Conflict Serializability | Practice Problems

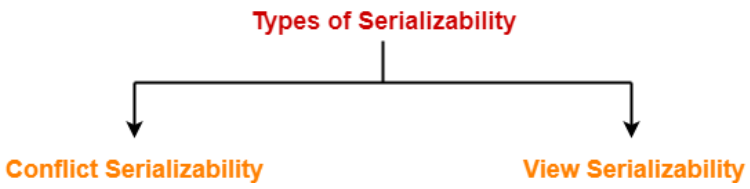
Database Management System

Conflict Serializability-

Before you go through this article, make sure that you have gone through the previous article on **Conflict Serializability**.

We have discussed-

- The concept of serializability helps to identify the correct non-serial schedules that will maintain the consistency of the database.
- There are two types of serializability-



In this article, we will discuss practice problems based on conflict serializability.

PRACTICE PROBLEMS BASED ON CONFLICT SERIALIZABILITY-

Problem-01:

Check whether the given schedule S is conflict serializable or not-

S : R_1(A) , R_2(A) , R_1(B) , R_2(B) , R_3(B) , W_1(A) , W_2(B)

Solution-

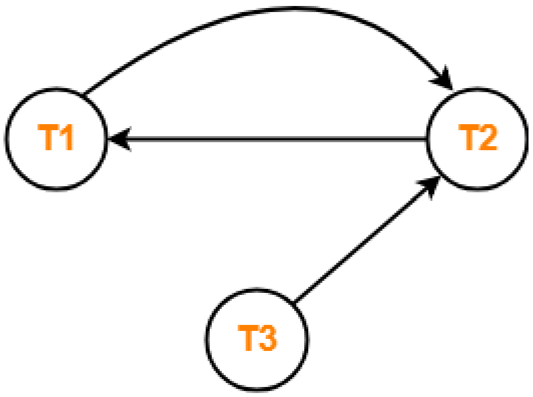
Step-01:

List all the conflicting operations and determine the dependency between the transactions-

- R_2(A) , W_1(A) (T_2 → T_1)
- R_1(B) , W_2(B) (T_1 → T_2)
- R_3(B) , W_2(B) (T_3 → T_2)

Step-02:

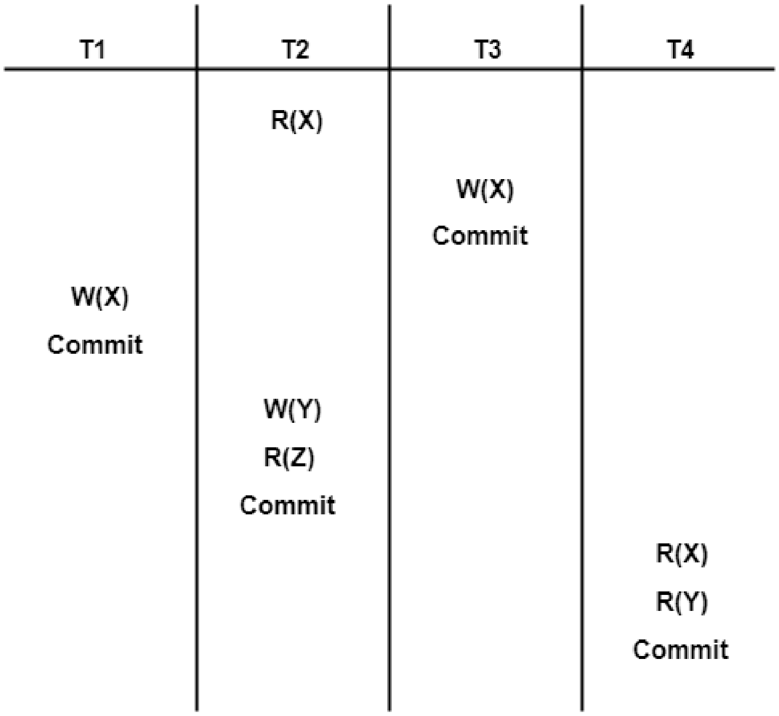
Draw the precedence graph-



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

Problem-02:

Check whether the given schedule S is conflict serializable and recoverable or not-



T1	T2	T3	T4
	R(X)		
		W(X)	
		Commit	
W(X)			
Commit			
	W(Y)		
	R(Z)		
	Commit		
			R(X)
			R(Y)
			Commit

Solution-

Checking Whether S is Conflict Serializable Or Not-

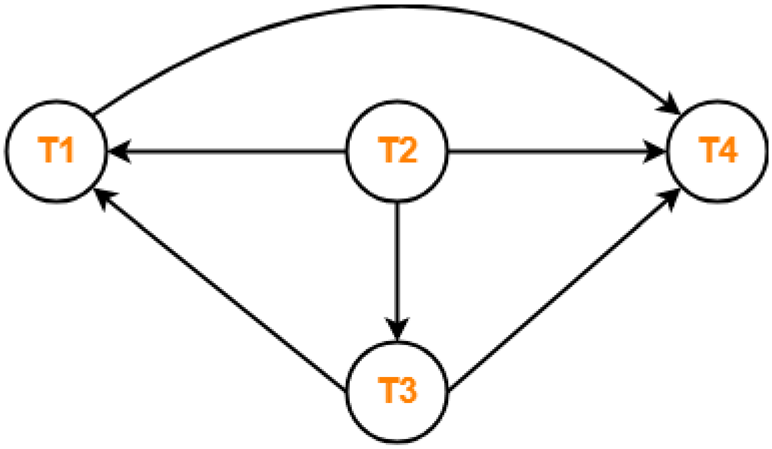
Step-01:

List all the conflicting operations and determine the dependency between the transactions-

- $R_2(X)$, $W_3(X)$ ($T_2 \rightarrow T_3$)
- $R_2(X)$, $W_1(X)$ ($T_2 \rightarrow T_1$)
- $W_3(X)$, $W_1(X)$ ($T_3 \rightarrow T_1$)
- $W_3(X)$, $R_4(X)$ ($T_3 \rightarrow T_4$)
- $W_1(X)$, $R_4(X)$ ($T_1 \rightarrow T_4$)
- $W_2(Y)$, $R_4(Y)$ ($T_2 \rightarrow T_4$)

Step-02:

Draw the precedence graph-



- Clearly, there exists no cycle in the precedence graph.
- Therefore, the given schedule S is conflict serializable.

Checking Whether S is Recoverable Or Not-

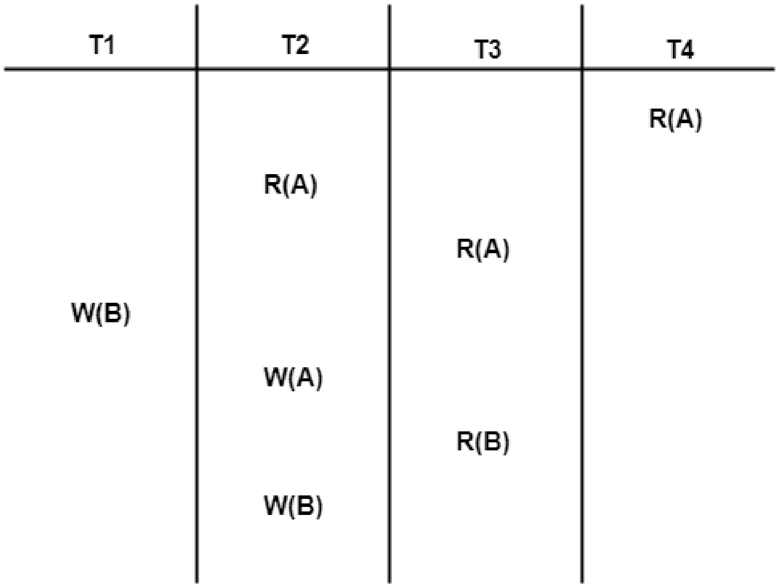
- Conflict serializable schedules are always recoverable.
- Therefore, the given schedule S is recoverable.

Alternatively,

- There exists no dirty read operation.
- This is because all the transactions which update the values commits immediately.
- Therefore, the given schedule S is recoverable.
- Also, S is a **Cascadeless Schedule**.

Problem-03:

Check whether the given schedule S is conflict serializable or not. If yes, then determine all the possible serialized schedules-



T1	T2	T3	T4
			R(A)
	R(A)		
		R(A)	
W(B)			
	W(A)		
		R(B)	
	W(B)		

Solution-

Checking Whether S is Conflict Serializable Or Not-

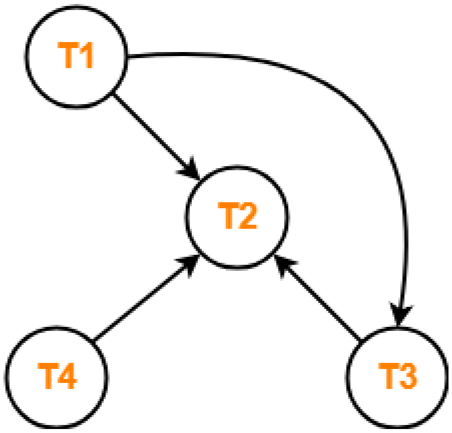
Step-01:

List all the conflicting operations and determine the dependency between the transactions-

- $R_4(A)$, $W_2(A)$ ($T_4 \rightarrow T_2$)
- $R_3(A)$, $W_2(A)$ ($T_3 \rightarrow T_2$)
- $W_1(B)$, $R_3(B)$ ($T_1 \rightarrow T_3$)
- $W_1(B)$, $W_2(B)$ ($T_1 \rightarrow T_2$)
- $R_3(B)$, $W_2(B)$ ($T_3 \rightarrow T_2$)

Step-02:

Draw the precedence graph-



- Clearly, there exists no cycle in the precedence graph.
- Therefore, the given schedule S is conflict serializable.

Finding the Serialized Schedules-

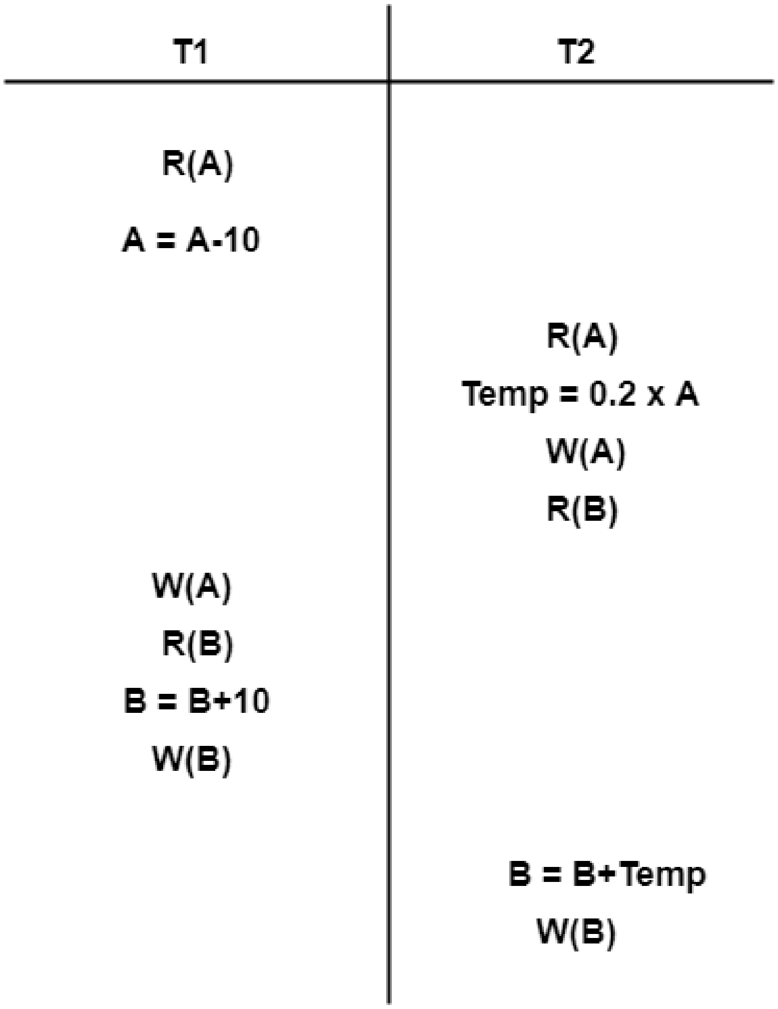
- All the possible topological orderings of the above precedence graph will be the possible serialized schedules.
- The topological orderings can be found by performing the **Topological Sort** of the above precedence graph.

After performing the topological sort, the possible serialized schedules are-

1. $T_1 \rightarrow T_3 \rightarrow T_4 \rightarrow T_2$
2. $T_1 \rightarrow T_4 \rightarrow T_3 \rightarrow T_2$
3. $T_4 \rightarrow T_1 \rightarrow T_3 \rightarrow T_2$

Problem-04:

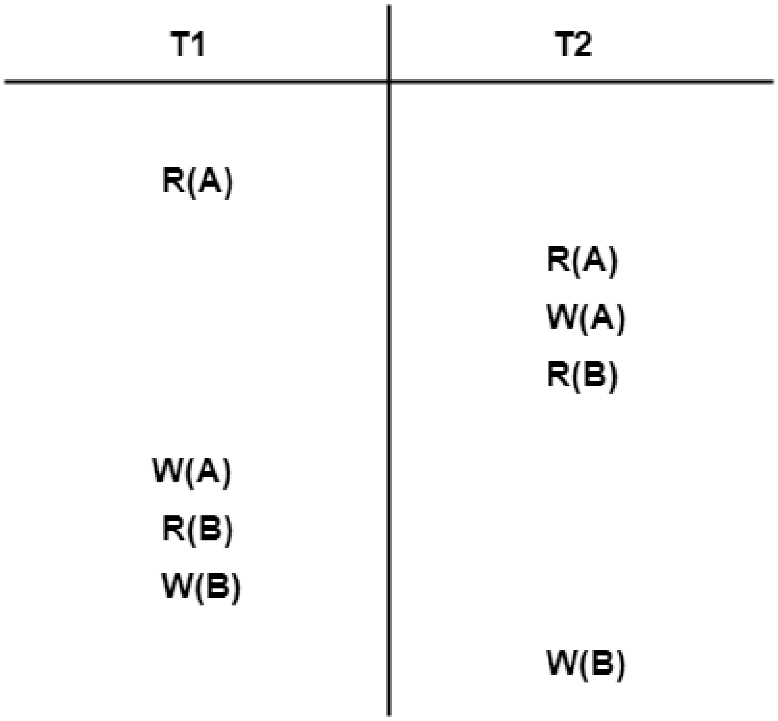
Determine all the possible serialized schedules for the given schedule-



T1	T2
R(A)	
A = A-10	
	R(A)
	Temp = 0.2 x A
	W(A)
	R(B)
W(A)	
R(B)	
B = B+10	
W(B)	
	B = B+Temp
	W(B)

Solution-

The given schedule S can be rewritten as-



T1	T2
R(A)	
	R(A)
	W(A)
	R(B)
W(A)	
R(B)	
W(B)	
	W(B)

This is because we are only concerned about the read and write operations taking place on the database.

Checking Whether S is Conflict Serializable Or Not-

Step-01:

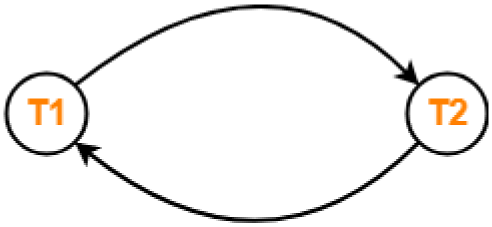
List all the conflicting operations and determine the dependency between the transactions-

- $R_1(A)$, $W_2(A)$ ($T_1 \rightarrow T_2$)
- $R_2(A)$, $W_1(A)$ ($T_2 \rightarrow T_1$)

- $W_2(A)$, $W_1(A)$ ($T_2 \rightarrow T_1$)
- $R_2(B)$, $W_1(B)$ ($T_2 \rightarrow T_1$)
- $R_1(B)$, $W_2(B)$ ($T_1 \rightarrow T_2$)
- $W_1(B)$, $W_2(B)$ ($T_1 \rightarrow T_2$)

Step-02:

Draw the precedence graph-



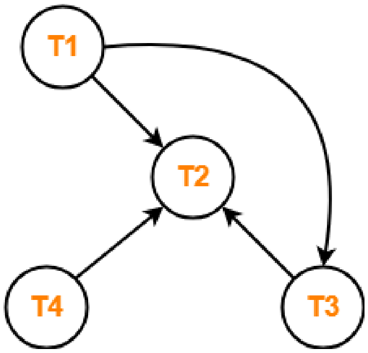
- Clearly, there exists a cycle in the precedence graph.
- Therefore, the given schedule S is not conflict serializable.
- Thus, Number of possible serialized schedules = 0.

Next Article- View Serializability in DBMS

Get more notes and other study material of **Database Management System (DBMS)**.

Watch video lectures by visiting our YouTube channel **LearnVidFun**.

Summary



Article Name	Conflict Serializability | Practice Problems
Description	Practice Problems based on Conflict Serializability and How to check whether a given schedule is conflict serializable or not. Serializability in DBMS is a concept that helps to identify the correct non-serial schedules that will maintain the consistency of the database.
Author	Akshay Singhal
Publisher Name	Gate Vidyalay
Publisher Logo	GateVidyalay A temple of learning