CMPT 454 (Spring 2010)
Assignment 1: Transaction Management

Due: Friday, 2010-01-29, 11:29AM (at the beginning of class)

Instructions on Assignment Submission: Hard copy only! Bring your assignments to the class on the due day, and TA will collect the assignments **at the beginning of class**. Please write legibly or typeset your answers using your favorite word processor. Late assignments will not be accepted unless there is a documented medical reason.

Problem 1.

(25 points) In the following table you are given the contents of the transaction log and the state of (a part of) the database after several system crashes. For each case, indicate what type of logging might have been used.

A	B	Log
6	0	<START T>,<T,A,5>,<COMMIT T>
6	6	<START T>,<T,A,6>,<T,B,6>,<COMMIT T>
6	5	<START T>,<T,A,6>,<T,B,6>,<COMMIT T>
5	5	<START T>,<T,A,6>,<T,B,6>
5	5	<START T>,<T,A,6>,<T,B,6>,<ABORT T>

Your answer, for each case, should be one of the following: (1) Undo; (2) Redo; (3) Either; (4) Neither. Moreover, briefly explain you reasons.

Problem 2.

(25 points) Consider the following transaction log from the start of the execution of a database system that is capable of running undo/redo logging with non-quiescent checkpointing:

(1) < START T_1 >
(2) <T_1, A, 50, 25>
(3) <T_1, B, 250, 25>
(4) <START T_2>
(5) <T_1, A, 75, 50>
(6) <T_2, C, 35, 25>
(7) <COMMIT T_1>
(8) <START T_3>
(9) <T_3, E, 55, 25>
(10) <T_2, D, 45, 25>
(11) <START CKPT (T_2,T_3)>
(12) <T_2, C, 65, 35>
(13) <COMMIT T_2>
(14) <START T_4>
(15) <T_4, F, 100, 25>
(16) <COMMIT T_3>
(17) <END CKPT>
(18) <T_4, F, 150, 100>
(19) <COMMIT T_4>

Suppose the log entries are in the format $< Tid, Variable, Newvalue, Oldvalue >$. What is the value of the data items A, B, C, D, E, and F on disk after recovery:

(1) if the system crashes just before line 10 is written to disk?

(2) if the system crashes just before line 13 is written to disk?

(3) if the system crashes just before line 14 is written to disk?

(4) if the system crashes just before line 19 is written to disk?

(5) if the system crashes just after line 19 is written to disk?

Problem 3.

(25 points) For each of the following schedules, state whether they are serializable and whether they are conflict-serializable. For each schedule, draw the corresponding precedence graph. If the schedule is conflict-serializable, show **all** the conflict-equivalent serial schedules. If the schedule is not serializable, provide an initial DB state and some semantics (pseudo-code) for the transactions for which no serial schedule has the same net effect.

(1) $S_1 : r_1(A); w_1(B); r_2(B); w_2(C); r_3(C); w_3(A);$

(2) $S_2 : r_1(A); r_2(A); r_1(B); r_2(B); r_3(A); r_4(B); w_1(A); w_2(B);$

Problem 4.

(25 points) Consider the following two transactions:

$$T_1 = w_1(C); r_1(A); w_1(A); r_1(B); w_1(B);$$
$$T_2 = r_2(B); w_2(B); r_2(A); w_2(A);$$

Suppose that the scheduler only performs exclusive locking (i.e., no shared locks). For each of the following three instances of transactions T_1 and T_2 annotated with lock and unlock actions, decide whether the annotated transactions:

(1) obey two-phase locking;

(2) will necessarily result in a serial schedule (if no deadlock occurs);

(3) will necessarily result in a conflict-serializable schedule (if no deadlock occurs);

(4) will necessarily result in a recoverable schedule (if no deadlock occurs);

(5) will necessarily result in a schedule that avoids cascading rollback (if no deadlock occurs);

(6) will necessarily result in a strict schedule (if no deadlock occurs);

(7) may result in a deadlock.

(a)	$T_1 = l_1(C); w_1(C); l_1(A); r_1(A); w_1(A); l_1(B); r_1(B); w_1(B); Commit; u_1(A); u_1(C); u_1(B);$ $T_2 = l_2(B); r_2(B); w_2(B); l_2(A); r_2(A); w_2(A); Commit; u_2(A); u_2(B);$
(b)	$T_1 = l_1(B); l_1(C); w_1(C); l_1(A); r_1(A); w_1(A); r_1(B); w_1(B); Commit; u_1(A); u_1(C); u_1(B);$ $T_2 = l_2(B); r_2(B); w_2(B); l_2(A); r_2(A); w_2(A); Commit; u_2(A); u_2(B);$
(c)	$T_1 = l_1(C); l_1(A); w_1(C); r_1(A); w_1(A); l_1(B); r_1(B); w_1(B); u_1(A); u_1(C); u_1(B); Commit;$ $T_2 = l_2(B); r_2(B); w_2(B); l_2(A); r_2(A); w_2(A); Commit; u_2(A); u_2(B);$

Format your answers in a table with Yes/No entries (a table as following).

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(a)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
(b)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
(c)	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N