University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Spring 2003

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!

In addition to technical correctness, we will consider the quality of your writing including the support for your answer when grading your answers.

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. Since we have already asked about a dozen screening exam questions on set-valued attributes, we can't do that again. Instead, 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. Query Optimization

For each of the following queries, (i) identify one possible reason why an optimizer might not find a good plan, and (ii) rewrite the query so that a good plan is likely to be found. Any available indexes or known constraints are listed before each query.

a) An index is available on the *age* attribute:

```
SELECT E.dno
FROM Employee E
WHERE E.age=20 OR E.age=10
```

b) A B+ tree index is available on the *age* attribute:

```
SELECT E.dno
FROM Employee E
WHERE E.age < 20 AND E.age > 10
```

c) An index is available on the *age* attribute:

```
SELECT E.dno
FROM Employee E
WHERE 2*E.age < 20
```

d) No index is available:

```
SELECT DISTINCT *
FROM Employee E
```

e) No index is available:

```
SELECT AVG (E.sal)
FROM Employee E
GROUPBY E.dno
HAVING E.dno=22
```

f) The *sid* in Reserves is a foreign key that refers to Sailors:

```
SELECT S.sid
FROM Sailors S, Reserves R
WHERE S.sid=R.sid
```

3. Data Mining

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

b) Explain why association rules cannot be used directly for prediction, without further analysis or domain knowledge.

c) 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.)

4. B-trees in 2010

We are rapidly approach the point where all essential indices will be memory resident at all times.

a) How will this change effect how B-trees should be structured. For example, if the index is totally memory resident what should the node size be? Why?

b) Discuss the impact that a memory resident index will have on the implementation of how locking in B-trees should be performed.

5. Distributed database systems and replication

a) Give three reasons why replication is of interest in a distributed database system.

b) Describe lazy-master replication. Be sure to explain what happens when updates are performed to disconnected copies and explain how conflicts are resolved.

c) How well does lazy-master replication scale as the number of replicas is increased?

d) Suppose that the Employees relation is stored in Madison and that Employee tuples with *sal* <= 100,000 are replicated at New York. Consider the following three options for lock management: all locks managed at a *single site*, say, Milwaukee; *primary copy* with Madison being the primary for Employees; and *fully distributed.* For each of these lock management options, explain what locks are set (and at which site) for the following queries. Also state from which site the page is read

A query submitted at Austin wants to read a page of Employees tuples with *sal* <= 50,000.

A query submitted at Madison wants to read a page of Employees tuples with *sal* <= 50,000.

A query submitted at New York wants to read a page of Employees tuples with *sal* <= 50,000.