DB Qual 2009

This exam consists of three parts. Please answer each part in a separate blue booklet. You have a total of 60 minutes for the entire exam. You can allocate time among the parts as you like, but be sure not to get stuck in one part without spending some time in the other parts.

The exam is open book and notes. No laptops are allowed. Simple calculators are allowed.

If you feel that a question is ambiguous or unclear, state any reasonable assumptions you need to make in order to answer the question.

DB Qual 2009; Part A

Problem A1 (12 points)

(a) [**4 points**] Relation $R(A, B, C)$ has no NULL's and satisfies the functional dependencies $AB \rightarrow C$ and $C \rightarrow B$. Let a, b, and c be the numbers of different values found in R for attributes A, B, and C, respectively. What numerical constraints on a, b, and c must hold? Justify your answer briefly.

(b) [**3 points**] Suppose that R from part (a) also satisfies the functional dependency $B \rightarrow A$. What constraints on a, b, and c must now hold?

(c) [**3 points**] A *minimal basis* for a set of functional dependencies is an equivalent set of FD's such that

1. If you remove any FD from the set, then the new set is not equivalent to the old set.
2. If you remove any attribute from the left side of any FD, then the new set is not equivalent to the old set.

Find a minimal basis for the set of FD's from part (b), that is, $AB \rightarrow C$, $C \rightarrow B$, and $B \rightarrow A$.

(d) [**2 points**] Sometimes, a minimal basis is not unique. Are there any other minimal bases for the FD's of part (c)? Either give an example of another, or explain briefly why there cannot be any other. *Note*: a minimal basis need not be constructed from the given set; it just needs to be minimal and equivalent to the given set. However, we should not view as different two sets that differ only in the order of the FD's or in the order of attributes within an FD.

Problem A2 (8 points)

Suppose there are relations $R(A)$, $S(A)$, and $T(A)$. We can express a constraint in relational algebra by writing an expression whose value is empty if and only if the constraint is satisfied. In what follows, you can write complex expressions by a sequence of assignment statements, in which the left side is a relation with named attributes (e.g., $U(B, C)$) and the right side is an expression of relational algebra. Develop expressions for the following constraints:

(a) [**3 points**] Every tuple of R is in either S or T (or both).

(b) [**5 points**] There exists a tuple of R that is in neither S nor T. *Hint*: Remember that the expression must be empty when the constraint is satisfied, not when it fails to be satisfied. Also, one capability of relational algebra, not often used, is the ability to create a constant relation. In this problem, it helps to create $U(B) := \{(1)\}$, that is, a unary relation with one tuple, having value 1.

DB Qual 2009; Part B

Problem B1 (9 points)

Suppose we have an XML database containing information about Ph.D. students in the CS and EE Departments. CS students have a name, advisor, and office, while EE students have a name, advisor, and quals-year. Here is some sample data:

```
<Students>
  <Student Dept="CS">
    <Name> Abby </Name>
    <Advisor> Aiken </Advisor>
    <Office> Gates A1 </Office>
  </Student>
  <Student Dept="CS">
    <Name> Ben </Name>
    <Advisor> Boneh </Advisor>
    <Office> Gates B1 </Office>
  </Student>
  ... more CS and EE students ...
  <Student Dept="EE">
    <Name> Carol </Name>
    <Advisor> Cover </Advisor>
    <Quals> 2008 </Quals>
  </Student>
  <Student Dept="EE">
    <Name> Dave </Name>
    <Advisor> Dutton </Advisor>
    <Quals> 2009 </Quals>
  </Student>
  ... more CS and EE students ...
</Students>
```

(a) [**5 points**] Specify a DTD for this data, conforming as closely as you can to the sample above. If you make any additional assumptions beyond those stated or illustrated, please explain them. Don't worry too much about exact syntactic details in the DTD.

(b) [**4 points**] Suppose you are to create a relational database for storing the same data. Suggest two different plausible relational schemas. For each of the two schemas, state one significant advantage it has over the other one.

Problem B2 (5 points)

Consider a table `Taking(student,advisor,subject)` containing information about what subjects students are taking on the Systems Qual. To keep things simple, assume each student takes only one subject, i.e., `student` is a key for table `Taking`. No other attributes are a key. Write a SQL query that returns those advisors whose advisees are all taking different subjects on the qual. Your solution will be graded on conciseness and clarity, as well as on correctness.

Problem B3 (6 points)

Each of the following questions about query optimization can be answered correctly in 1–3 short sentences.

(a) Consider the join of two tables R and S on join condition $R.A = S.B$, and suppose the result is to be ordered by B. Describe a scenario where it's better to use a nested-loops join rather than a sort-merge join, even though the result needs to be ordered on the join attribute. Your scenario may include cardinalities, selectivities, and/or indexes, but keep it as simple as possible.

(b) In a relational DBMS, when query *precompilation* is used, a query is compiled once into an execution plan, then the plan is stored in the database and used whenever the query is issued. It is a big savings to avoid compilation each time a query is issued, since query compilation is typically very time-consuming. Unfortunately precompiled queries may execute with poor performance. Briefly explain why.

(c) Some of the worst inaccuracies in query plan cost estimation are due to hidden "correlations" between values of attributes in a relation. Consider a relation $R(A_1, A_2, \ldots, A_n)$, and suppose you want to keep one statistic (one integer or float, say) for each pair of attributes A_i and A_j in R, in an attempt to capture correlation between those two attributes. What statistic would you keep? (You may assume that typical statistics are already kept for R and its single attributes, such as cardinality and number of distinct values.)

DB Qual 2009; Part C

Problem C1 (10 points)

We wish to study the performance of the logging system of a database management system. The system uses undo-redo physical logging, and uses non-quiescent checkpoints. We are given the following performance numbers:

- All transactions combined write X database objects per second.
- A checkpoint is started every Z seconds.
- The checkpoint process flushes to disk Y objects per second.
- The buffer pool holds K objects.

To make our computations simple, we can view the above numbers as fixed. For instance, Y is not an average or worst case number; we assume the system flushes exactly Y objects every second during the checkpoint.

(a) How long does a checkpoint take? (You can ignore the time taken to write the start and end checkpoint records.)

(b) During recovery after a crash, what is the maximum number of redo actions we may need to perform? Assume that at the time of the crash, all writes were for committed transactions.

(c) Consider all the redo log entries made while a checkpoint is in progress (i.e., those between a start checkpoint and an end checkpoint entry). We say that one of these redo entries is *redundant* if the new value reported by the undo action was flushed to disk by the current checkpoint process. (If an entry is redundant, it is not necessary to redo that action during recovery. Of course, the recovery process will redo the action regardless of whether it is redundant or not.) On average, how many redo entries made during a given checkpoint are redundant?

Problem C2 (10 points)

Consider a schedule S_1 with the following arcs in its precedence graph (and only these arcs): $T_4 \rightarrow T_5$, $T_1 \rightarrow T_5$, $T_3 \rightarrow T_4$, $T_3 \rightarrow T_1$, $T_4 \rightarrow T_2$.

(a) Give a serial schedule that is equivalent to S_1.

(b) Consider a second schedule S_2 whose precedence graph happens to have exactly the same vertices and arcs given above. Do we know for sure that S_2 is conflict-equivalent to S_1?

(c) Schedule S_3 has the same arcs in its precedence graph as does S_1, plus the arc $T_2 \rightarrow T_3$. Schedule S_4 has the same precedence graph as S_3. Do we know for sure that S_3 is conflict-equivalent to S_4?

(d) Consider a schedule S_5 that we know is conflict-equivalent to S_3. What is the precedence graph of S_5?

(e) Let $Sh(T_i)$ be the shrink point of T_i in S_1, i.e., the position in S_1 of the first unlock action of T_i. If S_1 is a legal schedule, and all transactions are well formed and two-phase, what can we say about $Sh(T_3)$ and $Sh(T_5)$ with respect to $Sh(T_1)$?