



ACID

(Atomicity, Consistency, Isolation, Durability)

.NET CORE

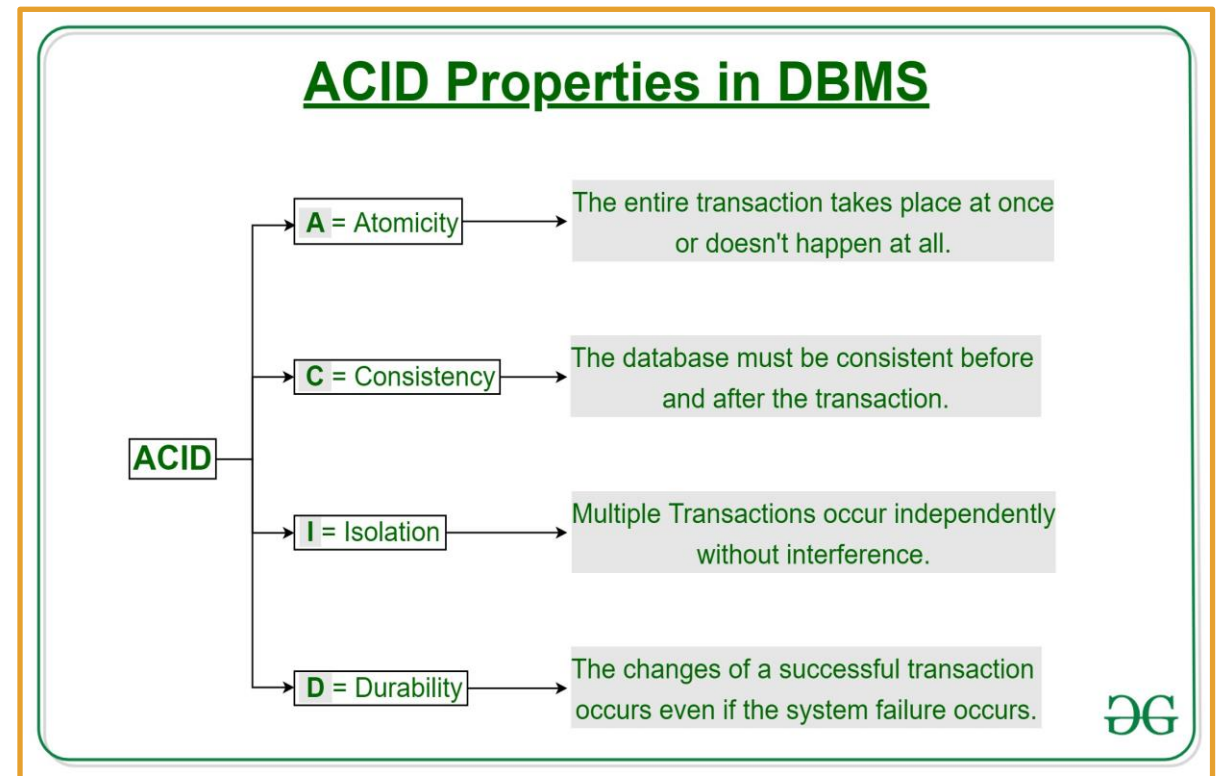
ACID (atomicity, consistency, isolation, durability) is a set of properties of database transactions intended to guarantee validity even in the event of errors or power failures.

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/ACID](https://en.wikipedia.org/wiki/ACID)

ACID – Atomicity

<https://docs.microsoft.com/en-us/windows/win32/cos-sdk/acid-properties>

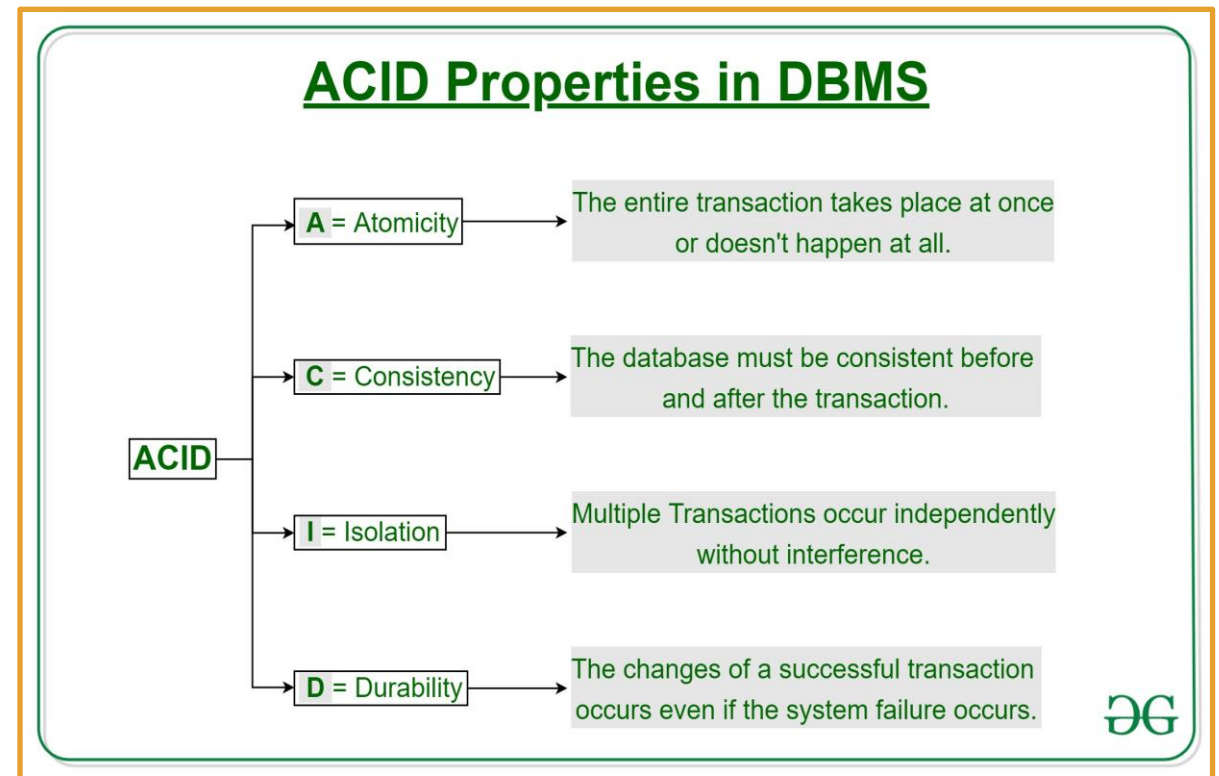
- A *transaction* must execute exactly once and must be *atomic*.
- An *Atomic Transaction* is indivisible and irreducible. Either all operations occur, or none do.
- By performing only a subset of interdependent operations, the overall intent of a *transaction* would be compromised.
- *Atomicity* eliminates the chance of processing only a subset of operations.



ACID – Consistency

<https://docs.microsoft.com/en-us/windows/win32/cos-sdk/acid-properties>

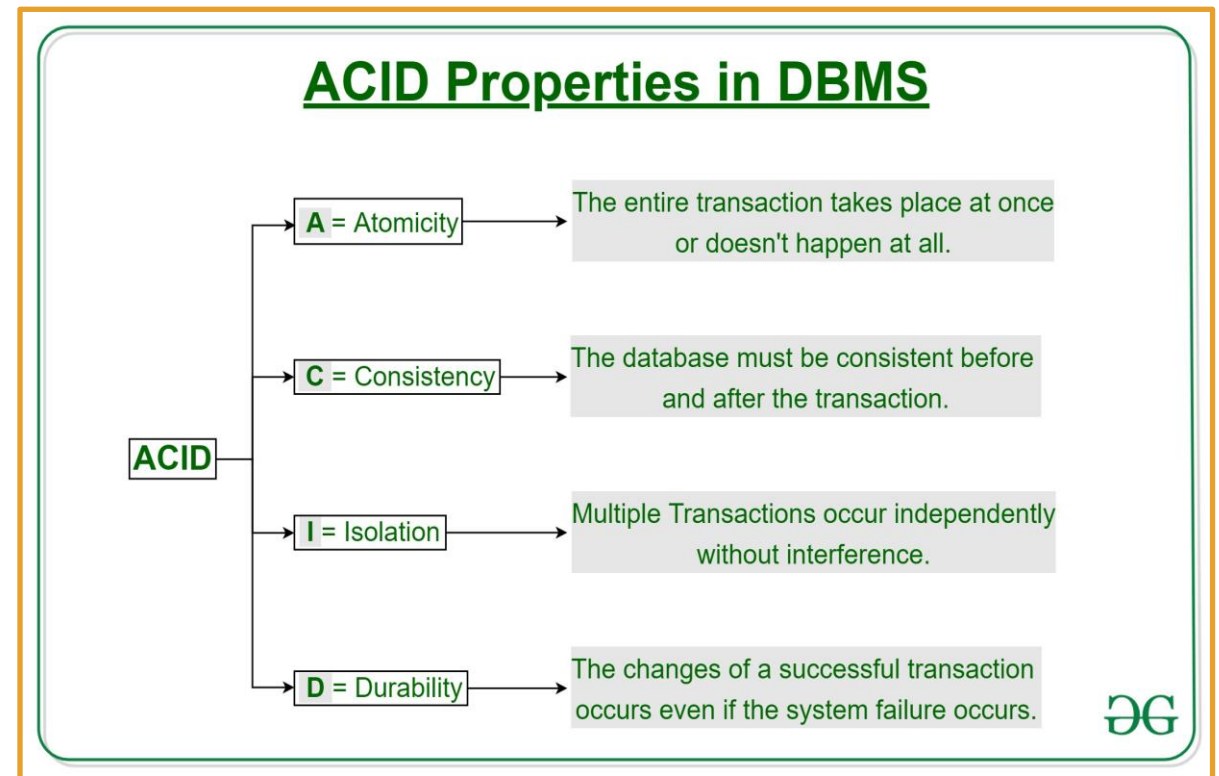
- A *transaction* must transform data from one *consistent* state to another *consistent* state.
- The responsibility for maintaining *consistency* falls to the application developer.
- *Referential Integrity* must be maintained when transforming one consistent state of data into another consistent state of data.



ACID – Isolation

<https://docs.microsoft.com/en-us/windows/win32/cosssdk/acid-properties>

- A *transaction* must be completed in isolation.
- **Concurrent Transactions** should behave as if each were the only *transaction* running in the system.
- A high degree of *isolation* can limit the number of *concurrent transactions*.
- There are 5 levels of Isolation.



ACID – Durability

<https://docs.microsoft.com/en-us/windows/win32/cos-sdk/acid-properties>

- A *transaction* must have durability (be recoverable).
- If a *transaction* ‘commits’, its updates will *persist* even if the system fails immediately after.
- Specialized logging allows the system's restart procedure to complete unfinished operations required by the *transaction*. This makes the *transaction durable*.

