Database Management System: Assignment 7

Total Marks : 20

August 26, 2023

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(Y)		
W(Y)				
R(X)				
			R(X)	
			W(X)	
				R(Y)

`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`.

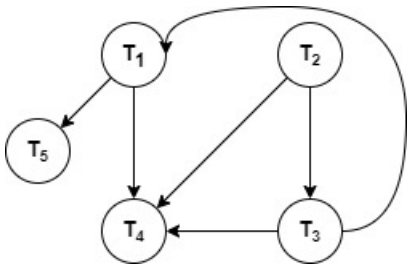
a) The `schedule` is neither `conflict serializable` nor `view serializable schedule`.

b) The `schedule` is both `conflict and view serializable schedule`.

c) The `schedule` is only `view serializable schedule`.

d) The `schedule` is only `conflict serializable schedule`.

Answer: b)
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 (b) 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)				
		R(X)		
		W(X)		
	W(X)			
				W(Z)
			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.
Identify the correct option(s) that represent the order of execution of all transactions of the above `schedule S`.

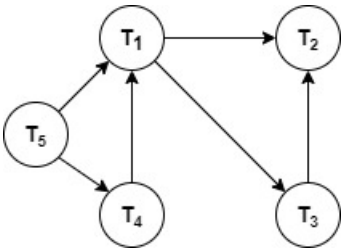
a) T4→T5→T1→T3→T2

b) T5→T4→T1→T3→T2

c) T1→T2→T3→T5→T4

d) T4→T1→T4→T3→T2

Answer: b)
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: (T5→T4→T1→T3→T2)
So, option (b) is correct

Question 3

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)		
W(Y)				
				W(Y)
			R(X)	
			W(X)	
	W(Y)			

`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`.

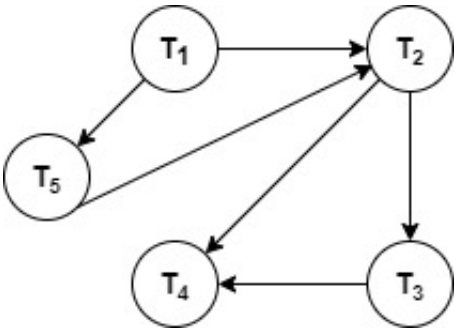
a) 0

b) 1

c) 3

d) 5

Answer: b)
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`.



Only one possible `topological orderings` of the above `precedence graph` will be the possible `conflict serializable schedule`.
1. T1 → T5 → T2 → T3 → T4
Hence, option b) is correct.

Question 4

Consider the following schedule S. *Marks: 2* **MCQ**

T1	T2	T3
R(X)		
	W(X)	
		R(X)
W(X)		
		W(X)
		W(Y)

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 possible number of view serializable schedules of the schedule S.

a) 5

b) 3

c) 2

d) 1

Answer: d)
Explanation: Step 1: Final Update on data item: This is done on data item X,Y by T3.
Since the final update on X, Y is made by T3, the transaction T3 must execute after all transactions T1 and T2. So, (T1,T2) → T3.
Step 2: Transaction T1 does the initial read on X; then, T2 updates on it.
Hence, the dependency is: T1 → T2
Step 3: Write Read Sequence: T2 writes X and then T3 reads X.
Hence, the dependency is: T2 → T3
Thus, the only possible way, this is possible is: T1 → T2 → T3.
Hence, total possible view serializable schedule of S = 1
So, option (d) is correct.

Question 5

Identify the correct statement(s) from the followings. In this context, *older* implies *smaller timestamp*.

Marks:2 **MSQ**

a) In `Wound-Wait` Deadlock Prevention scheme, `Older` transaction forces rollback of younger transaction instead of waiting for it.

b) In `Wound-Wait` Deadlock Prevention scheme, `Younger` transactions never wait for older ones; they are `rolled back` instead.

c) In `Wait-Die` Deadlock Prevention scheme, `Older` transaction forces rollback of younger transaction instead of waiting for it.

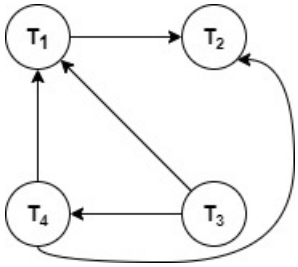
d) In `Wait-Die` Deadlock Prevention scheme, `Younger` transactions never wait for older ones; they are `rolled back` instead.

Answer: a), d)
Explanation: As per the deadlock prevention scheme, options a) and d) are correct.
For more details refer to Module 35.7.

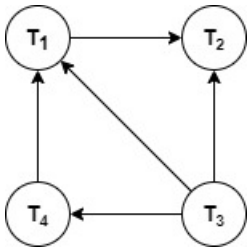
Question 6

Suppose that four `transactions` T_1, T_2, T_3 and T_4 are being applied on a database. `Transaction` T_1 is waiting for `transactions` T_3 and T_4; `transaction` T_2 is waiting for `transaction` T_1, 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**

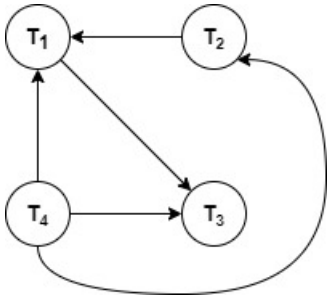
a)



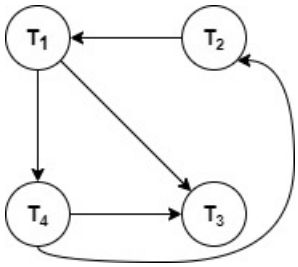
b)



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`. An edge $T_i \rightarrow T_j$ implies that T_i is waiting for T_j to release a data item.
Hence, option d) is correct.

Question 7

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

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

S2	
T_1	T_2
R(X)	
W(X)	
	R(X)
	W(X)
COMMIT	
	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 statements 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 `Cascadeless Schedule` but the schedule `S2` is `a Recoverable Schedule`.

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

Answer: a), c)
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_1 reads the data item X that was previously written by T_2, T_2 committed before T_1 reads X. Hence, the schedule is `recoverable schedule` as well as `Cascadeless schedule`.

In S2, T_2 reads the data item X that was previously written by T_1, T_1 committed before the commit operation of T_2. Hence, the schedule is only `recoverable schedule` but not `Cascadeless schedule`.
Hence, options (a) and (c) are correct.

Question 8

Consider two transactions given below where `lock-X(A)` denotes that T_i has obtained an `Exclusive-mode` lock on data item `A` and `lock-S(A)` denotes that 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)
lock-X(B)
lock-S(C)
write(A)
read(B)
write(B)
read(C)
unlock(C)
commit
unlock(A)
unlock(B)

T_2
lock-S(A)
read(A)
unlock(A)
lock-S(B)
read(B)
commit
unlock(B)

Which of the following statements is (are) `true`?

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

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

c) Only T_1 follows `strict two-phase locking protocol`, but T_2 does not follow `two-phase locking protocol`.

d) Only T_1 follows `rigorous two-phase locking protocol` but T_2 does not follow `two-phase locking protocol`.

Answer: c)
Explanation: Transaction T_1 unlocks the `Shared-mode lock` before commit and unlocks all `Exclusive-mode lock` after commit . That is why, it follows the `strict two phase locking protocol` but not `rigorous two-phase locking protocol`.
Transactions T_2 does not follow any `two-phase locking protocol`. Because there no growing phase and shrinking phase, after unlock(A) again granted a lock-S(B).
Hence, option (c) is correct.

Question 9

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 an `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 an `S lock` on a data item, other transactions will not be allowed to obtain an `S lock` on the same data item.

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

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

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

Question 10

Consider the following `schedule S`. *Marks: 2* **MCQ**

T1	T2	T3
R(Y)		
R(Z)		
W(Y)		
commit		
	R(Y)	
	W(Y)	
		R(Y)
		W(Y)
	abort	

`R(Y)` denotes read operation on data item Y by Transaction T_i.
`W(Y)` denotes write operation on data item Y by Transaction T_i.
Identify the correct statement(s) based on the above schedule S.

a) If T2 fails (aborted), both transactions T1, and T3 must also be rolled back.

b) If T2 fails (aborted), transaction T3 must also be rolled back.

c) If T2 fails (aborted), only transaction T1 must be rolled back.

d) If T2 fails (aborted), no transaction will be rolled back.

Answer: b)
Explanation: As per the cascading rollback a single transaction failure leads to a series of transaction rollbacks.
Hence, if T2 fails (aborted), T3 must also be rolled back.
So, option (b) is correct.