DBMS Lec 9 (Transaction) :

A transaction in databases is a set of operations that must be completed together to keep data safe and accurate, especially in systems like banking. Transactions follow four main rules—Atomicity, Consistency, Isolation, and Durability (ACID)—to make sure money and data are never lost or corrupted, even if something goes wrong.

What is a Transaction?

A transaction is a group of actions (like transferring money) that must all succeed for the database to change. If any part fails, the whole transaction is canceled, so no partial actions are left.

Example: Bank Money Transfer

When transferring money between accounts, the system reads the balance, deducts from one account, and adds to another. Each step must be done in order and together, or the transaction is rolled back.

Transactions: Logical Steps and Sequence

Transactions are made of logical steps that must be performed in sequence. If any step fails, previous steps are undone to keep the database correct.

Atomicity Explained

Atomicity means a transaction is all or nothing—either every step happens, or none do. This prevents data from being left in an incomplete state.

Consistency in Transactions

Consistency ensures that a transaction brings the database from one valid state to another, following all rules and constraints. Money cannot be created or lost.

Read and Write Operations

Reading gets data from the database, and writing updates it. Changes are first made in memory and only saved to the database when the transaction is committed.

Commit Operation and Data Saving

Committing a transaction means all changes are made permanent in the database. If not committed, changes can be rolled back and not affect the real data.

Recovery and Rollback

If a transaction fails, the system rolls back to the previous safe state, undoing any partial changes. This helps recover from errors or crashes.

Integrity Constraints

Integrity constraints are rules that data must follow (like total money before and after a transaction should match). They prevent mistakes or fraud.

Isolation in Concurrent Transactions

Isolation means that even if many transactions happen at the same time, each one acts as if it's alone, so data doesn't get mixed up or corrupted.

Durability of Transactions

Durability means that once a transaction is committed, its changes stay in the database even if the system crashes. Recovery systems and logs help ensure this.

Logging and Recovery Mechanisms

The system keeps logs of all actions in a transaction. If something fails, it uses these logs to restore the correct state or to redo completed actions.

Transaction States and Status

Transactions go through states like active, partially complete, committed, or rolled back. Only committed transactions change the database.

Atomicity Revisited and Conclusion

Atomicity is revisited as a key part of transaction safety, ensuring no partial updates happen. The session ends with a summary of why ACID properties matter.