12/6/23 Unit-4 TRANSACTION MANAGER

Transaction :

→ A collection of operations that form a single logical atomic unit.

→ A transaction is a unit of program execution or an executing program that forms a logical unit of database processing i.e. accessing and updating data items.

→ A transaction includes one or more database access operations., this includes insertion, deletion, updating or retrieval operations

→ A transaction can be bounded by putting/ specifying the boundaries explicity by BEGIN TRANSACTION and END TRANSACTION statements in an application program

Steps in Executing a Transaction

→ For a transaction to get executed it must be scheduled and it is the job of the scheduler to schedule and regulate transaction.

Role of schedular :

Transaction Manger
↓ Read/write requests
Schedular
↓ Read and write
Buffer

→ As soon as a transaction starts to execute, the transaction manager takes the read/write requests and sends it to the schedular.

→ The schedular takes the read/write request from transaction manager and performs read operation from the buffer (or) writes the value of buffer

read(x) :

Steps when a read function is executed.

1, Finds the address of the disk block that contains 'x'.

2, Copies that block into a buffer in main memory.

3, Copies the item x from buffer to program variable named again x (read(x))

write(x) :

1, Finds the address of the disk block that contains x.

2, Copies that block into a buffer in main memory.

3, Copies item x from program variable named x or t into its correct location in buffer

4, Store the updated disk block from the buffer back to disk either immediately or at some later point of time

Transaction Example

→ Transaction to transfer ₹10000/- from A to B

StepNo	Algorithm	Statements
1	Begin the Transaction	Begin Transaction
2	Read you balance i.e A	read (A)
3	Deduce the amount from A	A := A - 10000
4	Write/Update the remaining balance to A	write (A)
5	Read your friends balance i.e B	read (B)
6	Add the amount to B	B: B + 10000
7	Write/Update the remaining balance to B	write (B)
8	End the transaction	End Transaction

13/6/23

→ Before transaction execution, the transaction must see a consistent database

→ During a transaction execution the database may become inconsistent. When transaction is committed the database must be consistent.

Issues that a transaction must deal with:

1, Failures of various kinds, such as hardware failures and system crashes

2, Concurrent execution of multiple transactions accessing same shared memory

Properties of Transaction

→ To deal with the issues that a transaction faces and to maintain the integrity of database during transaction processing we have some concepts / properties of Transactions namely ACID properties.

A > Atomicity C > Consistency I > Isolation

D > durability

Atomicity

→ This property states that all of the instructions with in a transaction must be executed or none

→ If atomicity is present, all actions of transaction are reflected in database or none is reflected.

Consistency:

→ To maintain integrity. some constraints are maint-ained so that the database is consistend before & after the transaction

→ The execution of transaction will leave a database in either of its prior stable state (old stable db) or its updated new stable state.

→ The consistency property of database states that every transaction sees a consistent database

→ Ensuring consistency for an indivial transaction is responsibility of an Application programmer who write the code.

Isolation :

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

→ An isolation if transaction T_k is being executed & using the data item X, then the data item cannot be used by any other transaction T_m until T_k ends.

→ This property ensures that multiple transaction can occur concurrently without leading to inconsistency.

→ Ensuring isolation is the responsibility of concurrency sub system.

A ↓

Before ₹ 100
Debit B 20
Available 80
Debit C 20
Available 60
↓ T_2
Before 70
Credit A 20
Available 90

T_1 →

value read by B = 100
Before 50
Credit 20
Available 70

Durability (recovery manager)

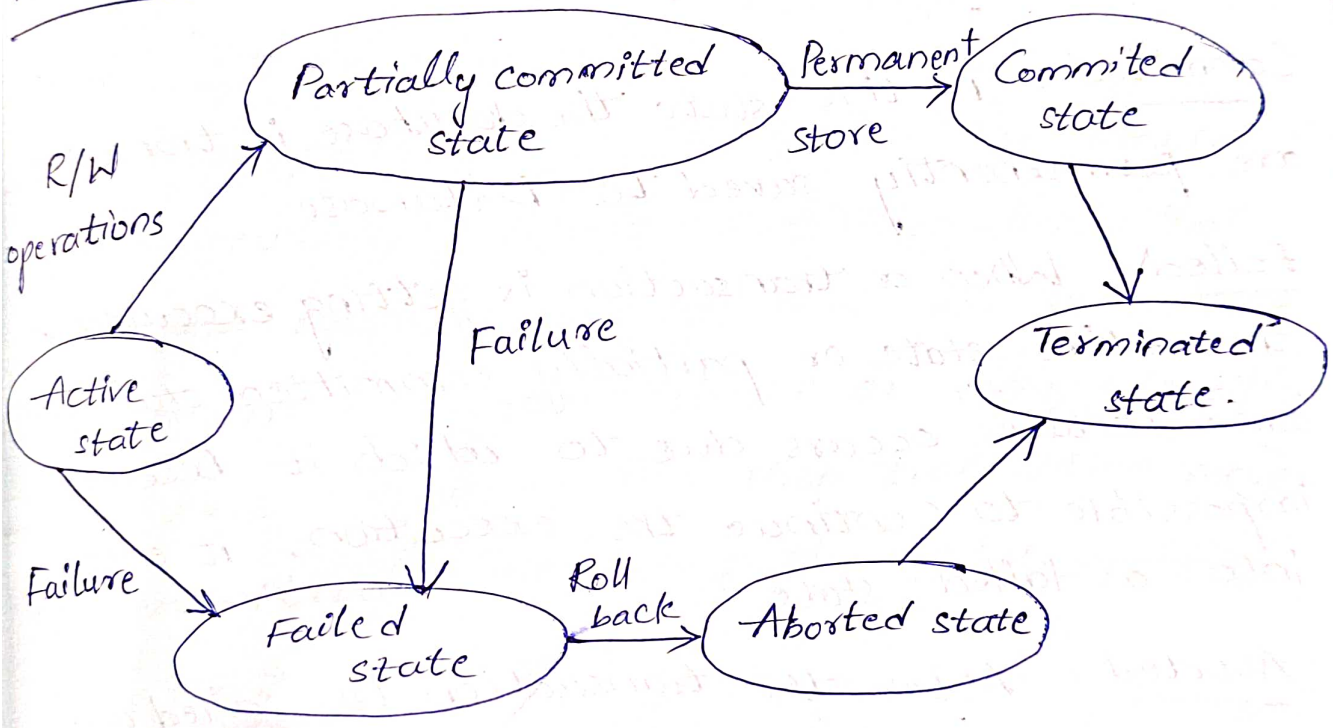
→ This property says that once after the transaction is completed the changes that has been made must be permanent and should be recoverable even after system crashes and power failure.

11/6/23

Acid properties :

Property	Responsibility for maintaining properties
Atomicity	Transaction Manger
Consistency	Application programmer
Isolation	Concurrency control Manager
Durability	Recovery Manager

Transaction States :



Active : (1st state of every transaction)

→ Transaction will ~~start~~ to begin the execution (i.e start stage is being executed) and get executed till the last line of transaction

Partially committed : This is an execution state where the last step of transaction is executed.

Last line can have 2 statements : 1, commit
2, rollback

1, commit : This signals a successful end of transaction so that any changes made while executing an transactions are now updated permanently in database.

2, rollback : This signals are unsuccessfully end of transaction so that any changes made while executing a transaction are now need get undone and have the prior stable state.

Commited : In this state the database is transactions are permanently saved to Database

Failed : When a transaction is getting executed in the active state or partially committed state and some failure occurs due to which it becomes impossible to continue the execution, it enters into a failed state

Aborted : After the transaction has failed and entered into a failed state, all the changes made by it have to be undone.

→ To undo the changes made by the transaction if becomes necessary to rollback the transaction

→ After the transaction has rolled back completely it enters into an aborted state.

Terminated: This is the last state in the life cycle of a transaction. After entering the committed state or aborted state, the transaction finally execut enters into a terminated state where its life cycle finally comes to and end.

→ A good transaction execution starts in active state.

→ When it comes near final statement it enters to partially committed state where the last statement will be executed.

→ At this point, the transaction has completed its execution, but it is still possible that it may have to be aborted due to rollback statement or system failure or some exceptions

→ If there are no crashes then transaction moves to committed state where the value updates are made permanent.