UNIVERSITY OF WISCONSIN-MADISON
COMPUTER SCIENCES DEPARTMENT

DATABASE SCREENING EXAMS
FALL 1991

BREADTH EXAM

Answer any *four* (4) of the following *five* (5) questions. You may *not* select questions from the Depth Exam. 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. Sorting

Suppose that in file F to be sorted, tuples are 100 bytes long, and that F contains 40M tuples. Furthermore, suppose that the system doing the sorting uses 4K byte pages (memory and disk).

(a) What is the minimal number of memory pages needed to sort F in two passes? How much memory is this (in bytes)?

(b) Let M be the minimal amount of memory you calculated in part (a). Suppose that M is now partitioned into 1K byte pages. Can you still sort F in two passes? What if M is partitioned into 8K byte pages?

(c) Now consider a two-pass external sort of an arbitrary file. Suppose that the initial internal sort of a run of p records takes $p \log p$ comparisons, and that the merge phase is done in such a way that each operation "choose next smallest record" can be accomplished in a number of comparisons that is logarithmic in the number of runs. Let the file to be sorted have n records. What is the total number of comparisons in the sorting algorithm as a function of k, the number of runs generated by the initial sorting phase?

2. SQL

Due to the incredible turnout for the database screening exam, the database faculty has decided to build a database of information they find useful in grading the exams and recording and justifying the results. After much debate they settled upon the following schema:

STUDENT(StudentID, HairColor, Height)
ANSWERS(ProblemNumber, StudentID, Grade)

The intent here is that STUDENT represents important information about the students and ANSWERS records the grades given to each answer written by the students (that is, ANSWERS contains a tuple for each answer written by a student). However, the faculty needs your help in translating the following four queries into SQL:

(a) Give all brown-haired students an "A" on all questions after the fifth.

(b) Find the average height of the students who received "A"s, "B"s, and "C"s (not "D"s or "F"s) on problem 6. (Note: the answer to this query should consist of three averages.)

(c) Find all problems such that the only students who received an "A" on the problem were over 6 feet tall.

(d) For each problem on the exam, print the number of students who answered the problem with a grade of "C" or better.

3. Conceptual Database Design

Notown Records has hundreds of musicians that perform on their albums, so they have decided to migrate their records about musicians and albums into a database. They have offered you a consulting job as a database designer. In exchange, you will receive 0.001% of the net income from Jackal Mikeson's next album (making your projected consulting rate approximately $100/msec). Here is a description of the information that Notown Records needs to manage:

- Each musician that records at Notown has a social security number, a name, an address, and a phone number.
- Some of musicians are full-time employees of Notown, in which case they also have a salary and a length of employment (in years).
- The remaining musicians are employed part-time, on a per-album basis, in which case they have a per-album contract number and a per-album fee for each album on which they perform.
- Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, saxophone, voice, nose flute) and a musical key (e.g., C, B-flat).
- Each album that is recorded on the Notown label has a title, a copyright date, an album identifier, and a style (e.g., R&B, heavy metal, new age, folk).
- Each song recorded at Notown has a title and a list of composer names (since a given song may have been co-written by more than one composer).
- Each song on any given album is performed by one or more musicians, and a musician may perform a number of songs; associated with each such song performance are the instruments that the musicians play in that performance.
- For each album, there is exactly one full-time Notown musician that acts as its producer, though a given musician may produce multiple albums.

Given this information, your job is to help Notown to arrive at a schema that will nicely capture the nature of their data.

(a) Design a conceptual schema for Notown and draw an E-R diagram for your schema. Be sure to indicate whether relationships are 1:1, 1:N, or M:N on your E-R diagram, and state any assumptions that you make along the way. You can use all features in the Extended ER Model.

(b) If you had to design a schema using only the Basic ER Model (i.e., if you could only use the concepts of "entity", "relationship" and "attribute"), are there any integrity constraints, based on the information above, that you could not capture?

(c) Given that you own stock in IBM, you have decided to recommend the use of a relational DBMS for Notown data management. Translate your E-R schema from (a) into a good relational schema.

4. Convoys

You are walking through the Database Systems Boothill, and you notice the inscription "Convoy was here."

(a) What is a convoy?

(b) Why is one formed?

(c) What can be done to make them go away?

(d) What sort of data and/or internal system resources are likely to be subject to the convoy problem? Why?

5. Referential Integrity

Consider the specific type of integrity constraints that are used to ensure what is termed the *referential integrity* of the data.

(a) What is a referential integrity constraint, and how is it different from a "regular" integrity constraint?

(b) Explain how updates could violate referential integrity constraints, and discuss the options available for enforcing them.

(c) Query modification is the technique that most systems use to ensure that updates do not violate any integrity constraints.

  (i) Describe how query modification works for regular integrity constraints.

  (ii) Discuss the viability of using query modification for referential integrity constraints.

DEPTH EXAM

Begin by taking the breadth exam, following the instructions given there. Then answer *three* (3) of the following *five* (5) questions.

6. Join Algorithms

This question asks you to compare two algorithms for computing the equijoin of relations R and S: the GRACE hashing algorithm, and sort-merge.

(a) Let relation R contain |R| pages, and S contain |S| pages. Suppose that there is sufficient memory in the system that both the sort-merge and the GRACE hashing algorithm can be done in two passes. Derive expressions for the number of sequential I/Os and random I/Os (in terms of |R| and |S|) for both join algorithms.

(b) Suppose that values in the join attributes for the relations are not uniformly distributed, e.g., they are skewed so that some values appear many times and others only once. How would you expect this skew to affect the relative performance of the two algorithms?

(c) Now suppose that R has 1/10 as many tuples as S. How would you expect this to affect the relative performance of the two algorithms? (Hint: you may want to consider the amount of memory required by sort-merge and by GRACE hash for this case.)

7. Concurrency Control

In contrast to traditional database concurrency control algorithms, where accesses to the current version of the database are mediated by a mechanism that detects and resolves conflicts, Reed proposed an approach involving the (temporary) retention of multiple versions of data. One motivation for the use of a multiversion concurrency control algorithm like Reed's is a desire to provide a higher degree of concurrency for transactions.

(a) Briefly explain how Reed's multiversion concurrency control algorithm works. Present the algorithm from a concurrency control standpoint, ignoring the recovery-related aspects of his scheme. Explain how both read and write requests are processed, and describe the conditions under which transactions must wait and/or abort.

(b) Support or dispute the claim that Reed's algorithm provides higher concurrency than a single-version algorithm (e.g., basic timestamp ordering). Use example histories (i.e., transaction executions) where appropriate to illustrate your point, and be sure to point out the sorts of transactions that benefit (if any).

(d) Reed suggests limiting the storage overhead of his scheme by reclaiming space from versions that are older than a certain threshold range of timestamp time. What potential problem(s) does that cause? Propose an alternative criteria for garbage-collecting prior versions of data that eliminates this problem.

8. Advanced Data Models

Depending on the phase of the moon and the audience, the designers of the Postgres data model occasionally claim that it is an object-oriented data model. Identify the characteristics of the model that support the claim and those that do not. Substantiate your answers by comparing it to your favorite core object-oriented data model, e.g., O2 or Orion.

9. Database Design Theory

It has been suggested that the logical database design problem is to take a set of attributes and a set of data dependencies over these attributes and to find an *equivalent* database scheme (i.e. a collection of relation schemes) that is *better* in some sense.

(a) In determining whether two database schemes are equivalent, it is natural to ask if there is a bijective mapping between the two schemes. That is, (i) given a valid relation instance over one scheme (i.e. an instance that obeys all the data dependencies), there should be a corresponding valid instance over the other scheme. Further, (ii) given a valid instance over one scheme, we should be able to uniquely construct the "corresponding" instance over the other scheme. If the data dependencies are all functional dependencies, formally define properties (i) and (ii).

(b) Impressed by your knowledge of database normalization theory, the Tux_R_Us Formal Wear company, located in Normal, Indiana, has hired you to design their database for them. Their information concerns employees, managers and departments. Every employee (key: Eid) belongs in exactly one department (key: Dname), and every department is managed by exactly one of the employees in it. You are asked to do the following:

(i) Formalize the data dependencies, and

(ii) Produce a database scheme in BCNF to capture the relationships between employees, departments and managers.

You do this in short order and are heading to the bank to deposit your fat consultant fee when Tux people call you and complain that there is still some redundancy in your design.

(iii) How can a BCNF scheme have a redundancy? If so, can it be fixed?

10. View Maintenance

You have worked for many years to produce a thesis on the view update problem, and one fine morning, you wake up with the brilliant realization that most people really just want to deal with the view maintenance problem. Anyway, and this is the important point, you convince your advisor of this fact, and change your thesis topic.

(a) Your first task is to determine how to incrementally update (i) join, (ii) project and (iii) union views when tuples are inserted into the base tables.

(b) You happily present these results to your advisor, and he ruins your day by telling you to consider deletions of tuples from the base tables. Moreover, he tells you not to consider any solutions that require you to maintain extra columns in either the base tables or the views. Being a nice guy, though, he tells you that he will approve your thesis if you can deal incrementally with view definitions such that the materialized views never contain duplicates. (The view definitions are in SQL and do not contain the keyword *distinct*; so duplicates are not automatically removed.) Can you do it?