

# Assignment No. 4

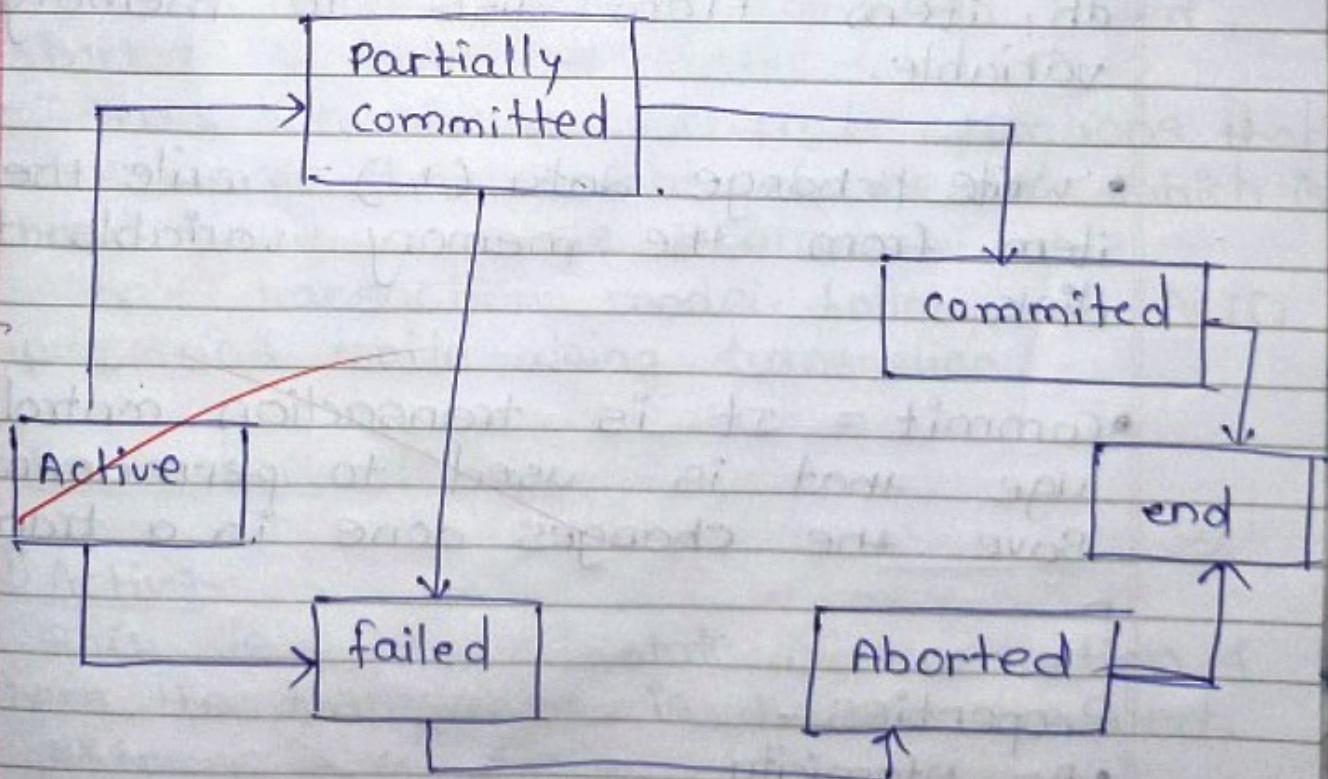
(Q.1) Explain Transaction.

→ A transaction is a single logical unit of work which access & possibly modifiers the considered of database.  
Tr

Transaction access data using read & write operations.

In order to maintain consistency in a database before & after the transaction certain example,

A transaction is any process a user perform after successfully logging in



Active -

The transaction is being exerted that is the initial state of every transaction.

Partially Committed -

When a transaction executes its final operation. It is said to be in partially committed state.

Transactions in the state are called Basic database access operation -

- Read or access data (R) = Accessing the db item from disk to memory variable.
- write | change data (w) = write the data item from the memory variable to the disk.
- Commit = It is transaction control language that is used to permanently save the changes done in a transaction.

Properties -

- A = Atomicity
- C = consistency
- I = Isolation

• D = Durability

Advantages -

- ① flexibility
- ② improved data sharing
- ③ Data integration
- ④ Data security
- ⑤ Data consistency
- ⑥ Data integrity

Q.2) Write short note on simple transaction model.

- It is model of transaction how it must be.
- It has active, partially committed, failed, aborted & committed States.
- Transaction is a several operations that can change the content of the db which is handled by a single program.
- simple transaction model follows all ACID properties while doing transaction.

### Transaction states

#### ① Active-

This is the 1<sup>st</sup> state of transaction & here the transaction is being executed.

ex:-

updating | inserting | deleting a record is done. here, but it is still not saved to

the db.

## ② Partially Committed -

- This is also an execution phase of transaction.

- Here last step of transaction is executed.

- But data is still not saved to the db.

Ex -

calculating total marks, final display, the total marks step is executed in this state.

## ③ Failed

- Because of the failure of the system or db if a transaction can't proceed to the execution state, then the transaction is said to be in failed state.

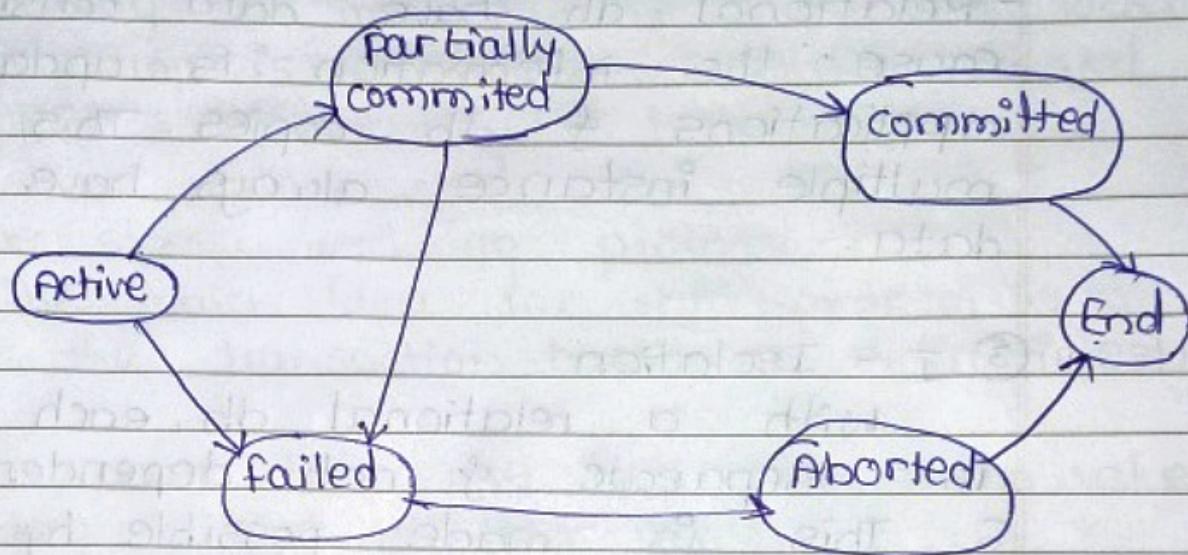
Ex -

If the db is not able fire a query to fetch the marks.

## ④ Aborted

- If a transaction is failed to execute, then the db recovery system will ensure that the db is in its previous consistent state.

If not , it brings the db to consistent state by ab or rolling back the transaction.



Q.3) Explain in details ACID properties .

### ① A : Atomicity

- Atomicity keeps data accurate.
- It makes sure all data is compliant with the rules, regulations & policies of the business.
- It also requires all tasks to succeed , or the transaction will roll back.
- Atomicity defines all the elements in a complete db transaction.

### ② C : consistency

The state of the db must remains consistent throughout the transaction.

- consistency defines the rule for

maintaining db points.

- This ensures they remain in a correct state after a transaction.
- Relational db have data consistency because the information is updated across applications & db copies. This means multiple instances always have the same data.

### ③ I - Isolation

- With a relational db, each transaction is separate & not dependent on others. This is made possible by isolation.
- Isolation keeps the effect of a transaction invisible until it is committed. This reduces the risk of confusion.

### ④ D - Durability

Durability means that you can recover data from a failed transaction.

It also ensures the data changes are permanent.

disadvantages -

- ① performance
- ② Scalability
- ③ Complexity
- ④ Management.

Q.4] Time stamp based protocol.

→ Concurrency control can be implemented in different ways.

- One way to implement it is by using locks
- Time stamp is a unique identifier created by DBMS to identify a transaction.

① Time stamp ordering protocol -

The main idea for this protocol is to order the transaction based on their timestamps.

To ensure this, use two time stamp values relating to each db item  $x$ .

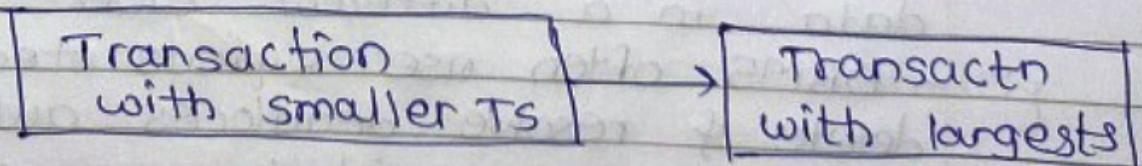
•  $W-TS(x)$  is the largest TS of any transaction that executed  $write(x)$  successfully.

•  $R-TS(x)$  is the largest TS of any transaction that executed  $read(x)$  successfully.

② Basic Time stamp ordering -

- Every transaction is issued a TS based on when it enters the system.

- The protocol manages concurrent execution such that the TS determine the serializability order.



### ③ strict TS ordering -

A variation of basic TO is called strict to ensures that the schedules are both strict & conflict serializable.

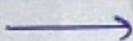
#### \* Advantages -

- ① High concurrency
- ② Efficient
- ③ No Deadlocks
- ④ Improved performance

#### \* Disadvantages -

- limited Granularity
- TS ordering
- TS synchronization
- TS allocation.

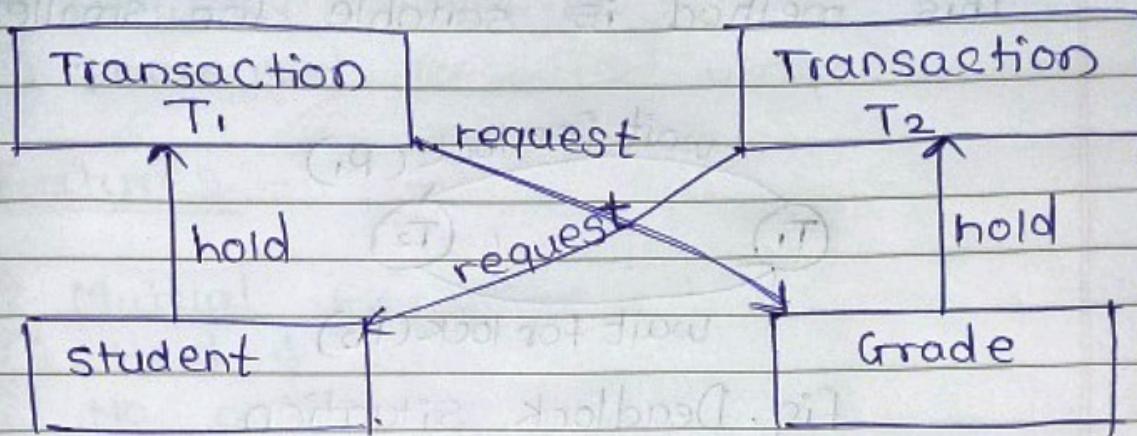
### Q.5) Explain Deadlock Handling Techniques.



- It occurs when two or more transactions are waiting for each other to release resources.
- It is in multi-user environment when two or more transactions are running concurrently & try to access the same data in a different order.
- DBMS often use various techniques to detect & resolve deadlocks automatically.
- Deadlock is said to be one of the most feared computations in DBMS

as it brings the whole system to a halt.

Ex -



## ① Deadlock Avoidance -

- When db is stuck in a deadlock, it is always better to avoid the deadlock rather than restarting.
- It is suitable for smaller db where as the deadlock prevention method is suitable for larger db.
- One method of avoiding deadlock is using application-consistent logic.
- It does not guarantee to remove deadlocks completely.

## ② Deadlock Detection -

when a transaction waits indefinitely to obtain a lock.

The db management system [DBMS] should detect whether the transaction is

involved in a deadlock or not.

- wait-for-graph - It is one of the methods for detecting the deadlock situation. This method is suitable for smaller db.

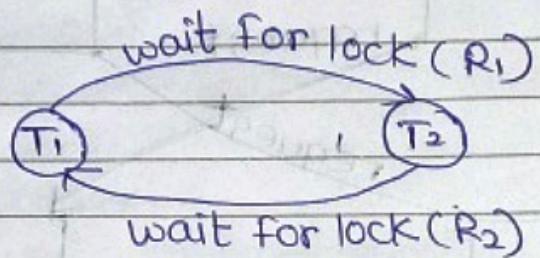


fig. Deadlock situation

### ③ Deadlock prevention -

For a large db, the deadlock prevention method is suitable.

- wait-Die scheme - If a transaction request a resource that is locked by another transaction, then the DBMS simply checks the TS of both transactions & allows the older transaction to wait until the resources is available for execution.

### • wound-wait schema -

If an older transaction requests a resource held by a younger transaction, then an older transaction force a younger transaction to kill the transaction & release the resource.

## Application -

- ① Delayed transaction
- ② lost transaction
- ③ Reduced concurrency
- ④ Increased Resource usage.

## Features -

- ① Mutual Exclusion
- ② Hold & wait
- ③ No preemption
- ④ Circular wait
- ⑤ Indefinite Blocking

## \* Disadvantages

- ① System downtime
- ② Resource waste
- ③ Reduced Concurrency
- ④ Complex resolution
- ⑤ Increased System overhead.

B  
8/11/29