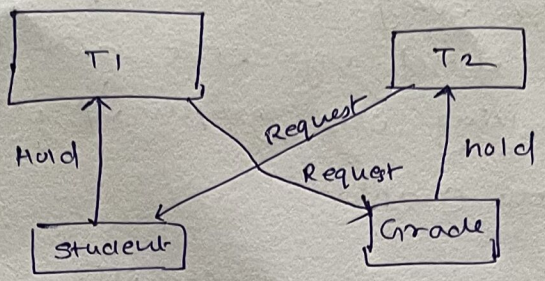
Deadlock in DBMS:- A deadlock is a condition where two or more transactions are waiting indefinitely for one another to give up locks.

Deadlock is said to be one of the most feared complications of DBMS as no task ever gets finished and is in waiting forever.



Deadlock in DBMS.

Deadlock avoidance ⇒ This method is used to detect any deadlock situation in advance. A method like "wait for graph" is used for detecting the deadlock situation but this method is suitable only for the smaller database. For the larger database, deadlock prevention method can be used.

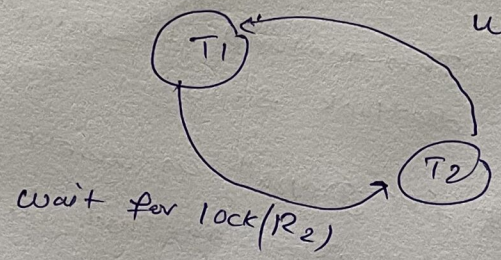
Deadlock detection:- In a database, when a transaction waits indefinitely to obtain a lock, then the DBMS should detect whether the transaction is involved in a deadlock or not. The lock manager maintains a wait for the graph to detect the deadlock cycle in the database.

Wait for Graph:-

* This is the suitable method for deadlock detection. In this method, a graph is created based on the transaction and their lock. If the created graph has a cycle or closed loop, then there is a deadlock.

* The wait for the graph is maintained by the system for every transaction which is waiting for some data held by the others. The system keeps checking the graph if there is any cycle in the graph.

The wait for a graph for the above scenario is shown below,



* Deadlock prevention:- is suitable for a large database. If the resources are allocated in such a way that deadlock never occurs, then the deadlock can be prevented.

T_1 : A $\xrightarrow{50}$ B

```
Lock-X(A)
Read(A)
A = A - 50
Write(A)
Unlock(A)
Lock-X(B)
Read(B)
B = B + 50
Write(B)
Unlock(B)
```

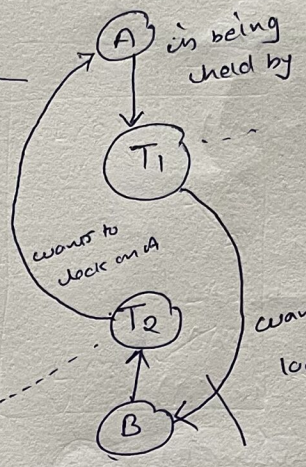
T_2 : Display (B+A)

```
Lock-S(B)
Read(B)
Unlock(B)
Lock-S(A)
Read(A)
Unlock(A)
Display(B+A)
```

T_1	T_2
X(A)	
R(A)	
A = A-50	
W(A)	
U(A)	
	S(B)
	R(B)
	U(B)
	S(A)
	R(A)
	U(A)
	Display (B+A)
X(B)	
R(B)	
B = B+50	
W(B)	
U(B)	

	A	B	3000
Before T_1 :	1000	2000	
After T_1 :	950	2050	

T_1	T_2
X(A)	
R(A)	
A = A-50	
W(A)	
	S(B)
	R(B)
	S(A)
	R(A)
	display (B+A)
	commit
	U(B)
	U(A)
X(B)	
R(B)	
B = B+50	
W(B)	
Commit	
U(B)	
U(A)	



There is a cycle.

T_1	T_2
L(A)	
U(A)	
	L(B)
	U(B)
L(B)	
U(B)	

$T_2 \rightarrow T_1$

T_1	T_2
L(A)	
U(A)	
	L(A)
	U(A)
	L(B)
	U(B)
L(B)	
U(B)