Assignment 1 – Data Storage Essentials

Database Management Systems – COMP 3753 X1 [2018-2019]

Due: September 17, 2018 at midnight

The main purpose of this assignment is to refresh your memory on how to store, retrieve, and manipulate data in files. Secondarily, it rewards devising ways of doing this in an efficient manner, which anyone registered in this course does have the background to do.

This will be a programming assignment, and requires you to write a program. You may choose which language you prefer, from C, C++, Python, or Java.

The program will take one mandatory command-line argument, the name of a folder/directory which will be where a collection of data is stored. You get to choose the structure of data in that folder, *except* that the folder may not contain (directly or indirectly) more than 50 files.

When it runs, it will read commands from the user through standard input. The commands it must accept are:

ADD <name>; <email>; <year>; <country>
SHOW <attr> = <value>
SHOW <attr> > <value>
SHOW <attr> < <value>
DELETE <attr> = <value>
SET <attr> = <value> WHEN <attr2> = <value2>
QUIT

This input will be kept very simple. Each line in the input will contain one command (i.e. no command will be split over multiple lines). Each line will be at most 1000 characters in length. These will be very rigidly formatted such that where a space occurs above, only a single space is valid. For example, an add command will be formatted specifically as "ADD", followed by a space, followed by characters up to but not including a semi-colon (which will be the name), followed by a semi-colon, followed by a space, followed by characters up to but not including a semi-colon (which will be the email), followed by a semi-colon, followed by a space, followed by digits up to but not including a semi-colon (which will be the year), followed by a semi-colon, followed by a space, followed by digits up to but not including the end of line (which will be the country).

When the QUIT command is executed, it will exit the program, making sure any data is saved.

When the ADD command is executed, it will add a record to the collection. The record's values will be the four items read (the name, email, year, and country).

When the DELETE command is executed, it will delete *all* records from the collection where the attribute matches the value specified.

When the SET command is executed, it will find all records in the collection whose attribute (attr2) matches the given value (value2), and sets those records' attribute (attr) to the value give (value).

When the SHOW command is executed, it will find any records in the collection where the given attribute is equal to, less than, or greater than (depending on the operator used) the value provided,

and will print their results to the standard output. The results will be printed such that each record found is printed on a separate line. Each line of results will contain the name, followed by a semi-colon, followed by the email, followed by a semi-colon, followed by the year, followed by a semi-colon, followed by the country. The rows output from the SHOW command may occur in any order (it doesn't require them to be sorted).

An example input:

```
ADD Duane Currie; duane.currie@acadiau.ca; 1976; Canada
ADD Tux; tux_rules@penguin.net; 2000; Antarctica
SHOW name = Tux
SET year = 1996 WHEN name = Tux
SHOW name = Tux
ADD Darcy; darcy.benoit@acadiau.ca; 1972; Canada
SHOW year < 1990
DELETE name = Duane Currie
SHOW year < 1990
QUIT
```

Would produce the following output (rows 3 and 4 might be in a different order):

```
Tux;tux_rules@penguin.net;2000;Antarctica
Tux;tux_rules@penguin.net;1996;Antarctica
Duane Currie;duane.currie@acadiau.ca;1976;Canada
Darcy;darcy.benoit@acadiau.ca;1972;Canada
Darcy;darcy.benoit@acadiau.ca;1972;Canada
```

Note:

- Your program will be tested with a memory limit of 512M, but large test cases will be used. That means your program must be able to handle more data than can fit in memory all at once.
- If the end of input (end of file) is found, it should do the same as quitting the program with the QUIT command.
- If the program is terminated abnormally (through a CTRL+C), data must still be preserved. Thus, if the program terminates that way, and then is run again, any data changes must not be lost.

I will be testing this program on a very large test suite. This will help me test for correctness. The assignment mark will be based upon correctness of the program, and upon the quality of the code submitted (sensible code structure, readability, and appropriate error/exception maanagement).

Also, the submissions will be divided into groups based upon which language was used for implementation. C and C++ submissions will be in the same group, while Python and Java will each form their own group. Each program will be run first on an initialization script (a file containing a large number of ADD commands), and then on a test script (a large set of commands, where about 90% are SHOW commands, 3% are ADD, 4% are SET, and 3% are DELETE, and it will end with a QUIT command).

The correctly working submission in each group with the shortest runtime on the test script will receive 3 bonus points on the term.

Assignment submission:

Submit the assignment in the form of a zip file in the Assignment drop-off box on Acorn. The zip file must contain all source files, and a file called README.txt. The README.txt file should identify the commands required to build your program, and how to run your program.

Suggestion:

Get the main part of the assignment done first, and make a copy of it before trying to apply speed optimizations for the potential bonus marks. That way, in case you change anything that breaks your program, you still have a working copy ready for submission. Alternatively, if you know how to use a version control system (e.g. git), that's even more powerful.