Branch : CSE & IT Batch : Hinglish

Database Management System

Transaction & Concurrency Control

DPP 03

[NAT]

1. Consider the following schedule

S: $w_1(X)$; $w_1(Y)$; $r_2(X)$; $w_2(Y)$; $r_3(X)$; $w_3(Y)$

How many schedules are conflict equivalent to given schedule (S) _____?

[NAT]

2. Consider the following schedule

S = $r_1(P)$; $r_3(S)$; $w_1(Q)$; $r_2(Q)$ $r_4(Q)$, $w_2(R)$ $r_5(R)$; $w_4(T)$; $r_5(T)$; $w_5(Q)$

How many serial schedules conflict equal to schedules(S)? _____.

[NAT]

3. Consider the following schedule

S = $r_1(P)$; $r_3(S)$; $w_1(Q)$; $r_2(Q)$ $r_4(Q)$, $w_2(R)$; $r_5(R)$; $w_4(T)$; $r_5(T)$; $w_5(Q)$

How many serial schedules view equal to schedule(S)_____?

[MCQ]

4. Consider the following transactions

T_1: $r_1(P)$; $w_1(P)$; $r_1(Q)$; $w_1(Q)$

T_2: $r_2(P)$; $r_2(Q)$

T_3: $w_3(P)$; $w_3(Q)$

How many concurrent schedules between T_1, T_2 and T_3 transactions _____?

(a) 400

(b) 410

(c) 420

(d) None

[NAT]

5. How many views equivalent serial schedules are possible for the given schedules below _____

S: $w_1(P)$ $r_2(P)$ $w_3(P)$ $r_4(P)w_5(P)$ $r_6(P)$

[MCQ]

6. The goal of concurrency control on database system is to

(a) Only allow concurrent execution of transaction that correspond to serial execution of some of the transactions.

(b) Allow only transactions that don't access common relationship to run concurrently.

(c) Execute transactions serially.

(d) None of the above.

[MCQ]

7. What problem can occur when a DBMS executes multiple transactions concurrently?

(a) Lost update problem.

(b) Dirty read problem.

(c) Incorrect summary problem.

(d) All of the above.

[MCQ]

8. Consider the following statements

S_1: Every view serializable schedule is conflict serializable.

S_2: Some view serializable schedules are conflict serializable.

(a) Only S_1 is true.

(b) Only S_2 is true

(c) Both S_1 & S_2 are true

(s) Neither S_1 nor S_2 is true

[MCQ]

9. Consider the following schedule involving two transactions

S_1: $r_1(A)$; $r_2(A)$; $w_2(A)$; $r_3(A)$; $w_1(A)$; $w_2(B)$; $r_3(B)$, c_2, $w_3(A)$; c_1, c_3

S_2: $r_2(A)$; $r_1(A)$; $w_1(A)$; $w_2(A)$; $w_2(A)$; $r_3(A)$; $w_3(A)$, $r_2(B)$; c_1, c_3; c_2

Which one of the following statements is TRUE?

(a) S_1 is recoverable and S_2 is not recoverable.

(b) S_1 is not recoverable and S_2 is recoverable.

(c) Both S_1 and S_2 are recoverable.

(d) Both S_1 and S_2 are not recoverable.

[MCQ]

10. Consider the following schedule:

S: $r_1(A)$; $r_2(C)$; $w_1(A)$; $r_3(A)$ $r_2(B)$; $w_2(B)$, $w_3(A)$; $r_3(B)$; $r_2(A)$

for the schedule S given above two orderings of commits (c_i) operations are specified.

I. c_1; c_3; c_2

II. c_1; c_2; c_3

Which of the above ordering ensures recoverability of schedule S?

(a) Only I

(b) Both I and II

(c) Only II

(d) None of these

[MCQ]

11. Consider the following partial schedule 'S' involving two transaction T_1 and T_2

Time	T_1	T_2
t_0	read(P);	
t_1	write(P);	
t_2		read(R);
t_3		write(R);
t_4		read(Q);
t_5		write(Q);
t_6		read(P);
t_7		commit;
t_8	read(Q);	

Suppose that the transaction T_1 fails immediately after time instance 8. Which one of the following is correct?

S_1: Schedule S is non recoverable and cannot ensure transaction atomicity

S_2: Only T_2 should be aborted and then restarted to ensure truncation atomicity

(a) Only S_1 is true

(b) Only S_2 is true

(c) Both S_1 and S_2 are true

(d) Both S_1 and S_2 are false

Answer Key

1. **(8)**
2. **(10)**
3. **(10)**
4. **(c)**
5. **(2)**
6. **(a)**
7. **(d)**
8. **(b)**
9. **(a)**
10. **(d)**
11. **(a)**

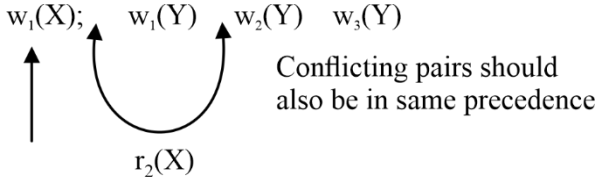
Hints & Solutions

1. (8)

Given schedule

S: $w_1(X)$; $w_1(Y)$; $r_2(X)$; $w_2(Y)$;$r_3(X)$;$w_3(Y)$

Conflict equivalent schedules to above schedules.



Transaction T_1 operations must be in same order

There are 2 ways $r_2(X)$ placed such that it must be before $w_2(Y)$ and conflicting pairs should be in precedence.

Hence 2 possibilities to place $r_2(x)$ to avoid conflict equivalence in above schedule.

$S = w_1(X)\ r_2(X)\ w_1(Y)\ w_2(Y)\ w_3(Y)$

$r_3(X)$

4 ways

$S = w_1(X);\ w_1(Y)\ r_2(X)\ w_2(Y)\ w_3(Y)$

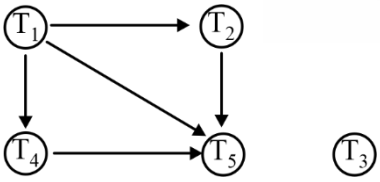
$r_3(X)$

4 ways

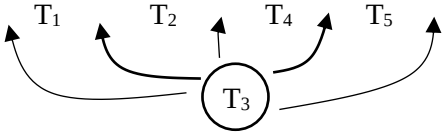
Total 8 conflict equal schedules to the given schedule.

2. (10)

Constructing the precedence graph

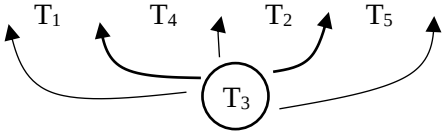


Topological orders



5 ways

Topological orders



5 ways

Total 10 topological orders.

As we know that number of serial schedule conflict 'S' is equal to number of topological orders.

3. (10)

Final write Q: $T_1\ T_5$

Initial Reads

Data item	Initial reads	Writes
P	T_1	-
Q	-	$T_1\ T_5$
R	-	T_2
S	T_3	-
T	-	T_4

Updated reads

$w_1(Q) \rightarrow r_2(Q)$

$r_4(Q)$

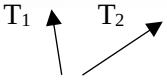
T_5 also writes Q

$w_2(R) \rightarrow r_5(Q)$

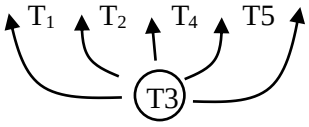
$w_4(T) \rightarrow r_5(T)$

T_1	T_5
-	
$T_1 \rightarrow T2$	
$T1 \rightarrow T4$	
$T2 \rightarrow T5$	
$T4 \rightarrow T5$	

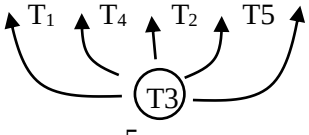
view equal serial orders



T_4 can be placed in two ways

1) 

5 ways

2) 

5 ways

$\therefore$ There are 10 serial orders.

4. (c)

$$\frac{8!}{4! * 2! * 2!} = 420$$

5. (2)

S: $w_1(P)r_2(P)$ $w_3(P)r_4(P)$ $w_5(P)r_6(P)$

Final write

- Can exchange
- without violation view equal conditions

View equal serial schedules are

T_1 T_2 T_3 T_4 T_5 T_6

T_3 T_4 T_1 T_2 T_5 T_6

6. (a)

only allow concurrent execution of transaction that correspond to serial execution of some of the transactions.

7. (d)

All the problems mentioned in option are potential when a DBMS executes multiple transactions concurrently.

8. (b)

Every conflict serializable schedules are view serializable but vice versa is not true. However, some view serializable schedules are conflict serializable.

9. (a)

S_1:

T_1	T_2	T_3
$r_1(A)$		
	$r_2(A)$	
	$w_2(A)$	
		$r_3(A)$
$w_1(A)$		
	$w_2(B)$	
		$r_3(B)$
	c_2	
		$w_3(A)$
c_1		
		c_3

Here the transaction reads the changes of uncommitted transaction but commits itself. So schedule is recoverable.

S_2:

T_1	T_2	T_3
	$r_2(A)$	
$r_1(A)$		
$w_1(A)$		
	$w_2(A)$	
	$w_2(B)$	
		$r_3(A)$
		$w_3(A)$
		$r_3(B)$
c_1		
		c_3
	c_2	

Here T_3 reads the changes of T_2 but T_3 commits before T_2. So it is not recoverable schedule. Hence, correct option is (a).

10. (d)

T_1	T_2	T_3
$r_1(A)$		
	$r_2(c)$	
$w_1(A)$		
		$r_3(A)$
	$r_2(B)$	
	$w_2(B)$	
		$w_3(A)$
		$r_3(B)$
	$r_2(A)$	
c_1		
		c_3
	c_2	

T_3 reads the changes of T_2 corresponding to B and commits itself before T_2. So this schedule is not recoverable

T_1	T_2	T_3
$r_1(A)$		
	$r_2(c)$	
$w_1(A)$		
		$r_3(A)$
	$r_2(B)$	
	$w_2(B)$	
		$w_3(A)$
		$r_3(B)$
	$r_2(A)$	
c_1		
	c_2	
		c_3

T_2 reads the changes of T_3 Corresponding to A and commits itself before committing of T_3. So this schedule is also not recoverable. Hence answer is option (d).

11. (a)

T_1 gets faild after t_8, as T_1 is uncommitted at t_1 time and write (P) and T_2 reads P at t_6 and gets committed. Hence uncommitted transaction changes are got read by other transaction and then gets committed itself hence it is non recoverable.

T_1 gets failed hence not atomic. Hence correct option is a.