What is ACID Properties

Acid is an acronym thatrefers to the set of 4 key properties that define a transaction: Atomicity, consistency, Isolation, and Durability. If a database operation has theses ACID properties, it can be called an ACID transaction, and data storage systems that apply these operations are called transactional systems. ACID transactions guarantee that reach read, write, or modification of a table has the following properties:

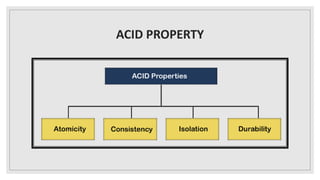
A transaction is a single logical unit of work that accesses and possibly modifies the contents of a database. Transactions access data using read and write operations. In order to maintain consistency in a database, before and after the transaction, certain properties are followed.

There are called ACID properties.

ACID Properties

(4key properties that define a transaction)

1. Atomicity
2. Consistency
3. Isolation
4. Durability



**1)Atomicity**

The term atomicity defines that data remains atomic. It means if any operation is performed on the data, either it should be performed or executed completely or should not be executed at all. It further means that the operation should not break in between or execute partially. In the case of executing operation on the transaction, the operation should be completely executed and not partially.

Example

**B**

**A**

**Before – 30**

**Debit -10**

**Available - 20**

**Transfer**

**Before -100**

**Credit – 10**

**Available -100**

**Process**

**Debited Succesfully Credited failure**

**Partial Execution no Atomicity excution termination**

**A**

**B**

**Before -100**

**Credit – 10**

**Available - 110**

**Before – 30**

**Debit – 10**

**Available - 20**

**Transfer**

trans

**Process**

transfer

**Debited Successfully Credited Failure**

**Complete Execution Atomicity Execution Successfull**

**2)Consistency**

The word consistency means that the value should remain preserved always. In DBMS, the integrity of the data should be maintained, which means if a change in the database is made, it should remain preserved always. In the case of transactions, the integrity of the data is very essential so that database remains consistent before and after the transaction. The data should always be correct.

**A**

**Example:**

**Before – 300**

**Debit to B – 50**

**Available – 250**

**Debit to C – 20**

**Available - 230**

**Available -**

**T**

**Value Read by B = 250 Before T**

**Value Read by B = 300 Before T**

**B**

**C**

**Before - 100**

**Credit - 50**

**Available - 150**

**Before – 50**

**Credit – 20**

**Available - 70**

**Data consistent**

**3)Isolation**

The term isolation means separation.

* In DBMS Isolation is the property of a database where no data should affect the other one and may occur concurrently.
* In short, the operation on one database should begin when the operation on the first database gets complete.
* It means if two operations are being performed on two different
* databases, they may not affect the value of one another.
* In the case of transactions, when two or more transaction occur simultaneously, the consistency should remain maintained.
* Any changes that occur in any particular transaction will not be seen by other transaction until the change is not committed in the memory.

**Example:** Isolation independent Execution T1 and T2 and A

**Value read by B Before T1 was 100**

**A**

**B**

**Before – 100**

**Debit to B – 20**

**Available -80**

**Debit to C = 20**

**Available - 60**

**Before – 50**

**Credit by A – 20**

**Available - 70**

**Before – 70**

**Credit by A – 20**

**Available - 90**

**C**

T1

**Value read by T2**

**C before T2**

**was 80**

4)Durability

* Durability ensures the permanency of something
* In DBMS, the term durability ensures that the data after the successful execution of the operation becomes permanent in the database.
* The durability of the data should be so perfect that even if the system fails or leads to a crush, the database still survives.
* However, if gets lost, it becomes the responsibility of the recovery manager for ensuring the durability of the database.
* For committing the values, the COMMIT command must be used every time we make changes.

**ACID properties (uses):**

* Maintain integrity constraints defined on the database schema
* Prevent concurrent transaction anomalies like dirty, non-repeatable, and phantom reads
* Provide reliable recoverability from system crashes and database failures
* Ensure overall consistency of the database after each transaction
* Eliminate partial or unsuccessful writes to the database
* Enforce business rules and relational integrity
* Support auditing and regulatory compliance

Advantages and Disadvantages of ACID Properties in

Advantages of ACID Properties

* Preserve data consistency
* Guarantee transactional integrity
* Provide fault tolerance and recovery through durability
* Enable concurrency control between transactions via isolation
* Maintain constraints and business rules through consistency
* Eliminate partial transactions with atomicity

Disadvantages of ACID Properties

* Lead to processing overhead and reduced throughput
* Cause poor performance in high-volume OLTP systems
* Limit scalability due to restrictive locking
* Decrease availability by blocking transactions to uphold isolation
* Make distributed transactions more complex

ACID properties come with significant benefits but also performance tradeoffs. Tuning isolation levels and using alternative NoSQL databases can help mitigate the disadvantages.

**Conclusion:**

The ACID properties in DBMS ensure reliable and robust data management

1) Atomicity: Transactions are treated as indivisible units, ensuring all changes are committed or none at all

2)Consistency: The database remains in a valid state before and after transactions, enforcing predefined rules and constraints.

3)Isolation: Concurrent transactions don't interfere with each other, preventing data inconsistencies.

4)Durability: Committed changes are permanently saved and survive system failures, ensuring data reliability

Adhering to these properties ensure data integrity, consistency, isolation, and durability in a database system.