# PRACTICE PROBLEMS BASED ON VIEW SERIALIZABILITY-

## Problem-01:

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

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

**Solution-**

- We know, if a schedule is conflict serializable, then it is surely view serializable.
- So, let us check whether the given schedule is conflict serializable or not.

**Checking Whether S is Conflict Serializable Or Not-**

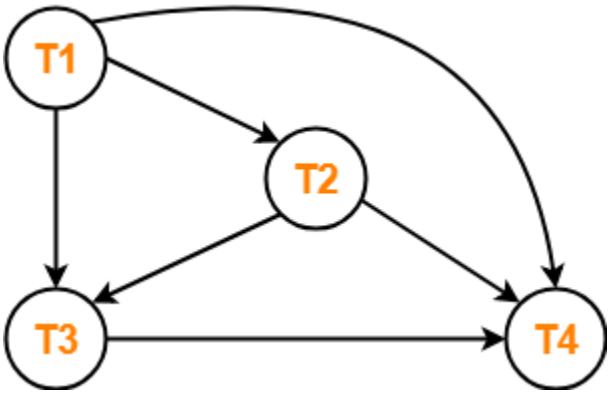
**Step-01:**

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

- $W_1(B)$ , $W_2(B)$ $(T_1 \rightarrow T_2)$
- $W_1(B)$ , $W_3(B)$ $(T_1 \rightarrow T_3)$
- $W_1(B)$ , $W_4(B)$ $(T_1 \rightarrow T_4)$
- $W_2(B)$ , $W_3(B)$ $(T_2 \rightarrow T_3)$
- $W_2(B)$ , $W_4(B)$ $(T_2 \rightarrow T_4)$
- $W_3(B)$ , $W_4(B)$ $(T_3 \rightarrow T_4)$

**Step-02:**

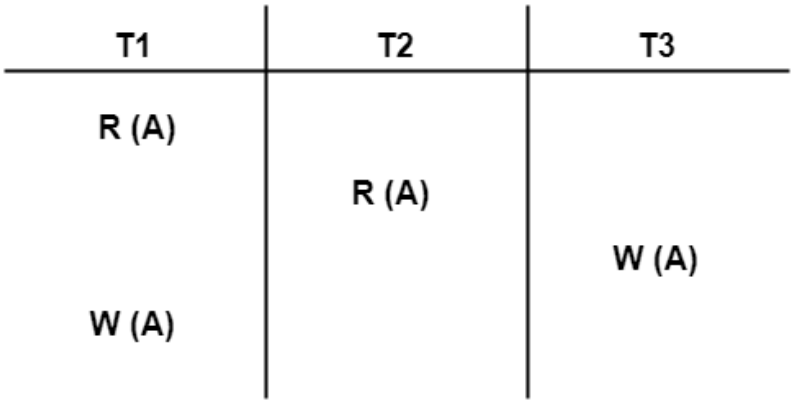
Draw the precedence graph-




- Clearly, there exists no cycle in the precedence graph.
- Therefore, the given schedule S is conflict serializable.
- Thus, **we conclude that the given schedule is also view serializable.**

**Problem-02:**

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




**Solution-**

- We know, if a schedule is conflict serializable, then it is surely view serializable.
- So, let us check whether the given schedule is conflict serializable or not.

**Checking Whether S is Conflict Serializable Or Not-**

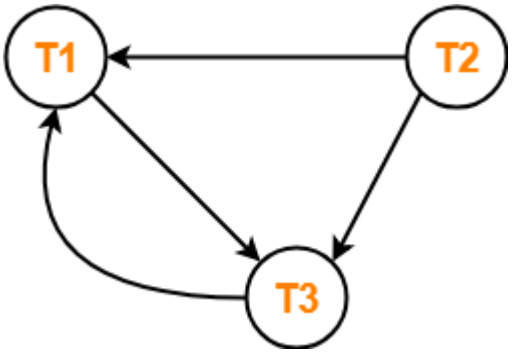
**Step-01:**

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

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

**Step-02:**

Draw the precedence graph-




- 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.

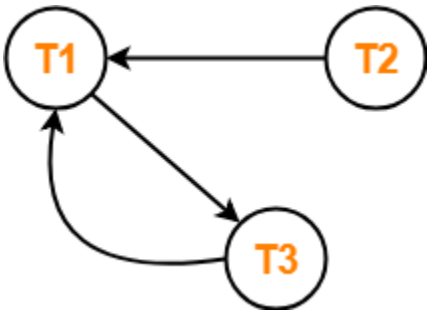
Now,

- 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.

Now, let us draw a dependency graph using these dependencies-




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

## Problem-03:

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

| T1 | T2 |
|---|---|
| R (A) | |
| A = A + 10 | |
| | R (A) |
| | A = A + 10 |
| W (A) | |
| | W (A) |
| R (B) | |
| B = B + 20 | |
| | R (B) |
| | B = B x 1.1 |
| W (B) | |
| | W (B) |

## Solution-

- We know, if a schedule is conflict serializable, then it is surely view serializable.
- So, let us check whether the given schedule is conflict serializable or not.

**Checking Whether S is Conflict Serializable Or Not-**

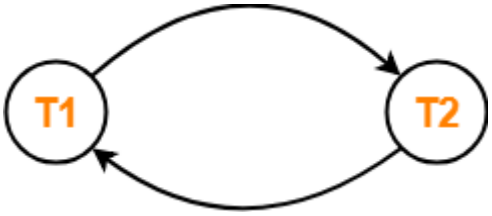
**Step-01:**

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

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

**Step-02:**

Draw the precedence graph-




- 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 no blind write in the given schedule S.
- Therefore, it is surely not view serializable.

**Alternatively**,

- You could directly declare that the given schedule S is not view serializable.
- This is because there exists no blind write in the schedule.
- You need not check for conflict serializability.

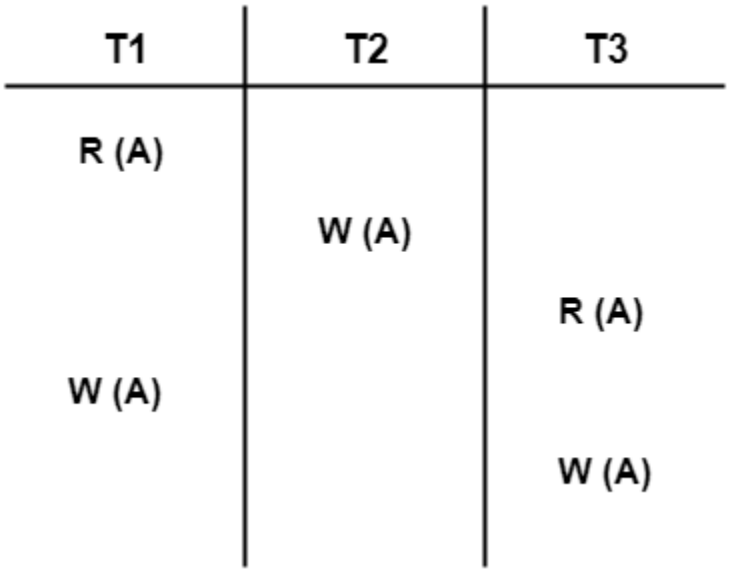
## Problem-04:

Check whether the given schedule S is view serializable or not. If yes, then give the serial schedule.

$$S : R_1(A) , W_2(A) , R_3(A) , W_1(A) , W_3(A)$$

## Solution-

For simplicity and better understanding, we can represent the given schedule pictorially as-




- We know, if a schedule is conflict serializable, then it is surely view serializable.
- So, let us check whether the given schedule is conflict serializable or not.

## Checking Whether S is Conflict Serializable Or Not-

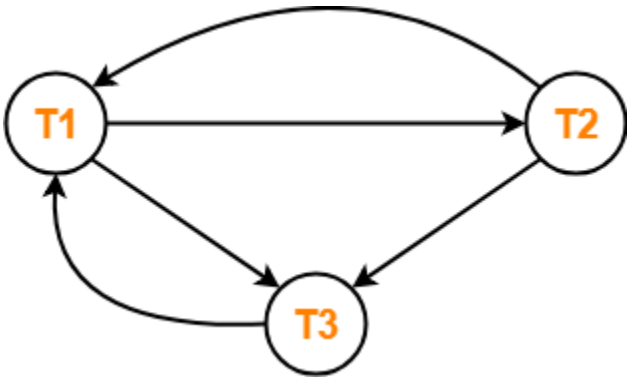
### Step-01:

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

- $R_1(A)$ , $W_2(A)$ $(T_1 \rightarrow T_2)$
- $R_1(A)$ , $W_3(A)$ $(T_1 \rightarrow T_3)$
- $W_2(A)$ , $R_3(A)$ $(T_2 \rightarrow T_3)$
- $W_2(A)$ , $W_1(A)$ $(T_2 \rightarrow T_1)$
- $W_2(A)$ , $W_3(A)$ $(T_2 \rightarrow T_3)$
- $R_3(A)$ , $W_1(A)$ $(T_3 \rightarrow T_1)$
- $W_1(A)$ , $W_3(A)$ $(T_1 \rightarrow T_3)$

**Step-02:**

Draw the precedence graph-




- 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 $W_2(A)$ in the given schedule S.
- Therefore, the given schedule S may or may not be view serializable.

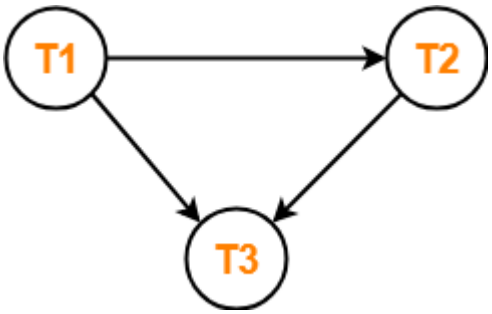
Now,

- 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 T2 firstly updates A.
- So, T1 must execute before T2.
- Thus, we get the dependency **T1 → T2**.
- Final updation on A is made by the transaction T3.
- So, T3 must execute after all other transactions.
- Thus, we get the dependency **(T1, T2) → T3**.
- From write-read sequence, we get the dependency **T2 → T3**

Now, let us draw a dependency graph using these dependencies-




- Clearly, there exists no cycle in the dependency graph.
- Therefore, the given schedule S is view serializable.
- The serialization order T1 → T2 → T3.