

Database Qualifier Spring 2001

Answer seven (7) of the eight (8) questions. If you make any assumptions about a question, be sure that you state them clearly.

1. Database Design

Part I: Relational Databases

In relational database design, the two approaches proposed in the literature fall into design by successive decomposition of relations and design by synthesis of functional dependencies.

- (a) What are the practical difficulties in using the synthesis approach? Why do we not see any commercial design tools that follow this approach?
- (b) Synthesis algorithms start by first computing the minimum cover of the given set of functional dependencies. Does this guarantee that the 3NF relations resulting from it are unique? Why or why not? Is it possible to design different algorithms with different criteria of minimality for the resulting design? Can you suggest some such criteria?
- (c) Is it always possible to construct a BCNF design from the given functional dependencies? If yes, what is the sacrifice made in the process?
- (d) Illustrate the above sacrifice by appropriately converting the following 3NF relation into a BCNF relation: *PATIENT_VISIT*(*Patient*, *Hospital*, *Doctor*) Where: *Patient*, *Hospital* \rightarrow *Doctor* and *Doctor* \rightarrow *Hospital*. (make sure that this is a lossless or non-additive decomposition).

Part II: Object-Oriented Databases

Assume that you are a DBA managing a large relational DB installation at a hospital that includes data on patients, doctors, procedures and lab tests etc. It has been decided to transfer over to an OO DBMS environment such as ObjectStore. (Do not question whether or not this is a good decision).

- (e) From a modeling standpoint, what challenges does this open for you? In other words, what will be your approach to enhancing the relational design to make sure that you will benefit from the O-O approach?
- (f) Would you embark on a reverse engineering project to first abstract the existing design into a conceptual design (such as extended ER schemas?) - defend your answer.
- (g) Develop a generalized checklist of things to be accounted for in going from a relational DB design into an O-O DB design.

2. Database Architecture

Consider the following components (modules) of current relational database management systems:

- (a) Storage management including storage organization, indexing and caching
- (b) Catalog management
- (c) Query Language processing and optimization
- (d) Run-time transaction processing including concurrency control and recovery

Consider the following ongoing efforts to include additional features into database management systems:

1. Rule Processing - data based rules as well as external "business" rules
2. Interoperability with other DB systems, possibly with different models
3. Web access and web-based query and transaction processing
4. Mobile operation of clients

Choose any **TWO** features from the lower list (items 1 thru 4). For those two features, write a few sentences each on what impact it has on the design of the database modules a thru d (discuss all modules). Your answer can be structured in terms of issues, difficulties or problems and solution techniques under each category. Feel free to give literature references (i.e., give some idea of ongoing research).

3. Data Models

A data model provides capabilities for describing data structures, constraints and operations on the data. The emerging XML standard provides capabilities for describing XML schemata as well as XML documents. Evaluate XML as a data model.

4. Query Optimization

Consider the relations, $R1(\underline{A},B,C,D)$ and $R2(\underline{A},B,D)$ and the following pair of SQL queries:

```
SELECT S.A
FROM R1 AS S
WHERE S.B = 'X' AND S.C = 'Y' AND NOT EXISTS(
    SELECT T.A
    FROM R1 AS T
    WHERE T.B = 'Z' AND T.C = 'Y' AND T.D < S.D);
```

```
SELECT A
FROM R1
WHERE B = 'X' AND C = 'Y' AND D < ALL(
    SELECT D
    FROM R1
    WHERE B = 'Z' AND C = 'Y');
```

Now consider this pair of SQL queries:

```
SELECT E.B,E.C
FROM R1 AS E
WHERE E.A IN (
    SELECT A
    FROM R2
    WHERE E.B = B AND E.D = D);
```

```
SELECT E.B,E.C
FROM R1 AS E, R2 AS F
WHERE E.A = F.A AND E.D = F.D AND E.B = F.B);
```

Nested queries, in particular correlated nested queries, are known to be difficult to optimize. The two pairs of example queries above illustrate how to remove correlation and how to remove nesting. Consider the different types of nested SQL queries (ignore queries involving built-in aggregate functions).

- (a) Draft an algorithm for removing correlation. Justify!
- (b) Draft an algorithm for removing a single level of nesting. Justify!

5. Physical Storage Structures

Suppose we have a file whose schema is Employee(SSN,Name,Age,Dept,City,Salary) and want to process queries like

- (i) Select * from Employee Where Dept = 100
- (ii) Select * from Employee Where Age = 50
- (iii) Select * from Employee Where Dept = 200 And Age = 25

One storage structure that might provide efficient retrieval performance for the above queries is the partitioned hash file organization. For the partitioned hash file organization, you generate a 10 bit hash bucket address from the two fields, Dept and Age, where Dept contributes the high order 6 bits and Age the remaining 4 bits. The buckets are of size 4K bytes and each record of size 200 bytes is stored in the appropriate bucket. The buckets are stored contiguously on disk from address 0000000000 to 1111111111 (in binary). We can assume that there are no overflow buckets.

- (a) If we process the above three queries, then how many buckets will be accessed for each? Can any of these three queries benefit from sequential I/O processing? Explain/defend your answers.
- (b) In our example, we were given the number of bits associated with each of the fields with regard to the hash function. One aspect of designing a partitioned hash file is to determine the number of bits, b_i , associated with each field so that we minimize the expected number of bucket accesses. Assume that each query specifies only one field, there are k fields and the number of buckets is 2^B . Also, associated with each field is the probability, p_i that the field appears in a query. With this said, determine the general formula for the expected number of bucket accesses.

6. Concurrency Control

(a) Here is a schedule for two transactions, with one action missing (denoted by ???):

$$r_1(A), r_2(B), ???, w_1(C), w_2(A)$$

What actions of certain types could replace the ??? and make the schedule not serializable. Tell all possible nonserializable replacements for each of the following types of action:

- Read actions
- Write actions
- Increment actions

(b) For each of the two schedules of transactions T_1, T_2 and T_3 shown below, insert shared, exclusive and update locks, along with unlock operations.

- (i) $r_1(A), r_2(B), r_3(C), r_1(B), r_2(C), r_3(D), w_1(C), w_2(D), w_3(E)$
- (ii) $r_1(A), r_2(B), r_3(C), r_1(B), r_2(C), r_3(A), w_1(A), w_2(B), w_3(C)$

You can place a shared lock in front of every read action that is not going to be upgraded, place an update lock in front of every read action that will be upgraded and place an exclusive lock in front of every write action. Place unlocks at the ends of the transactions.

Tell what happens when each of the above two schedules is run by a two phase locking scheduler that supports the three lock types.

7. Recovery

A given transaction T updates database objects A and B in this order. As some point during or after execution of T, the system crashes. For each of the two objects, there are four possible cases:

1. T's update was not written to the log, and it was not propagated to the database.
2. T's update was written to the log (on disk) but it was not propagated to the database.
3. T's update was not written to the log, but it was propagated to the database (on disk).
4. T's update was written to the log (on disk) and it was propagated to the database (on disk).

(Hints: when we say that T's update was not propagated to the database, it simply means that the update could have been reflected in the in-memory log-tail, or could have been made to a copy of the object in the buffer pool, but these changes did not make it to disk.)

Thus, we can represent the state of affairs (after the crash) by two numbers (one for each object) referring to the cases above. For example, the presentation of A:1, B:2 means that A's update was neither logged nor flushed to the database, while B's update was logged but not written to the database. Please answer the following questions:

- (a) Assume that pure undo logging is used and that T did not commit before the crash. List all possible states using the A:n, B:m notation.
- (b) Assume that pure undo logging is used and that T did commit before the crash. List all possible states.
- (a) Assume that pure redo logging is used and that T did not commit before the crash. List all possible states.
- (d) Assume that pure redo logging is used and that T did commit before the crash. List all possible states.

8. Distributed Databases

Part I: Basic Questions

- (a) Describe the main differences between traditional distributed database systems and federated database systems?
- (b) Describe how the semi-join operator works? Are semi-join operators always beneficial?
- (c) Describe the main steps in distributed query processing, especially the key problems addressed and the common techniques used in each of the steps.

Part II: Advanced Questions

- (d) Describe how mediator-based database systems work? List three main differences between distributed database systems (classical or federated) and mediator-based database systems?
- (e) Describe how you envision XML and the second generation Web. You are encouraged to use real world examples or experiences to back up your vision
- (f) How do you see the role of database research in the Internet data management? List three most important database technologies that can be applied to the Internet Data Management applications. Explain your answer.