DBMS Lec 10 (ATOMICITY IMPLEMENTATION) :

Atomicity in databases means that a group of operations within a transaction are treated as a single unit: either all happen or none happen. This ensures data consistency and prevents partial updates, even if a system fails during the transaction.

What is Atomicity?

Atomicity is a key property of database transactions, ensuring that either all operations within a transaction are completed successfully, or none are applied at all. This prevents data from being left in an inconsistent state.

Making Copies Before Updates

Before making any changes, a copy of the current data is created. All updates are performed on this copy, not directly on the original data. This allows for easy rollback if something goes wrong.

Committing the Transaction

Once all operations on the copy are successful, the changes are committed, meaning the new copy replaces the old data. This step finalizes the transaction and makes the changes permanent.

Rolling Back on Failure

If any operation fails before the commit, the system discards the copy and keeps the original data unchanged. This rollback mechanism ensures that incomplete transactions do not affect the database.

Importance of Complete Copies

Creating a full copy of the data ensures that the system has a reliable backup to revert to, supporting both rollback and recovery processes.

Logging and Recovery

Transaction logs record all changes and operations. These logs are crucial for restoring the database to a consistent state after a crash or failure, as they allow the system to reapply or undo operations as needed.

The Role of Locks in Atomicity

Locks are used to prevent other operations from interfering with a transaction in progress, ensuring that the atomicity and consistency of the data are preserved.

Updating and Maintaining Consistency

When updates are made, both old and new values are tracked to ensure that the database can revert to the original state if necessary, maintaining consistency and integrity.

Checkpoints and Recovery Methods

Checkpoints are markers in the transaction log that indicate a consistent state of the database. They help in speeding up recovery by providing a recent safe point to start from after a failure.

Real-world Application Example

Applying atomicity in real-world systems, such as electronic stores, ensures that complex operations (like inventory updates and sales) are processed reliably without risk of partial completion.