Database Management System: Assignment 7

Total Marks : 100

September 26, 2022

Question 1

Consider the following schedule S involving five transactions T_1, T_2, T_3, T_4 and T_5:

Marks: 2 **MCQ**

T_1	T_2	T_3	T_4	T_5
R(X)				
	W(X)			
		R(X)		
		W(X)		
			R(Z)	
				W(Z)
R(Z)				

R(X) denotes read operation on data item X by transaction T_i.
W(X) denotes write operation on data item X by transaction T_i.
Choose the correct option for the above transaction schedule.

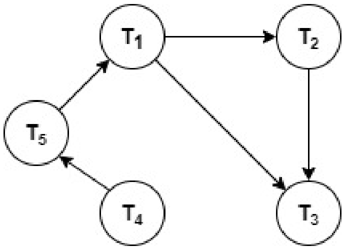
a) The schedule is only view serializable schedule.

b) The schedule is only conflict serializable schedule.

c) The schedule is both view and conflict serializable schedule.

d) The schedule is neither conflict serializable nor view serializable schedule.

Answer: c)
Explanation: If we draw the precedence graph of the transactions as shown in the following, we can observe that the graph has no cycle.
So, the above schedule is a conflict serializable schedule.
All conflict serializable schedules are view serializable too.



So, option (c) is correct

Question 2

Consider the following `schedule S` involving five transactions T_1, T_2, T_3, T_4 and T_5:

Marks: 2 **MCQ**

T_1	T_2	T_3	T_4	T_5
R(X)				
	W(X)			
		R(X)		
		W(X)		
			R(Z)	
			W(Z)	
				W(Z)
R(Z)				

`R(X)` denotes read operation on data item `X` by `transaction` T_i.
`W(X)` denotes write operation on data item `X` by `transaction` T_i.
Identify the correct option(s) that represent the order of execution of all transactions of the above `schedule S`.

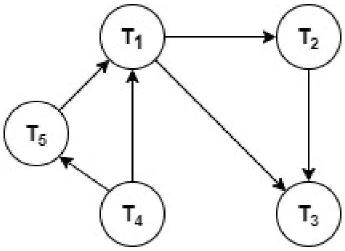
a) T1→T2→T3→T4→T5

b) T4→T5→T1→T3→T2

c) T4→T1→T5→T2→T3

d) T4→T5→T1→T2→T3

Answer: d)
Explanation: If we draw the `precedence graph` of the `transactions` as shown in the following, we can observe that the graph has no cycle.
So, the above schedule is a `conflict serializable schedule`.
All `conflict serializable schedules` are `view serializable` too.



All possible `topological orderings` of the above `precedence graph` will be the possible `conflict serializable schedule`.
Hence, the correct order of execution of all transactions is: (T4→T5→T1→T2→T3)
So, option (d) is correct

Question 3

Consider the following `schedule S` involving four transactions T_1, T_2, T_3 and T_4.

Marks: 2 **MCQ**

T_1	T_2	T_3	T_4
R(X)			
W(X)			
	W(X)		
	R(Z)		
			R(Z)
		W(Z)	

`R(X)` denotes read operation on data item `X` by `transaction` T_i.
`W(Y)` denotes write operation on data item `Y` by `transaction` T_i.
Identify the possible number of `conflict serializable` schedules of the above `schedule S`.

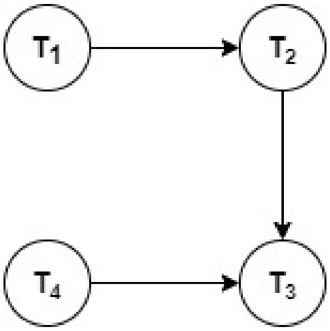
a) 1

b) 2

c) 3

d) 4

Answer: c)
Explanation: If we draw the `precedence graph` of the `schedule`, we can observe that the graph has no cycle. Hence, the above schedule is `conflict serializable schedules`.

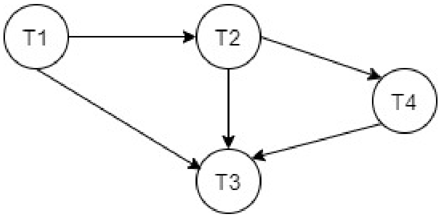


All possible `topological orderings` of the above `precedence graph` will be the possible `conflict serializable schedule`.
1. T1 → T2 → T4 → T3
2. T1 → T4 → T2 → T3
3. T4 → T1 → T2 → T3
Hence, option c) is correct.

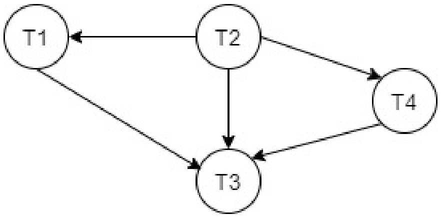
Question 4

Suppose in a database, there are four `transactions` T_1, T_2, T_3 and T_4. `Transaction` T_1 is waiting for `transactions` T_2 and T_3, `transaction` T_2 is waiting for `transaction` T_3, and `transaction` T_4 is waiting for `transactions` T_2 and T_3 to release a data item.
Identify the correct `wait-for graph` for the above scenario. *Marks: 2* **MCQ**

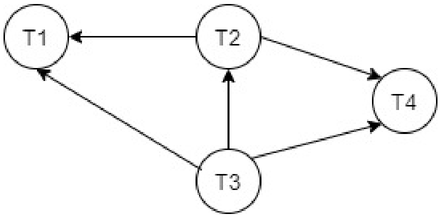
a)



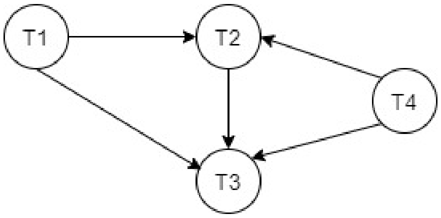
b)



c)



d)



Answer: d)
Explanation: When T_i requests a `data item` currently being held by T_j, then the edge $T_i \rightarrow T_j$ is inserted in the `wait-for graph`. $T_i \rightarrow T_j$, implying that T_i waiting for T_j to release a data item.
Hence, option d) is correct.

Question 5

Consider the following `schedule S` of transactions T_1 and T_2. *Marks: 2* **MCQ**
The read operation on data item `A` is denoted by `read(A)` and the write operation on data item `A` is denoted by `write(A)`.

T_1	T_2
`read(A)`	
`A:=A-200`	
`write(A)`	
	`read(C)`
	`temp:=C*0.1`
	`C:=C-temp`
`read(B)`	
`B:=B+200`	
`write(B)`	
	`write(C)`
	`read(D)`
	`D:=D+temp`
	`write(D)`

Which of the following is `TRUE` about the `schedule S`?

a) S is serializable only as T_1, T_2.

b) S is serializable only as T_2, T_1.

c) S is not serializable either as T_1, T_2 or T_2, T_1.

d) S is serializable both as T_1, T_2 and T_2, T_1.

Answer: d)
Explanation: First, swap all non-conflicting instruction of the above schedule `S`.
Here, T_1 and T_2 both are working on different data items. So, S is serializable both as T_1, T_2 and T_2, T_1
Hence, option (d) is correct.

Question 6

Identify the incorrect statement(s) about the `lock compatibility matrix` given below, where S denotes a `shared mode lock` and X denotes an `exclusive mode lock`.

Marks:2 **MCQ**

	S	X
S	True	False
X	False	False

a) If a transaction holds a `S lock` on a data item, other transactions will not be allowed to obtain a `X lock` on the same data item.

b) If a transaction holds a `S lock` on a data item, other transactions will be allowed to obtain a `S lock` on the same data item.

c) If a transaction holds an `X lock` on a data item, other transactions are not allowed to obtain a `X lock` on the same data item.

d) If a transaction holds an `X lock` on an item, other transactions may be allowed to obtain a `S lock` on the same data item.

Answer: d)
Explanation: As per lock based protocols. Refer Module 34 slide 10.

Question 7

Suppose in a database, there are four transactions T_1, T_2, T_3 and T_4 with timestamp 10, 20, 30 and 40 respectively. T_3 is holding some data items which T_1, T_2 and T_4 are requesting to acquire. Which of the following statement(s) is (are) correct in respect of `Wait-Die` Deadlock Prevention scheme? *Marks: 2* **MSQ**

a) Transaction T_1 and T_4 will wait for T_3 to release the data item.

b) Transaction T_1 and T_2 will wait for T_3 to release the data item.

c) Transaction T_2 and T_4 will wait for T_3 to release the data item.

d) Transaction T_4 will rollback.

Answer: b), d)
Explanation: In `Wait-Die` Deadlock Prevention scheme:
`Older` transaction may `wait` for younger one to release data item. (older means smaller timestamp)
`Younger` transactions never wait for older ones; they are `rolled back` instead.
Transactions T_1 and T_2 are older and T_4 is younger to Transaction T_3.
Hence, options b) and d) are correct.

Question 8

Consider two transactions given below where `lock-X(A)` denotes T_i has obtained an `Exclusive-mode` lock on data item A and `lock-S(A)` denotes T_i has obtained a `Shared-mode` lock on data item A. `read(A)` denotes read operation on data item A by the transaction T_i. `write(A)` denotes write operation on data item A by the transaction T_i. *Marks:2* **MCQ**

T_1
`lock-X(A)`
`read(A)`
`write(A)`
`lock-X(B)`
`read(B)`
`write(B)`
`unlock(B)`
`commit`
`unlock(A)`

T_2
`lock-X(A)`
`read(A)`
`write(A)`
`lock-S(B)`
`read(B)`
`unlock(B)`
`commit`
`unlock(A)`

Which of the following statement(s) is/are `true`?

a) Both T_1 and T_2 follow the `strict two-phase locking protocol`.

b) Both T_1 and T_2 do not follow the `strict two-phase locking protocol`.

c) T_1 follows the `2-phase locking protocol` only but T_2 follows the `strict two-phase locking protocol`.

d) T_1 follows the `strict two-phase locking protocol` only but T_2 follows the `rigorous two-phase locking protocol`.

Answer: c)
Explanation: Transaction T_1 unlocks one `Exclusive-mode lock` before commit. That is why, it does not follow the `strict two phase locking protocol` as well as the `rigorous two-phase locking protocol`. It is following `two-phase locking protocol` only. The first is the growing phase in which it is acquiring locks, the second is one in which it is releasing locks. But transaction T_2 unlocks the `Shared-mode lock` before commit and unlocks the `Exclusive-mode lock` after commit. That is why, it follows the `strict two phase locking protocol` but not the `rigorous two-phase locking protocol`.
Hence, option (c) is correct.

Question 9

Consider two schedules S_1 and S_2 as follows. *Marks*: 2 **MCQ**

S_1	
T_1	T_2
lock-X(A)	
read(A)	
write(A)	
	lock-X(B)
	read(B)
	write(B)
	lock-S(A)
	read(A)
lock-S(B)	
read(B)	
unlock(B)	
	unlock(A)
unlock(A)	
	unlock(B)

S_2	
T_1	T_2
lock-S(A)	
read(A)	
lock-S(B)	
read(B)	
unlock(A)	
unlock(B)	
	lock-X(A)
	read(A)
	write(A)
	lock-X(B)
	read(B)
	write(B)
	unlock(A)
	unlock(B)

Identify the correct statement from the following which relates to whether the schedules are deadlock free. Please note that if any schedule suffers from deadlock, some operations of the transactions in that schedule may not be executed.

a) Both S_1 and S_2 will suffer from deadlock.

b) S_1 will suffer from deadlock, S_2 will not suffer from deadlock.

c) S_1 will not suffer from deadlock, S_2 will suffer from deadlock.

d) Neither S_1 nor S_2 will suffer from deadlock.

Answer: b)

Explanation: In S_1, T_1 is holding exclusive mode lock on (A) and T_2 has requested shared mode lock on (A). While one transaction is holding exclusive mode lock on a particular database, no other transaction can acquire any shared mode lock on (A) unless the lock is released by the former transaction (which is holding exclusive mode lock on the data item).
Similarly, T_2 is holding exclusive lock on (B) and T_1 has requested shared mode lock on (B).
Unless transaction T_2 gets shared mode lock on (A), it will not proceed to the next operations and will not release the exclusive mode lock on (B).
This, in turn, restricts transaction T_2 to acquire shared mode lock on (A).
Similarly, transaction T_1 cannot acquire shared mode lock on (B).
Thus, both T_1 and T_2 are waiting for each other to release resources. Hence, S_1 is going to suffer from a deadlock.

In S_2, T_1 has acquired shared mode lock on (A) and (B) and T_1 released locks on data items A and B. Then, T_2 wants to acquire exclusive mode lock on (A) and (B) and granted to T_2 and no deadlock occurs in S2.
Hence, option (b) is correct.

Question 10

Consider the following two schedules S1 and S2. *Marks: 2* **MSQ**

S1	
T_1	T_2
R(X)	
W(X)	
	R(X)
	W(X)
	COMMIT
R(Y)	

S2	
T_1	T_2
R(X)	
W(X)	
COMMIT	
	R(X)
	W(X)
	COMMIT

R(X) denotes read operation on data item X by Transaction T_i.
W(X) denotes write operation on data item X by Transaction T_i.

Which of the following statement(s) is/are true for the above two schedules S1 and S2?

a) Both schedules S1 and S2 are Recoverable Schedule.

b) Both schedules S1 and S2 are Cascadeless Schedule.

c) The schedule S1 is not a Recoverable Schedule but the schedule S2 is Cascadeless Schedule.

d) The schedule S1 is not a Recoverable Schedule but the schedule S2 is Recoverable Schedule.

Answer: c), d)
Explanation: Recoverable Schedule: If a transaction T_j reads a data item previously written by a transaction T_i , the commit operation of T_i must appear before the commit operation of T_j.
Cascadeless schedules: For each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i, the commit operation of T_i appears before the read operation of T_j.
In S1, T_2 reads the data item X which was previously written by T_1; T_2 committed immediately after the read(X) and write(X) operations . Hence, the schedule is non recoverable schedule.
In S2, T_2 reads the data item X which was previously written by T_1' T_1 commited before read X operation by T_2. Hence, the schedule is only recoverable schedule as well as Cascadeless schedule.
Hence, options (c) and (d) are correct.