CS 245
Final Exam – Winter 2009

This exam is open book and notes. You have 150 minutes (2 hours, 30 minutes) to complete it.

Print your name:_____

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

Signed:_____

Problem	Points	Maximum
1		10
2		10
3		10
4		10
5		10
6		10
7		10
8		10
Total		80

Problem 1 (10 points)

State if the following statements are true or false. Please write TRUE or FALSE in the space provided. (You will be penalized for incorrect answers, so leave the box blank if you don't know.)

(a) One key benefit of undo/redo logging is that it gives great flexibility when deciding when to flush modified database pages to disk.

ANSWER:_____

(b) Frequent checkpoints can reduce the amount of data lost if a disk is destroyed.

ANSWER:_____

(c) Keeping full histograms can result in more accurate estimates than just keeping a value count, but histograms requires more processing to maintain, so they can result in worse performance.

ANSWER:_____

(d) All schedules produced by a tree-based concurrency control scheme are two-phase.

ANSWER:_____

(e) In associating rule mining, an efficient strategy for computing k-sets (i.e., high support sets with k items) is to compute $k+1$-sets first.

ANSWER:_____

(f) Extensible hashing makes searching over an encrypted database very efficient.

ANSWER:_____

(g) Validation is most effective when it is rare for concurrent transactions to interact with the same data.

ANSWER:_____

(h) Logging logical actions may reduce the size of a log, as compared to logging the physical changes made to the database.

ANSWER:_____

(i) Timeouts are a good way of avoiding deadlock if the running time of a transaction can be predicted accurately.

ANSWER:_____

(j) Dense indexes are always first-level indexes.

ANSWER:_____

Problem 2 (10 points)

Suppose we have a relation $R(x,y,z)$ where the following holds:

- The pair of attributes x and y together form a key.
- Relation R holds 10,000 tuples.
- There are 1,000 distinct x values in R, specifically the values 1, 2, 3, ... 1000.
- There are 100 distinct values in Y, specifically 1, 2, 3, ..., 100.
- For each x value, there are on the average 10 y values, uniformly distributed between 1 and 100.
- For each y value, there are on the average 100 x values, uniformly distributed between 1 and 1000.

We are considering two types of multiple-key indexes for R. In either case, the lowest leaf level is dense, and the depth of the tree is such that there is a single root block. Assume that a block can hold ten key-pointer pairs.

1. **First x:** We first build an index for attribute x. A leaf pointer in this index for $x = k$ links to a y index for all records with $x = k$.
2. **First y:** We first build an index for attribute y. A leaf pointer in this index for $y = k$ links to an x index for all records with $y = k$.

(a) Suppose that we wish to minimize the number of IOs for this query:
SELECT z FROM R WHERE $x = 1$ AND $y \leq 10$.
Which of the two multiple-key indexes should we build? Assume that initially all index and data blocks are on disk.

First index is on (x or y):_____

Expected number of IOs for this index and query:_____

(b) Suppose now that we wish to minimize the number of IOs for this query:
SELECT z FROM R WHERE $x \leq 10$ AND $y = 1$.
Which of the two multiple-key indexes should we build? Again assume that initially all index and data blocks are on disk.

First index is on (x or y):_____

Expected number of IOs for this index and query:_____

(c) Consider the following query
SELECT z FROM R WHERE $x \leq 10$ AND $y \leq 10$
where we use the "First x" index. Determine the expected number of IOs to answer this query, assuming all data is initially on disk.

Number of IOs for First x:_____

Problem 3 (10 points)

Consider the following undo/redo log with checkpointing. The third and fourth values in entries that have 4 values are the old and the new value, respectively.

<START, T_1>
< T_1, A, 4, 5>
<START,T_2>
< T_2, B, 9, 10>
<START CKPT (T_1, T_2)>
<COMMIT T_2 >
<START T_3 >
< T_3, C, 14, 15>
<END CKPT>
<COMMIT T_3 >
<COMMIT T_1 >

(a) Suppose that there is a crash, and the last log entry to appear on disk is < T_2, B, 9, 10>. That is, only the first 4 log entries shown above actually appear in the log. The remaining 7 entries are never recorded on the log. Describe the actions of the recovery manager, including changes to both the disk and log.

Actions on disk : _____

Actions on log : _____

(b) Now suppose that the last entry made is <COMMIT T_2 >. That is, only the first 6 entries are actually recorded before the crash. Describe the actions at recovery time.

Actions on disk : _____

Actions on log : _____

(c) Same question except that now last recorded action is <END CKPT>.

Actions on disk : _____

Actions on log : _____

(d) Same question except that last recorded action is <COMMIT T_1 >.

Actions on disk : _____

Actions on log : _____

Problem 4 (10 points)

The following two transactions are run concurrently:
T_1: $r_1(A); w_1(B); c_1$
T_2: $w_2(A); r_2(B); c_2$

Depending on the scheduler, we may get various combinations of the actions of T_1 and T_2. One example of a resulting schedule could be

$S : r_1(A); w_1(B); w_2(A); r_2(B); c_1; c_2$

(a) List below all 10 possible schedules where T_2 *reads from* T_1. We have listed some of the actions in each schedule to simplify your job. You just need to fill in the missing actions. In each gap shown (underscore), there can be zero, one or two actions.

S_A : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_B : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_C : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_D : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_E : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_F : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_G : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_H : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_I : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

S_J : _____ $r_1(A)$; _____ $w_1(B)$; _____ ; $r_2(B)$; _____ c_2 _____

(b) How many of the schedules of part (a) are recoverable?

Number of schedules: _____

(c) How many of the schedules of part (a) are ACR?

Number of schedules: _____

(d) How many schedules of part (a) are strict?

Number of schedules: _____

(e) How many schedules of part (a) are conflict serializable?

Number of schedules: _____

Problem 5 (10 points)

We want to join the two relations $R(a, b)$ and $S(a, c)$ on a using a merge join. Suppose $B(R)$=10,000 and $T(R)$=50,000 while $B(S)$=5,000 and $V(S, a)$=1000. Assume a memory size of 101 blocks.

Except for part (d) below, assume that if you need to sort a relation, then you generate and store on disk the sorted relation.

(a) If both R and S are already sorted on a, how many disk IOs are needed for a merge join?

IOs: _____

(b) If R is already sorted on b and S is already sorted on c, how many disk IOs are needed for a merge join?

IOs: _____

(c) If R is already sorted on a and S is already sorted on c, how many disk IOs are needed for a merge join?

IOs: _____

(d) We can improve the result in (c) by not storing (materializing) sorted versions of relations, and just storing sorted buckets/segments. How many disk IOs are needed for the merge join if we implement this optimization?

IOs: _____

(e) Now suppose we have a clustering index on $S.a$. (With this structure, all the S records that share the same a value are contiguous on disk.) The disk IOs used for accessing the clustering index can be ignored. How many disk IOs are needed for joining R and S using the index?

IOs: _____

Problem 6 (10 points)

Consider a four-dimensional data cube R(A, B, C, D) with 10,000 cells. Each cell $R(v_1, v_2, v_3, v_4)$ of the cube is an integer. The aggregation operation we are considering is a sum and the following sum aggregates have been materialized:

- $R(*, B, C, D)$ with 200 cells
- $R(A, B, C, *)$ with 500 cells
- $R(A, B, *, *)$ with 250 cells

(a) How many distinct values does each dimension have?

Distinct A values: _____

Distinct B values: _____

Distinct C values: _____

Distinct D values: _____

For each of the following queries, choose a materialized aggregate cube from which you can compute the query using *minimum* number of sums. (Example: adding 1+2 takes 1 sum; adding 1+2+3 takes 2 sums.)

(b) $R(*, v_2, *, v_4)$

Materialized cube we should use: _____

Number of sums needed to answer query: _____

(c) $R(v_1, v_2, *, *)$

Materialized cube we should use: _____

Number of sums needed to answer query: _____

(d) $R(*, v_2, v_3, *)$

Materialized cube we should use: _____

Number of sums needed to answer query: _____

(e) $R(*, v_1, *, *)$

Materialized cube we should use: _____

Number of sums needed to answer query: _____

Problem 7 (10 points)

For each of the following schedules answer these two questions:

- Is the schedule conflict-serializable? If yes, describe all the conflict-equivalent serial schedules. (You do not need to explicitly write out the serial schedules, just describe them.) If no, explain why it is not.
- Could the schedule be created by the 2PL protocol? If yes, augment the schedule with lock (shared, exclusive) and unlock actions to show it can be generated by 2PL. If not, explain briefly why.

Here are the schedules:

(a) $S_1 : w_1(Y); w_2(X); r_3(X); r_1(Z); w_4(Z); r_4(X) r_2(Y);$

Is schedule conflict-serializable?: _____

Equivalent serial schedules, or explanation why not:

Creatable by 2PL?: _____

Augmented schedule or explanation why not:

(b) $S_2 : r_1(X); r_4(V); r_3(X); w_3(Z); w_5(V); w_2(Y); w_2(X); r_4(V); r_3(Y) r_1(Z);$

Is schedule conflict-serializable?: _____

Equivalent serial schedules, or explanation why not:

Creatable by 2PL?: _____

Augmented schedule or explanation why not:

Problem 8 (10 points)

(a) Consider a validation concurrency control mechanism. We know the following facts about two transactions U and T:

- `T` started reading at time 50;
- `T` successfully validated at time 100; at that time `U` is in `VAL`.
- `RS(T) = {A, B}` ;
- `WS(T) = {B, C}` ;

Assume that U finished at time 25. What objects *cannot* be in the read and write sets of U? If any object can be in the sets, write "All allowed."

OBJECTS THAT CANNOT BE IN RS(U):_____

OBJECTS THAT CANNOT BE IN WS(U):_____

(b) For the same scenario of part (a), assume that U starts reading at time 25 and finishes at time 75. What objects *cannot* be in the read and write sets of U? If any object can be in the sets, write "All allowed."

OBJECTS THAT CANNOT BE IN RS(U):_____

OBJECTS THAT CANNOT BE IN WS(U):_____

(c) For the same scenario of part (a), assume that U starts reading at time 60 and finishes at time 75. What objects *cannot* be in the read and write sets of U? If any object can be in the sets, write "All allowed."

OBJECTS THAT CANNOT BE IN RS(U):_____

OBJECTS THAT CANNOT BE IN WS(U):_____

(d) For the same scenario of part (a), assume that U finishes at time 125. What objects *cannot* be in the read and write sets of U? If any object can be in the sets, write "All allowed."

OBJECTS THAT CANNOT BE IN RS(U):_____

OBJECTS THAT CANNOT BE IN WS(U):_____