CPS216 Advanced Database Systems - Fall 2008
Assignment 4

- Due date: Nov 25, 2008, 2.50 PM. Late submissions will not be accepted.
- Submission: In class, or email solution in pdf or plain text to shivnath@cs.duke.edu.
- Do not forget to indicate your name on your submission.
- State all assumptions. For questions where descriptive solutions are required, you will be graded both on the correctness and clarity of your reasoning.
- Email questions to shivnath@cs.duke.edu.

Question 1 **Points 5**

This question is based on the Ioannidis paper. Suppose you have a database with three tables R, S, and T, and the following query on this database:

```
Select  R.A, S.B, T.C, T.D
From    R, S, T
Where   R.num = S.num and S.val = R.val and R.val = T.val and T.id > 100
Groupby R.E
Orderby R.F
```

What histograms (e.g., what type?, on which attributes?) would you recommend for improving the performance of this query? Justify briefly.

Question 2 **Points 15**

Consider a table R(A,B) where the values of the attributes have the following data distribution:

- A is an integer attribute. There are 20 distinct values 1,2,...,20 of A in R.
- For each value v in 1,2,...,20, there are 2×v records in R having R.A=v.
- B is also an integer attribute, and there are 20 distinct values 1,2,...,20 of B in R.
- For each value v in 1,2,...,10, there are 2×v records in R having R.B=v. For each value v in 11,12,...,20, there are v records in R having R.B=v.

1. Give an equi-width histogram on R.A with 5 buckets.
2. Give an equi-depth histogram on R.A with 5 buckets.
3. Give a serial histogram on R.A with 5 buckets.
4. Give an end-biased histogram on R.A with 5 buckets.
5. Give a serial histogram on R.B with 5 buckets.
6. Give an end-biased histogram on R.B with 5 buckets.

Question 3 **Points 5**

Consider a database system with three types of locks: S (shared), I (increment), X (exclusive). We wish to extend the system to handle multiple-granularity locks by adding "intention" locks IS, II, and IX. Locks IS and IX are the same as discussed in class. Intention lock II on an object at level i indicates the intention of the lock holder to lock objects at level i + 1 in I mode. Give the compatibility matrix for the proposed scheme.

Question 4 **Points 15**

Assume that a database using Undo/Redo logging and nonquiescent checkpointing crashes with the log records on disk given below. Record $< T, X, v, w >$ means that transaction T changed the value of database element X; its former value was v, and its new value is w.

```
<START T1>
<T1, X, 14, 28>
<T1, Y, 15, 5>
<START T2>
<T2, Z, 20, 10>
<COMMIT T1>
<START CKPT (T2)>
<T2, W, 4, 7>
<START T3>
<END CKPT>
<T3, X, 28, 17>
<COMMIT T2>
```

1. What are all of the possible values on disk for each of the database elements W, X, Y and Z?
2. Which, if any, transactions will need to be redone in the recovery process?
3. How would your answers to parts (1) and (2) change if <END CKPT> were not present in the log?

Question 5 **Points 5**

For each of the following schedules, indicate if the schedule is conflict-serializable or not. Justify your answer.

(a) r1 (A), w1 (B), r2 (B), w2 (C), r3 (C), w3 (A)
(b) w3 (A), r1 (A), w1 (B), r2 (B), w2 (C), r3 (C)
(c) r1 (A), r2 (A), w1 (B), w2 (B), r1 (B), r2 (B), w2 (C), w1 (D)
(d) r1 (A), r2 (A), r1 (B), r2 (B), r3 (A), r4 (B), w1 (A), w2 (B)

Question 6 **Points 10**

In the following sequences of events, we use Ri(X) to mean "transaction Ti starts, and its read set is the list of the database elements X." Also, Vi means "Ti attempts to validate," and Wi(X) means that "Ti finishes, and its write set was X." State what happens when each sequence is processed by a validation-based scheduler. In particular, for each Vj action, indicate if the validation is successful or not.

1. R1(B,C); R2(A,C); R3(C); V1; V2; V3; W1(A); W2(C); W3(B);
2. R1(B,C); R2(A,C); V1; W1(A); R3(A); V2; V3; W2(C); W3(B);

Question 7 **Points 15**

Two transactions are not interleaved in a schedule if every action of one transaction precedes every action of the other. For example, in the schedule r3(A), r1(A), r1(B), r3(B), r2(A), r2(B), transactions T1 and T2 are not interleaved, while transaction T1 and T3 are interleaved. A schedule is a serial schedule if no two transactions in the schedule are interleaved. Schedule S1 is conflict-equivalent to schedule S2 if S2 can be derived from S1 by a sequence of swaps of non-conflicting actions. For example, the schedule S1 = r1(A), r2(A), w2(A), w1(A), r2(B), w2(B) is conflict-equivalent to the schedule S2 = r2(A), r1(A), w2(A), r2(B), w1(A), w2(B), and S2 can be derived from S1 as shown below:

```
S1 = r1(A), r2(A), w2(A), w1(A), r2(B), w2(B); swap(r1(A),r2(A))
   = r2(A), r1(A), w2(A), w1(A), r2(B), w2(B); swap(w1(A), r2(B))
S2 = r2(A), r1(A), w2(A), r2(B), w1(A), w2(B)
```

Trivially, every schedule is conflict-equivalent to itself. A schedule is conflict-serializable if it is conflict-equivalent to some serial schedule. Give an example of a conflict-serializable schedule S which has all the following properties:

1. S involves transactions T1 and T2 and possibly others.
2. Transactions T1 and T2 are not interleaved in S, and all actions of T1 precede all actions of T2 in S.
3. In every serial schedule that is conflict-equivalent to S, all actions of transaction T2 precede all actions of transaction T1.

Question 8 **Points 20**

Prove or disprove each of the following statements. See Question 7 for definition of conflict-equivalence.

1. If two schedules are conflict equivalent, then their precedence graphs are identical.
2. If two schedules involve the same set of transactions, and have identical precedence graphs, then they are conflict equivalent.

Question 9 **Points 10**

Consider the locks required by two transactions T_1 and T_2.
T_1: $s(A), u(B), x(B)$
T_2: $u(A), x(A)$

where s, u, and x stand for shared, update, and exclusive locks respectively. Each transaction has to acquire its locks in the respective order shown.

(1) If T_1 and T_2 obey two-phase locking, then what is the number of legal sequences of the five lock steps above? Give the legal sequences in this case.

(2) Suppose T_1 and T_2 are not required to obey two-phase locking. However, they cannot acquire a lock on a database item X after having unlocked any lock they held previously on X. Then, what is the number of legal sequences of the five lock steps above? Give the legal sequences in this case.