RollNo: _____ Name: _____

National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Advanced Database Concepts**	**Course Code:**	**CS451**
	Program:	**BS(Computer Science)**	**Semester:**	**Spring 2018**
	Duration:	**60 Minutes**	**Total Marks:**	**34**
	Paper Date:	**27-Feb-18**	**Weight**	**12.5%**
	Section:	**CS**	**Page(s):**	**5**
	Exam:	**Midterm-I**		

Instruction/Notes: Scratch sheet can be used for rough work however, all the questions and steps are to be shown on question paper. ***No extra/rough sheets should be submitted with question paper***.
You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements.

Q1. *(3 points)* Given these transactions, find a cascade-free but not strict schedule, if possible.
T1: r1(A), r1(B), w1(B), w1(A), c1.
T2: r2(B), w2(B), c2.
T3: w3(B), c3.

Ans:
r1(A), r1(B), r2(B), w3(B), w1(B), w2(B), w1(A), c1, c2, c3.

Q2. *(6 points)*
a) Describe how fuzzy checkpointing is used in ARIES.
b) What is the difference between the UNDO/REDO and the UNDO/NO-REDO algorithms for recovery with immediate update?
c) Differentiate between non-repeatable read and phantom problems

Ans:
Get from text book.

RollNo: _____ Name: _____

Q3. *(6 points)* Consider the following classes of schedules: conflict-serializable, view-serializable, recoverable, cascadeless, and strict. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs in a certain class based on the listed actions, explain briefly. The actions are listed in the order they are scheduled. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions. Also draw precedence graph for each schedule.

a) S1: r2(X), w3(X), c3, w1(Y), r2(Y), r2(Z), c2, r1(Z), c1.
b) S2: r2(X), w3(X), w1(Y), r2(Y), w2(Z)

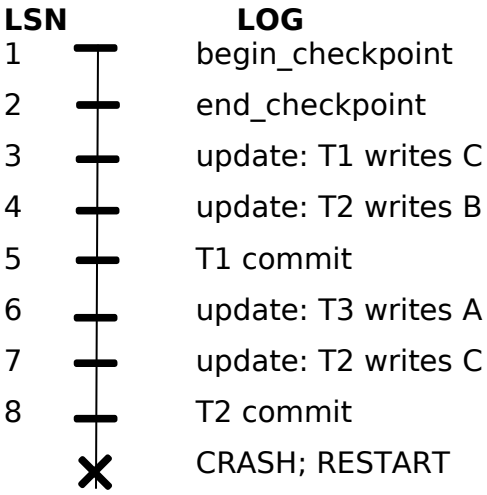
Ans:
a) S1: r2(X), w3(X), c3, w1(Y), r2(Y), r2(Z), c2, r1(Z), c1.
S1: It is conflict-serializable, view serializable, not strict , not cascadeless, non-recoverable as T2 read the dirty value of Y which is updated by T1 and T2 commit before T1. Equivalent serial schedule is T1→(Y)T2→(X)T3.

b) S2: r2(X), w3(X), w1(Y), r2(Y), w2(Z)
S2: It is conflict-serializable, view serializable, not strict , not cascadeless. We cannot decide whether it's recoverable or not, since the abort/commit sequence of these two transactions (i.e. T1 & T2 where T2 read dirty value of Y which is updated by T1) are not specified. Equivalent serial schedule is T1→T2→T3.

RollNo: _____ Name: _____

Q4. *(1+4+1+1= 7 points)* Consider the execution shown in below Figure. Assume that the Dirty Page Table and Transaction Table were empty before the start of the log.

a) What is the value of the LSN stored in the master log record?
b) What is done during Analysis? (Be precise about the points at which Analysis begins and ends and show the contents of Dirty Page Table and Transaction Table constructed in this phase.)
c) What is done during Redo? (Be precise about the points at which Redo begins and ends.)
d) What is done during Undo? (Be precise about the points at which Undo begins and ends.)



Ans:
a) LSN = 1
b) Analysis phase starts from LSN=2 record until it reaches the end i.e. LSN=8. The end checkpoint contains the empty transaction table and dirty page table. The analysis phase will further reconstruct these tables as follows:

Transaction Table (TT)

TID	Last_LSN	Status
T1	~~3~~ 5	~~inProcess~~ commit
T2	~~4~~ ~~7~~ 8	~~inProcess~~ commit
T3	6	inProcess

Dirty Page Table (DPT)

Page_ID	LSN
C	3
B	4
A	6

c) REDO begins from LSN=3 (smallest LSN in DPT) and proceed with the REDO of updates. The LSNs {3, 4, 6, 7} corresponding to the updates for pages C, B, A, and C, respectively, are not less than the LSNs of pages in initial DPT. So those data pages will be read again and all the updates reapplied from the LOG. At this point REDO phase is finished and the UNDO phase starts.
d) UNDO is applied only to the active transaction T3. UNDO phase starts at LSN=6 (the last update of active transaction T3) and proceeds backward in the LOG. The backward chain of updates for T3 (only LSN=6 record) is followed and undone.

RollNo: _____ Name: _____

Q5. *(12 points)* For the schedule S: r2(X), w3(X), c3, w1(Y), r2(Y), r2(Z), c2, r1(Z), c1. Show that the schedule S will be accepted/rejected in exactly the order shown by the below protocols. Provide proper reason and show your working.

a) Basic 2PL (add locks to the transactions)
b) Basic Timestamp Ordering (Assume T1 < T2 < T3)
c) Strict Timestamp Ordering (Assume T1 < T2 < T3)
d) Optimistic Concurrency Control

Ans:

a) Basic 2PL (Reject);
Here xl3(X) cannot acquire due to sl2(X).
1- ABORT T3 for Deadlock Avoidance using wait-die scheme
2- WAIT T3 for sl2(X) to release for Deadlock Avoidance using wound-wait scheme
3- WAIT T3 for sl2(X) to release for Deadlock Detection (use wait-for-graph, in case of deadlock)

b) Basic TO (ACCEPT);
- w3(X) allow as T3 is younger than RTS of X (i.e. TS(T3) > RTS(X))
- r2(Y) allow as T2 is younger than WTS of Y (i.e. TS(T2) > WTS(Y))

c) Strict TO (REJECT);
- w3(X) allow as T3 is younger than RTS of X (i.e. TS(T3) > RTS(X))
- r2(Y) delay until c1/a1; as T2 is younger than WTS of Y (i.e. TS(T2) > WTS(Y))

d) **Optimistic(REJECT);**
- T3 Forward Validation fails due to item X conflict with T2; so it may be delay or abort T3/T2.
- But the validation of T1 & T2 is successful, if T3 abort.

RollNo: _____ Name: _____