National University of Computer and Emerging Sciences
Lahore Campus

Advance Database Concepts (CS4064)

Date: Sat, 01 Mar 2025

Course Instructor(s)

Muhammad Ishaq Raza, Muhammad Naveed

Sessional-1 Exam

Total Time (Hrs.):	**1**
Total Marks:	**30**
Total Questions:	**3**

____________ ____________ ________________________

Roll No Section Student Signature

Note: Please make sure to attempt all questions and their respective parts in the specified order. Failure to do so may result in a deduction of one mark for each incorrect part of a question.

SOLUTION

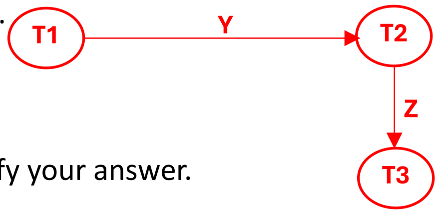
CLO # 1: *Understanding advance data models, technologies, and approaches for building database systems.*

Q. No 1: Consider the following schedule of action: [4+2+2= 8]

S: r1(X), w2(Z), r3(X), r1(Y), r3(Z), w1(Y), r2(Y).

a. Draw the serializability (precedence) graph for this schedule. State whether this schedule is conflict-serializable (correct) or not. If the schedule is conflict-serializable, write down the equivalent serial schedule(s) otherwise explain why it is not.

It is conflict-serializable as there is no-cycle in the graph.
Equivalent serail schedule is T1→T2→T3.



b. State whether this schedule is view-serializable or not. Justify your answer.

It is also view-serializable as it is conflict-serializable.

c. Describe the conflict equivalence of two schedules and a serializable schedule.

Two schedules are said to be conflict-equivalent if the relevant order of any two conflicting-operations is the same in both schedules.

Schedule (non-serial) S is said to be serializable if it is (conflict) equivalent to some serial schedule S'.

CLO # 1: *Understanding advance data models, technologies, and approaches for building database systems.*

Q. No 2: Consider the following schedule of action: [15]

S: r1(X), w2(Z), r3(X), r1(Y), r3(Z), c3, w1(Y), r2(Y), c1, c2.

For each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the schedule. Assume that the timestamp of transaction *Ti* is *i*. For lock-based concurrency control mechanisms, add lock and unlock requests to the above schedule of actions as per the locking protocol. The DBMS processes actions in the order shown. If a transaction is blocked, assume that all its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed schedule) of an unblocked transaction.

a. Rigorous 2PL with protocol based on a timestamp for deadlock avoidance (Use wound-wait policy)

T_1	T_2	T_3
s1-lock(X) r1(X)		
	x2-lock(Z) w2(Z)	
		s3-lock(X) r3(X)
s1-lock(Y) r1(Y)		
		s3-lock(Z) ...**wait** for T2
x1-lock(Y) ...**upgrade-lock** w1(Y)		
	S2-lock(Y) ...**wait** for T1	
c1, release all locks. unlock1(X) unlock1(Y)		
	r2(Y) ...**wake-up** **c2**, release all locks. unlock2(Z) unlock2(Y)	
		r3(Z) ...**wake-up** **c3**, release all locks. unlock3(X) unlock3(Z)

b. Strict Timestamp Ordering (Assume T1 < T2 < T3)

T1	T2	T3	X		Y		Z	
			RTS	*WTS*	*RTS*	*WTS*	*RTS*	*WTS*
			T0	{ }	**T0**	{ }	**T0**	{ }
r1(X)			{T1}					
	w2(Z)							T2
		r3(X)	{T3}					
r1(Y)					{T1}			
		r3(Z) …wait for T2						
w1(Y)						T1		
	r2(Y) …wait for T1							
c1								
	r2(Y) …**wake-up**				{T2}			
	c3							
		r3(Z) …**wake-up**					{T3}	
		c3						

c. Optimistic concurrency control technique (Use defer the validation until a later time when the conflicting transactions have finished.)

T3 Validation: Successful

Backward: True (no overlapping transaction)
Forward: True (no write-set of T3 i.e. transaction being validated)

T1 Validation: Fail and Deferred until T2 end

Backward: True (no write-set of T3 i.e. committed overlapping transaction)
Forward: False; WS(T1) ∩ RS(T2) = {Y} ∩ {Y} = {Y}

T2 Validation: Successful

Backward: True (no write-set of T3 i.e. committed overlapping transaction)
Forward: True; WS(T2) ∩ RS(T1) = {Z} ∩ {X, Y} = Ø

T1 Re-validation: Successful

Backward: True; RS(T1) ∩ (WS(T2) ∪ WS(T3)) = {X, Y} ∩ ({Z} ∪ {Ø}) = Ø
Forward: True (no overlapping active transaction)

National University of Computer and Emerging Sciences
Lahore Campus

CLO # 1: *Understanding advance data models, technologies, and approaches for building database systems.*

Q. No 3: Consider the following log at the point of system crash. Suppose that we use ARIES recovery algorithm to answer the following questions. [1+2+1+1+2= 7]

LSN	Last_LSN	Trans_ID	Type	Page_ID	Other_Info
1	0	T1	Update	W	...
2	0	T2	Update	X	...
3	1	T1	Update	Y	...
4	0	T3	Update	X	...
5	2	T2	Commit		
6	begin checkpoint				
7	end checkpoint				...
8	4	T3	Update	Y	...
9	3	T1	Update	Z	...
10	0	T4	Update	W	...
11	9	T1	Commit		...

a. Show the contents of transaction and dirty page table at the time of checkpoint.

Transection Table

Trans_ID	LSN	Status
T1	**3**	**in-progress**
T2	**5**	**Commit**
T3	**4**	**in-progress**

Dirty Page Table

Page_ID	LSN
W	**1**
X	**2**
Y	**3**

b. What is done during Analysis? Be precise about the points at which Analysis begins and ends and show the contents of transaction and dirty page table reconstructed in this phase.

Analysis Phase: Start from LSN# 6 *(i.e. begin checkpoint)* till end of log file (i.e. LSN# 11).

Transection Table

Trans_ID	LSN	Status
T1	**11**	**Commit**
T2	**5**	**Commit**
T3	**8**	**in-progress**
T4	**10**	**in-progress**

Dirty Page Table

Page_ID	LSN
W	**1**
X	**2**
Y	**3**
Z	**9**

c. What is done during Redo? Be precise about the points at which Redo begins and ends.

Redo Phase: Start from LSN# 1 *(i.e. smallest LSN in DPT)* till last update-operation (i.e. LSN# 10).

LSNs (1,2,3,4,8,9,10) update the corresponding pages (W, X, Y, X, Y, Z, W).

d. What is done during Undo? Be precise about the points at which Undo begins and ends.

Undo Phase: Start from LSN# 10 *(i.e. last update-operation of active transaction)* till first update-operation of active transaction (i.e. LSN# 4).

Undo will perform on T4 and T3 transactions.

T4 undo operations: LSN# 10 (W) and T3 undo operations: LSN# 8, 4 (Y, X).

e. What do the terms steal/no-steal and force/no-force mean with regard to buffer management for transaction processing?

See Textbook DB Recovery Chapter.