University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Fall 1999

Answer any four (4) of the following 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!

1. Object-relational database systems.

Object-relational database systems extend the relational data model in a number of ways including set-valued attributes, inheritance hierarchies for row tables, table hierarchies for tables and type extensibility through user defined data types and methods. In this question, consider the issue of type extensibility in which users or database administrators may customize their database system by adding new types and methods/functions on those types. Extended types and methods can include basically any type that can be coded in a programming language. Instances of these types can be large (e.g. images) and the associated functions can be time consuming to execute (and possibly buggy).

a) Discuss how type extensibility effects query optimization.

b) Since user-defined functions may be buggy, discuss alternatives for preventing such functions from corrupting/crashing the database server.

c) Generally, an object-relational DBMS will not store such instances "in-line" with the other attributes of the tuple. Instead, the attribute is stored as a separate object in the database with the tuple containing a pointer to the object. Discuss how the existence of such external attribute values complicates the processing of queries in a parallel object-relational database system. Focus on the case where the attribute in question is not processed as the first operator in the query tree. Propose and discuss two alternative strategies for dealing with this situation.

2. Hybrid Hash Join

In his paper "Join Processing in Database Systems with Large Main Memories", L. Shapiro describes the simple hash, GRACE hash and hybrid hash join algorithms. Given that we want to join two relations R and S, where |R| = 100 and |S| = 1000, compare the respective maximum memory requirements of each of the following algorithms:

(a) (i) Page level nested loops join.
(ii) Sort merge join.
(iii) Grace hash join.
(iv) Hybrid hash join.

(b) Both GRACE hash and hybrid hash join partition the input relations using a hash function such that each partition of the build relation fits in memory. Can you think of situations where this might not work very well?

(c) How would you rectify the problems you mentioned in (b)?

(d) By looking at the cost formula of hybrid hash join as given in Shapiro's paper, how would the execution time of a hybrid hash join (joining the above mentioned R and S) vary as we increase the amount of memory available to the join from 10 pages to 1000 pages? (A good way to answer this is to draw a graph of the performance, with the x-axis labeled by the number of pages, and the y-axis some unspecified time unit.)

(e) Suppose we want to evaluate the expression (R JOIN S) JOIN T using two hybrid hash joins working concurrently in a pipeline. |R| = 100, |S| = 1000, |T| = 600, |R JOIN S| = 500. Suppose we have only 80 pages of memory available to us. How would you allocate this memory to the two joins (assuming that each join should not make more than two passes over the `data`) such that the total response time is minimized?

3. Multidimensional Index Structures

a) Describe the differences and similarities between a B+-tree and a one-dimensional R-tree. When would you prefer one over the other? Explain.
b) Describe the differences and similarities between a two-dimensional R-tree and a two-dimensional Grid-file. When would you prefer one over the other? Explain.

4. Expressive Power of Languages

Consider the following schema:

KnownPeople(knowingPerson: string,, knownPerson: string, likesThem: real);

With the interpretation that a tuple (p1, p2, r) means that person p1 knows p2, and that p1 likes p2 “r” much, where “r” ranges from 0.0 (doesn’t like them at all) to 1.0 (really likes them!)

For each of the following queries, state whether it can be written in each of these languages: Tableau Queries, Relational Algebra, Datalog, Chandra-Harel's C(L,G), SQL.

(a) Is there a sequence of 2 people from DeWitt to Bill Gates? (That is, does DeWitt know anyone who knows Bill?)
(b) Is there some sequence of people from DeWitt to Bill Gates?
(c) Is there a person who is known by at least one other person but who knows nobody?
(d) Find the person DeWitt likes least.
(e) List all people DeWitt knows, ordered by how well he likes them.

5. Distributed Database Systems

Traditionally work in distributed database systems focused on systems involving 5-10 sites. Many papers with solutions to a number of problems including data placement, query execution, concurrency control, and recovery were published and several prototypes were built (including Ingres* and R*). While the commercial versions of these products were failures, they failed mainly for non-technical reasons (too hard to administer, slow adoption of the relational data model...).

The web itself is itself rapidly becoming a giant distributed database. Much of the data on the web is actually stored in database systems and web clicks are frequently turned into SQL queries by CGI bin scripts. There are several things that impede being able to run queries that span multiple web sites. First, current web sites expose their data as HTML pages and not tables with associated schema information. Second, there is no standard for describing the semantics of the data.

Assume for the purposes of this question that the advent of XML (a markup language which allows data to be tagged with semantic information) will provide at least a partial solution to the issue of semantics. That is, assume that in the future all web pages will contain data elements tagged by a description. For example, a bib.xml file might contain an author element that is labeled

<author>
author goes here
</author>

and that there will be an agreed upon intergalactic standard of these labels and what they mean.

Given that the semantics issue has been solved, discuss what additional mechanisms will be needed to enable the construction of a distributed query mechanism capable of running queries that span multiple web sites. In discussing your answer consider issues of security and scalability, especially with respect to query optimization and execution.