University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Spring 2016

GENERAL INSTRUCTIONS

Answer each question in a separate book.

Indicate on the cover of each book the area of the exam, your code number, and the question answered in that book. On one of your books list the numbers of all the questions answered. Return all answer books in the folder provided. Additional answer books are available if needed.

Do not write your name on any answer book.

SPECIFIC INSTRUCTIONS

Answer all five (5) questions. Before beginning to answer a question make sure that you read it carefully. If you are confused about what the question means, state any assumptions that you have made in formulating your answer. Good luck!

The grade you will receive for each question will depend on both the correctness of your answer and the quality of the writing of your answer.

Policy on misprints and ambiguities:

The Exam Committee tries to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the first hour of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is nontrivial.

1. ARIES.

During the REDO pass, when we are deciding whether to redo the operation in a log record *l* for a page *p*, the claim is that we don't even have to check the pageLSN for *p* if *p* is not in the Dirty Page Table or if the page's recoveryLSN in the DPT is greater than the LSN of *l*.

a) Why is this true?

b) How is it even possible that we are considering an update log record for a page *p* but *p* is not in the DPT?

c) Consider the three points in the log:
   i. the beginning of analysis phase,
   ii. the beginning of the REDO phase,
   iii. the earliest log record considered by the UNDO phase.

Is it possible that they could appear, from earliest to latest, in the order (i), (ii), (iii)? Explain your answer.

2. Are DBMS Really Necessary?

Many people who deal with a lot of data do not use database systems. Even scientists whose experiments produce vast quantities of data and analysts exploring huge amounts of customer data often avoid the use of database systems for storing their data, relying instead on file system files for the task. This trend seems to be getting worse, with the advent of NoSQL systems.

This question asks you to give your insights into this issue.

a. What database system factors do you believe most limit the use of database systems for data management?

b. Do you believe the inclusion of Map-Reduce as an interface to DBMS (that is, allowing users to use generic Map-Reduce programs to access DBMS-resident data) will significantly increase the use of database technology? Why or why not?

c. One could argue that NoSQL systems and relational systems are becoming more similar over time, that is, they are adopting more and more of each other's features. Do you suspect that we will always have both kinds of systems, or will they converge to a single, unified system? Explain your answer.

3. Join Algorithms

In Shapiro's paper on join processing algorithms, he does not distinguish between sequential and random I/O. (Here by "sequential I/O" we mean that consecutive reads/writes go to adjacent disk pages, whereas by "random I/O" we mean consecutive reads/writes that likely go to different parts of the disk.) Assume for simplicity that the unit of reading/ writing is a single page (that is, reading or writing a run of *k* pages will take *k* I/Os.)

a. For the hybrid hash and sort-merge join algorithms give approximate formulas for the number of sequential and random I/Os for each algorithm. Break down this cost by each phase of the algorithms, assuming that you have only one disk available. (Assume that the join is of relations R and S, and that both R and S are smaller than M^2, where M is the number of pages of memory.)

b. Now assume that you have two disks. How would you use them in the join algorithms, and what are the new formulas for random and sequential I/Os?

4. Data Mining

a. Define "frequent itemsets" and describe their role in identifying association rules. Describe the Apriori algorithm for computing frequent itemsets.

b. One of the strengths of a relational DBMS is that operations can be composed to write a rich variety of queries. How would you extend SQL to support the creation and manipulation of association rules? (Concentrate on how you would represent and manipulate the input and output, not how the underlying algorithms are implemented.)

5. Datalog and conjunctive queries:

Consider the following three conjunctive queries. For each pair of queries *q* and *q'*, state if *q* is contained in *q'*, q' is contained in *q*, or they are incommensurate (neither is contained in the other.)

```
  i.   q1(X,Y,Z) :- a(X,W), b(Z,Y), c(X,Z).
 ii.   q2(X,Y,Z) :- a(X,W), b(W,Y), c(X,Z).
```

```
iii.   q3(X,Y,Z) :- a(X,Y), b(U,Y), c(X,Z).
```

b) Does the following recursive Datalog program have an equivalent conjunctive query? If your answer is yes, give an equivalent query; if your answer is no, argue (informally is fine) why it does not.

```
t(X,Y) :- e(X,W), t(Z,Y).
t(X,Y) :- e(X,Y).
```