ACID

(Atomicity, Consistency, Isolation, Durability)

.NET

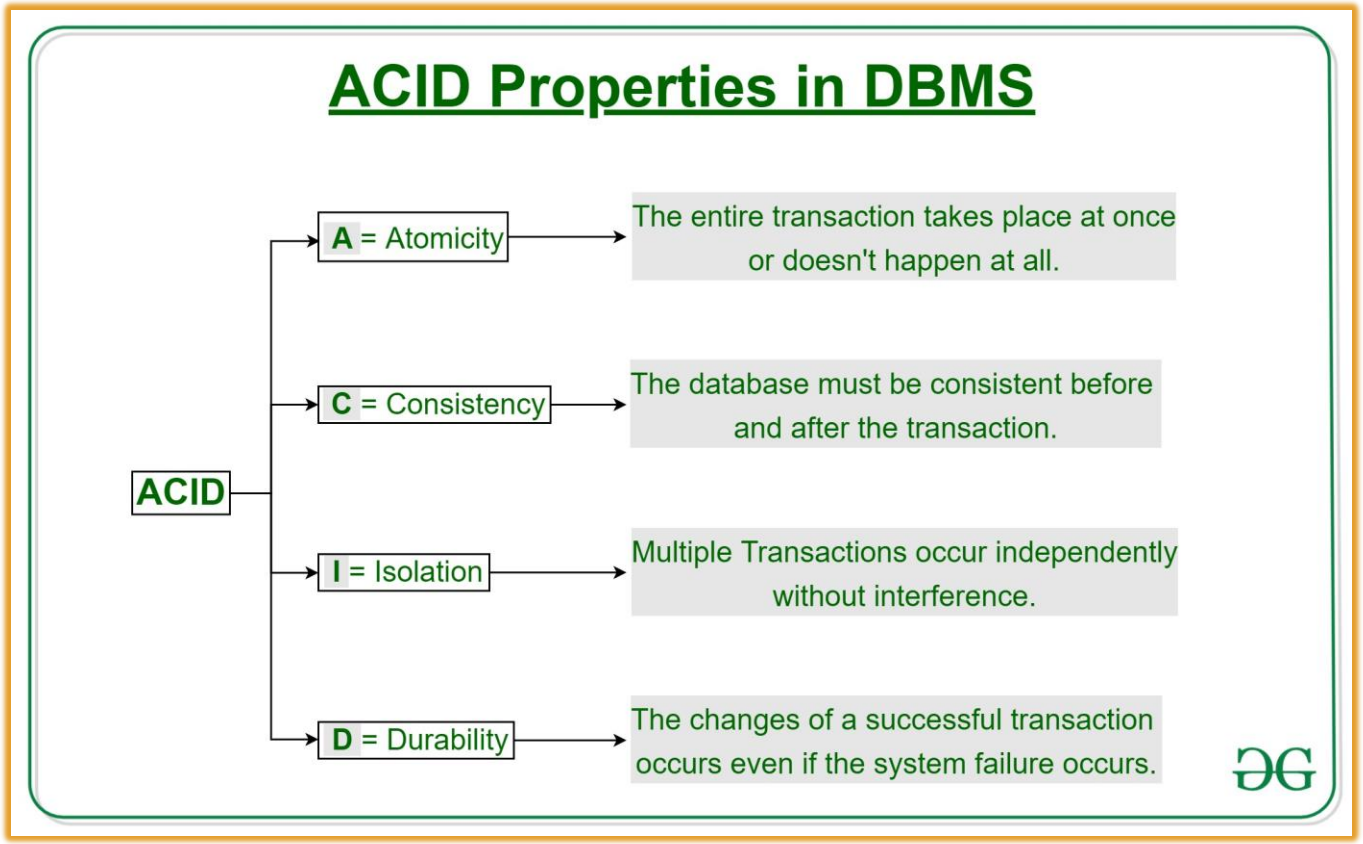
ACID (atomicity, consistency, isolation, durability) *is a set of properties of database transactions intended to guarantee validity even in the event of errors or power failures.*

HTTPS://EN.WIKIPEDIA.ORG/WIKI/ACID

ACID – Atomicity

https://docs.microsoft.com/en-us/windows/win32/cossdk/acid-properties

- A transaction must execute exactly once and must be atomic (either all of the work is done or none of it is).
- 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.

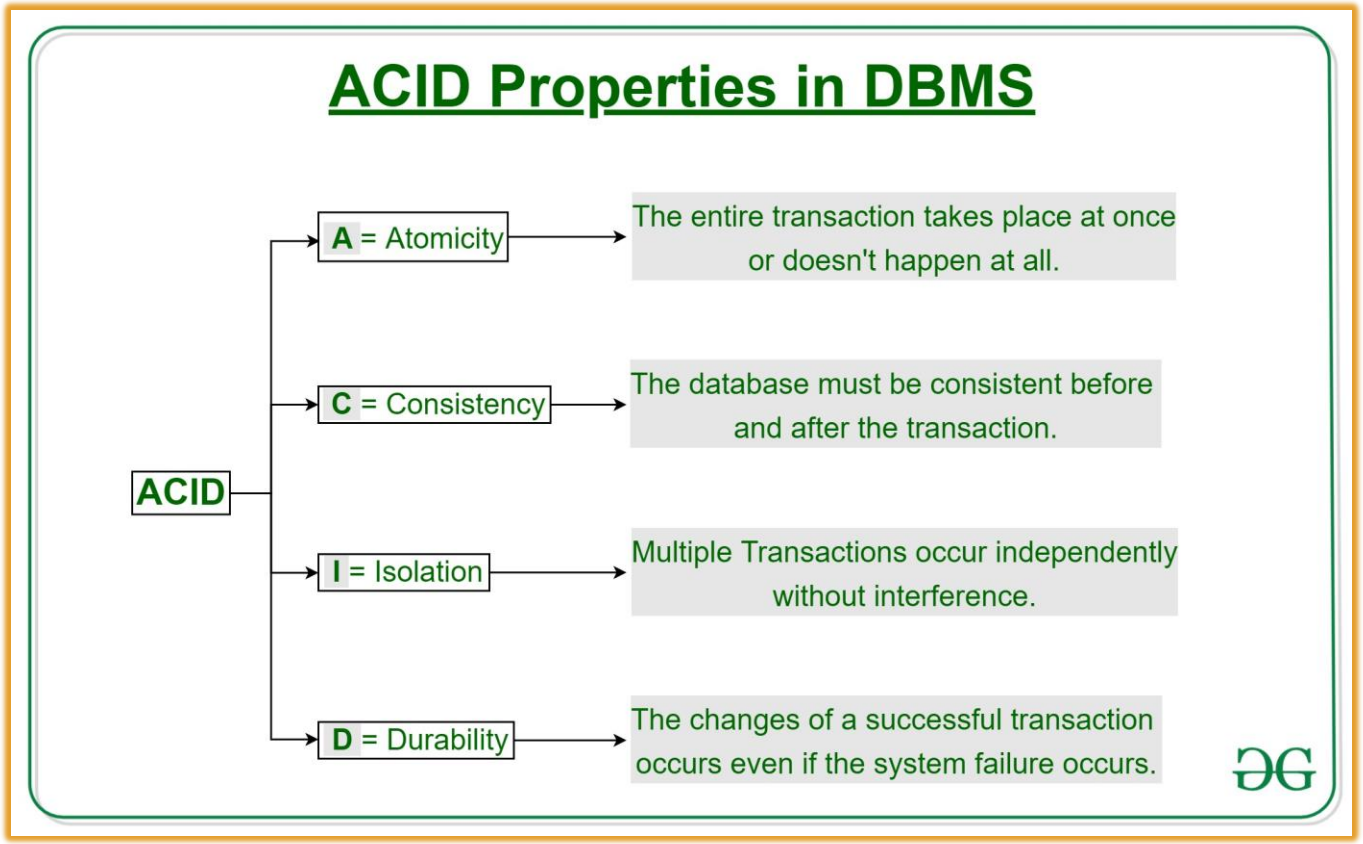


ACID – Consistency

https://docs.microsoft.com/en-us/windows/win32/cossdk/acid-properties

A transaction must preserve the consistency of data.

Referential Integrity must be maintained when transforming one consistent state of data into another consistent state of data.

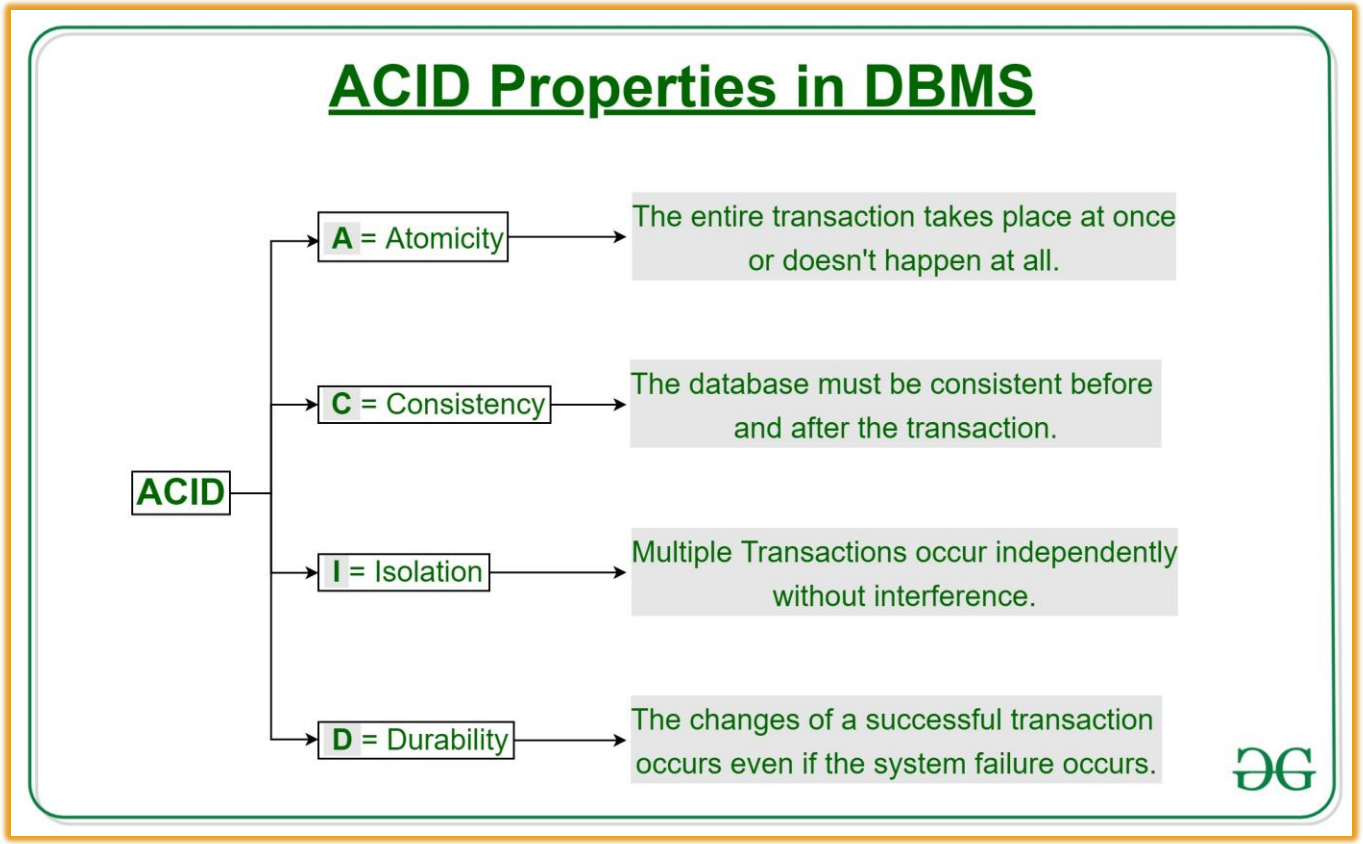


ACID – Isolation

https://docs.microsoft.com/en-us/windows/win32/cossdk/acid-properties

A transaction must be completed in isolation. This means that 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.***



ACID – Durability

https://docs.microsoft.com/en-us/windows/win32/cossdk/acid-properties

A ***transaction*** must have durability (i.e., be recoverable.

If a ***transaction commits***, its updates will ***persist*** even if the computer crashes immediately after the ***commit***.

Specialized logging allows the system's restart procedure to complete unfinished operations required by the ***transaction***. This makes the ***transaction durable***.