

Assignment 6: Query Optimization, Transaction Management,
& Concurrency Control

Deadline: Jan 4 (Wednesday), 2006

This is an individual assignment, that is, no group submissions are allowed.

Cheating Policy: If you are caught cheating, your grade is 0.

Late Policy: You may hand in your late assignment at TA's office hour on Thu (1/5/2006) for 80% of original grade, or at TA's office hour on Fri (1/6/2006) for 70%. We will not accept any assignment submissions after Fri.

Questions

1. Let relations $r1(A,B,C)$ and $r2(C,D,E)$ have the following properties: $r1$ has 20,000 tuples, $r2$ has 45,000 tuples, 25 tuples of $r1$ fit on one page, and 30 tuples of $r2$ fit on one page. The buffer size is 102 pages.

Estimate the number of page accesses required and explain in detail how you obtain it, using each of the following join strategies:

- a. Simple Nested Loop Join (Use $r1$ as outer relation.)
- b. Block Nested Loop Join (Use $r1$ as outer relation.)
- c. Hash-Join (NO overflow occurs)

2. Consider the following two transactions:

```
T1: read(A);
    read(B);
    if A = 0 then B := B + 1;
    write(B);
T2: read(B);
    read(A);
    if B = 0 then A := A + 1;
    write(A);
```

Let the $A = B = 0$ the initial values.

- a. Show a concurrent execution of $T1$ and $T2$ that produces a non-serializable schedule.
- b. Is there a concurrent execution of $T1$ and $T2$ that produces a serializable schedule? Why? (Hint: For a serializable schedule, it must produce identical result as some serial schedule.)

Consider the following three instructions:

```
S(O): request for a shared lock of object O.
X(O): request for a exclusive lock of object O.
release(O): release all held locks of object O.
```

- c. Add $S(O)$, $X(O)$, and $release(O)$ instructions to transactions $T1$ and $T2$, so that they observe the two-phase locking protocol.
- d. Can the execution of these transactions result in a deadlock? Why?

Submission

Hand in PAPER PRINTOUT that contains your answers to the three questions. Please include your name and ID.