Concordia University
Dept. of Computer Science and Software Engineering
COMP 6521: Advanced Database Technology and Applications
Winter 2019-2020

Lab Assignment 2

Total Points: 10
Codes & Reports: Wed. April 8th at 12:00 noon
Demos: Wed. April 8th during the lab hours

Project Description: Consider again the data files T1 and T2 defined in Lab Assignment 1, with the following record structure:

EmpID INT (8)
LastUpdate CHAR (10)
EmName CHAR(25)
Gender INT(1)
Dept. INT(3)
Social Insurance Number INT(9)
Address CHAR(43)

Date is a string of length 10 characters of the form 'YYYY-MM-DD'. Each of these two files is stored as a text file, and each tuple is of size 100 bytes, stored as a separate line/record/row in the file. Each line is ended with the "return" key. Suppose each block is of size 4K, holding 40 tuples. The rest of the block is left unused. Furthermore, the blocks of each file are stored in consecutive disk blocks. There is not delimiter separating the values of different fields/attribute. An example of a tuple in these files is as follows (but would be on a single line):

```
123456782020-01-24John                     12223333333331455 de
Maisonneuve West, Montreal, QC, H3G 1M8, Canada
```

Each of these files could contain tuples with duplicated Employee IDs. Recall that in LA1, tuples t1 and t2 are considered as "duplicates" if they had the same EmpID's. A tuple in T1 and/or T2 could have any number of duplicates. Assume the positions of tuples in T1 and T2 are fixed. In LA2, you and your team are required to create and use bit map indexes on attributes EmpID, Dept and Gender.

(1) Report the time required to create the bit map indexes (3 on T1 and 3 on T2).
(2) Report the actual and compressed sizes of these bit map indexes.
(3) Use the actual (uncompressed) indexes to find the tuples that are "duplicated" in T1 and/or T2.
(4) Use the information in part 3 to merge T1 and T2, and as you did in LA1, for each EmpID, return the most update record, and discard the previous ones, if any.
(5) Comment on the solution using bit map indexes with using two phase,

multiway merge-sort method used in LA1. For the comparison, consider the total number of disk I/O’s, and the processing time. As in LA1, ignore the number of disk I/O’s and the time required to write the result back to the disk. The amount of main memory available is 10MB (so you do not need to consider and compare the results when MM=20 MB).

Use instances of T1 and T2 from LA1. Your report should include the test results, at least on these instances. You may include additional test results using the instances created by your team.

To further evaluate the performance of your implementation, consider instances of T1 and T2 with about 500,000 and 1,000,000 tuples, respectively.

What tools you should use?

Use VM argument Xmx5m in Eclipse to restrict the main memory size to 10 MB in Java Virtual Machine.

What to submit by the due date?

Through Moodle, submit your report (in a single PDF format) and the source codes (a single zip file). Also please include instructions to compile and run your code. Make sure your program compiles and runs on the computers in the labs H-903 and H-907 assigned to the course.

Demos: Book a time slot as soon as we post the demo schedule in early April. Each member of a team must be present in their project demo.

Bonus: The lab instructors may recommend additional 2 points
for an implementation which has the best performance and
additional 1 point for the next best.