UNIT :- 4

What is Transaction?

⇒ A transaction is a logical unit of work of database processing that includes one or more database access operations.

⇒ A transaction can be defined as an action or series of actions that's carried out by a single user or application program to perform operations for accessing the contents of the database.

⇒ The operations can include vertical (Read), insertion (write), deletion and modification.

⇒ A transaction must be either completed or aborted.

* Different operations performed in a transaction :-

Operations	Descriptions
Read (x)	Read operation is used to read the value of x from the database and stores it in a buffer in main memory.
Write (x)	Write operation is used to write the value back to the database from the buffer.
Commit	It is used to save the workdone permanently in the harddisk.
Roll back	It is used to undo the workdone.

Properties of Transaction :-

A transaction is having four properties. Such as

i) Atomicity
ii) Consistency
iii) Isolation
iv) Durability

i) Atomicity :-

= Atomicity means all successful or none.
= It states that all operations of the transaction take place at once if not, the transaction is aborted.
= There is no midway i.e. the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

ii) Consisten.

= Atomicity involves the following two operations:

i) Abort : If a transaction aborts then all the changes made are not visible.

ii) Commit : If a transaction commits then all the changes made are visible,

ii) Consistency :-

⇒ The integrity constraints are maintained so that the database is consistent before and after the transaction.

⇒ The execution of a transaction will leave a database in either it's prior stable state or a new stable state.

⇒ The consistent property of database states that every transaction sees a consistent database instance.

⇒ The transaction is used to transform the database from one consistent state to

iii) ISOLATION :-

= It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.

= In isolation, if the transaction T_1 is being executed and using the data items X, then that data item can't be accessed by any other transaction T_2 until the transaction T_1 ends.

= The concurrency control subsystem of the DBMS enforced the isolation property.

(Converting parallel to Serial Schedule transaction)

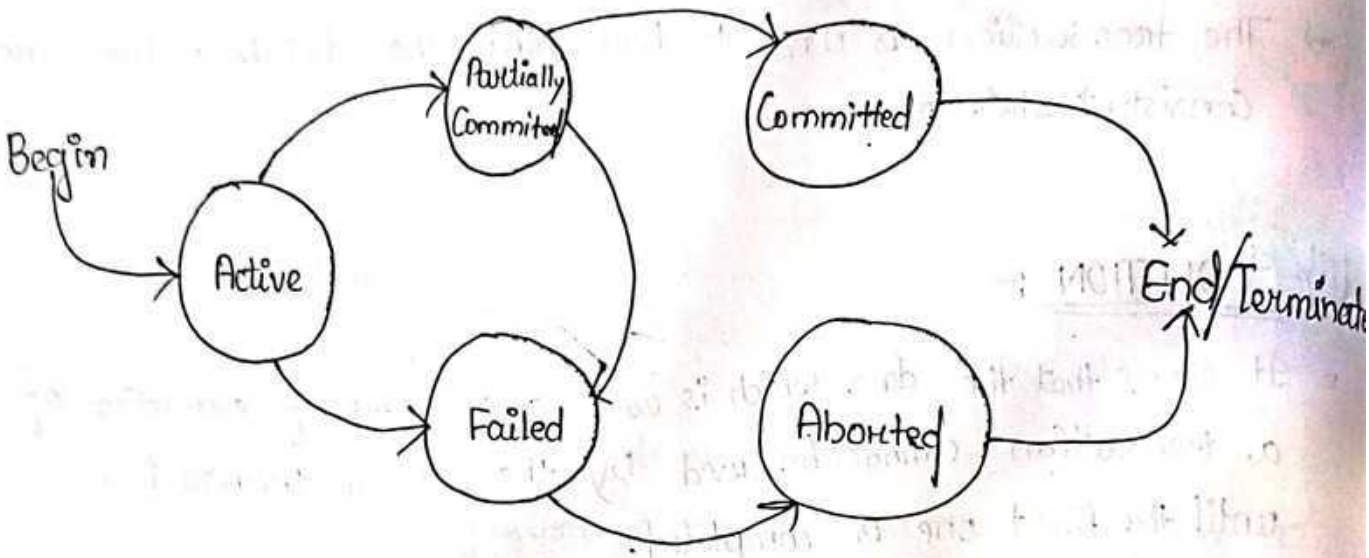
iv) Durability :-

= The database should be durable enough to hold all its latest updates even if the system fails or restards.

= If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data.

= If a transaction commits but the system fails before the data could be written on the disk, then that data will be updated once the system springs back into action.

Stages Of Transaction :-



Stages of Transaction -

1) **Active State :**
In this state, the transaction is being executed. This is the initial state of every transaction.

11) **Partially Committed :**
When a transaction executes its final operation, but the data is not saved in disk, it is said to be in a partially committed state.

111) **Failed :**
A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.

IV) **Aborted :**
If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all it's write operations on the database to bring the database back to it's original state where it was prior to the execution of the transaction. Transactions in this state are called aborted.

The database recovery module can select one of the two operators after a transaction aborts -

* Re-start the transaction
* Kill the transaction

v) Committed :

If a transaction execute all its operations successfully, it's said to be committed. All it's effect are now permanently

Schedule :-

When several transaction are executing concurrently then the order of the execution of various transaction is known as Schedule.

= There are 2 types of schedule :

1) Serial Schedule -: It is a type of schedule where one transaction executed completely before starting of another transaction.

Ex :-

T_1	T_2
R(A)	
A = A - 100	
W(A)	
R(B)	
B = B + 100	
W(B)	
	R(A)
	A = A - 200
	W(A)

$T_1 \longrightarrow T_2$.

Non Serial Schedule :- In this multiple transaction execute simultaneously. The other transaction proceed without completion of previous transaction.

= All the transaction are interleaved.

T_1	T_2
R(A)	
A = A - 10	
W(A)	
	R(B)
	B = B + 10
	W(B)
R(A)	
A = A + 100	
W(A)	

Advantages :-

i) Waiting time is decrease

ii) Responsive time will decrease

iii) Resource will increase

iv) Efficiency will increase.

Disadvantages :-

i) WR conflict.

ii) RW conflict.

iii) WW conflict.

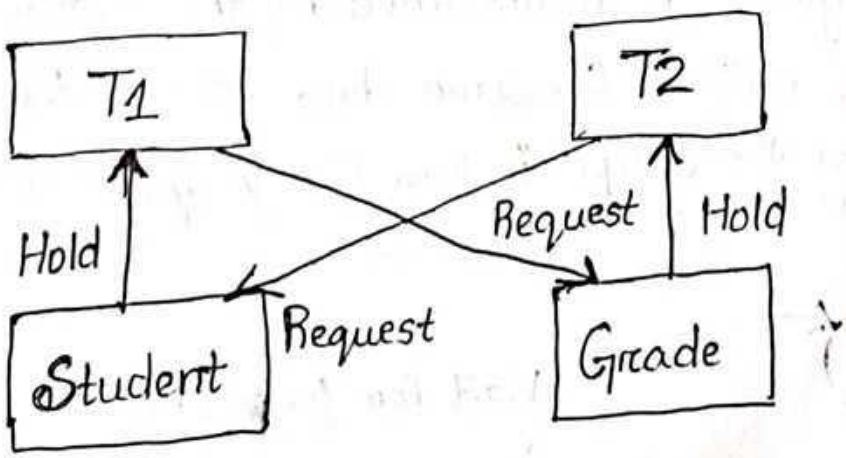
DEADLOCK :- L.8

= 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 in DBMS as no task ever gets finished and is in waiting state forever.

= Example of Deadlock -

In the student table, transaction T_1 holds a lock on some rows and needs to update some rows in the grade table. Simultaneously, transaction T2 holds locks on some rows in the grade table and needs to update the rows in the Student table held by Transaction T_1.



Deadlock Avoidance :-

= When a database is stuck in a deadlock state, then it is better to avoid the database

Deadlock Detection :-

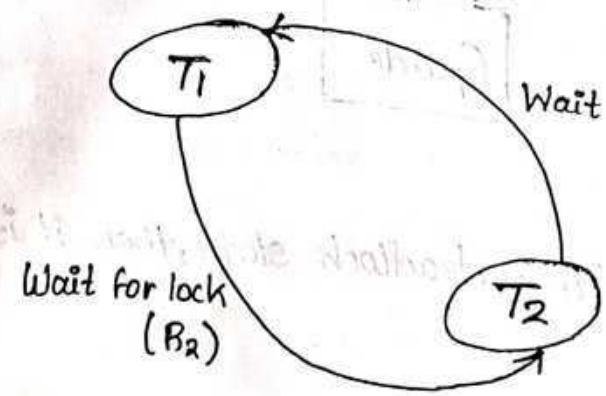
= In a database, when a transaction waits indefinitely to obtain a lock, then the DBMS should detect whether the transaction is involved in a deadlock or not.

= 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.



Deadlock Prevention :-

= Deadlock prevention method 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.

= The Database management system analyzes the operations of the transaction whether they can create a deadlock situation or not. If they do, then the DBMS never allowed that transaction to be executed.

WAIT - DIE SCHEMA :-

= In this schema, if a transaction requests for a resource which is already held with a conflicting lock by another transaction then the DBMS simply checks the timestamp of both transaction. It allows the older transaction to wait until the resource is available for execution, otherwise it will rollback (dies).

WOUND WAIT SCHEMA :-

= In wound wait schema, if the older transaction requests for a resource which is held by the younger transaction, then older transaction forces younger one to kill the transaction and release the resource. After the minute delay, the younger transaction is restarted but with the same timestamp.

= If the older transaction has held a resource which is requested by the younger transaction, then the younger transaction is asked to wait until older releases it.

Deadlock Recovery :-

= When a detection algorithm determines that a deadlock exists, the system must recover from the deadlock.

= The most common solution is to rollback one or more transactions to break the deadlock.

= Three actions need to be taken -

i) Selection of a Victim :

⇒ Given a set of deadlocked, we must determine which transaction transactions to roll back to break the deadlock.

⇒ We should rollback those transactions that will give the minimum cost.

ii) Rollback :

= Once we have decided that a particular transaction must be rolled back, we must determine how far this transaction should be rolled back.

= The simplest solution is a total rollback; Aborct the transaction and then restart it.

= Partial rollback requires the system to maintain additional information about the state of all ruining transactions.

iii) Starvation :

= In the case of a selection of victim, it may happen that the same transaction is always picked as a victim.

= As a result, this traction never completes its designated task, thus this is a starvation..

= We must ensures that a transaction can be picked as a victim only a small (Finite) number of times. The most common solution is to include the number of rollback in the cost factor.

DATABASE RECOVERY :-

= Database Systems like any other computer system, are subjected to failures but the data stored in them must be available as and when required.

= Database recovery is the process of restoring the database to a consistent state in the event of a failure.

= It is the process of restoring the database to the most recent consistent state that existed shortly before the time of system failure.

= Types of Recovery Techniques :- There are mainly two types of recovery techniques used in DBMS :-

i) Rollback/Undo Recovery Technique.

ii) Commit/Redo Recovery Technique.

i) Rollback / Undo Recovery Technique :-

= The rollback / undo recovery technique is based on the principle of backing out or undoing the effects of a transaction that has not been completed successfully due to a system failure or error.

= This technique is accomplished by undoing the changes made by the transaction using the log records stored in the transaction log.

ii) Commit / Redo Recovery Technique :-

= The commit/redo recovery technique is based on the principle of reapplying the changes made by a transaction that has been completed successfully to the database.

= This technique is accomplished by using the log records stored in the transaction log to redo the changes made by the transaction that was in progress at the time of the failure or error.

CHECKPOINT RECOVERY :-

= Checkpoint Recovery is a technique used to reduce the recovery time by periodically saving the state of the database in a checkpoint file.

= In the event of a failure, the system can use the checkpoint file to restore the database to the most recent consistent state before the failure occured, rather than going through the entire log to recover the database.

BACKUP :-

= Database backup is the process of creating a copy (backing up) of an organization's structured data utilized by popular databases. Different types of backups are —

i) Full Database Backup.
ii) Differential Backup.
iii) Transaction Log Backup.

Backup Techniques :-

- Full database Backup – In this full database including data and database.
- Differential Backup – It stores only the data changes that have occured since th last full database backup.

Approaches to modify database :-

1) Deferred database modification :-

= The deferred modification technique occurs if the transaction does not modify the database until it has committed.

= In this method, all the logs are created and stored in the stable storage, and the database is updated when a transactions commits.

2) Immediate database modification :-

= The immediate modification technique occurs if database modification occurs while the transaction is still active

= In this technique, the database is modified immediately after every operation. It follows an actual database modification.

Example 1) Deferred database transaction

Database	T_1	
A = ~~100~~ 200	R(A)	$\langle T_1, Start \rangle$
B = ~~200~~ 400	A = A+100 = 200	$\langle T_1, A, 200 \rangle$
	W(A)	$\langle T_1, B, 400 \rangle$
	R(B) – 200	$\langle T_1, Commit \rangle$
	B = B+200 = 400	Redo.
	W(B)	
	Commit	

Immediate database modifications -

A = ~~100~~ 200
B = ~~200~~ 400.

Database

T_1	
R(A) — 100	$\langle T_1, \text{Start} \rangle$
A = A+100 = 200	$\langle T_1, A, 100, 200 \rangle$
W(A)	
R(B) — 200	$\langle T_1, B, 200, 400 \rangle$
B = B+200 = 400.	
W(B)	$\langle T_1, \text{Commit} \rangle$
Commit	Redo.

* If commit will not present, then it will perform the undo operations. else Redo.

3) Shadow Paging :-