CONCURRENT EXECUTIONS

A transaction is a unit of database processing which contains a set of operations. For example, deposit of money, balance enquiry, reservation of tickets etc.

Every transaction starts with **begin** transaction and terminates with **end** transaction.

The set of operations within these two delimiters constitute one transaction.

Example:

```
main()
{
  begin transaction
} end transaction
```

There are three possible ways in which a transaction can be executed. These are as follows:

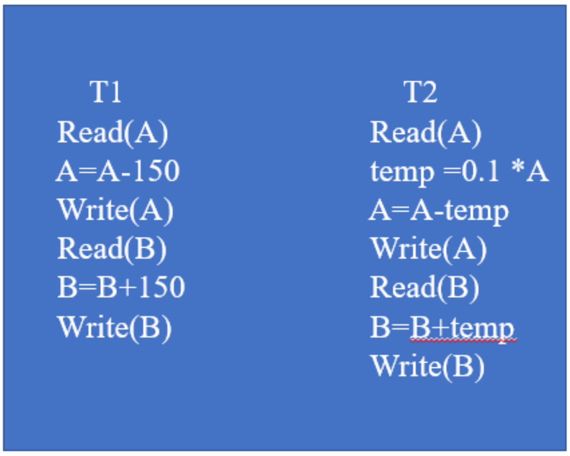
1. Serial execution.
2. Parallel execution.
3. Concurrent execution.

CONCURRENT EXECUTION OF TRANSACTION

1. In the transaction process, a system usually allows executing more than one transaction simultaneously.
2. Concurrent transaction or execution includes multiple transactions which are executed concurrently or simultaneously in the system.
3. This process is called a concurrent execution.
4. Concurrent schedules do not keep the database in the consistent state. The concurrent execution of the transactions has to be carried out in a controlled environment.
5. If a system consists of ‘n’ number of transactions, we will have more than ‘n’ number of concurrent schedules, out of which only a few of them are keeping the database in consistent state.

Example

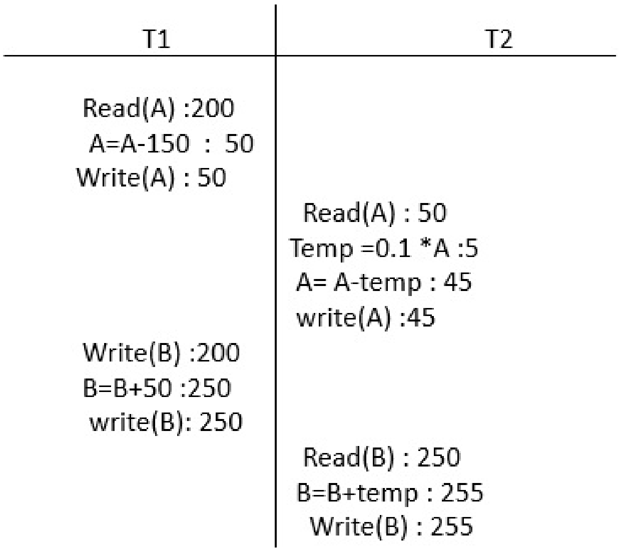
6. Given below is an example of serial execution −
7. Let us consider two transactions T1 and T2 where T1 performs transfer Rs.150 from account A to account B and similarly T2 transfers 10% of balance from A to B.



T1	T2
Read(A)	Read(A)
A=A-150	temp =0.1 *A
Write(A)	A=A-temp
Read(B)	Write(A)
B=B+150	Read(B)
Write(B)	B=B+temp
	Write(B)

Example

The concurrent execution given below is in a consistent state.



T1	T2
Read(A) :200 A=A-150 : 50 Write(A) : 50	

	Read(A) : 50
	Temp =0.1 *A :5
	A= A-temp : 45
	write(A) :45

Write(B) :200 B=B+50 :250 write(B): 250	
	Read(B) : 250 B=B+temp : 255 Write(B) : 255

ADVANTAGES

The advantages of the concurrent transactions are as follows:

1. Increases throughput which is nothing but number of transactions completed per unit time.
2. It reduces the waiting time.

Example

T1= 90sec

T2= 500sec

T3= 5sec.

If we execute serially by T1->T2->T3 then transaction T3 waits for 590 sec, so we go for non-serial or concurrent transactions to reduce waiting time.

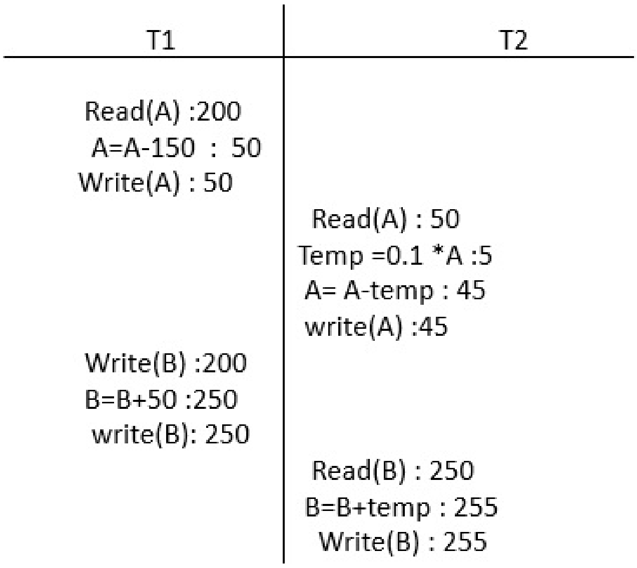
i.e. T3 -> T1 -> T2.

DISADVANTAGE

The disadvantage is that the execution of concurrent transactions may result in inconsistency.

Example

The concurrent execution given below is in a consistent state.



T1	T2
Read(A) :200	
A=A-150 : 50	
Write(A) : 50	
	Read(A) : 50
	Temp =0.1 *A :5
	A= A-temp : 45
	write(A) :45
Write(B) :200	
B=B+50 :250	
write(B): 250	
	Read(B) : 250
	B=B+temp : 255
	Write(B) : 255