DB Qual 2007

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 2007; Part A

Problem A1 (10 points)

In the following Datalog rules, E and F are EDB predicates and P and Q are IDB predicates.

```
P(x,y) :- E(x) AND E(y)
Q(x,y) :- F(x,y) AND NOT P(x,y)
```

a) (3 pts.) If E contains the tuples (1) and (2), and F contains the tuples (1,2), (1,3), and (2,3), what is the stratified model (stratified fixedpoint) for this Datalog program?

b) (4 pts.) If E and F are as in part (a), what are all the nonstratified models (least fixedpoints) for this Datalog program?

c) (3 pts.) Write a relational-algebra expression that produces Q from E and F. You should not assume that E and F have any particular value, as you did in parts (a) and (b).

Problem A2 (5 points)

Let F be a set of functional dependencies for a relation R. The *closure* of a set X of attributes of R, denoted X^+, is the set of attributes A such that $X \rightarrow A$ follows from F. Note that $X \subseteq X^+$ always holds. A set X of attributes of R is said to be *closed* (with respect to the FD's F) if $X^+ = X$. We know that the relation $R(A, B, C)$ below satisfies F:

A	B	C
1	2	3
1	2	4
5	6	3

However, we do not know exactly what F is. As a result, we may or may not be able to conclude that a set S of R's attributes (*i*) is definitely closed, (*ii*) may or may not be closed, or (*iii*) is definitely not closed. Classify each of the seven nonempty subsets of $\{A, B, C\}$ as (*i*), (*ii*), or (*iii*).

Problem A3 (5 points)

Draw an entity-relationship diagram that reflects the following information:

1. There are countries, each with a name and population. No two countries have the same name.
2. There are rivers, each with a name and length. It is possible for two rivers to have the same name or the same length, or both.
3. Each river flows through one or more countries. Sometimes, a river forms part of the border of a country through which it flows. We consider that a river ”flows through” a country even if it only forms part of the border of that country, but we want to know whether a river is only border, only interior, or both border and interior for a country.

Make your design as simple as possible, consistent with the need to represent all the information above.

DB Qual 2007; Part B

Problem B1 (10 points)

Consider a table `Player(name,hand,skill)` listing table-tennis players.

- Attribute `name` is a key.
- Attribute `hand` is either "left" or "right."
- Attribute `skill` is an integer specifying the player's skill level.

You are to write a query that returns pairs (n_1,n_2) of player names such that n_2 is a potential doubles partner for n_1:

(1) n_2.`hand` complements n_1.`hand`: one has `hand` = "left" and the other has `hand` = "right".

(2) Among the players satisfying (1), n_2's skill level is the closest to n_1's. That is, there is no player n_3 satisfying (1) such that n_3.`skill` and n_1.`skill` are closer together than n_2.`skill` and n_1.`skill`.

1. [**2 points**] Given these assumptions on the data and specifications for the query, and no others, is the first attribute in the query result (the one referred to as n_1) guaranteed to be a key? Briefly justify your answer.

2. [**2 points**] Given these assumptions on the data and specifications for the query, and no others, is the query result symmetric? That is, if (n_1,n_2) is in the query result, is (n_2,n_1) guaranteed to be in it too? Briefly justify your answer.

3. [**6 points**] Write the query in SQL. Your query will be graded on clarity and conciseness, as well as on correctness. (You may assume a SQL function `dist`(a, b) implementing $|a - b|$, if you find it useful.)

Problem B2 (4 points)

Consider two tables $R(A)$ and $S(B)$. Let A be the primary key for R and suppose we wish to enforce referential integrity from $S.B$ to $R.A$. We write the following SQL-99 tuple-based constraint on S:

```
check (exists (select * from R where A = B))
```

Is this constraint valid in SQL-99? If not, what's wrong? If so, does it properly enforce referential integrity? Briefly justify your answer.

Problem B3 (4 points)

Consider executing the following query on table $R(A, B)$:

```
Select *
From R
Where R.A = 5
```

You are deciding whether to use an index scan on a 3-level B+ tree for the $R.A = 5$ condition, or use a full table scan on R.

- R has 100 pages.
- On average there are 10 records on a page.
- Record values are distributed randomly.
- You expect approximately 1% of the records to satisfy $R.A = 5$.
- A linear table scan is 5 times as fast per page as random page accesses.

Assuming a typical I/O cost metric, should you use a table scan or an index scan? Briefly justify your answer.

Problem B4 (2 points)

When a query plan is comprised primarily of nested-loop joins, which is most likely to be selected: a left-deep plan, a right-deep plan, or a bushy plan? (Just state the answer, no need to justify.)

DB Qual 2007; Part C

Problem C1 (10 points)

(a) Consider a B+ Tree of order $n = 5$ (each node can hold n keys and $n + 1$ pointers). The tree currently has 3 levels. What is the *minimum* number of records that can be indexed by this tree? Assume there are no duplicate keys.

(b) Consider the same B+ Tree of order $n = 5$ with 3 levels. What is the *maximum* number of records that can be indexed by this tree? Assume there are no duplicate keys.

(c) Consider an extensible hash table where each bucket can hold 5 records. The directory currently has 8 entries ($i = 3$) and cannot be made any smaller. What is the *minumum* number of records that can be stored in this table? Assume that all keys hash to distinct values.

(d) Consider the same extensible hash table where each bucket can hold 5 records, where the directory currently has 8 entries ($i = 3$) and cannot be made any smaller. What is the *maximum* number of records that can be stored in this table? Assume that all keys hash to distinct values.

Problem C2 (10 points)

(a) Consider a conflict-serializable schedule S of a set of transactions T. Formally prove that there exists a transaction $T_i \in T$ such that T_i has no incomming edges in the precedence graph $P(S)$.

(b) Formally prove that all serial schedules are recoverable.