K.K.D.A.K.Indrajith -16000579 (2016cs057)
Assignment 01

1. Because of the concept of Multiprogramming OS allows executing multiple programs at the same time. Hence many users can access the database at the same time. If there is one CPU multiprogramming is achieved by suspend one process in CPU and execute another one. This is also called **interleaving of processes**. If there are multiple CPUs, multiprogramming is achieved by **parallel processing**.

2.
   a. **Atomicity**
   Transaction should be either complete entirely or not (atomic unit of processing). **Transaction recovery subsystem** ensures the atomicity. It should redone or undone when failures occurs.

   b. **Consistency Preservation**
   Transaction should take database from one consistent state to another consistent state. If a transaction executed from its beginning to end without interference from other transaction.
   Ensures by the programmers who written database programs or DBMS modules that enforces integrity constraints (primary key, foreign key etc).

   c. **Isolation**
   Transaction should be appearing as an isolated unit even though many transactions are executing concurrently. Execution of transaction should not be interfered with any other transaction executing concurrently.
   Enforce by the **concurrency control subsystem** in DBMS.

   d. **Durability or Permanency**
   Changes applied by a committed transaction must not be lost because of any failure.

3.
   a. Serial Schedule
      i. Schedule S is serial if, for every transaction T participating in the schedule, all operations in T should executed one after the other. Therefore in a serial schedule only one transaction is active at a time. Commit of the active transaction initiates the execution of the next transaction. There are no interleaving operations occurs. If we consider transactions to be independent every serial schedule is consider correct and does not matter which transaction occur first. And operations in one transaction are isolated from other transactions.
      ii. Problems
         1. If a transaction waits for I/O we cannot switch the CPU for another transaction.
         2. If some transaction T is quite long others must wait for T to complete.

   b. Serializable Schedule
      i. A schedule S of n transaction is Serializable if it is equivalent to some serial schedule of the same n transactions. Schedules considered **result equivalent** if they produce the same final state in the database. And two Schedules considered **conflict equivalent** if the order of any two conflicting operations is same in both schedules. We can reorder non-conflicting operations until it form equivalent serial schedule.

4. K
   a.

T1	T2	T3
read_item(X)		
		read_item(X)
write_item(X)		
	read_item(X)	

		write_item(X)

Answer- We cannot reorder non-conflicting operations in this schedule because r3(X) and w1(X), w1(X) and w3(X) are conflicting. Hence this is a non-Serializable schedule.

b.

T1	T2	T3
read_item(X)		
		read_item(X)
		write_item(X)
write_item(X)		
	read_item(X)	

Answer - r3(X),w3(X),r1(X),w1(X),r2(X).

c.

T1	T2	T3
		read_item(X)
	read_item(X)	
		write_item(X)
read_item(X)		
write_item(X)		

Answer – r2(X),r3(X),w3(X),r1(X),w1(X)

d.

T1	T2	T3
		read_item(X)
	read_item(X)	
read_item(X)		
		write_item(X)
write_item(X)		

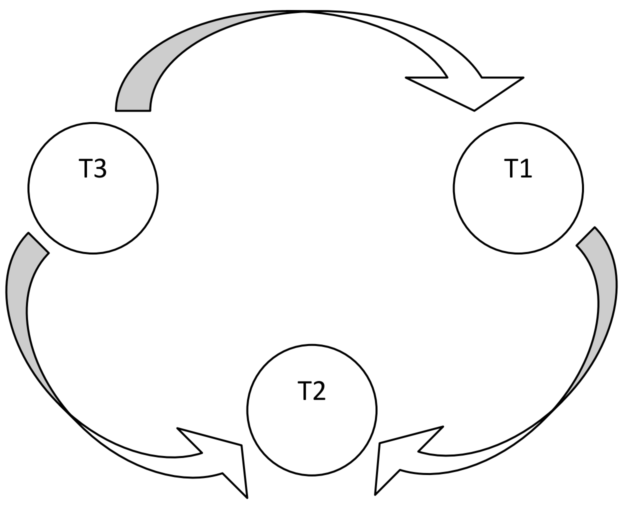
Answer - We cannot reorder non-conflicting operations in this schedule because {r1(X) and w3(X)}, {w3(X) and r1(X)} are conflicting. Hence this is a non-Serializable schedule.

5.

a.

T1	T2	T3
read_item(X)		
	read_item(Z)	
read_item(Z)*		
		read_item(X)-
		read_item(Y)+
write_item(X)-		
		write_item(Y)
	read_item(Y)	
	write_item(Z)*	
	write_item(Y)+	

K.K.D.A.K.Indrajith -16000579 (2016cs057)



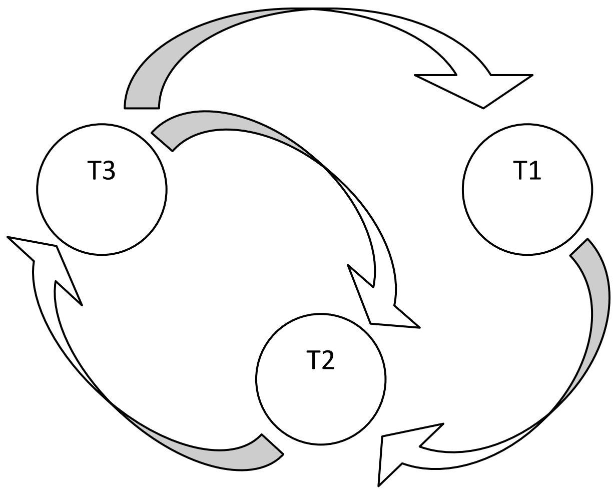
Answer – Schedule is serializable. There are no cycles contain.

T1	T2	T3
		read_item(X)-
		read_item(Y)+
		write_item(Y)
read_item(X)		
read_item(Z)*		
write_item(X)-		
	read_item(Z)	
	read_item(Y)	
	write_item(Z)*	
	write_item(Y)+	

R3(X), R3(Y), W3(Y), R1(X), R1(Z), W1(X), R2(Z), R2(Y), W2(Z), W2(Y)

b. F

T1	T2	T3
read_item(X)		
	read_item(Z)	
		read_item(X)*
read_item(Z)-		
	read_item(Y)+	
		read_item(Y)^
write_item(X)*		
	write_item(Z)-	
		write_item(Y)+$
	write_item(Y)^$	



K.K.D.A.K.Indrajith -16000579 (2016cs057)

Answer – Schedule is not Serializable.