Database Management System: Assignment 7

Total Marks : 20

July 31, 2024

Question 1

Marks: 2 **MCQ**

Consider the following `schedule S` of transactions T_1 and T_2.
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-1000	
write(A)	
	read(A)
read(B)	
B:=B+1000	
	temp:=A∗0.8
	A:=A-temp
write(B)	
	write(A)
	read(B)
	B:=B+temp
	write(B)

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

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

b) S is not serializable neither as T_1, T_2 nor T_2, T_1.

c) S is serializable only as T_1, T_2.

d) S is serializable only as T_2, T_1.

Answer: c)
Explanation: First, swap all non-conflicting instructions of the above schedule S.
T_1 and T_2 both are working on the same data items.
So, S is serializable as T_1, T_2 only
Hence, option (c) is correct.

Question 2

Marks:2 **MCQ**

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

T_1	T_2	T_3	T_4	T_5
R(X)				
	W(X)			
		R(Y)		
		W(Y)		
			W(X)	
			W(Y)	
	R(Z)			
		W(Z)		
				W(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`.

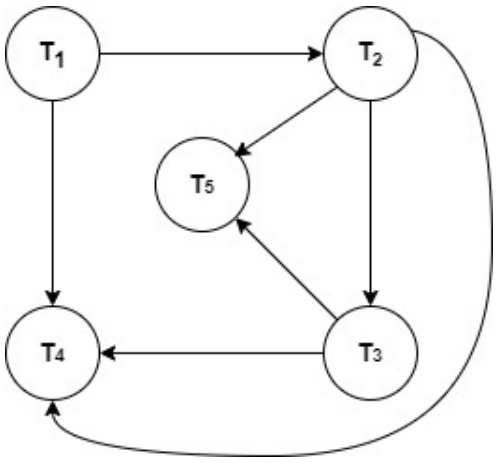
a) The `schedule` is only `view serializable schedule`.

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 3

Marks:2 **MCQ**

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

T_1	T_2	T_3	T_4
R(X)			
	W(X)		
		W(Z)	
			R(Z)
R(Z)			
W(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 possible number of `conflict serializable` schedules of the above `schedule S`.

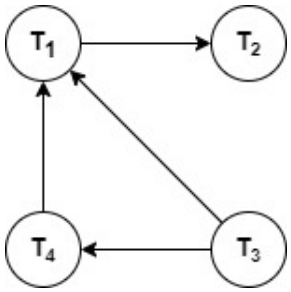
a) 1

b) 2

c) 3

d) 4

Answer: a)
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. T3→T4→T1→T2
Hence, option a) is correct.

Question 4

Marks:2 **MCQ**

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

T_1	T_2	T_3	T_4
R(X)			
R(Z)			
	W(X)		
		R(Y)	
		W(Y)	
			W(X)
			W(Y)
			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 option(s) that do not represent the correct order of execution of all transactions of the above schedule `S`.

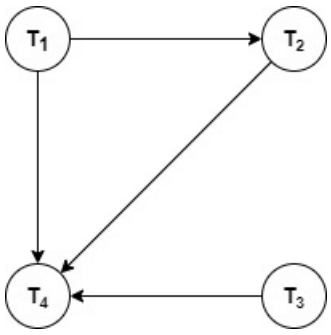
a) T1 → T3 → T2 → T4

b) T1 → T2 → T3 → T4

c) T3 → T1 → T2 → T4

d) T3 → T2 → T1 → T4

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:
1. T1 → T3 → T2 → T4
2. T1 → T2 → T3 → T4
3. T3 → T1 → T2 → T4
So, option (d) is the answer.

Question 5

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

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

`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 possible number of `view serializable` schedule of the above schedule S.

a) 7

b) 5

c) 3

d) 1

Answer: d)
Explanation: Step 1: Final Update on data item: Y-T1
Final Update on data item: X-T1
Since the final update on Y and X is made by T1, the transaction T1 must execute after all the transactions: `(T2,T3)` → `T1`.
Step 2: Initial Read + Which transaction updates after read?
Y : `T3` (initial read) then `T2` firstly updates it.
Hence, the dependency is: T3 →T2
X : `T3` (initial read) then `T1` firstly updates it as well as final update also.
Hence, `T1` will execute later than `T3`: T3 →T1
Step 3: Write Read Sequence: `T2` firstly writes Y and then `T1` reads Y.
Hence, the dependency is: T2 →T1
The ways we can arrange (T3→T2 → T1).
Hence, total possible `view serializable schedule of S`= 1.
Hence, option (d) is correct.

Question 6

Marks:2 **MCQ**

Consider two transactions given below where `lock-X(A)` denotes T_i has obtained an `Exclusive-mode` lock on item A and `lock-S(A)` denotes T_i has obtained a `Shared-mode` lock on item A.

T_1
lock-S(A)
read(A)
lock-X(B)
read(B)
B:= B+100
write(B)
lock-S(C)
read(C)
unlock(A)
unlock(C)
commit
unlock(B)

T_2
lock-X(A)
read(A)
A:= A-100
write(A)
lock-S(B)
read(B)
commit
unlock(A)
unlock(B)

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

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

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

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

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

Answer: a)
Explanation: Transaction T_1 first unlocks all `Shared-mode lock` and then commits. After commit, it unlocks `Exclusive-mode lock` . That is why, it follows the `strict two phase locking protocol` only.
Transaction T_2 first commits and then unlocks all locks(`Exclusive-mode lock and Shared-mode lock`). That is why, T_2 follows the `rigorous two-phase locking protocol`.
So, option (a) is correct.

Question 7

Marks: 2 **MSQ**

Suppose in a database, there are four transactions T_1, T_2, T_3, and T_4 with timestamp 10, 14, 18, and 22 respectively. T_2 is holding a data item which T_1 and T_3 are requesting to acquire. Which of the following statement(s) is (are) correct in respect of `Wound-Wait` Deadlock Prevention scheme?

a) Transaction T_1 will wait to release the data item.

b) Transaction T_3 will wait to release the data item.

c) Transaction T_1 will wounds T_2, Transaction T_2 will be aborted.

d) Transaction T_3 will wounds T_2, Transaction T_2 will be aborted.

Answer: b), c)
Explanation: `Wound-Wait` Deadlock Prevention scheme: When `TA1` requests data item held by `TA2` (older means smaller timestamp), two cases may arise.

- If `TA1` is older than `TA2` then `TA1` wounds `TA2` (`TA2 will be aborted`).
- If `TA1` younger to `TA2` then `TA1` will wait for release the data item held by TA2.

In this case `Transaction` T_2 and T_3 are younger in respect of `Transaction` T_1.
Hence, options b) and c) are correct.

Question 8

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

T_1	T_2
R(X)	
W(X)	
R(Y)	
W(Y)	
	R(X)
COMMIT	
	W(X)
	R(Y)
	W(Y)
	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.
Choose the correct options for the above schedule.

a) The schedule is only `recoverable schedule`.

b) The schedule is only `cascadeless schedule`.

c) The schedule is `recoverable schedule` and `cascadeless schedule` both.

d) The schedule is neither `recoverable nor cascadeless schedule`.

Answer: a)
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.
Here, T_2 read data item X before commit of T_1. Hence, the schedule is recoverable.
So, option (a) is correct.

Question 9

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

T1	T2	T3
		R(X)
		W(X)
		R(Y)
		commit
	R(X)	
	W(X)	
R(X)		
W(X)		
	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 will be rolled back.

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

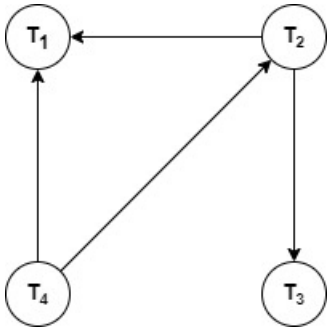
Answer: c)
Explanation: As per the cascading rollback a single transaction failure leads to a series of transaction rollbacks.
Hence, if T2 fails (aborted), T1 must also be rolled back.
Transaction T3 has already been committed, so no rollback will happen. So, option (c) is correct.

Question 10

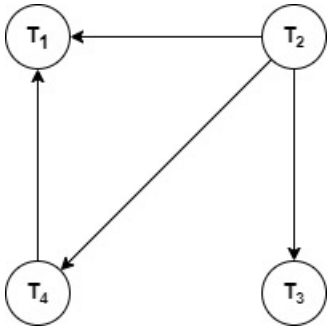
Marks: 2 **MCQ**

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_2; `transaction` T_2 is waiting for `transaction` T_3, and `transaction` T_4 is waiting for `transactions` T_1 and T_2 to release a data item.
Identify the correct `wait-for graph` for the above scenario.

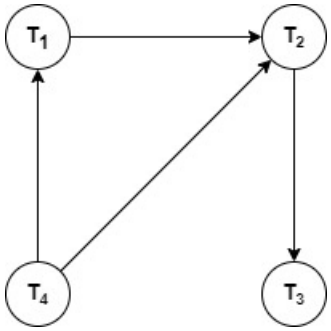
a)



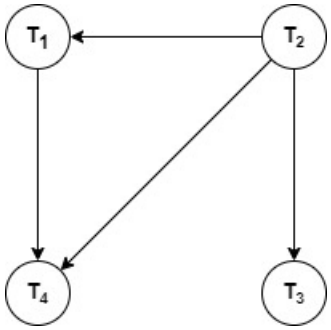
b)



c)



d)



Answer: c)
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 c) is correct.