University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Spring 2005

Answer all five (5) questions (NOTE: this is different from some previous years, when you were only asked to answer 4 of 5.) 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 both of the correctness of your answer and the quality of the writing of your answer.

1. Database Design

Consider a table R(X, Y, Info).

(a) For which workloads would you choose:

(i) two single column hash indexes, one on R.X, one on R.Y;

(ii) a single two-column hash index on the combined key (R.X,R.Y);

(iii) single column B+ tree indices on R.X and R.Y;

(iv) a two-column B+ tree index on the combined key (R.X,R.Y);

(v) a grid-file using (X,Y) as the key.

(b) Suppose you are told that the FD X → Y holds. How will this affect your answers in part (a)?

2. Concurrency Control

For each of the following locking protocols, assuming that every transaction follows that locking protocol, state which of these desirable properties are ensured: *serializability, conflict-serializability, recoverability, avoidance of cascading aborts*.

Protocol (1): Always obtain an exclusive lock before writing; hold exclusive locks until end-of-transaction. No shared locks are ever obtained.

Protocol (2): In addition to (1), obtain a shared lock before reading. Shared locks can be released at any time.

Protocol (3): As in (2), and in addition, locking is two-phase.

Protocol (4): As in (2), and in addition, all locks held until end-of-transaction.

3. Main Memory Database Systems

When compared to the memory sizes for which current RDBMS were architected, today's machines have enormous memories. Some researchers (and even some startup companies) think that now it is reasonable to assume that most of the relevant data fits in main memory, and that disk accesses are rather rare.

Joe RAMbo has an idea for exploiting memory more efficiently. He feels that the slotted page architecture, while good for disk-resident data, makes no sense if the data is in memory. So he proposes the following. On disk, he will use slotted pages, just like current practice. But when a page is brought into memory, he will copy all the tuples off the page and insert them into a hash table, then discard the original slotted page from the memory. All the tuples for relation R will go in a hash table for relation R; the hash key is the record id.

Your job in this question is to try to analyze what Joe is proposing. What if anything will his approach improve? What will it make worse? *Hint:* Your answer should probably discuss some of these issues—concurrency control, recovery, query processing, and buffer management.

You will be graded both on the technical quality of your answer and the quality of your exposition.

4. Storage Trends and Versioning

Disks are becoming so large and so cheap that very soon there will be no need to ever physically remove any deleted tuples or to update existing tuples in place.

(a) Why might we want to keep old versions of tuples as well as deleted tuples?

(b) Suggest how SQL might be changed to allow users to take full advantage of this capability.

(c) How would not deleting tuples and not doing updates in place affect the way database systems are architected in terms of how tuples are stored on disk and how indices are constructed? Discuss the impact of your proposed approach with respect to query processing, optimization, concurrency control, and recovery.

5. Web Search Engines

(a) How is a document represented in the vector-space model?

(b) What is TF*IDF? How is it related to the vector space model?

(c) Given a document d and a collection of documents D, how would you rank the documents in D by similarity to d using the vector space model?

(d) Google uses a measure called page rank to order search results. Explain what page rank is and describe the underlying intuition. Also, explain the relative roles of page rank and traditional vector-space ranking in Google.

(e) Search engines must scale to billions of documents and millions of queries per day. Describe how a typical search engine can be parallelized to achieve these goals, using a cluster of commodity computers (e.g., PCs running Linux).