S1 :

T_1	T_2
R(P)	
W(Q)	
	R(P)
	W(P)
R(Q)	
W(Q)	
	R(Q)
	W(Q)

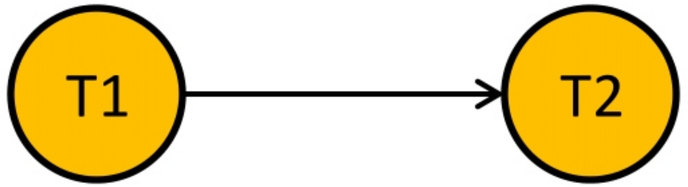
Conflict serializable?

S1 :

T_1	T_2
R(P)	
W(Q)	
	R(P)
	W(P)
R(Q)	
W(Q)	
	R(Q)
	W(Q)

Conflict serializable?

Answer - YES



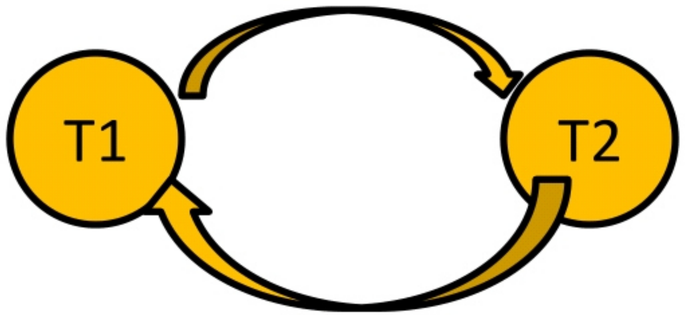
Conflict serializable?

S2 :

T_1	T_2
R(P)	
W(Q)	
	R(P)
	W(P)
	R(Q)
	W(Q)
R(Q)	
W(Q)	

Conflict serializable?

Answer - NO



S2 :

T_1	T_2
R(P)	
W(Q)	
	R(P)
	W(P)
	R(Q)
	W(Q)
R(Q)	
W(Q)	

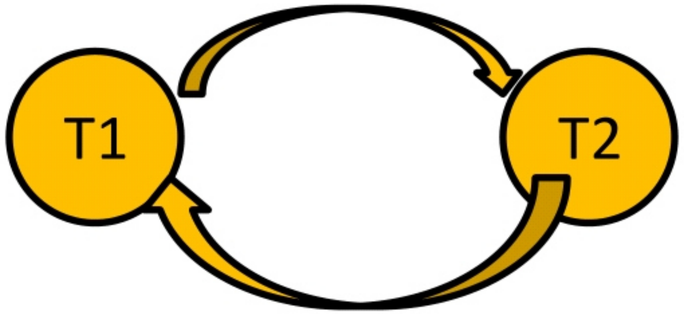
S3 :

T_1	T_2
R(P)	
R(Q)	
	R(P)
	R(Q)
	W(Q)
W(Q)	

Conflict serializable?

T_1	T_2
R(P)	
R(Q)	
	R(P)
	R(Q)
	W(Q)
W(Q)	

S3 :



Conflict serializable?

Answer - NO

Conflict serializable?

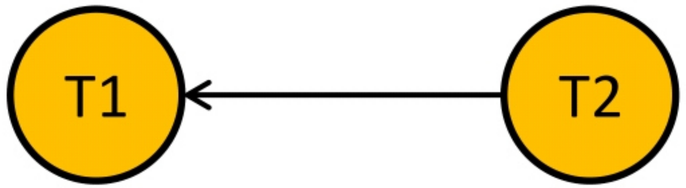
S4 :

T_1	T_2
R(P)	
	R(P)
	R(Q)
	W(Q)
R(Q)	
W(P)	

Conflict serializable?

S4 :

T_1	T_2
R(P)	
	R(P)
	R(Q)
	W(Q)
R(Q)	
W(P)	



Answer - YES

unacademy

View Serializable?

T1	T2	T3
R(A)		
	R(A)	
		W(A)
W(A)		

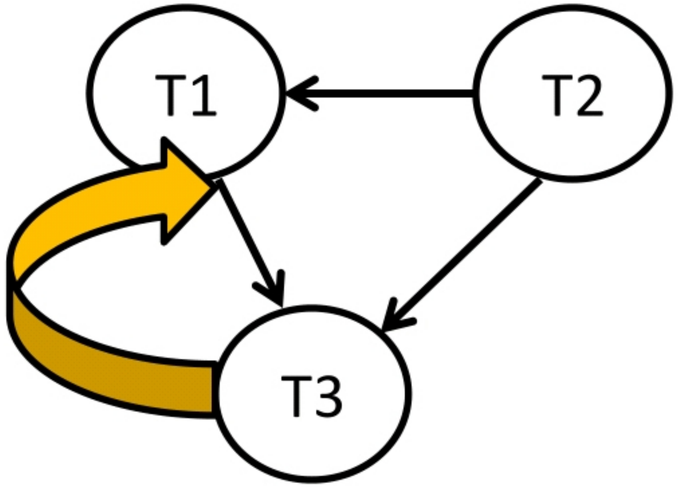
Checking Whether S is Conflict Serializable Or Not-

Step-1:

$R_1(A)$, $W_3(A)$ $(T_1 \rightarrow T_3)$
$R_2(A)$, $W_3(A)$ $(T_2 \rightarrow T_3)$
$R_2(A)$, $W_1(A)$ $(T_2 \rightarrow T_1)$
$W_3(A)$, $W_1(A)$ $(T_3 \rightarrow T_1)$

T1	T2	T3
R(A)		
	R(A)	
		W(A)
W(A)		

Step-2:



Clearly, there exists a cycle in the precedence graph. Therefore, the given schedule S is not conflict serializable.

Now,
Since, the given schedule S is not conflict serializable, so, it may or may not be view serializable.
To check whether S is view serializable or not, let us use another method.
Let us check for blind writes

Checking for Blind Writes-

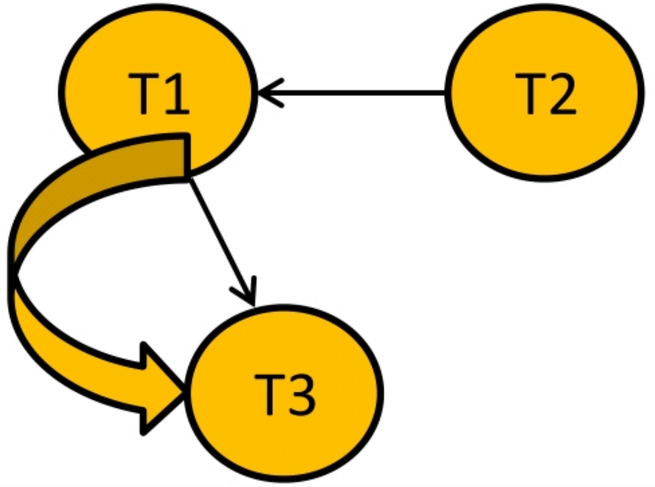
There exists a blind write W_3 (A) in the given schedule S.
Therefore, the given schedule S may or may not be view serializable.

Now,
To check whether S is view serializable or not, let us use another method.
Let us derive the dependencies and then draw a dependency graph.

Drawing a Dependency Graph-

T1 firstly reads A and T3 firstly updates A.
So, T1 must execute before T3.
Thus, we get the dependency **T1 → T3**.
Final updation on A is made by the transaction T1.
So, T1 must execute after all other transactions.
Thus, we get the dependency **(T2, T3) → T1**.
There exists no write-read sequence.

Clearly, there exists a cycle in the dependency graph.
Thus, we conclude that the given schedule S is not view serializable.



T1	T2	T3
R(A)		
	R(A)	
		W(A)
W(A)		

unacademy

THANK YOU !