Pig Latin Parser

We ask you to write a language parser based on a high-level language called pig-latin (http://hadoop.apache.org/pig/docs/r0.3.0/piglatin.html). This language is currently used by hadoop clustered file system to simplify the creation of map-reduce application.

For the purpose of this interview, we simplify the language specification so that it can be solved in a short time.

The simplified language has several criteria, as follows:

- Each line, except when it's a blank line, should represent a complete statement. There will be no incomplete statement within a line.
- All keywords/identifiers are case insensitive.
- Assume that all the inputs are in the correct syntax. You don't need to check for syntax error.

The language has some reserved keywords:

Keyword	**Description**
LOAD	Load data from a file. File specification: ● The file has to be in comma delimited format. ● Every row has to have the same number of column. ● Each element are integers, ranging between -100000 to 100000 Example usage: a = load abc.csv
DUMP	Dump the content of a variable. Example usage: a = load abc.csv dump a Example output: 1,2,2 2,2,3 5,1,3 3,4,5
FILTER - BY	Select each row from a variable that matches the criteria. The criteria are defined by comparing a column identifier with a specific value. The column identifier is always started with $ followed by the column number. The supported operator is >, >=, ==, <=, <. Example usage:

	a = load abc.csv b = filter a by $0 > 2 DUMP b Example output: 5,1,3 3,4,5
FOREACH - GENERATE	Process each variable, and construct a new data from it. Example usage: a = load abc.csv b = foreach a generate $0 + 1, $1 * 2 DUMP b Example output: 2,4 3,4 6,2 4,8

Please write a python code to implement this parser, and if possible, write the unit-test for your code as well.

Other Example

Example CSV myfile.csv:

1,3,5,2,4

2,3,4,1,2

1,2,3,5,7

3,3,3,3,3

Example pig script:

A = load myfile.csv

Dump a

B = foreach a generate $0, $0 * 2, $1

Dump b

C = filter b by $2 < 3

Dump c

Example output:

1,3,5,2,4

2,3,4,1,2

1,2,3,5,7

3,3,3,3,3

1,2,3

2,4,3

1,2,2

3,6,3

1,2,2