Unit - 6

Transaction Management

Transaction :- A transaction is a logical group of one or more sql statements.

Transaction
Select
update
insert
⋮

Transactions are used in banking, ecommerce, social networks, etc..,)

Generally a transaction can be accessed by using two operations namely read & write.

Eg:- T_1

A = 1000
R(A)
A = A - 500
W(A)
A = 500

* Before & after any transaction dB must be consistent.

Properties of a Transaction

in order to maintain the dB in consistent state, a trsctn follow some properties namely :-
A - Atomicity
C - Consistency
I - Isolation
D - durability

Faculty Signature : ....

Atomicity:-

The . transaction must be treated as Atomic Unit i.e., either all SQL statements or none are applied to the dB.

completed

Ex:-

successful	unsuccessful
✓ Debit	✓ debit
✓ Credit	x Credit
✓	x ....

Consistency :- The dB must be Consistent before & after the transaction.

1000

	Ram	Sam		Total
Before	10 K	5 K	=	15 K
Successful transaction	9 K	6 K	=	15 K
failure	10K	5K	=	15 K

Isolation :-

Multiple transaction can occur at the same time without adversely affecting the other.

Ram (10K)

debit (1K) Credit (25K) debit (9K)

(25K)

Durability :-

changes of a successful transaction persist even after a system crash.

Transaction states :-

① **Active :-**

The very 1st state of a transaction, transaction is being executed.

at this state, read & write operations can be performed.

② **Partially Committed :-**

A transaction executes its final operation, but data is still not saved to the database. [stored in buffer memory].

③ **Committed :-**

If ~~it~~ the trxtn executes all its operations successfully, then it is said to be in Committed state

- All the effects are now permanently saved on dB.

④ <u>Failed state</u>:

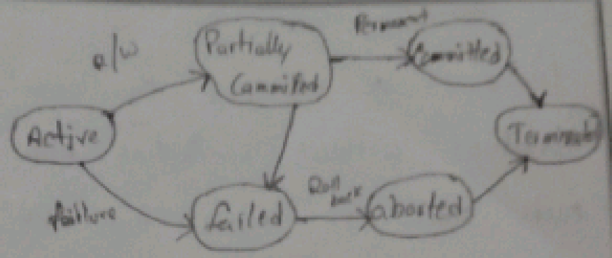
- If any failure occurs ~~at~~ in the active / partially committed state, then the transaction undergoes will enter into failed state, and there is no possibility to continue the execution.

⑤ <u>Aborted</u> : → if trxn enters into failed state, the changes made by it have to be undone

- If the transaction fails in the middle of the transaction then before executing transaction all the transactions are rollback to Consistency state.

⑥ <u>Terminated</u> :

- Terminated state refers to the transaction leaving the system.



<u>Concurrency in DBMS</u>

Concurrency is the ability to allow multiple users to affect multiple transactions within a dB

→ allows multiple users to access data all at same time.

<u>Ex</u>:- At our results time, all students tries to access the same dB at same time [approx]

it may lead to several prblms.

<u>need of Concurrency</u>

if we execute each transaction individually, it increases waiting time.

~~ex~~ if many transactions tries to access the same dB, then inconsistency arises

Concurrency Control

Procedure of managing simultaneous operations without Conflicting with each other. It ensures that DB transactions are performed concurrently and accuratly to produce correct results.

Several problems that arise when numerous transactions execute simultaneously in a random manner are referred to as Concurrency Control Problems.

Dirty Read Problem :- [W-R]

* It occurs when one transaction update the dB item and the transaction fail for some reason.
* The updated dB item is accessed by another transaction before it rollback to the Original Value.

Ex) A=1000

T₁	T₂
R(A)	
A=A+500	
W(A)	
	A=A+300
	Commit
Rollback	

Here, at first the amount is 1000 in dB, then in T₁ it gets updated to 1500 and then in T₂ as 1800 as it committed to dB. Now, assume that some server issues occured and the rollbact dB have to rollback, if the dB rollback happens, it dB will lost 800 and gets back to 1000.

Lost Update Problem :- [W-W]

The update made by one transaction is overwriten by other transaction.

A=1000

T₁	T₂
R(A)	
A=A+500	
W(A)	
	A=A+300
	W(A)
R(A)	

At first, the value is 1000, and in T_1, it is updated to 1500 and in T_2, it is updated to 1200. But at the end, when you read the T_1 amount, it'll return the ver value updated by T_2. i.e., we lost the update made by T_1.

Inconsistent Retrival :- [W - R] unrepeatable read

It occurs when two or more different values of the same data are read during the read operations in the same transaction

Eg:-

A=1000

T_1	T_2	
R(A)		
	R(A)	→ 1000

A= A+500
W(A)
		⇕ diff read values

	R(A)	→ 1500

Here, in T_2 transaction, both read operations retrives different values, even though they were done in a single transaction.

Phantom Read Problem :-

It occurs, where the transaction reads a variable once and when it tries to read the variable again it gets an error showing that variable does not exist, as the variable is deleted by another transaction.

A = 1000

Eg:-

T_1	T_2	
R(x)		
	R(x)	- 1000
D(x)		
	R(x)	- error

Here, in T_2, the first read returns 1000 and the second read returns an error ~~showo~~ stating that the data doesn't exist, as the data is deleted by T_1.