CIS560 – Database System Concepts **Name:_______________**

Homework Assignment 7 [20 points] – due November 1st (by midnight)

1. [3 points] Consider a database with objects X and Y and assume that there are two transactions T1 and T2. Transaction T1 reads objects X and Y and then writes X. Transaction T2 reads objects X and Y and then writes objects X and Y. Give three examples of schedules for the transactions T1, T2 such that:

   i. Your first schedule should contain a write-read conflict.

   ii. Your second schedule should contain a write-write conflict.

   iii. Your third schedule should contain a read-write conflict.

   In each case your schedule may contain additional conflicts, but should contain at least one conflict of the type indicated. (In particular you may give a single schedule, which illustrates all three conflicts!) In each case, indicate the conflict of the type you are illustrating.

2. [5 points] For the following schedules:

   (a) $r_2(A)$; $r_1(C)$; $r_2(B)$; $w_2(B)$; $r_3(B)$; $r_1(A)$; $r_3(C)$; $w_3(C)$; $w_1(A)$

   (b) $r_2(A)$; $r_1(C)$; $r_2(B)$; $r_3(B)$; $w_2(B)$; $r_1(A)$; $r_3(C)$; $w_3(C)$; $w_1(A)$

   Answer the following questions:

   i. What is the precedence graph for the schedule?

   ii. Is the schedule conflict-serializable? If so, what are the equivalent serial schedules?

3. [6 points]

i. Consider the following two transactions and schedule (time goes from top to bottom). Is this schedule conflict-serializable? Explain why or why not.

T0	T1
$R_0(A)$	
$W_0(A)$	
	$R_1(A)$
	$R_1(B)$
	c_1
$R_0(B)$	
$W_0(B)$	
c_0	

ii. Show how 2PL can ensure a conflict-serializable schedule for the transactions above. Use the notation $L_i(A)$ to indicate that transaction i acquires the lock on element A and $U_i(A)$ to indicate that transaction i releases its lock on A.

iii. Show how the use of locks without 2PL can lead to a schedule that is NOT conflict-serializable.

4. [6 points] The following schedules are presented to a timestamp-based scheduler. Assume that the read and write timestamps of each element start at 0 (RT(X) = WT(X) = 0), and the commit bits for each element are set (C(X)=1). Explain what happens as each schedule executes.

   a. st_1, st_2, st_3, $r_1(A)$, $r_2(B)$, $w_1(C)$, $r_3(B)$, $r_3(C)$, $w_2(B)$, $w_3(A)$

   b. st_1, st_2, $r_1(A)$, $r_2(B)$, $w_2(A)$, com_2, $w_1(B)$