ACID properties in database transactions

ACID is an acronym representing four key properties of database transactions: Atomicity, Consistency, Isolation, and Durability. These properties collectively guarantee the reliability and integrity of data within a database system, even in the face of system failures or concurrent operations.

Atomicity:

Atomicity ensures that a transaction is treated as a single, indivisible unit of work. This means that either all operations within a transaction are successfully completed and committed to the database, or none of them are. If any part of the transaction fails, the entire transaction is rolled back, leaving the database in its state before the transaction began. This "all or nothing" principle prevents partial updates and maintains data integrity.

Consistency:

Consistency dictates that a transaction must bring the database from one valid state to another. This means that all data written during a transaction must adhere to all defined rules, constraints, and triggers within the database schema. If a transaction attempts to violate any of these rules, it is rolled back to maintain the database's consistent state, preventing invalid data from being introduced.

Isolation:

Isolation ensures that concurrent transactions do not interfere with each other. Each transaction appears to execute in isolation, as if it were the only transaction running on the system. This prevents issues like dirty reads (reading uncommitted data), non-repeatable reads (reading the same data twice and getting different results), and phantom reads (re-running a query and finding new rows). Different isolation levels exist, offering varying degrees of concurrency and consistency based on application needs.

Durability:

Durability guarantees that once a transaction has been successfully committed, its changes are permanent and will survive any subsequent system failures, such as crashes or power outages. This is typically achieved by writing transaction logs or using write-ahead logging to persist the changes to stable storage (like disk) before acknowledging the commit. In distributed systems, durability often involves

replicating data across multiple nodes to ensure data persistence even if one node fails.