UNIVERSITY OF WISCONSIN - MADISON
COMPUTER SCIENCES DEPARTMENT

DATABASE QUALIFYING EXAM
SPRING 1997

Answer any *five* (5) of the following *six* (6) questions. Before beginning to answer a question, be sure to read it carefully. If you are confused about what the question means, state any assumptions you have made in formulating your answer. *Good luck*!

1. Recovery

Consider the following questions about the ARIES recovery algorithm.

(a) Is ARIES steal or no steal? Is it force or no force?

(b) What is meant by "during recovery ARIES repeats history" and why does it do so?

(c) Discuss the tradeoffs involved in choosing a checkpoint frequency.

(d) What is the impact on ARIES if arbitrary transactions are allowed to release their locks before the end of transaction?

2. Path Indices

In an OODBMS or Object/Relational DBMS we might have a schema

rel1(key1: int; p1: ref rel2);

rel2(key2: int; p2: ref rel3);

rel(K-1)(key(K-1): int; p(K-1): ref relK);

relK(keyK: int);

A path index could be defined to map "key1" to the "keyK"s to which it is associated by following a chain of pointers from rel1 to relK. That is, the index is useful for answering queries like "find all tuples in relK that are reachable from rel1 starting from a record in rel1 with key1 = 6."

One way to build such an index is to create a single index with search key key1 and values that are RIDs from relK. Another way is to create a set of K-1 indices, each indexing one step of the way. That is, the first index maps from rel1 to rel2; the second index maps from rel2 to rel3; and so forth.

Discuss the advantages and disadvantages of each.

3. File Access

R is organized as a sorted file with dense secondary indexes. Assume that R.a is a candidate key for R, with values lying in the range 0 to 4,999,999, and that R is stored in R.a order. For each of the following four relational algebra queries, briefly explain which of the following three approaches is most likely to be the cheapest, assuming the data follows a uniform distribution:

(a) Just access the sorted file for R.
(b) Use a (clustered) B+ tree index on attribute R.a.
(c) Use a linear hashed index on attribute R.a.

(1) $\sigma_{a<50,000}(R)$
(2) $\sigma_{a=50,000}(R)$
(3) $\sigma_{a>50,000 \text{ and } a<50,010}(R)$
(4) $\sigma_{a <> 50,000}(R)$

4. Grid Files and R-Trees

Typically discussions on how to use these data structures focus on how to use Grid Files for exact-match and range queries, and how to use R-Trees for overlap queries.

(a) Now consider using R-Trees for exact-match and range queries. For this part of the question, assume that the data you are storing are single points (rather than regions).

(i) What would the entries in the leaves of the R-Tree contain?

(ii) What would the entries in the parents of the leaves contain?

(iii) Would you expect to see 2-disk access performance for exact-match queries? Why or why not?

(iv) Would you expect to see range-query performance comparable to that of a Grid File? Why or why not?

(b) Next, consider using Grid Files for region overlap queries. You have a choice as to what you store in the Grid File. Suppose we are storing rectangles.

For each rectangle, would you store a single record with the coordinates of the center of the rectangle (along with its size in the x and y dimension), or would you store four records corresponding to the four corners of the rectangle? Explain your choice by describing (briefly!) how you would use your implementation for a lookup of all rectangles overlapping a specified rectangle.

5. SQL Consider the following relations for keeping track of which kids are hoping to get which toys during the coming holidays:

KID (ssno, name, age, address)
WANTS (kidno, toyno)
TOY (tno, tname, brand, price)

Write each of the following queries in SQL:

(a) Print the names of the kids under 10 years old who want a toy made by the "Nintendo" toy company that costs $100 or more.

(b) Print the name and brand of the most expensive toy. (In the event of a tie, print the name and brand of each such toy.)

(c) Print the name and average toy price for each toy company.

(d) Print the name of each toy that all of the kids want.

6. Theoretical Variety

For each of the following statements, provide a proof if it is correct or a counterexample if it is wrong.

(1) Assume that a relation schema is in BCNF and that some candidate key is simple (i.e., it consists of a single attribute). Then the relation schema is in 5NF.

(2) Assume that p is the optimal plan generated by the Selinger query optimization algorithm for a regular query Q that returns many tuples as an answer in general. Plan p is also optimal for the query that results from modifying Q so that it only asks for a single tuple, while keeping all other characteristics of Q (target list, qualification, etc.) unchanged.

(3) The relational algebra queries form a subset of Datalog queries.

(4) All polynomial-time queries are fixpoint queries.

(5) The magic rewriting of a Datalog program has always a more efficient bottom-up evaluation than the original program.