

[Back To Module Home](#)

# Distributed Systems

0% completed

## Introduction to Distributed Systems

### Basic Concepts and Theorems

Partitioning


Algorithms for Horizontal Partitioning

Replication

Single-Master Replication Algorithm

Multi-Master Replication Algorithm

Quorums in Distributed Systems



	Safety Guarantees in Distributed Systems
	ACID Transactions
	The CAP Theorem
	Consistency Models
	CAP Theorem's Consistency Model
	Isolation Levels and Anomalies
	Prevention of Anomalies in Isolation Levels
	Consistency and Isolation
	Hierarchy of Models
	Why All the Formalities?
	Quiz

## Conclusion

Mark Module as Completed

# ACID Transactions

Let's see the ACID properties.

### We'll cover the following

- Atomicity (A)
- Consistency (C)
- Isolation (I)
- Durability (D)

**ACID** is a set of properties of traditional database transactions that provide guarantees around the expected behavior of transactions during errors, power failures, etc. More specifically, these properties are the following.

## Atomicity (A)#

**Atomicity** guarantees that a transaction that comprises multiple operations is treated as a single unit. This means that either *all* operations of the transaction are executed, or *none* of them is.

This concept of atomicity extends to distributed systems, where the system might need to execute the same operation in multiple nodes of the system in an atomic way. So, the operation is either executed to all the nodes or none.

# Consistency (C)#

**Consistency** guarantees that a transaction only transitions the database from one valid state to another valid state, while maintaining any database invariants. However, these invariants are application-specific and defined by every application accordingly.

For example, consider an application that has a table A with records that refer to records in table B through a foreign key relationship. The database prevents a transaction from deleting a record from table A, unless any records in table B referenced from this record are already deleted.

Note that this is not the concept of consistency we refer to in the context of distributed systems.

# Isolation (I)#

**Isolation** guarantees that even though transactions might run concurrently and have data dependencies, the result is as if one of them was executed at a time and there was no interference between them. This prevents a large number of anomalies.

# Durability (D)#

**Durability** guarantees that once a transaction is committed, it remains committed even in the case of failure.

In the context of single-node, centralized systems, this usually means that completed transactions and their effects are recorded in non-volatile storage.

In the context of distributed systems, this means that transactions need to be durably stored in multiple nodes. This way, recovery is possible even in the presence of total

failures of a node, alongside its storage facilities.

**Back**

Safety Guarantees in Distributed Syst...

**Next**

The CAP Theorem

Mark as Completed

---

Report an Issue