

## 1. The Concept of a Transaction:

- **Definition:** A transaction is a logical unit of work that consists of one or more database operations.
- **Purpose:** To ensure data integrity and consistency by treating a series of operations as a single, indivisible unit.

## 2. The ACID Properties:

- **Atomicity (A):**
  - "All or nothing" principle. Either all operations within the transaction succeed, or none do.
  - Understanding how rollbacks work in case of failure.
- **Consistency (C):**
  - Transactions maintain the database's integrity by adhering to defined rules and constraints.
  - Knowing about database constraints, such as primary keys, foreign keys, and check constraints.
- **Isolation (I):**
  - Transactions operate independently of each other, preventing interference.
  - Understanding concurrency control and how it manages simultaneous transactions.
- **Durability (D):**
  - Once a transaction is committed, its changes are permanent and survive system failures.
  - Knowing about transaction logs and recovery mechanisms.

## 3. Database Operations within a Transaction:

- **Read (SELECT):** Retrieving data from the database.
- **Write (INSERT, UPDATE, DELETE):** Modifying data in the database.
- **Commit:** Making the changes permanent.
- **Rollback:** Reverting the database to its state before the transaction began.

## 4. Concurrency Control:

- **Locks:** Mechanisms to prevent multiple transactions from modifying the same data simultaneously.
- **Transaction Isolation Levels:** Different levels of isolation that control how transactions interact with each other.

## 5. Transaction Logs:

- How database systems keep track of transaction operations to ensure durability and enable recovery.

## **6. Practical Examples:**

- Understanding how transactions are used in real-world scenarios, such as bank transfers, online orders, or inventory management.