

Serializable schedules

Example 1

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

In this situation, the two transactions do not have operations on the same data item, so the schedule is serializable. A serializable schedule is a schedule whose effect on any consistent database instance is guaranteed to be identical to that of some complete serial schedule over S.

Example 2.

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

Is this schedule serializable?

Yes – because it is equivalent to the serial schedule T2, T1. The results are equivalent to T2 running first in its entirety, then T1 running.

Example 3

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

Is this schedule serializable? Yes, this is still equivalent to the serial schedule T2, T1.

Example 4.

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

Is the schedule serializable? Let's look at the 2 possibilities. The schedule cannot be T2, T1 because T2 is reading the result of T1's write. The schedule cannot be T1, T2 because T1 is reading the result of T2's write. So the schedule is not serializable.

This is an example of reading uncommitted data (called a dirty read).

