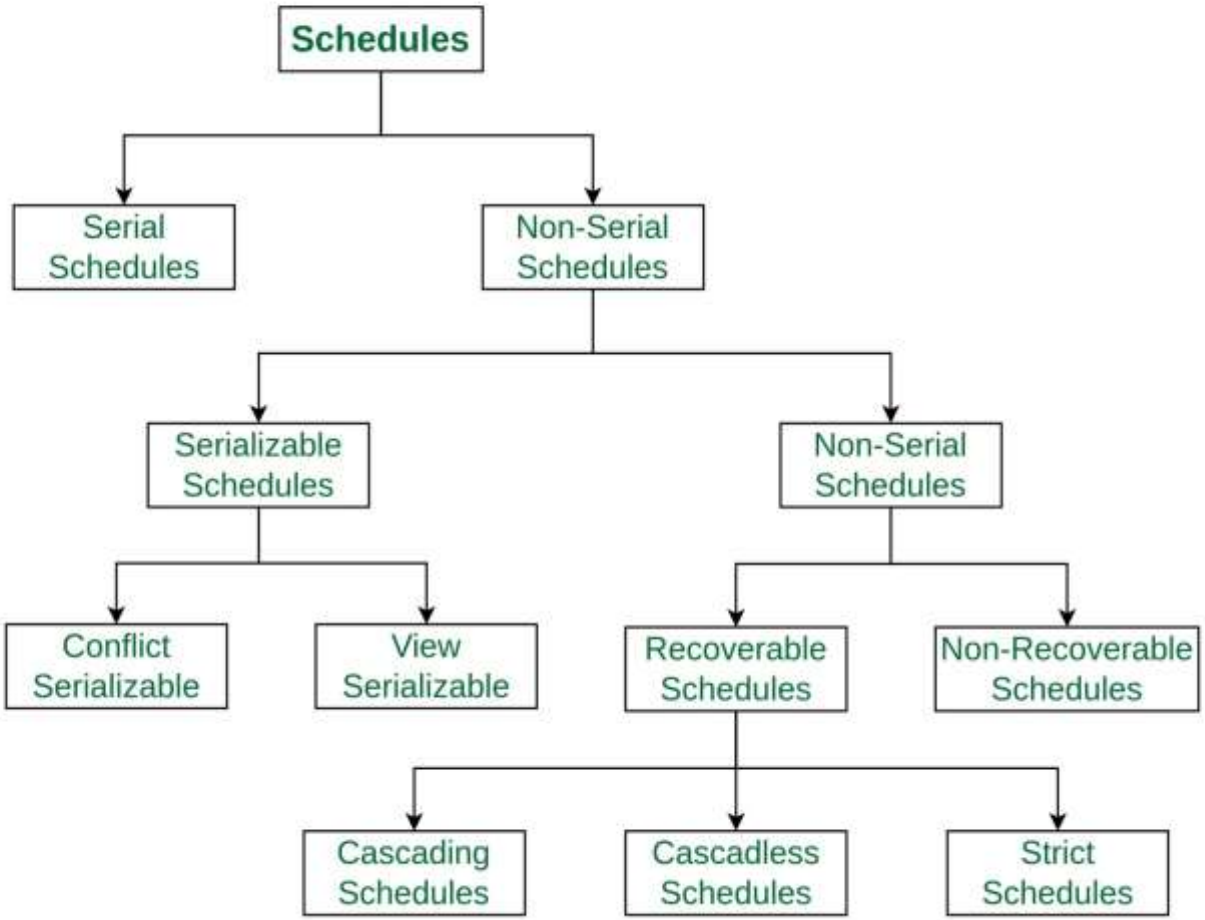
UNIT IV

TRANSACTIONS

Prepared by,
K.Revathi AP/IT, SNSCE

Types

Types of schedules in DBMS



Serializable

- The non-serial schedule is said to be in a serializable schedule.
- Multiple transactions can execute concurrently.

Types

(1) Conflict Serializable:

It can be transformed into a serial schedule by swapping non-conflicting operations.

T_1	T_2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

Before swapping

T_1	T_2
read(A)	
write(A)	
read(B)	
write(B)	
	read(A)
	write(A)
	read(B)
	write(B)

After swapping

(2) View Serializable:

It is view equal to a serial schedule (no overlapping transactions).

T1	T2
Read(A)	
	Write(A)

Schedule S1

T1	T2
	Write(A)
Read(A)	

Schedule S2

Non-Serializable

Schedules in which transactions commit only after all transactions whose changes they read commit are called recoverable schedules.

Types:

(1). Recoverable Schedules

(2). Irrecoverable Schedules

(1)Recoverable Schedules

T_1 commits before T_2, that makes the value read by T_2 correct.

T_1	T_2
R(A)	
W(A)	
	W(A)
	R(A)
commit	
	commit

(2) Irrecoverable Schedule

Tj is reading the value updated by Ti and Tj is committed before committing of Ti, the schedule will be irrecoverable.

T1	T1's buffer space	T2	T2's Buffer Space	Database
				A=5000
R(A);	A=5000			A=5000
A=A-100;	A=4000			A=5000
W(A);	A=4000			A=4000
		R(A);	A=4000	A=4000
		A=A+500;	A=4500	A=4000
		W(A);	A=4500	A=4500
		Commit;		
Failure Point				
Commit;				

Recoverable with Cascading Rollback

If Tj is reading value updated by Ti and commit of Tj is delayed till commit of Ti, the schedule is called recoverable with cascading rollback.

T1	T1's buffer space	T2	T2's Buffer Space	Database
				A=5000
R(A);	A=5000			A=5000
A=A-100;	A=4000			A=5000
W(A);	A=4000			A=4000
		R(A);	A=4000	A=4000
		A=A+500;	A=4500	A=4000
		W(A);	A=4500	A=4500
Failure Point				
Commit;				
		Commit;		

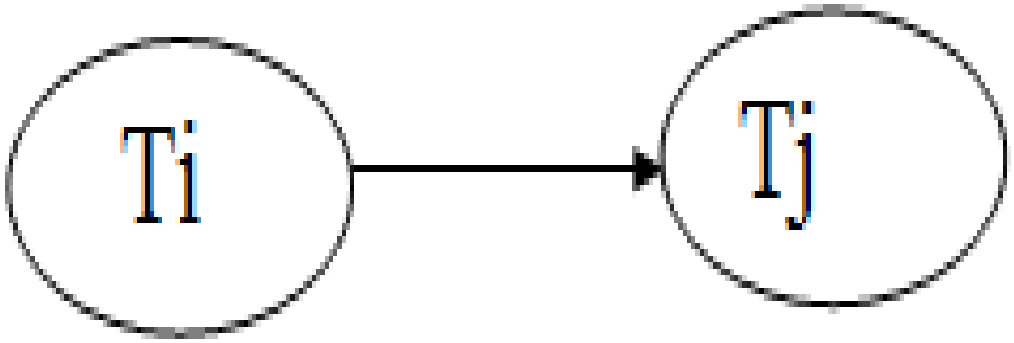
Cascadeless Recoverable Rollback:

If Tj reads value updated by Ti only after Ti is committed, the schedule will be cascadeless recoverable.

Testing of Serializability

- Serialization Graph is used to test the Serializability of a schedule.
- Three conditions holds: Ti → Tj
  - Create a node Ti → Tj if Ti executes write (Q) before Tj executes read (Q).
  - Create a node Ti → Tj if Ti executes read (Q) before Tj executes write (Q).
  - Create a node Ti → Tj if Ti executes write (Q) before Tj executes write (Q).

Precedence graph for Schedule S



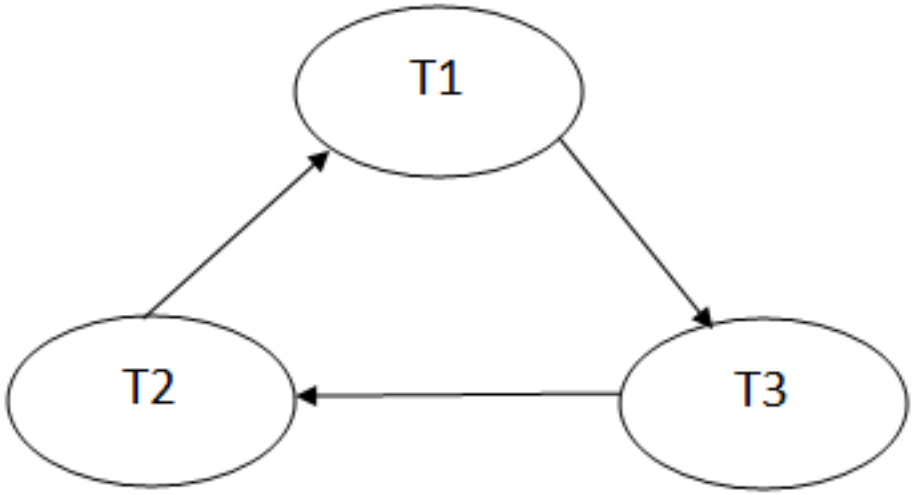
Example

Time

T1	T2	T3
Read(A)		
	Read(B)	
A:= f_1(A)		
		Read(C)
	B:= f_2(B)	
	Write(B)	
		C:= f_3(C)
		Write(C)
Write(A)		
		Read(B)
	Read(A)	
	A:= f_4(A)	
Read(C)		
	Write(A)	
C:= f_5(C)		
Write(C)		
		B:= f_6(B)
		Write(B)

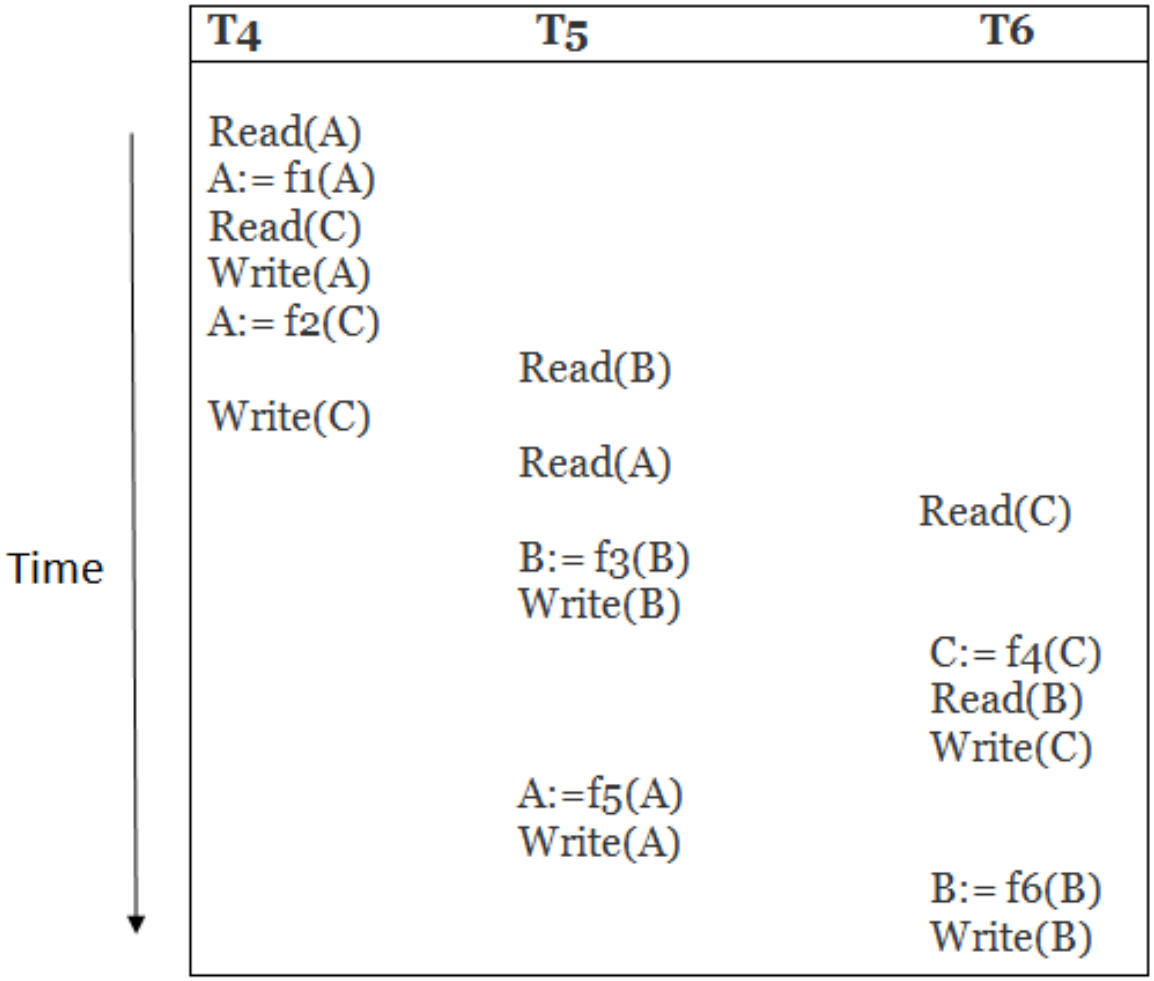
Schedule S1

Precedence graph for schedule S1:



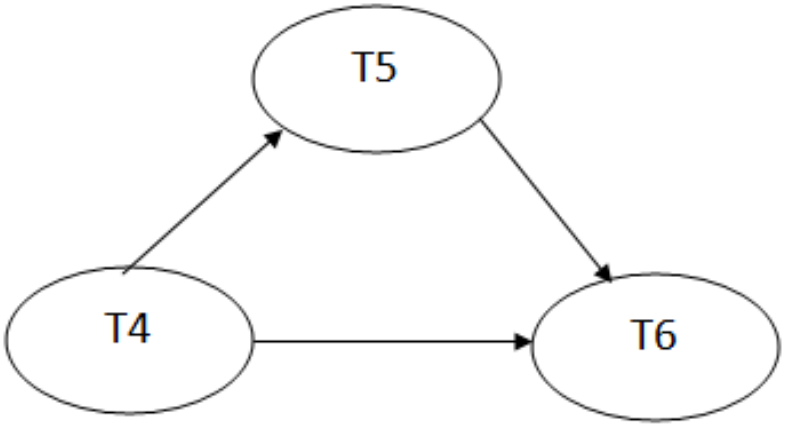
- The precedence graph for schedule S1 contains a cycle
- Schedule S1 is non-serializable.

Example 2



Schedule S2

Precedence graph for schedule S2:



- The precedence graph for schedule S2 contains no cycle
- ScheduleS2 is serializable.