

University of Toronto Mississauga
CSC 343 Winter 2018
Group Assignment 3
Due: Monday April 2nd, 2018 by 9:00am

Michael Liut

March 14, 2018

I. Indexes (10 marks)

Question 1 (10 marks)

For the workload given in Assignment 2, Part I, several queries have been showing poor performance (i.e., increasing response times). Your task is to improve the performance of this workload as much as possible by recommending two indexes that should be defined on the tables. What two indexes would you recommend?

For each index, state:

- The attribute(s) the index is defined on.
- Classification of the index.
- Which queries (q1 – q12) you think this index will help, and one sentence justifying why.

II. Database Design (40 marks)

Question 2 (22 marks)

Consider a relation schema R with attributes $\varphi = \{A, B, C, D, E, F, G, H\}$ and the set of functional dependencies $\lambda = \{DF \rightarrow AC, AEH \rightarrow BD, C \rightarrow AFH, C \rightarrow ACG, E \rightarrow F, BC \rightarrow EH, A \rightarrow B, DE \rightarrow HB, ABD \rightarrow FGH, H \rightarrow EA\}$

- (a) [7 marks] Find all candidate keys (i.e., minimal keys) of relation R . Show **all** the steps you took to derive each key, and clearly state which of Armstrongs axioms are used in each step.
- (b) [15 marks] Find a *minimal cover* (also known as a minimal basis, or a canonical cover). Give your FDs in alphabetical order. Provide a 3NF decomposition of relation R , and indicate the key(s) for each relation. You must include **all** steps in your solution. (Refer to Section 8.4.3 of the textbook for assistance on how to compute a minimal cover.)

Question 3 (18 marks)

Consider relation $R(A,B,C,D,E,F)$. Now, consider the decomposition of R into $R_1(ABCD)$ and $R_2(DEF)$.

- (a) [4 marks] Give a set of functional dependencies (FDs) such that the decomposition into R_1 and R_2 is lossless join and dependency preserving. Justify why your FDs satisfy the criteria.
- (b) [4 marks] Give a set of functional dependencies such that the decomposition into R_1 and R_2 is lossless join, but not dependency preserving. Justify why your FDs satisfy the criteria.
- (c) [4 marks] Give a set of functional dependencies such that the decomposition into R_1 and R_2 is not lossless join, but dependency preserving. Justify why your FDs satisfy the criteria.
- (d) [6 marks] Consider a relation $S(A,B,C,D,E,F)$, and functional dependency $F : A \rightarrow BCDEF$, that holds over S . Define two other functional dependencies F_1 and F_2 , which satisfy the following three properties:
 - i. Neither F_1 nor F_2 can be inferred from F using Armstrongs axioms.
 - ii. Relation S with functional dependencies F and F_1 is in BCNF.
 - iii. Relation S with functional dependencies F , F_1 , and F_2 is in 3NF but not in BCNF.

State your two FDs F_1 and F_2 , and justify how your FDs satisfy the above properties.

III. Transactions and Concurrency (20 marks)

Question 4 (10 marks)

- (a) [5 marks] Give an example of a transaction schedule that is conflict-serializable, but which is not possible using 2PL. In other words, the schedule should be conflict-serializable, but there should be no way to acquire and release read/write locks in a way that is consistent with the 2PL protocol.
- (b) [5 marks] Is every schedule that is allowed under the Strict 2PL locking protocol also allowed under the 2PL locking protocol? Or vice versa? Give an example schedule that is allowed in one of the two locking protocols, but not in the other locking protocol.

Question 5 (10 marks)

Consider the following classes of schedules: *serializable*, *conflict-serializable*, *view-serializable*, *recoverable*, *avoids-cascading-aborts*, and *strict*. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs in a certain class based on the listed actions, explain briefly.

The actions are listed in the order they are scheduled and prefixed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.

1. T1:W(A), T2:R(B), T1:R(B), T2:R(A)
 2. T1:R(A), T2:W(A), T1:W(A), T2:Abort, T1:Commit
 3. T1:W(A), T2:R(A), T1:W(A), T2:Abort, T1:Commit
 4. T2:R(A), T3:W(A), T3:Commit, T1:W(B), T1:Commit, T2:R(B), T2:W(C), T2:Commit
 5. T1:R(A), T2:W(A), T2:Commit, T1:W(A), T1:Commit, T3:R(A), T3:Commit
-

Grading

This is a group assignment to be completed in pairs (i.e. a team of 2 people). You will work with the same group as in Assignment 1 and 2. This assignment is worth 13.3% of your final grade in this course.

Submission

All files are to be submitted using the Blackboard platform (portal.utoronto.ca). Only one person from each group is required to submit the files. Please ensure your answers are typed and submissions are clearly legible. Include your and your partner's full name and student ID number.

Upload one files with the indicated file extensions (no compression based .tar, .zip, .rar files). Submit your solutions to all questions in a file called **asg3.pdf**.

Please note that late assignments will be docked 20% per day of lateness and after four (4) days, the assignment will no longer be accepted.

Plagiarism

Please refer to the course outline and introduction slides. To serve as a reminder: Turnitin will be used for all written work and MOSS for all code submissions. UTM's policy on Academic Integrity: <http://academicintegrity.utoronto.ca/>