CS2102 Database Systems

Semester 1 2019/2020

Assignment 01

DEADLINE: 31 August 2019 **1 MARK**

1 Introduction

A database management system (DBMS) is a specialized piece of software for managing data. Before DBMSs were invented, applications manage data using what is known as *file processing techniques*, where the data is stored on the operating system file system and is processed by application programs written in some programming language. The following illustrates the typical processing steps on a single data file.

```
initialize some book-keeping information I
open data file F
while (F is not empty)
  read next record r from F
    if (r satisfies some condition) then
      do something with r
    update I if necessary
do something with I if necessary
close file F
```

The objective of this assignment is to let you get the sense of what it takes to manage data without using a DBMS.

2 Database

In this assignment, we have a very simple database that consists of a single comma-separated values (CSV) file that records the value of cryptocurrency in a marketplace from December 2013 to November 2017. The file contains a total of 7688 records, where each record consists of the following sequence of 10 columns:

1. Year
2. Month
3. Day
4. Currency (e.g., BTC for bitcoin, LTC for litecoin, ETH for ethereum, etc)
5. Open (value during opening)
6. High (highest value during the day)
7. Low (lowest value during the day)
8. Close (value during closing)
9. Volume
10. Market cap

Your code should work for ***any data file with the same format***.

3 What to do?

You are to attempt ***AT LEAST ONE*** of the following four questions. Each question requires writing a program (using your favorite programming and/or scripting language) to perform a data management task.

Quetion 1. Find all `(Open, High, Low, Close)` records with `Currency = 'BTC'`, `Month = 'Jan'`, and `Year = 2016`. Output these records in *ascending order of* `Day` to a CSV file named `q1.csv`.

Quetion 2. Compute the following statistics for each `Currency`: (1) the highest `Volume` transacted in a day, (2) the average `Market cap` value, (3) the number of transactions, and (4) the lowest ***middle*** value where we define ***middle*** as `(Open + Close) / 2`. All average and ***middle*** should be rounded to the nearest integer values. Output each currency and its four statistics as a single record to a CSV file named `q2.csv`. The records are to be sorted in descending order of `Currency` followed ***middle*** values.

Quetion 3. We say that a given `Month` and `Currency` is a ***bearish*** or ***bullish*** month as follows. The month m is ***bullish*** for the given currency c if the average of the `Market cap` on a given month m is larger than the previous month $m-1$ (*if any*) and one of the following condition is satisfied:

a) The `Open` value on the first day of the month m is smaller than the `Close` value of the last day of the month m;
b) The `High` value of the first day of the month m is smaller than the `High` value of the last day of the month m; or
c) The `Low` value of the first day of the month m is smaller than the `Low` value of the last day of the month m.

All months that are not classified as ***bullish*** is considered ***bearish***. Find all the ***bearish*** `Month` and `Year` for `Currency` named `'XPR'` and output them to a CSV file named `q3.csv` sorted in ascending order of `Year` followed by `Month`.

Quetion 4. To reflect inflation, you are to update all records on `Year 2017` by increasing the `Market cap` and `Volume` by 10%, all records on `Year 2016` by increasing the `Market cap` and `Volume` by 15%, all records on `Year 2015` by increasing the `Market cap` and `Volume` by 20%, all records on `Year 2014` by increasing `Market cap` and `Volume` by 25%, and all records on `Year 2013` by increasing `Market cap` and `Volume` by 30%. You are to round all values to the nearest integer values. You are to *rewrite* `crypto.csv` file.

4 How to start?

Download the file `cs2102_assign01.zip` from ***Assignment*** folder in Luminus. Unzip the file and it should contain the following files:

- A CSV database file called crypto.csv
- A directory output-files/ which contains an output CSV file for each of the questions. You may use these files to compare against your program outputs.

You may use any programming language you want, but make sure it's the "**vanilla**" version. That means, if you are using Python, no `panda` or `numpy`. All major programming languages have file reading capability in their library (e.g., C File I/O[1], Java FileReader[2], JavaScript FileReader[3], or Python read[4]).

5 How to submit?

Submit your source code for each of the attempted questions. Name each source file using an appropriate question number (e.g., `q1.java`, `q2.py`, etc). Do not submit any CSV files.

Zip the file and name the file with your user ID *that starts with e* (e.g., `e0123456.zip`). Upload the zip file containing all your source files to ***Assignment 01*** folder in Luminus. The deadline for this assignment is *31 August 2019*.

In your code, you should read from `crypto.csv` file from the same folder as your code. You should not use absolute path to read `crypto.csv`. Instead, you should use relative path. For instance, you should not use "`C:\Users\<username>\CS2102\Assignment 01\crypto.csv`" or wherever your file is located when you read the CSV file.

6 Grading

You will be graded by your best score. If you managed to do any one of the questions above correctly, you will be given the mark. The maximum mark for this assignment is **1 mark**. As such, you just need to do one question well to score the mark. We will test your code with other CSV files, following the same format, but still named as `crypto.csv`.

[1] https://www.tutorialspoint.com/cprogramming/c_file_io.htm
[2] https://docs.oracle.com/javase/7/docs/api/java/io/FileReader.html
[3] https://developer.mozilla.org/en-US/docs/Web/API/FileReader
[4] https://www.tutorialspoint.com/python/file_read.htm