University of Wisconsin-Madison
Computer Sciences Department

Database Qualifying Exam
Spring 07

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 (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 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. Aho and Ullman "Universality of Data Retrieval Languages"

Consider the tables R_0(X:int, Y:int), R(X:int, Y:int), S(X:int, Y:int). Define the composition operator "$\circ$" as

$$R \circ R = \pi_{\$1,\$4}(R \, Join_{\$2=\$1} \, R)$$

In the following, U is the union operator from relational algebra, X is the cross-product, and – is the difference. Consider the equations

- f_1: $R = (R_0 \circ R_0) \; U \; (R \circ R)$.
- f_2: $R = \pi_{\$1,\$4}(R_0 \; X \; R_0) \; U \; (R \circ R)$
- f_3: $R = (R_0 - S) \; U \; ((R \circ R) - S)$

Note that the value of S is left unspecified in equation f_3.

a) Suppose $R_0 = \{(1,2),(2,3),(3,4),(4,5)\}$. What is the least fixed point of f_1?
b) Can f_1 be described in relational algebra? Argue informally why or why not.
c) Can f_2 be described in relational algebra? Argue informally why or why not.
d) Which of f_1, f_2, and f_3 always have least fixed points? For any function you think has a least fixed point, explain why; for any function you think does not have a least fixed point, explain why not.

2. The Coming of Flash

Storage devices made of Flash memory share many of the advantages of standard disk drives including non-volatility. In addition, since they are solid state devices they eliminate the seek and rotational delays that significantly impact conventional disk drives. Furthermore, they consume about a factor of 10 less power.

Currently Samsung is shipping a 32GB drive in a standard SATA form factor for laptops. Other vendors have announced drives that pair a smaller flash drive (say 8GB) with a conventional drive. In terms of cost/gigabyte such drives are much more expensive than conventional drives currently. However, they are predicted to get much cheaper in the years to come.

Assume that drives based on flash memory replace conventional disk drives five years from now.

Your task in this question is to speculate on how the switch from standard disk drives to Flash memory might impact the design and implementation of relational database systems. Obviously, this is an open ended question and there is no single correct answer. However, you may want to consider such issues as how the move to flash might impact query processing algorithms, indexes (clustered vs. unclustered), the WAL approach to recovery.

3. Parallel Database Systems

The parallelization of relational database queries takes the following path:

SQL -> Optimizer -> Parallelization -> Execution

That is, first the optimizer is invoked to produce the best serial plan and then this plan is parallelized to produce a parallel plan.

a) Speculate on why the phases of optimization and parallelization are separated.

b) Is this approach likely to lead to an optimal plan? Why or why not?

c) How might adaptive query optimization be exploited to improve the quality of the parallel plans produced? (Recall that "adaptive query optimization" refers to an approach in which the system can change the plan it is running during query execution if it becomes clear that the initial plan found by the optimizer is not a good one.)

4. Data Mining

a) Briefly describe the BIRCH clustering algorithm. What fundamental ideas allow it to scale to very large data sets?

b) In recent years, many data management applications increasingly must exploit domain integrity constraints to improve their accuracy. For clustering, a common kind of constraint is "these two data points must belong to the same cluster" or "these two data points do not belong to the same cluster". Describe how you would extend BIRCH to efficiently incorporate such constraints.

c) Give two other kinds of constraints you can imagine might be useful for such clustering settings. (Here the notion of constraint is very general and can include any property you might want to impose on the resulting clusters.) Discuss whether your extended algorithm can or cannot handle them, and why.

5. Query Optimization

Assume three tables

- Students(sid,name,address,gpa), with a clustered index on sid and an unclustered index on name;
- Courses(cid,title,room,building,stime,etime,max-credits), with a clustered index on cid and an unclustered index on room;
- Takes(sid,cid,num-credits), with a clustered index on sid and an unclustered index on cid.

a) Given the query:

```
SELECT sid, name
FROM Students, Courses, Takes
 WHERE (Students.gpa > 3.5) AND (Students.sid = Takes.sid) AND
       (Takes.cid = Courses.cid) AND (Courses.room = 325)
```

list all the "interesting orders" that will be considered by System R Optimizer in the first step of query optimization.

b) List all strategies for joining two relations that the System R Optimizer might consider for the query in Part a. (By "might" we mean that it is possible that the System R Optimizer could consider the strategy if the underlying single table access plans it uses are not pruned in the first step of optimization.) For each strategy, list which two relations are to be joined, which ones will be the outer and inner relations, respectively, the join method used, and how the relations are accessed.

c) Joe Qualtaker is writing a query optimizer to find the best plan to access and join data from multiple Web data sources. He finds that, similar to the relational setting, each such Web source can be viewed as a relational table with multiple access paths (e.g., one path allows accessing data via a query interface, another path allows browsing the data, etc.). But unlike the relational setting, here each access path may not have a full "coverage". That is, one path may be quite fast, but allows Joe to retrieve only 80% of the data in the source, whereas another path is slower, but allows Joe to retrieve 100% of the data.

To address this problem, Joe define the quality of a query execution plan P to be *quality(P) = alpha * (estimated time to execute P) + beta * (estimated coverage of P)*, where alpha and beta are co-efficients to provide tradeoffs between runtime and coverage. Given a SQL-like query Q over the Web sources, his goal is then to find the execution plan with the highest quality.

He plans to adapt the System R Query Optimizer to this problem. What do you think is the main problem he will encounter in this process?