CHAPTER 14
DISCUSSIONS 2

Discussion 14-6

Explain the distinction between the terms *serial schedule* and *serializable schedule*.

Discussion 14-7

Give an example of a *serializable schedule* with two transactions such that the order in which the transactions commit is different from the serialization order.

Discussion 14-8

What are the values of *A* and *B* after the execution of each of these schedules, with *A*=*B*=100 initially.

The two schedules are both serializable but yield different results. Is this a discrepancy?

T_1	T_2
read(A)	
$A := A - 50$	
write(A)	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
read(B)	
$B := B + 50$	
write(B)	
	read(B)
	$B := B + temp$
	write(B)

Schedule 1

T_1	T_2
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
read(A)	
$A := A - 50$	
write(A)	
	read(B)
	$B := B + temp$
	write(B)
read(B)	
$B := B + 50$	
write(B)	

Schedule 2

Discussion 14-9

Draw a *precedence graph* for the schedule shown on the right, and determine whether it is serializable.

T_1	T_2	T_3	T_4	T_5
read(Y) read(Z)	read(X)			
				read(V) read(W) read(W)
	read(Y) write(Y)	write(Z)		
read(U)			read(Y) write(Y) read(Z) write(Z)	
read(U) write(U)				

Discussion 14-10

Define *recoverable schedule*.

Is the following schedule recoverable?

When is a schedule recoverable?

T_1	T_2
read(A)	
$A := A - 50$	
write(A)	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
read(B)	
$B := B + 50$	
write(B)	
	read(B)
	$B := B + temp$
	write(B)
	commit
abort	

Discussion 14-11

Define *cascadeless schedule*.

Is the following schedule cascadeless?

When is a schedule cascadeless?

T_1	T_2
read(A)	
$A := A - 50$	
write(A)	
	read(A)
	$temp := A * 0.1$
	$A := A - temp$
	write(A)
read(B)	
$B := B + 50$	
write(B)	
	read(B)
	$B := B + temp$
	write(B)
abort	

Discussion 14-12

Show that *every cascadeless schedule is also recoverable.*