AUB American University of Beirut الجامعة الأميركية في بيروت

Faculty of Arts & Sciences
Department of Computer Science
CMPS 377—Internals of Database Systems
Spring 2004–2005

Final Exam

Date: Tuesday, June 7th, 2005 at 3:00 pm.
Duration: 120 minutes.
Instructor: Jihad Boulos

ID #: ------------------------------------

You are allowed to have your book and notes with you during this exam –although I highly advise you not to waist your time in searching in your documents for answers to the problems given here.

Your answers should be concise, and when possible should be a list of important points rather than prose. Solve as many problems as you can. I recommend that you start with problems you think the easiest. I also advise you to spend time on understanding what is being asked by each problem.

Beware, wordy and/or irrelevant answers might reduce your score for that problem. Your answers should be the summary of work done on scratch paper that you do not hand in. Also, do not expect that I will spend much time trying to decipher your hand writing. I will give a ZERO to **illegible** answers. The space allocated for answers should be sufficient. If not, use the back sides of the pages.

Wish you good work.

	Prob. 1	Prob. 2	Prob. 3	Prob. 4	Prob. 5	Prob. 6	Prob. 7
Max Grade	11	10	10	30	8	15	16
Your Grade							

Exercise I (11 points): Short Questions

1. ☐ Relation *R* has 4000 records. Suppose a clustered B+ tree index is built on R's key attribute. A block can hold a B+ tree node (interior or leaf node) with 20 keys and 21 pointers. In this case, at least 220 blocks are needed to hold the entire B+ tree index. (TRUE or FALSE)

2. ☐ Consider an extensible hash table where the hash function generates a 20-bit hash key, but only the high order 8 bits are used. In this case, the hash table can have at most 220-8 data blocks (containing records). (TRUE or FALSE)

3. ☐ If a schedule is conflict serializable then it must be serializable. (TRUE or FALSE)

4. ☐ It's possible to avoid deadlocks by assigning priorities to transactions, so that a higher priority transaction always gets the lock that it requests. (TRUE or FALSE)

5. ☐ Optimistic concurrency control completely avoids the need for any locking or synchronization. (TRUE or FALSE)

6. ☐ Any schedule produced by a lock scheduler using shared and exclusive locks is conflict serializable. (TRUE or FALSE)

7. ☐ No deadlocks can occur under 2PL. (TRUE or FALSE)

8. For which WHERE clauses might a hash index on attributes <a, b> be useful? Circle the correct WHERE clause (s).

   (a < X and b = Y) (a = X and b = Y) (a < X and b < Y) (a = X) (b = Y)

9. Which buffer manager algorithm would be best to use for the inner relation during Page-Oriented Nested Loops Join? Assume the inner relation is large. Circle your answer.

   LRU MRU CLOCK

10. Which transactional goal(s) does concurrency control ensure? (Circle your answer(s).)

    ATOMICITY CONSISTENCY ISOLATION DURABILITY

11. Which transactional goal(s) does recovery protocols ensure? (Circle your answer(s).)

    ATOMICITY CONSISTENCY ISOLATION DURABILITY

Exercise II (10 points):

Consider the following schema:

Students(StId, StName)
Courses(CId, CName)
Took(StId, CId)

Write the following queries in relational algebra:

a. Find the names of students who have taken CMPS256 and CMPS257.
b. Find the names of students who have taken CMPS256 or CMPS257 but not both.

Exercise III (10 points):

Which of the following queries might produce different results when evaluated using set semantics and bag semantics, even if the relations *R* and *S* are themselves sets (i.e., they contain no duplicate tuples)? Briefly justify your answer.

a. $R - S$
b. $(R \bowtie S) - (S \bowtie R)$
c. $R \cup S$
d. $\sigma_{a=5}(R)$

Exercise IV (30 points):

Consider the three relations R(A, B, D, E), S(A, C, F, G), and T(B, C, H, I), and the following query:

```
SELECT *
FROM R, S, T
WHERE R.A = S.A
      AND R.B = T.B
      AND T.C = S.C
      AND R.D = 100
      AND S.F = 200
```

1. Draw six Query Execution Plans with the following conditions:

a. They contain no cartesian product.
b. All selections have been pushed down.
c. Each join is associated with only one join predicate.

Assume the following statistics for the relations:
• R has 10,000 tuples, an index on D, V(D,R)=100, V(A,R)=10,000, V(B,R)=10,000.
• S has 20,000 tuples, an index on F, V(F,S)=100, V(A,S)=10,000, and V(C,S)=20,000
• T has 30,000 tuples, an index on B, with V(B,T)=10,000, V(C,T)=20,000.

Assume disk and memory page size is 4KB, the page header size is 48 bytes, each indexed attribute occupies 4 bytes and the tuple size of each relation is 200 bytes.
Assume also that there is one 1 MB of main memory as buffers.
Hint: You should use the main memory to avoid copying intermediate results to the disk, if there is enough space in the main memory.

2. Estimate the result size of this query. State any assumption you make and show as much analysis as you deem necessary.

3. Find the optimal physical plan, i.e., an algebraic expression where selections and projections are annotated with execution operators, in the following two cases:

a. The execution primitives are *Scan* and *Index* for the selection operator, and *Nested-Loops*, *Indexed-Nested-Loops*, and *Sort-Merge* for joins.
b. The execution primitives are *Scan* and *Index* for the selection operator, and *Nested-Loops*, *Indexed-Nested-Loops, Sort-Merge* and *Hash-Merge* for joins.

Compute an estimation of the optimal plan's cost (in terms in IO) in these two cases. State any assumption you make.

Exercise V (8 points):

Draw the precedence graph for the following schedules of reads and writes. For each schedule, is it conflict serializable?

a. S1 = R1(A), W2(A), W1(B), R2(A), W3(C), R2(C), W3(B)

b. S2 = R1(A), W3(B), R1(B), W2(A), W2(C), R3(C)

Exercise VI (15 points)

Assume we have a DBMS system that only has exclusive locking (L and U are for Locking and Unlocking, respectively). Consider the following pairs of transactions.
Mark a 'Y' in the table for each property that applies, and a 'N' for those that do not.

(a) T1 = L(A), W(A), R(A), L(B), L(C), R(B), W(B), U(C), U(A), U(B) **Commit**
T2 = L(B), W(B), L(C), L(A), R(A), W(A), U(B), U(C), **Commit**, U(A)

(b) T1 = L(A), W(A), R(A), L(C), R(C), **Commit**, U(C), U(A)
T2 = L(A), W(A), L(C), L(B), W(B), R(A), W(C), **Commit**, U(A), U(B), U(C)

(c) T1 = R(A), R(B), R(A), W(C), **Commit**
T2 = R(C), W(A), **Commit**

	Two Phase Locking	Strict Two-Phase Locking	May results in cascading aborts	May results in deadlock	May results in unrepeatable reads
(a)					
(b)					
(c)					

Exercise VII (16 points)

Examine the schedule given below. There are three transactions, T1, T2, and T3. Initially, the salary = 1 and the tax = 2. The assignments happen within the local memory space of the transactions and the effects of these assignments are not reflected in the database until the WRITE operation.

```
      T1                      T2                      T3
--------------------------------------------------------------------
0                                                     start
1                                                     READ tax
2                                                     tax := tax + 1
3     start
4     READ salary
5     salary := salary + 1
6                                                     WRITE tax
7                                                     commit
8                             start
9                             READ tax
10                            READ salary
11                            tax := tax + salary
12                            WRITE tax
13                            commit
14 ------                  checkpoint start
15    READ tax
16    tax := tax + salary
17    WRITE salary
18 ------------------- checkpoint end -------------------------
19    commit
```


a) Show the undo/redo log file entries that would be generated by this execution. Use the same notation used in class. For each log entry, indicate what line above generates it.

b) Assume that the undo/redo recovery algorithm with checkpoints is being used. The database crashes immediately after statement 7. (Assume that all the log records up to this point have been flushed to disk.)

b.1) Which transactions would have to be undone?

b.2) Which transactions would have to be redone?

c) Again assume that the undo/redo recovery algorithm with checkpoints is being used, but now the databases crashes just after statement 18. (Assume that all the log records up to this point have been flushed to disk.)

c.1) Which transactions would have to be undone?

c.2) Which transactions would have to be redone?