

Database Systems, CSCI 4380-01 Exam #3  
Monday May 9, 2010 at 2 pm

1 (20)	2 (20)	3 (20)	4 (20)	5 (20)	6 (20)	TOTAL (100)

**Note.** The exam is open book and open notes. Use your own book and notes only, sharing is not allowed. Electronic gadgets are NOT allowed during the exam. Write your answers clearly, legibly and explain your reasoning as much as you can. If I cannot read or understand your answers, you will not get points.

There are 6 questions in this exam. Answer any 5 of them. If you answer all 6, I will only read the first 5 answers. Mark clearly which question you do not want me to grade.

In SQL queries in Question 5, do not use views (i.e. using CREATE VIEW statements), triggers and other procedural elements. Make sure you use DISTINCT only when you have to.

**Question 1 (20 points).** You are given the following schedule:

$r_1(X)$   $r_1(Z)$   $r_2(Z)$   $w_2(Z)$   $r_3(W)$   $w_3(X)$   $w_1(Y)$   $r_4(K)$   $r_4(W)$   $w_4(Z)$   $w_4(Y)$   
 $commit_1$   $commit_2$   $commit_3$   $commit_4$

- (a) List all conflicts in this schedule and draw the conflict diagram. Is this schedule is serializable? Why or why not? If this schedule is serializable, then find a serial schedule that is equal to this schedule.
- (b) Is this schedule possible under the two phase locking scheme if all items are locked with a single type of lock? Explain why or why not in detail.
- (c) True or false: If a schedule is serializable and we are using a single type of locks, no other transaction can read the data written by another transaction X before X commits. Explain why or why not.

**Question 2 (20 points each).** Answer the following questions:

- (a) In order to avoid reading the log completely from the beginning during recovery, how can we modify the log? Explain briefly.

- (b) Suppose you are using the following protocol for transaction management:

When a transaction wants to commit, a commit record is written. But, the transaction is not yet allowed to commit.

The transaction manager periodically flushes the log to disk completely, then writes all the changed data pages to disk. When both of the write operations complete, all transactions that had written a commit log record are allowed to commit.

Under this protocol, if a crash occurs, is there a need to do REDO and/or UNDO? Explain why.

- (c) If you are using REDO/UNDO recovery, and the following are the contents of the log and the disk after crash, which log entries should be redone and which log entries should be undone? Explain why briefly.

LOG:	LSN	Entry	DATA:	Data Page	LSN of Last recorded log entry
	10	T1 update P1		P1	10
	11	T2 update P2		P2	5
	12	T2 update P3		P3	12
	13	T3 update P4		P4	13
	14	T1 update P5		P5	6
	15	T1 commit			

**Question 3 (20 points).** You are given the following statistics for `Students(id, gender, class)` with 500 tuples and `Transcript(studid, crn, semester, year, grade)` with 10,000 tuples:

Attribute	Values	Min	Max
Students.id	500	66001234	66119999
Students.gender	2	'F'	'M'
Students.class	5	1	5
Transcript.studid	480	66001234	66119999
Transcript.crn	50	12345	23456
Transcript.semester	3	1	3
Transcript.year	8	2004	2011
Transcript.grade	10	'A'	'W'

Q1.

```
SELECT
    *
FROM
    transcript
WHERE
    semester = 1
    AND year = 2010 ;
```

Q2.

```
SELECT
    *
FROM
    students
WHERE
    gender = 'F'
    AND class = 2;
```

Q3.

```
SELECT
    *
FROM
    students S
    , transcript T
WHERE
    S.id = T.studId
    and T.grade = 'A'
```

- Find the total number of expected tuples in the above queries Q1, Q2 and Q3 (write your answer next to each query above).
- What is the cost of answering query Q2 above using an index on *Student(class)* with height 1 (root and leaf level), and 40 leaf nodes?

**Question 4 (20 points).** Answer the following given  $PAGES(R) = 300$ ,  $PAGES(S) = 1,000$  and index on  $R(A)$  with height 1 (root and leaf) and 30 nodes at the leaf level.

- (a) What is the cost of sorting relation  $R$  with  $M = 10$  pages?
- (b) What is the cost of joining  $R \bowtie S$  using block-nested-loop join with  $M = 51$  pages ( $R$  is the outer relation and  $S$  is the inner relation)?
- (c) What is the cost of the query plan given in the figure below (index look up on  $R.A$  followed by a block nested loop join)? Note: Yes, you really have all the statistics you need for this question.

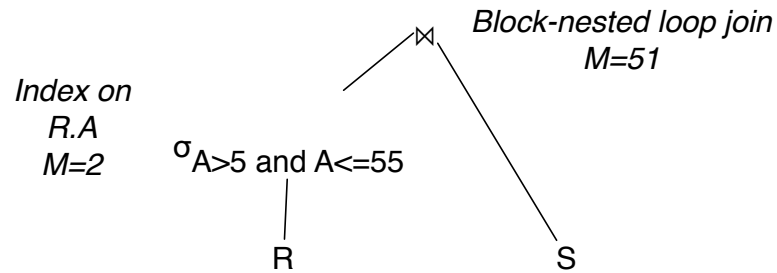


Figure 1: The query plan for question 4

**Data Model.**

Country(id, name)

Person(id, name, gender, birthday, birthCountryId, fatherId, motherId)

CitizenOf(personId, countryId)

In Person, fatherId and motherId are foreign keys to Person(id), but they can also be null if the values are not known. The attribute birthCountryId is a foreign key to Country.

In CitizenOf, personId is foreign key to person and countryId is a foreign key to Country. This relation stores which countries a person is a citizen of (some people may be dual or triple citizens).

**Question 5 (20 points).** Suppose you are given the data model above. Write the following queries using SQL:

- (a) Find people who have a mother and father with citizenship from completely different countries (if mother and father are known). Return their id of the found people.
- (b) Find people who have dual citizenship (exactly 2 countries, no more), and are born in a country different than either of their current countries of citizenship. Return their id.

**Question 6 (20 points).** Using the data model from Question 5, write a pl/pgsql procedure (or a procedure in pseudo-code that combines SQL queries with procedural code) for the following: Given the id of a person, find which generation of US citizen he/she is.

If she is not a US citizen, then she belongs to generation 0.

If she is a US citizen, but neither of her parents are US citizens, then she is of generation 1.

Similarly, any person who is a US citizen with parents belonging to generation  $i$  belongs to generation  $i+1$ .

If the parents of a person are not known, assume they are not US citizens.

For example, if the person is a US citizen and her parents are US citizens, but none of her grand parents are US citizens, then she is generation 2.

**Hint.** Finding which generation everyone is in first may make things much easier.



Blank page for answers