Week 9: ACID, Transactions and Concurrency Control

1. Consider the following schedules:

T1		R(A)	W(A)	R(B)					
T2					W(B)	R(C)	W(C)	W(A)	
T3	R(C)								W(D)

(a) Draw the dependency graph (precedence graph) for the schedule.

T3 -> T2 [(R(C), W(C)];
T1 -> T2 [R(A), W(A) and W(A),W(A) and R(B),W(B)]

(b) Is the schedule conflict serializable? If so, what are all the (conflict) equivalent serial schedules? If not, why not?

Yes. Serial schedules: T3, T1, T2; T1, T3, T2.

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

(a) Draw the dependency graph (precedence graph) for the schedule.

T3 -> T1 [R(A), W(A)];
T2 -> T1 [R(A), W(A)];
T1 -> T2 [R(B), W(B)];
T4 -> T2 [R(B), W(B)]

(b) Is the schedule conflict serializable? If so, what are all the (conflict) equivalent serial schedules? If not, why not?

No. Why not: cycle in the precedence graph (T1 must precede T2, T2 must precede T1)

2) a. What will be printed in the following execution (B=3, F=300)?

Lock_X(B)	
Read(B)	
B = B*10	
Write(B)	Lock_S(B)
Lock_X(F)	
F = B*100	
Write(F)	
Unlock(B)	
Unlock(F)	
	Lock_S(F)
	Read(F)
	Read(B)
	Print(F+B)
	Unlock(B)
	Unlock(F)

Answer: 3030 (because B=30, F=3000)

b. Does the execution use: (a) 2PL or (b) Strict 2PL?

Answer: Strict 2PL