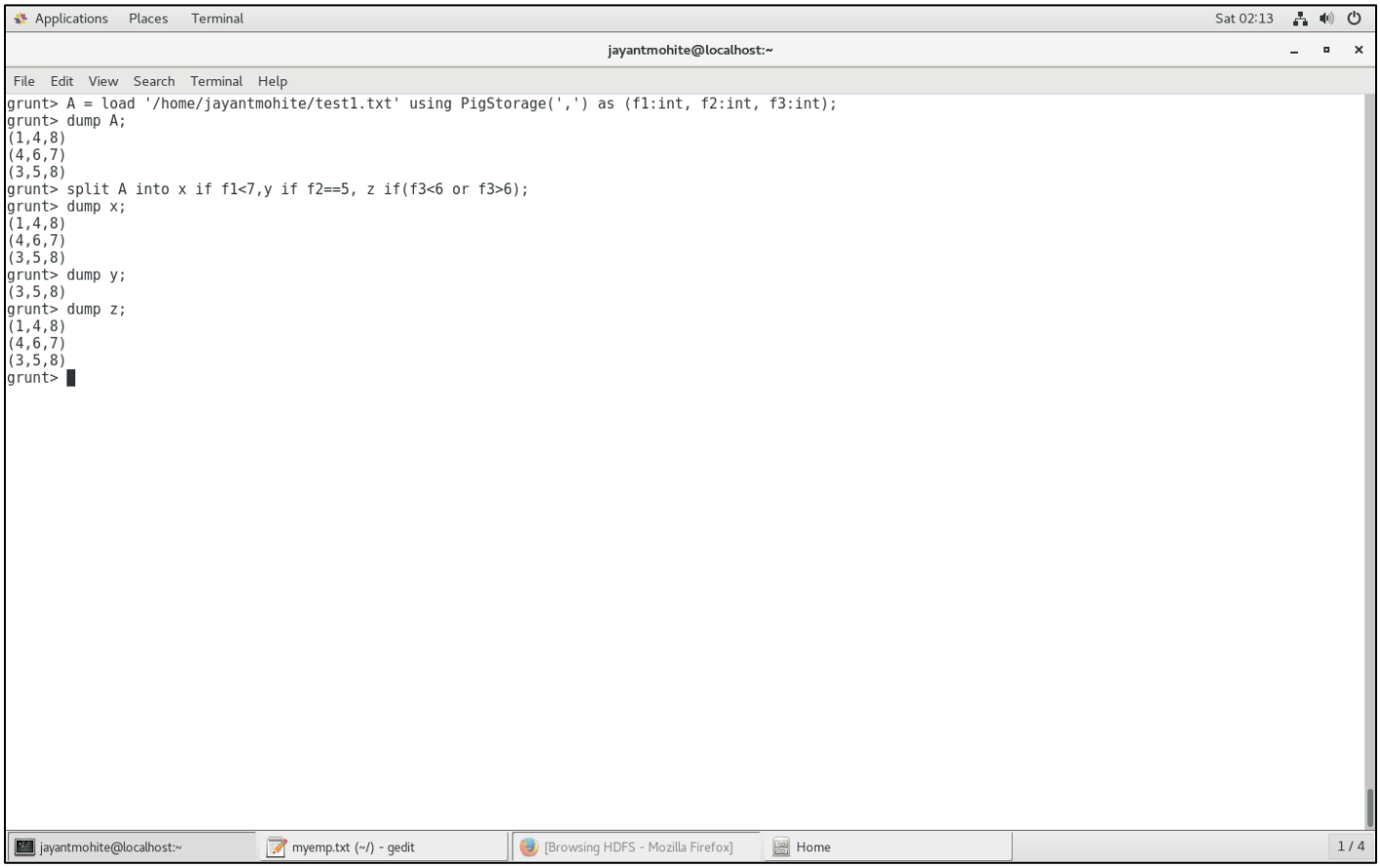
In this step we will have a look at the split command which allows you to create multiple bags from an existing bag in the same command based on some criteria

Commands:

grunt> A = load '/home/jayantmohite/test1.txt' using PigStorage(',') as (f1:int, f2:int, f3:int);

grunt> split A into x if f1 < 7, y if f2 == 5, z if (f3 < 6 or f3 > 6);
(this command will create the following 3 bags from data in bag A
bag x = all records from A that have the value of 1st column less than 7
bag y = all records from A that have value of 2nd column equal to 5
bag z = all records from A that have value of 3rd column not equal to 6)

Step Visualization:



Step Description:

In this step we will demonstrate some aggregate functions like Min, Max and Count

Commands:

grunt> A = load '/home/jayantmohite/myemp.txt' using PigStorage (',') as (id:int, name:chararray, salary:int, country:chararray);

grunt> B = group A by country;

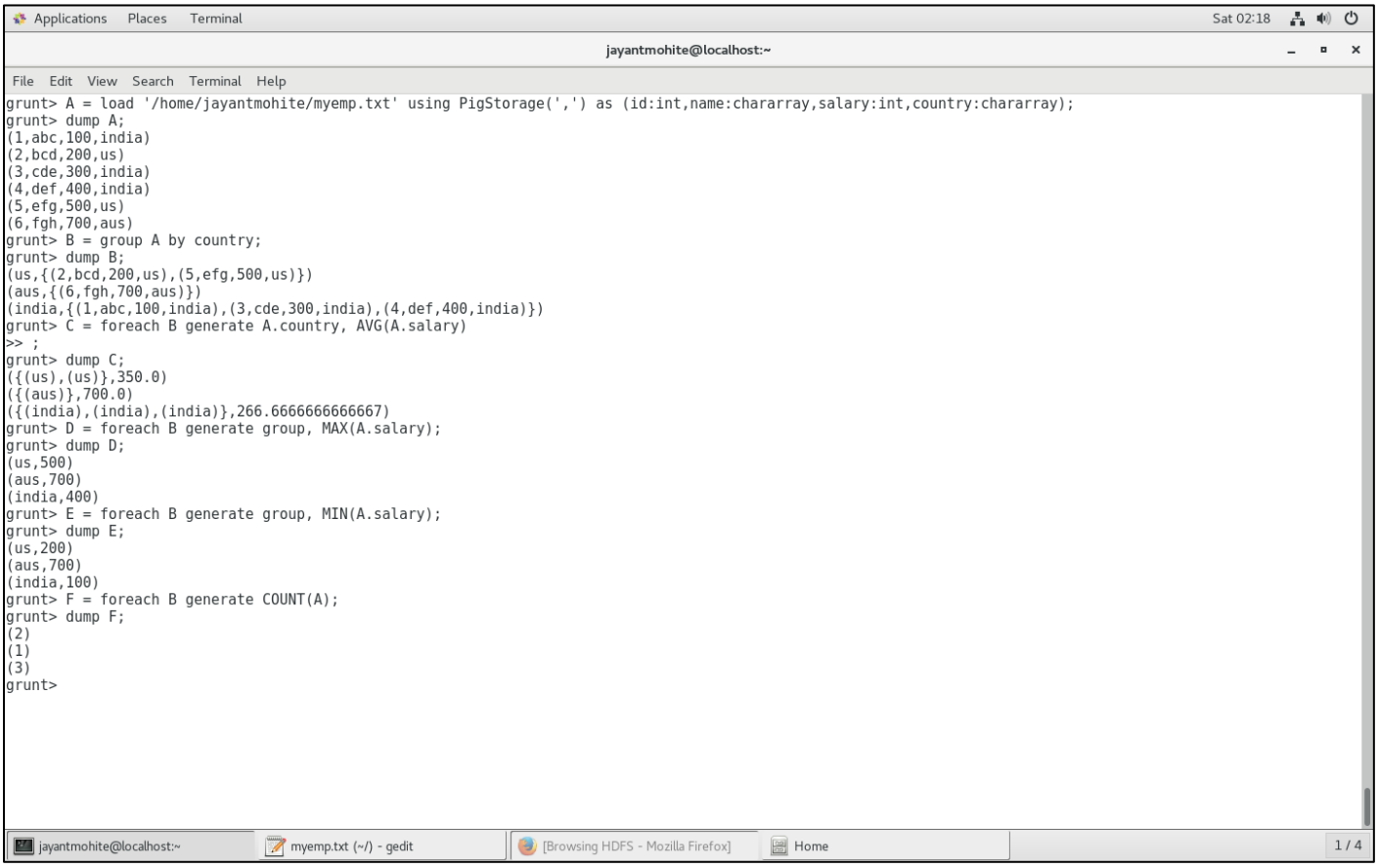
grunt> C = foreach B generate A.country, AVG(A.salary);

grunt> D = foreach B generate group, MAX(A.salary);

grunt> E = foreach B generate group, MIN(A.salary);

grunt> F = foreach B generate COUNT(A);

Step Visualization:



That is all about the Pig Latin Commands. Explore more in the next chapters.

Author: Jayant Mohite