Types of Schedules based Recoverability in DBMS

1. Recoverable
2. Cascadeless
3. Strict

Recoverable Schedule:

A schedule is said to be recoverable if it is recoverable as name suggest. Only reads are allowed before write operation on same data. Only reads (Ti->Tj) is permissible.

Example –
S1: R1(x), w1(x), R2(x), R1(y), R2(y)

W2(x), w1(y), C1,C2;

Given schedule follows order of **Ti->Tj => C1->C2**. Transaction T1 is executed before T2 hence there is no chances of conflict occur. R1(x) appears before W1(x) and transaction T1 is committed before T2 i.e. completion of first transaction performed first update on data item x, hence given schedule is recoverable.
Lets see example of **unrecoverable schedule** to clear the concept more:

S1: R1(x), R2(x), R1(z), R3(x) , R3(y), w1(x),
W3(y), R2(y) , W2(z) , w2(y), C1, C2, c3;

Ti->Tj => C2->C3 but W3(y) executed before W2(y) which leads to conflicts thus it must be committed before T2 transaction. So given schedule is unrecoverable.
if **Ti->Tj => C3->C2** is given in schedule then it will become recoverable schedule.

Note: A committed transaction should never be rollback. It means that reading value from uncommitted transaction and commit it will enter the current transaction into inconsistent or unrecoverable state this is called *Dirty Read problem*.
Example:

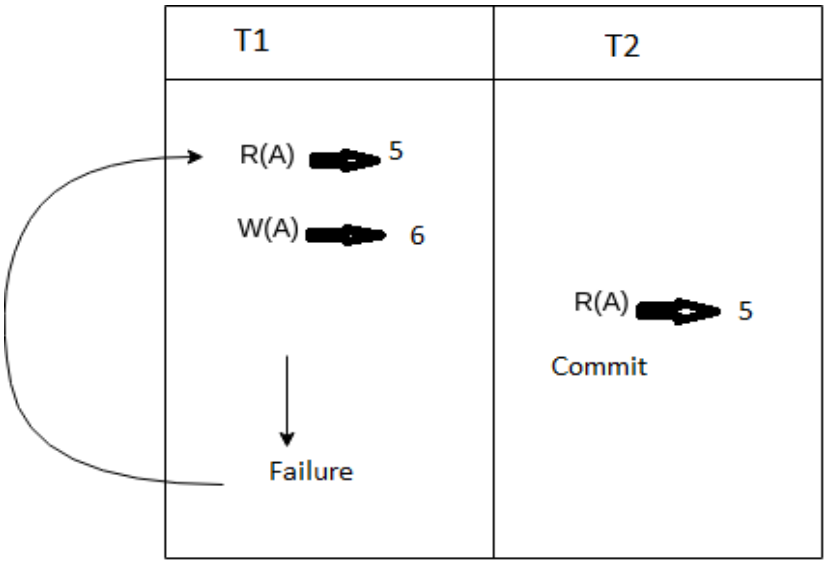
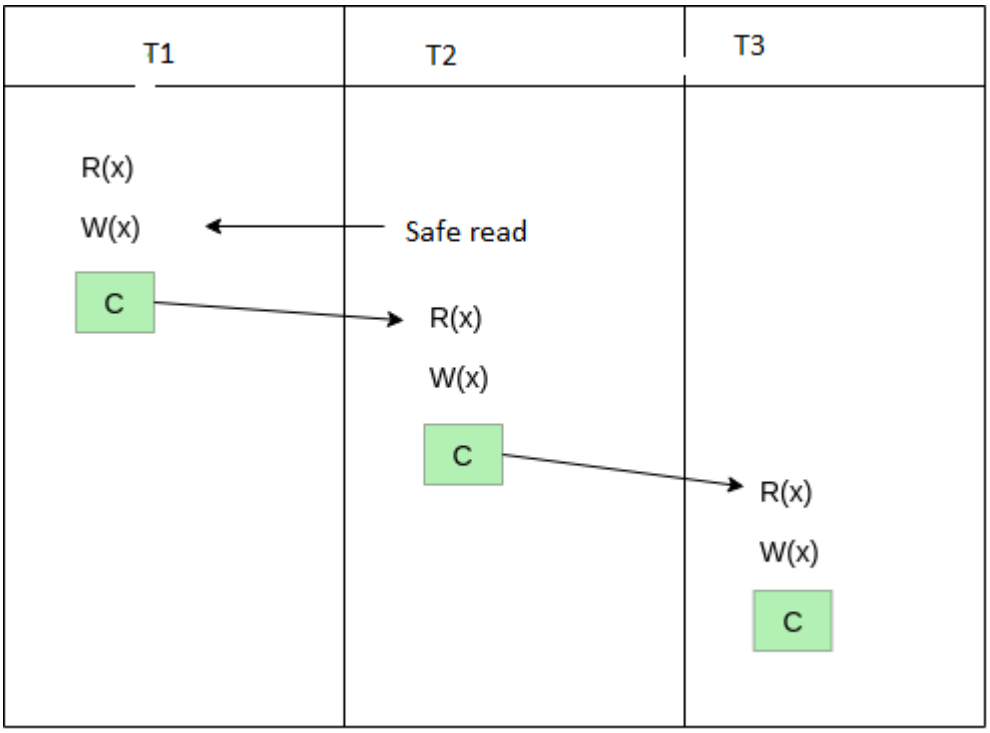


Figure - Dirty Read Problem

2. Cascadeless Schedule –
When no **read** or **write-write** occurs before execution of transaction then corresponding schedule is called cascadeless schedule.
Example –
S3: R1(x), R2(z), R3(x), R1(z), R2(y), R3(y), W1(x), C1,
W2(z), **W3(y)**, **W2(y)**, **C3**, **C2**;

In this schedule **W3(y)** and **W2(y)** overwrite conflicts and there is no read, therefore given schedule is cascadeless schedule.

As given below all the transactions are reading committed data hence it's cascadeless schedule.



3. Strict Schedule –
if schedule contains no **read** or **write** before commit then it is known as strict schedule. Strict schedule is strict in nature.
Example –
S4: R1(x), R2(x), R1(z), R3(x), R3(y),
(x), **C1**, **W3(y)**, **C3**, R2(y), **W2(z)**, **W2(y)**, **C2**;

In this schedule no read-write or write-write conflict arises before commit hence its strict schedule:

T1	T2	T3
R1(x)		
	R2(x)	
R1(z)		
		R3(x)
		R3(y)
W1(x)		
C1;		
		W3(y)
		C3;
	R2(y)	
	W2(z)	
	W2(y)	
	C2;	

Figure - Strict Schedule

4. Cascading Abort –
Cascading Abort can also be rollback. If transaction T1 abort as T2 read data that written by T1 which is not committed. Hence it's cascading rollback.
Example:

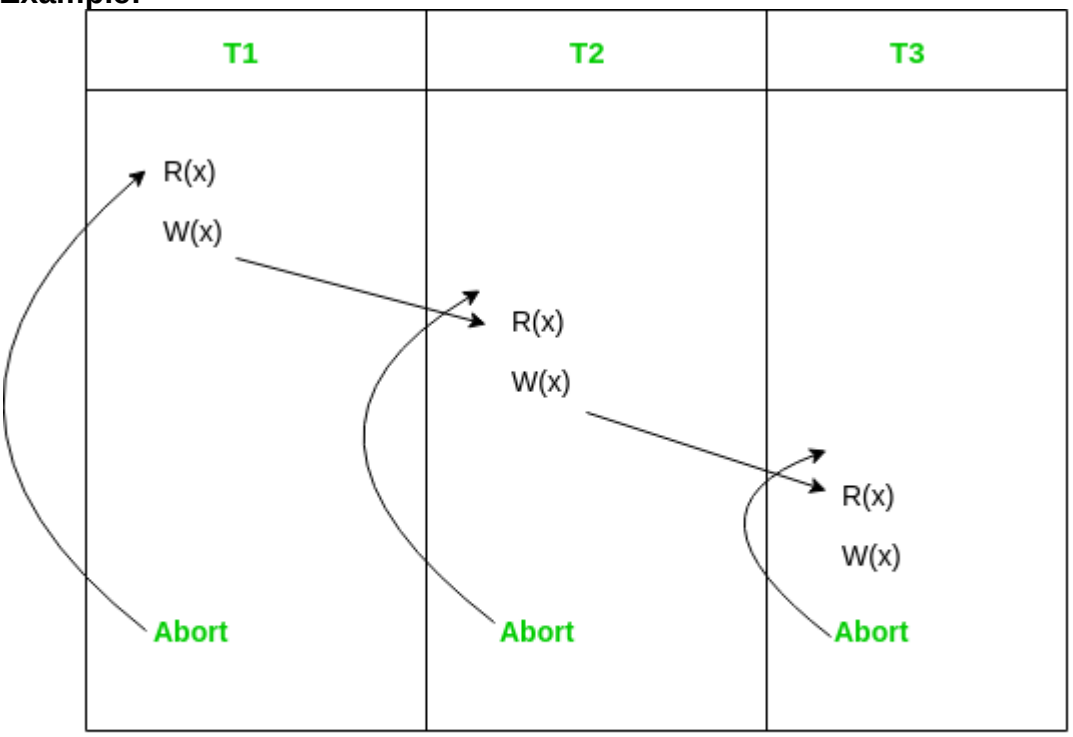


Figure - Cascading Abort

Corelation between Strict, Cascadeless and Recoverable schedule:

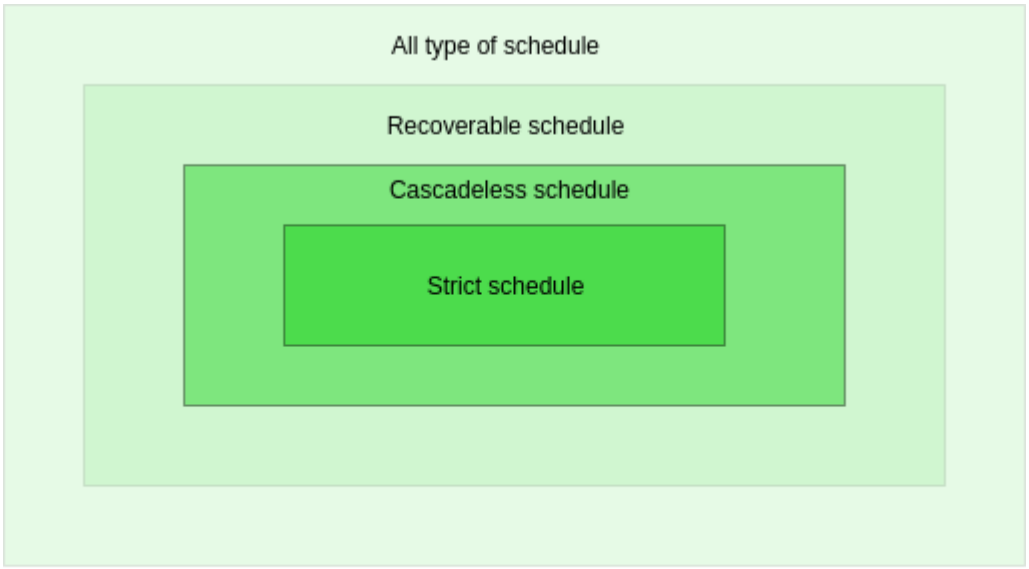


Figure - Venn diagram of these schedules

From above figure:

1. Strict schedules are all recoverable and cascadeless schedules
2. All cascadeless schedules are recoverable