Consistency

Consistency ensures that each transaction takes the database from one valid state to another valid state. It means that the database must adhere to predefined rules and constraints before and after a transaction. In other words, a transaction must preserve the integrity and correctness of the database.

Example:

Consider a bank database where a transaction transfers money from one account to another. Consistency requires that the total amount of money in the system remains the same before and after the transaction. If the transaction fails (e.g., due to insufficient funds), it must be rolled back to ensure that no money is lost or created.

Isolation

Isolation ensures that transactions operate independently of each other. Even though multiple transactions may be executed concurrently, each transaction should be executed as if it were the only transaction running in the system. The intermediate results of a transaction must not be visible to other transactions.

Example:

Imagine two transactions where one is updating a product's price and another is calculating discounts based on that price. If these transactions are isolated, the discount calculation will not use intermediate or uncommitted prices from the updating transaction. The discount calculation will either use the price before the update or the new price after the update, but not a partially updated price.