9.2 Explain the working of below pig script commands with an example :

1. LOAD

2. STORE

3. DUMP

4. FOREACH

5. FILTER

6. GROUP BY

7. ORDER BY

8. DESCRIBE

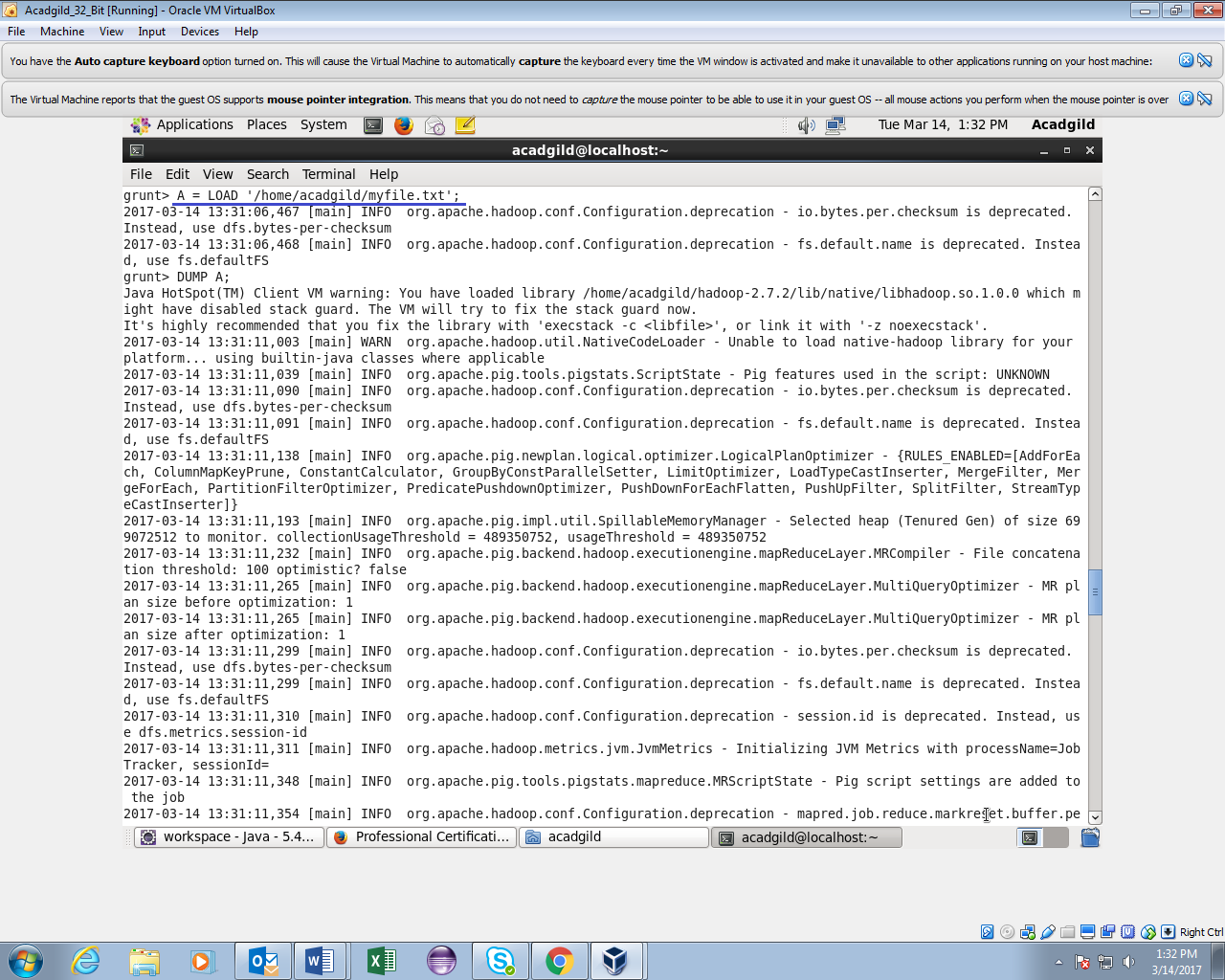
1.LOAD

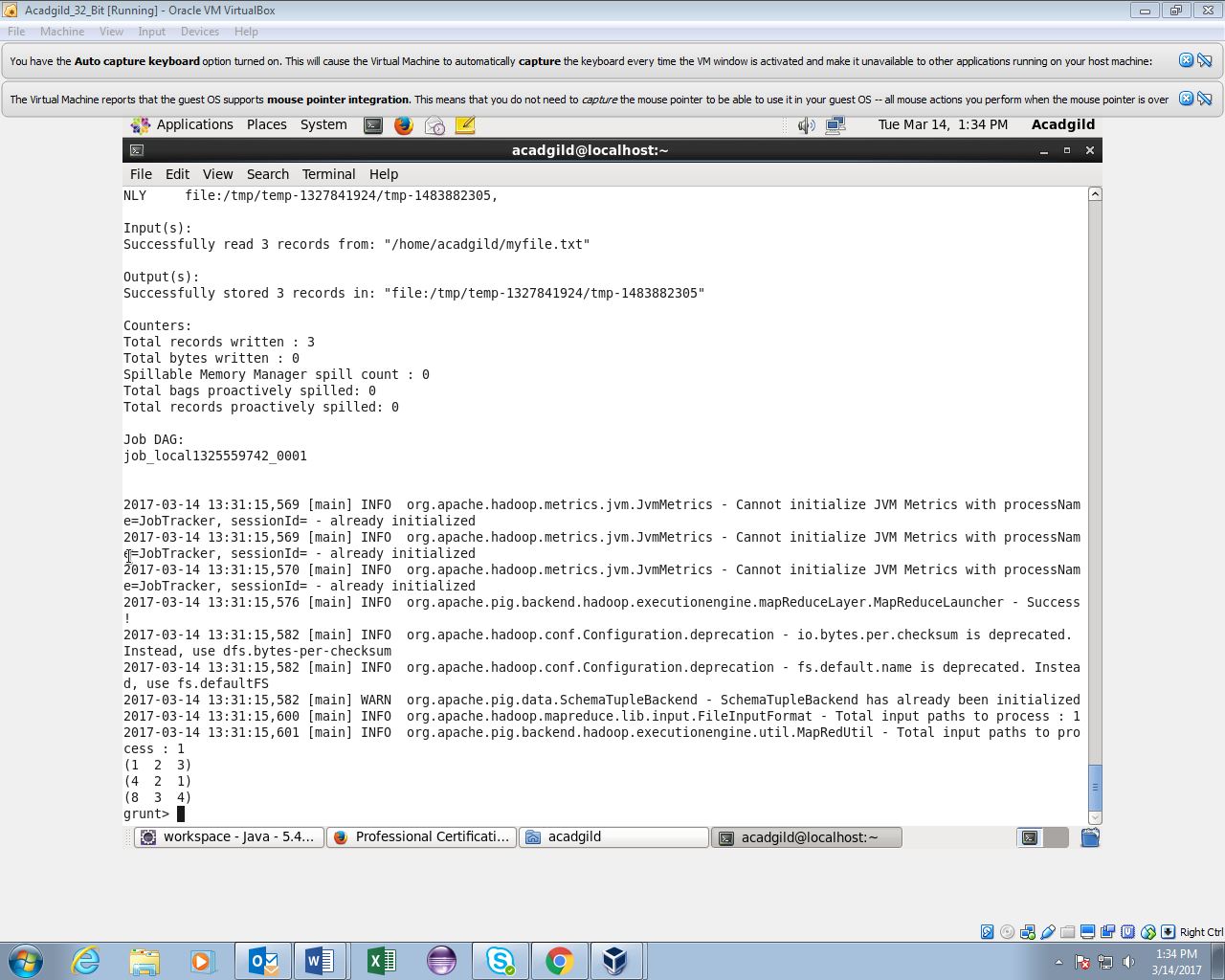
• Loads data from the file system.

Syntax • LOAD 'data' [USING function] [AS schema];

Suppose we have a data file called myfile.txt. The fields are tab-delimited. The records are newline-separated. 1 2 3 4 2 1 8 3 4

A = LOAD 'myfile.txt';



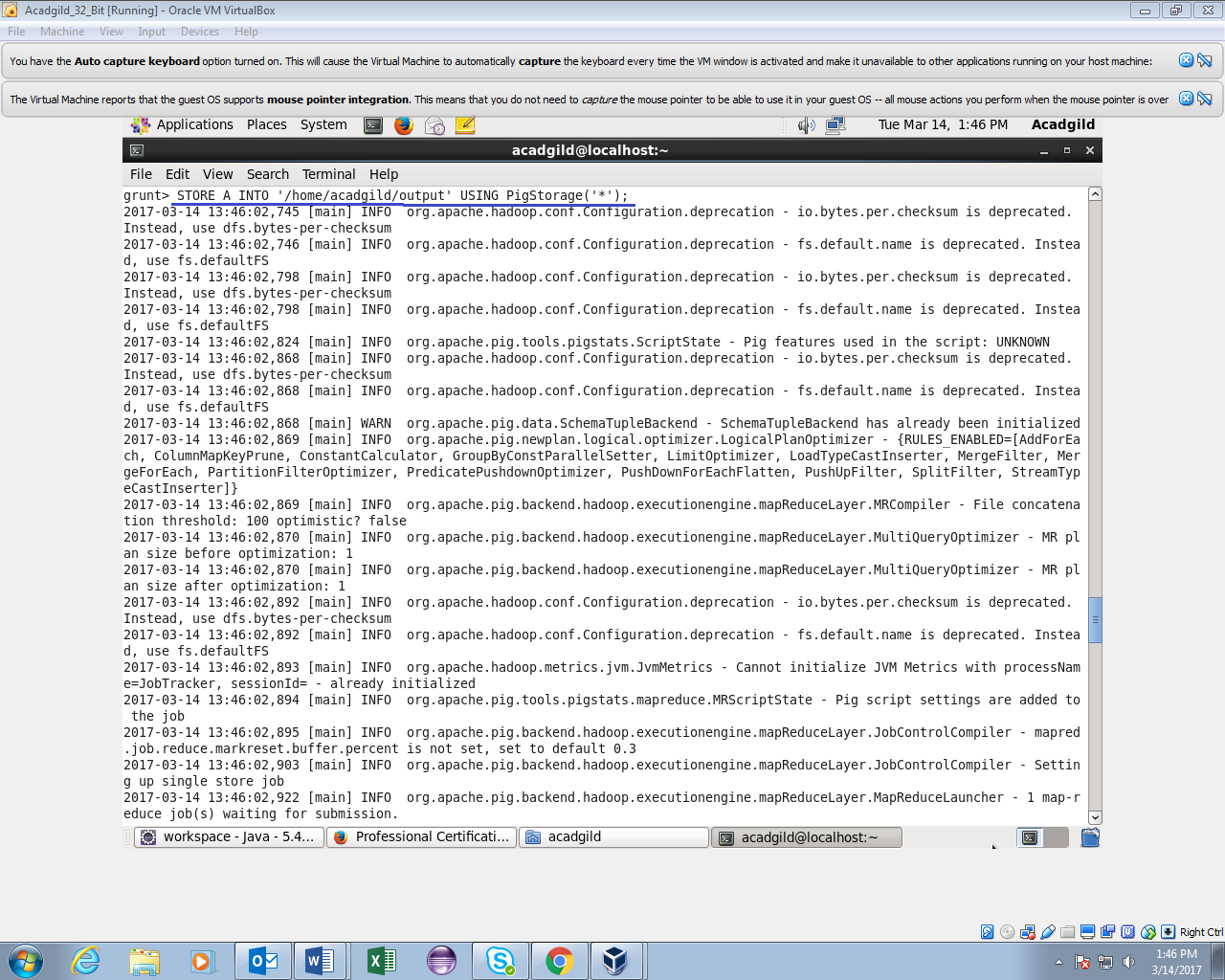


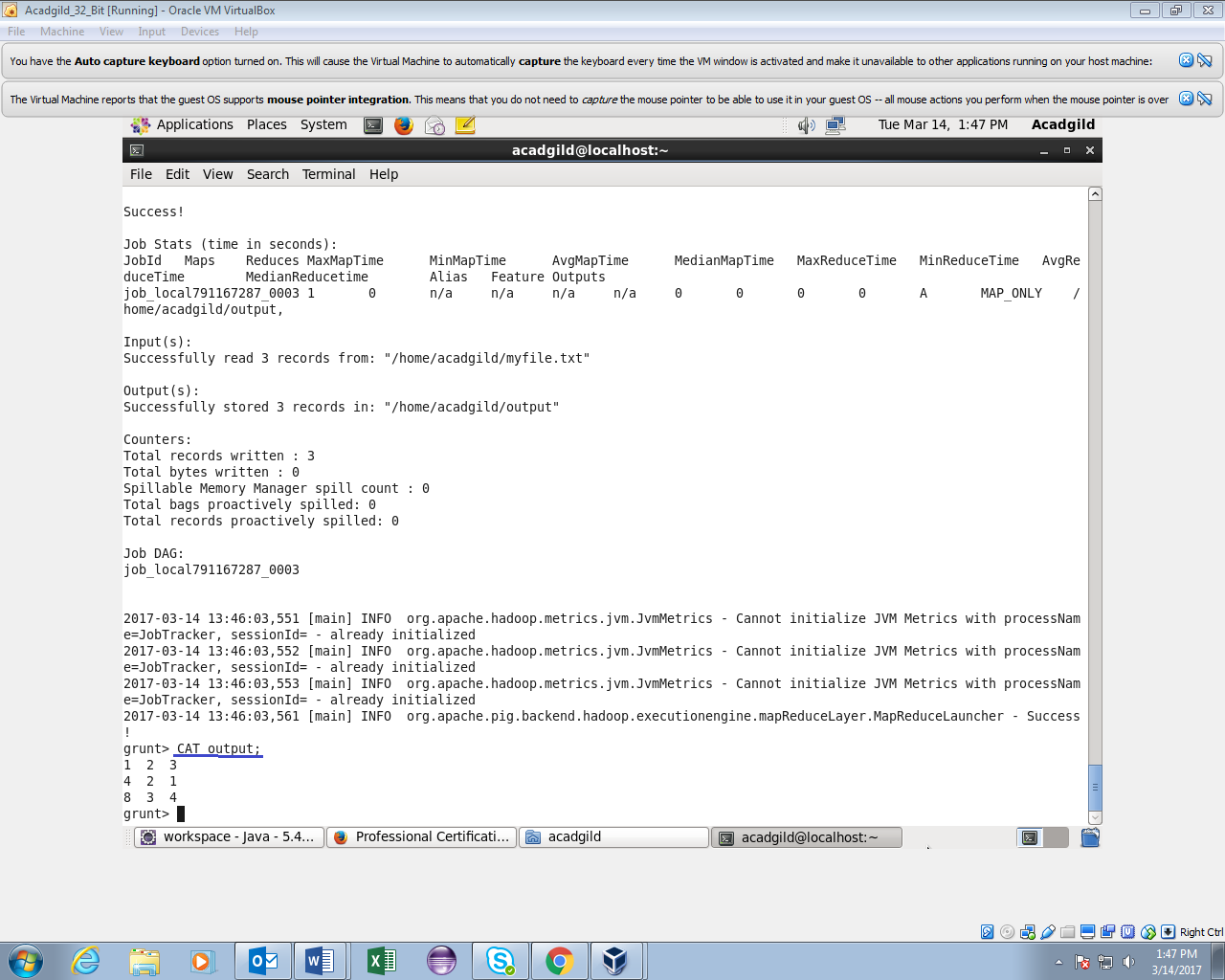
2.STORE

• Stores or saves results to the file system

Syntax • STORE alias INTO 'directory' [USING function];

STORE A INTO 'myoutput' USING PigStorage ('\*');

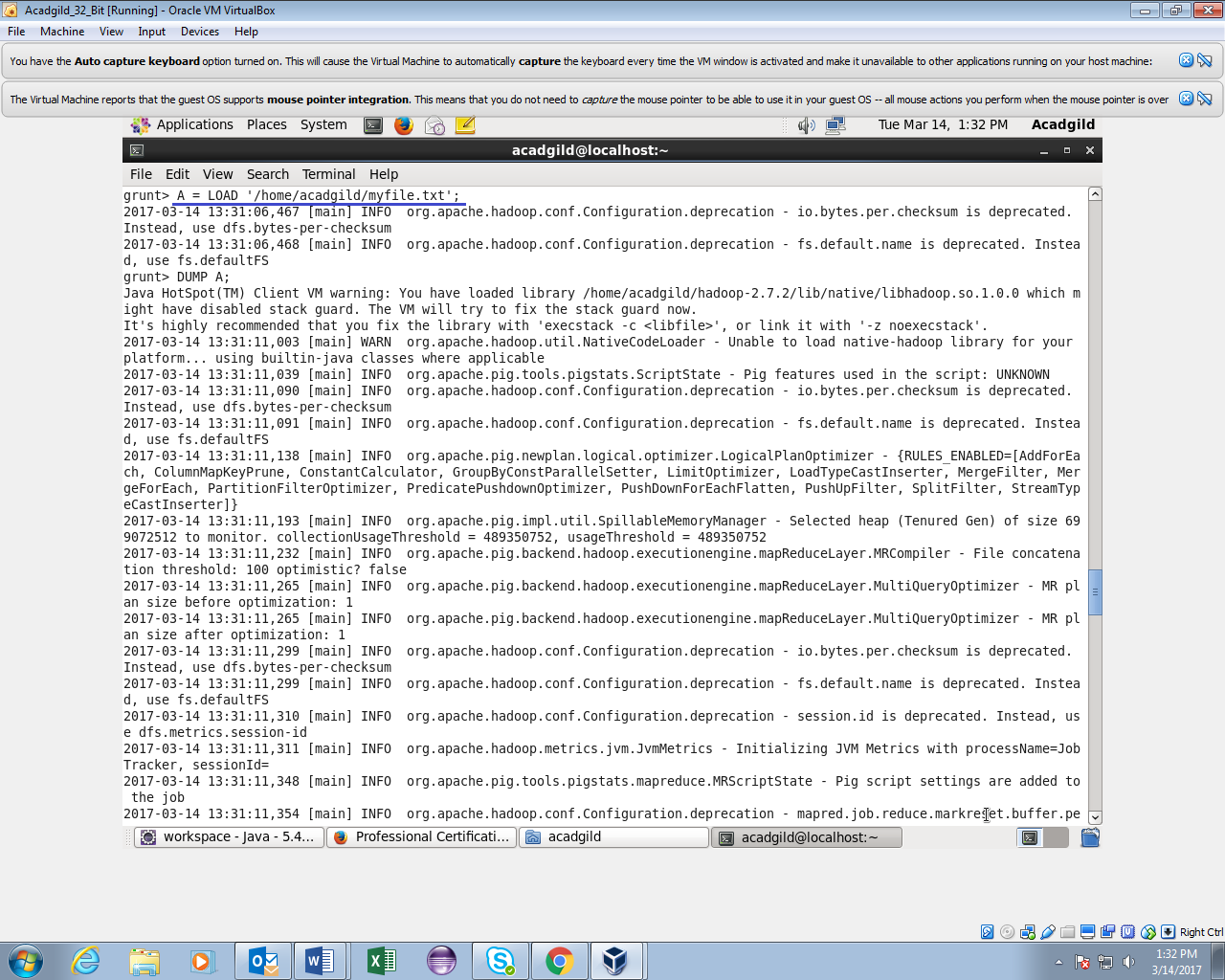


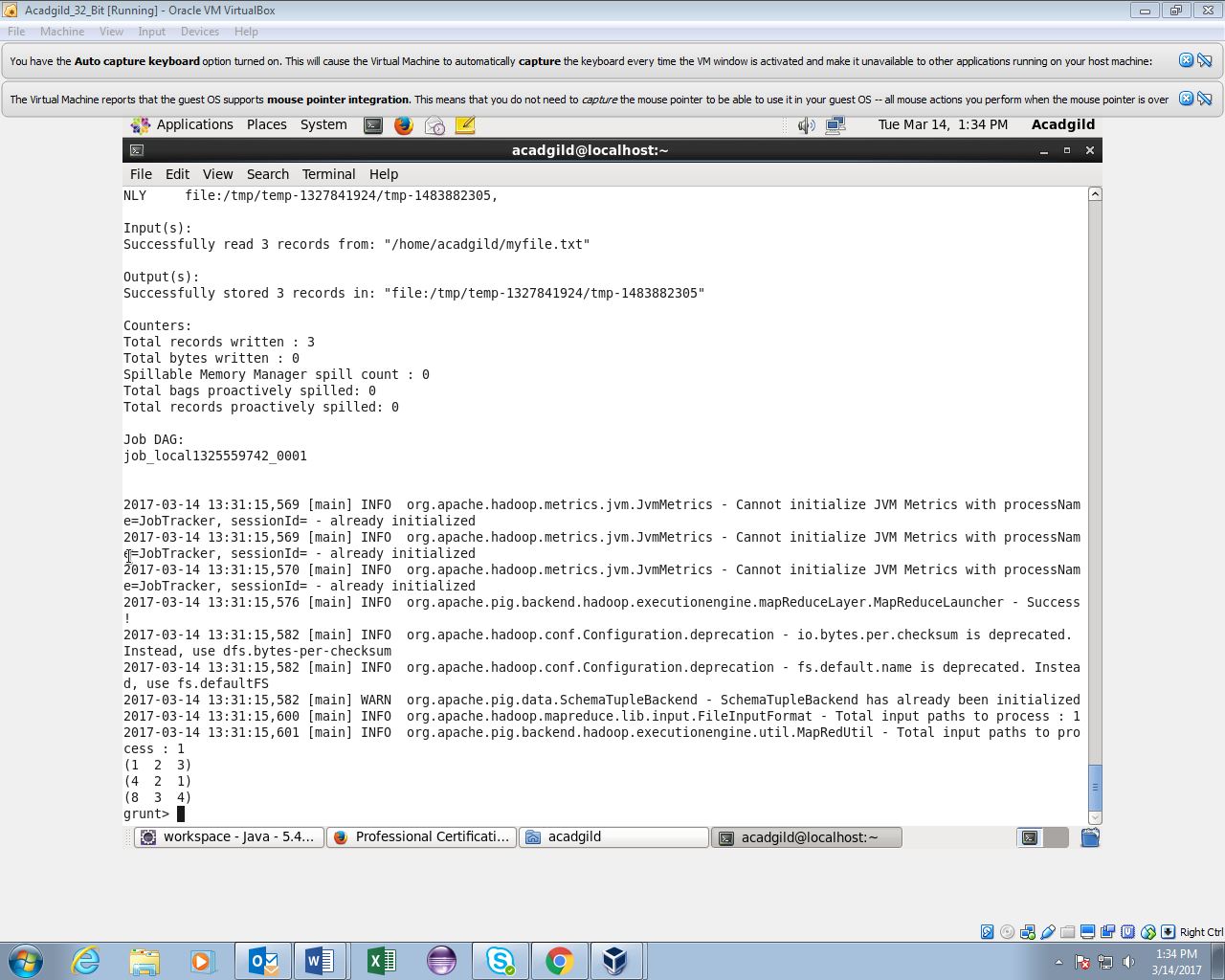


3. DUMP • Dumps or displays results to screen.

• Syntax • DUMP alias;

A = LOAD 'data' AS (a1:int,a2:int,a3:int); DUMP A; (1,2,3) (4,2,1) (8,3,4)





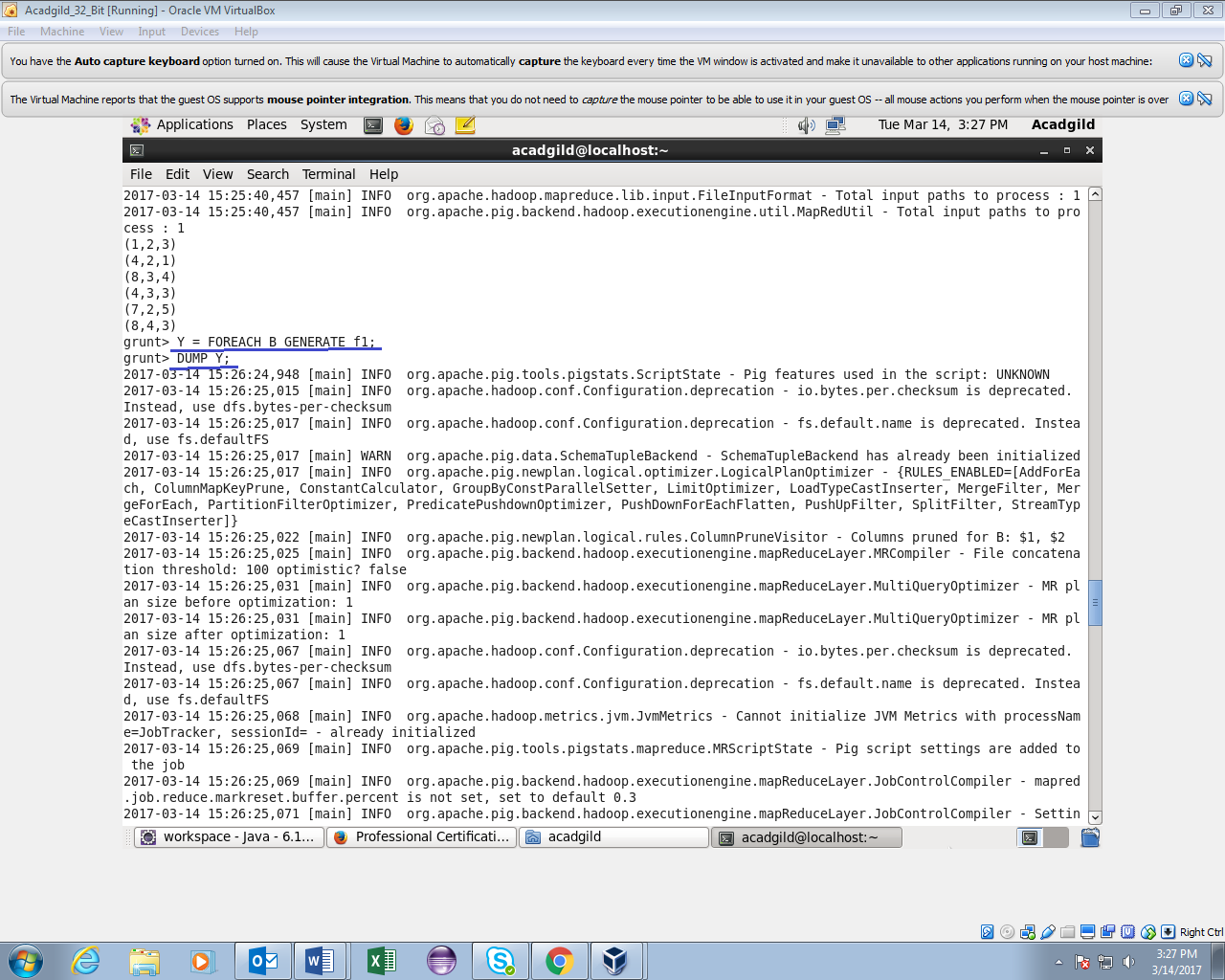
4.FOREACH

• Generates data transformations based on columns of data.

Syntax • alias = FOREACH generate\_operations [AS schema];

• X = FOREACH A GENERATE f1;





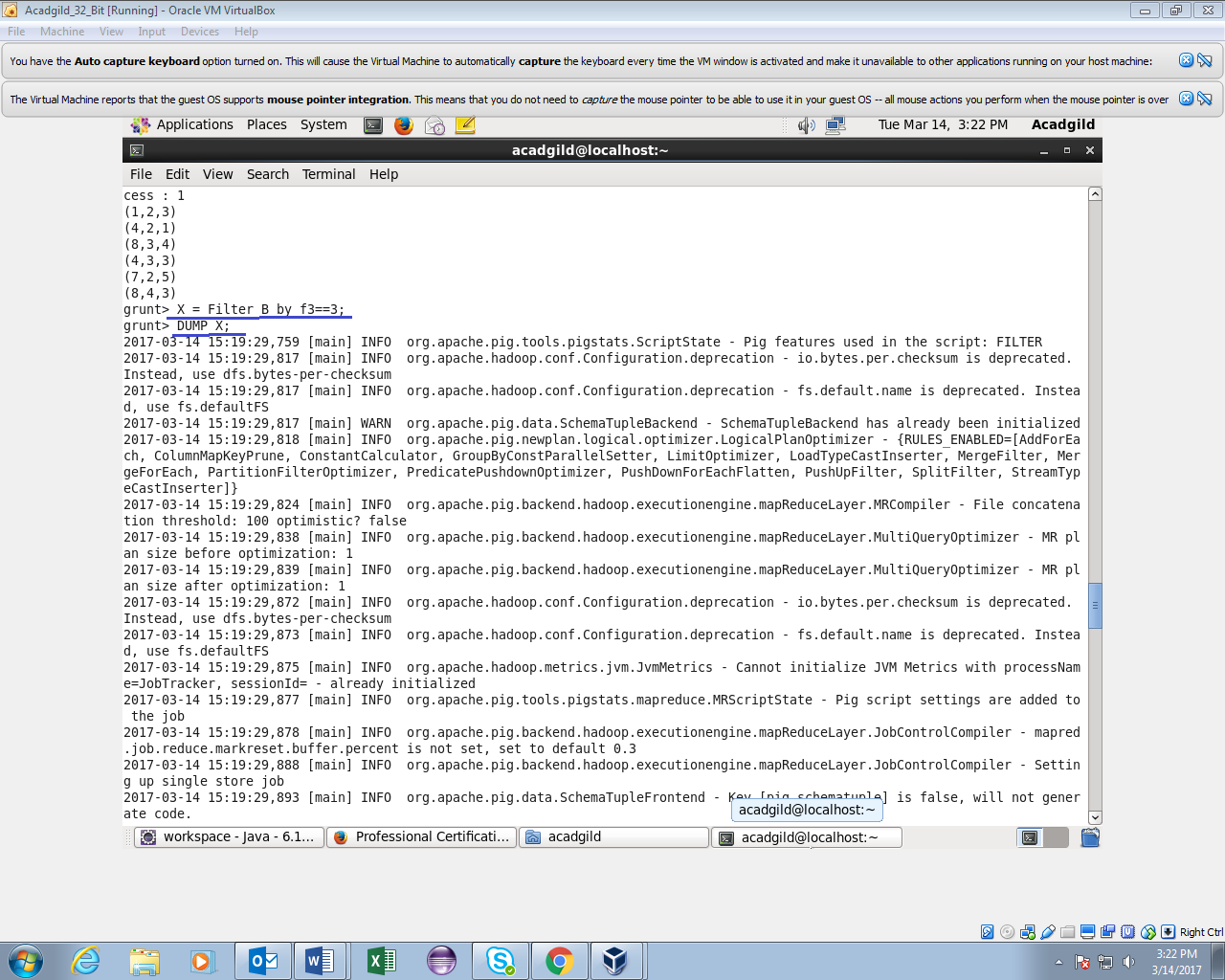
5.FILTER

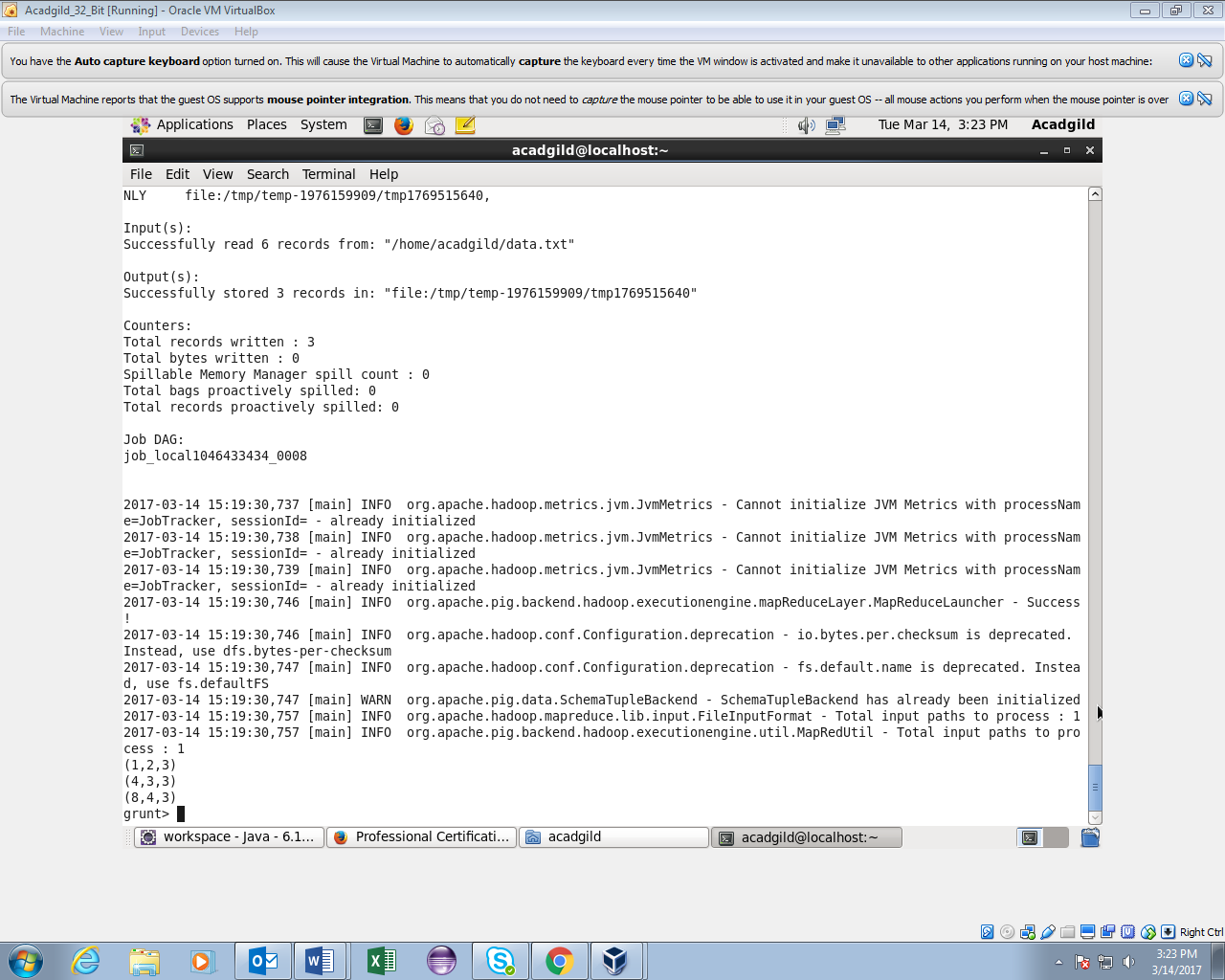
• Selects tuples from a relation based on some condition.

Syntax • alias = FILTER alias BY expression;

DUMP A; • (1,2,3) • (4,2,1) • (8,3,4) • (4,3,3) • (7,2,5) • (8,4,3) • In this example the condition states that if the third field equals 3, then include the tuple with relation X. X = FILTER A BY f3 == 3; • DUMP X; • (1,2,3) • (4,3,3) • (8,4,3)



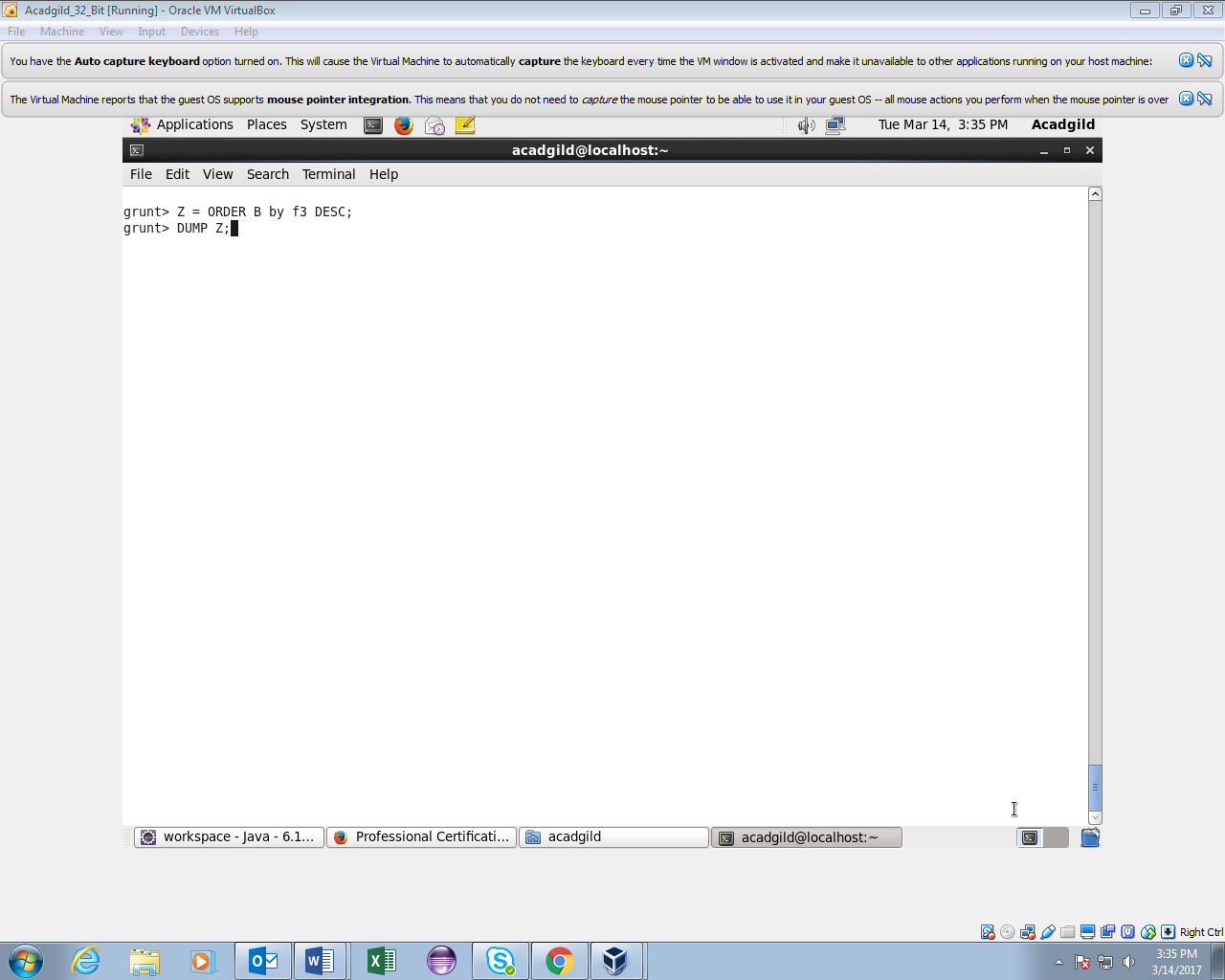


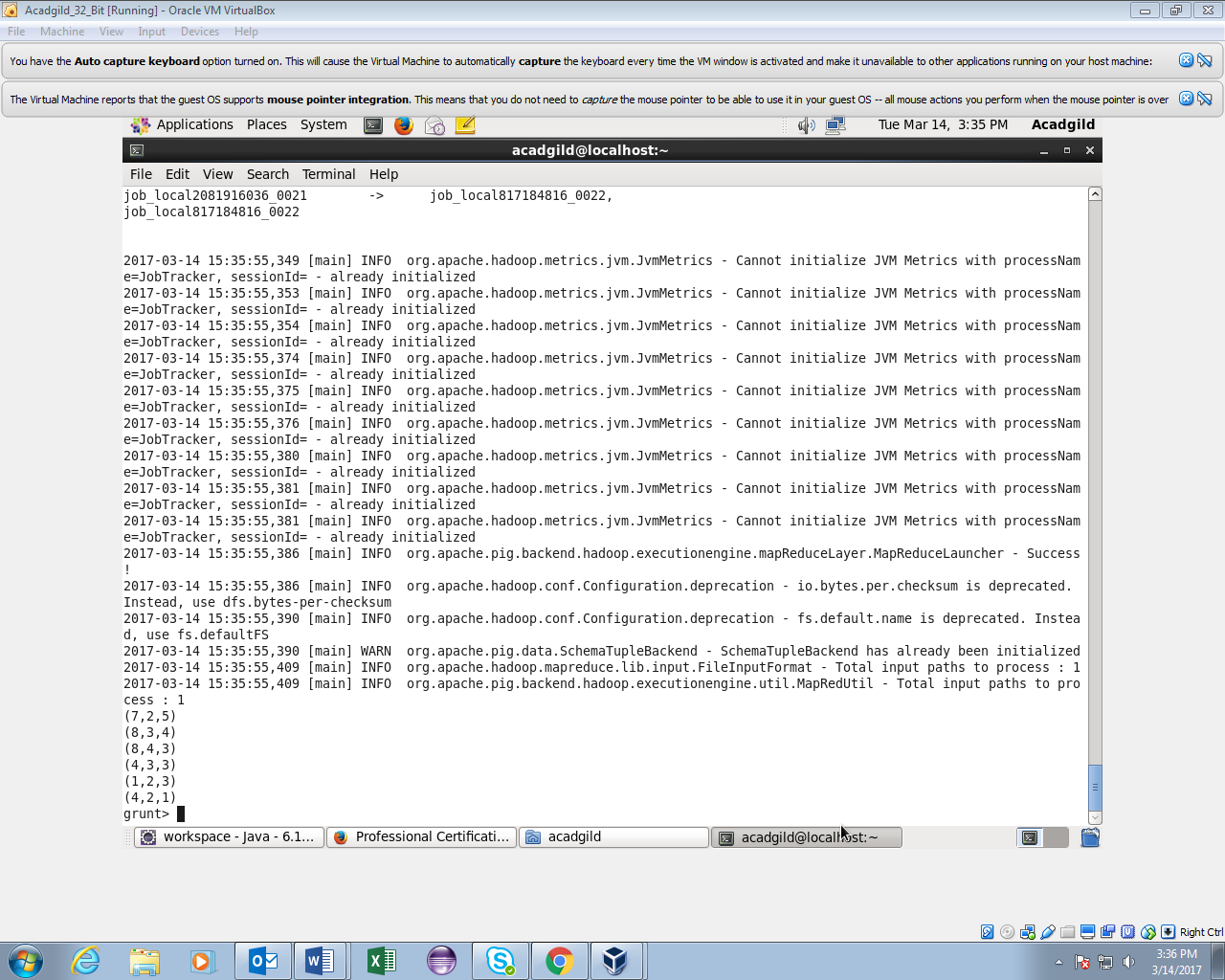


6.ORDER

• Sorts a relation based on one or more fields

Syntax • alias = ORDER alias BY { \* [ASC|DESC] | field\_alias [ASC|DESC] [, field\_alias [ASC|DESC] …] } [PARALLEL n] X = ORDER A BY a3 DESC; DUMP X; (7,2,5) (8,3,4) (1,2,3) (4,3,3) (8,4,3) (4,2,1)



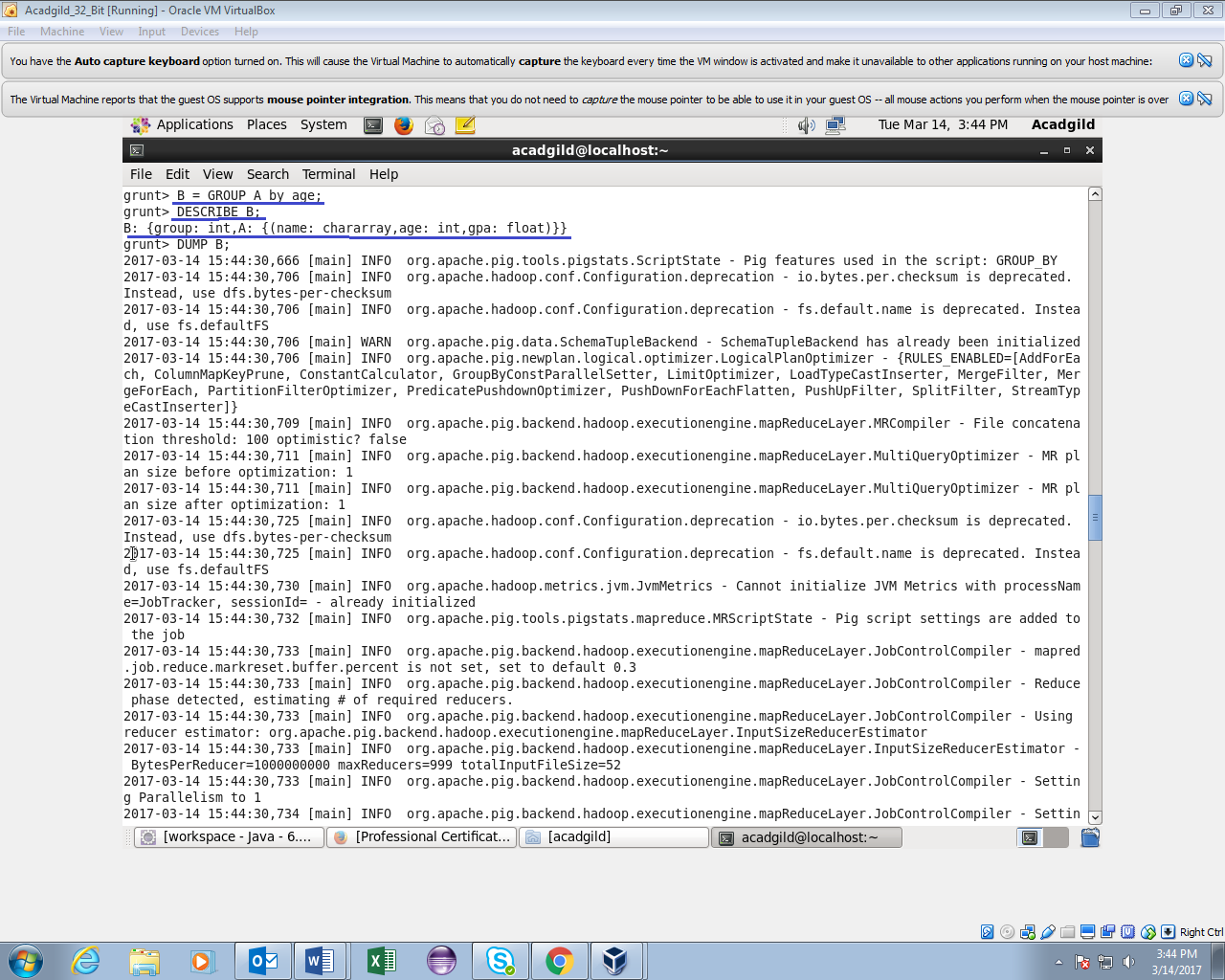


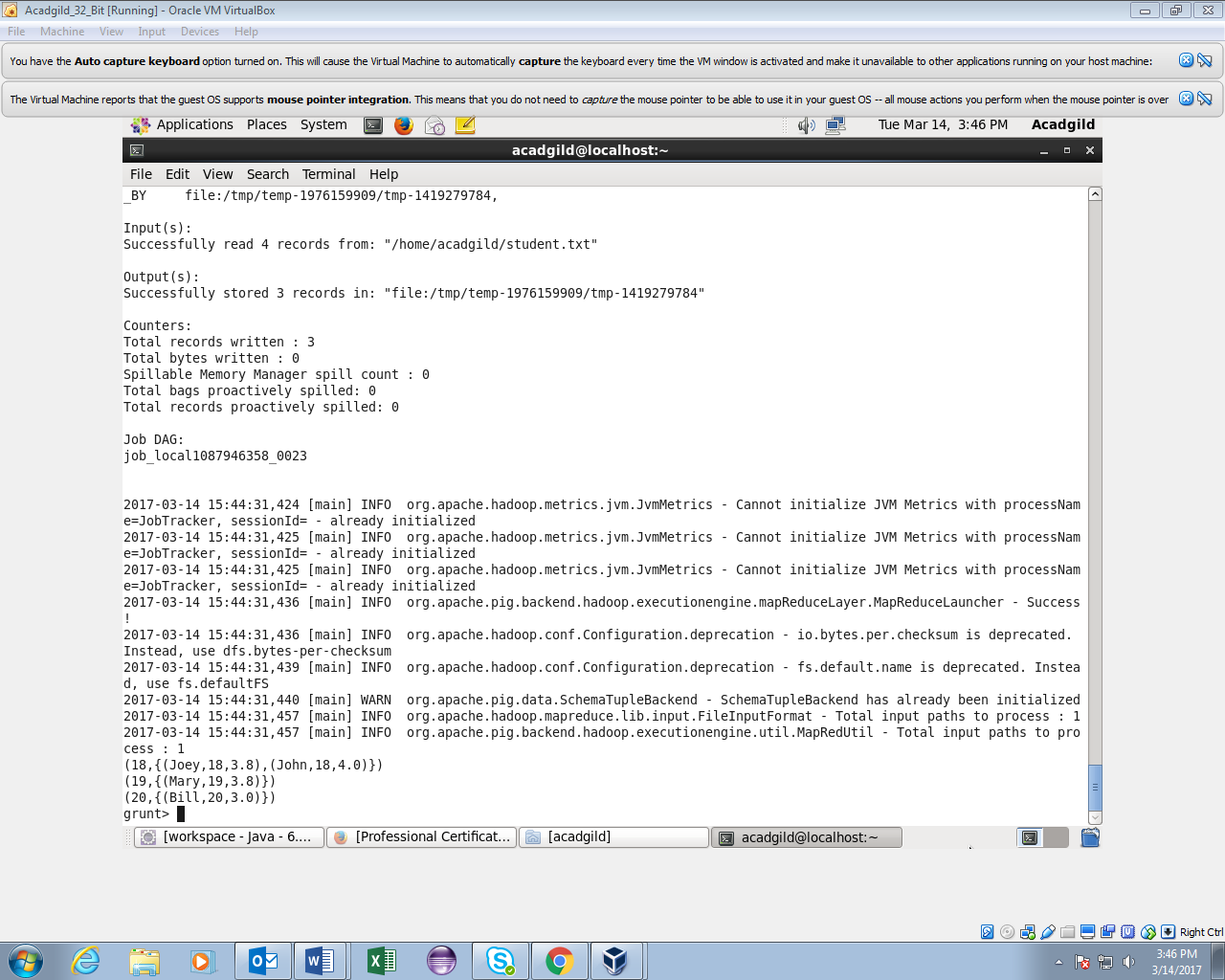
7.GROUP • Groups the data in one or multiple relations • alias = GROUP alias { ALL | BY expression} [, alias ALL | BY expression …] [PARALLEL n]; • The GROUP operator groups together tuples that have the same group key (key field) • The key field will be a tuple if the group key has more than one field, otherwise it will be the same type as that of the group key. The result of a GROUP operation is a relation that includes one tuple per group. This tuple contains two fields: • The first field is named "group" (do not confuse this with the GROUP operator) and is the same type as the group key • The second field takes the name of the original relation and is type bag • The names of both fields are generated by the system

DUMP A; (John,18,4.0F) (Mary,19,3.8F) (Bill,20,3.9F) (Joe,18,3.8F)

B = GROUP A BY age;

DUMP B; (18,{(John,18,4.0F),(Joe,18,3.8F)}) (19,{(Mary,19,3.8F)}) (20,{(Bill,20,3.9F)})





8. DESCRIBE

• Returns the schema of an alias.

• Syntax • DESCRIBE alias;

DESCRIBE A; A: {name: chararray,age: int,gpa: float} DUMP A; (John,18,4.0F) (Mary,19,3.8F) (Bill,20,3.9F) (Joe,18,3.8F)

