

Log In

Join

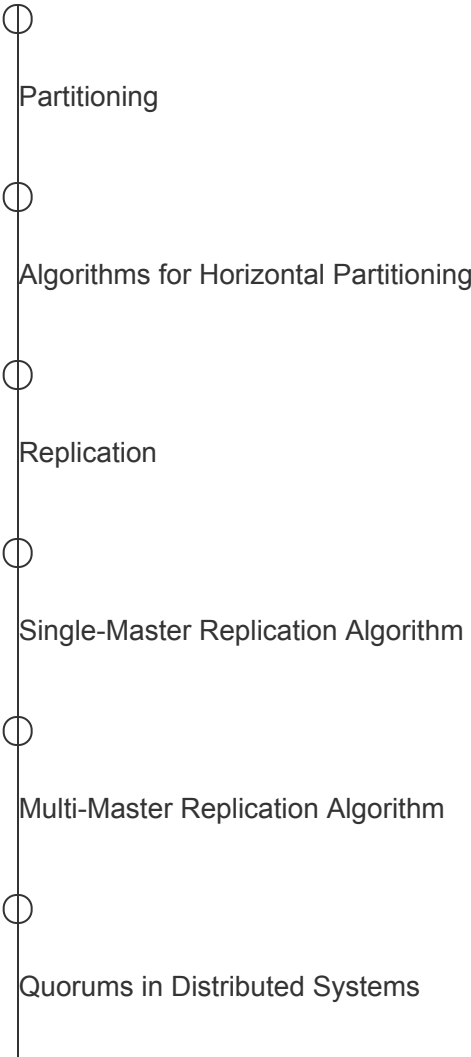
Back To Module Home


Distributed Systems

0% completed

Introduction to Distributed Systems

Basic Concepts and Theorems





○	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**

# Why All the Formalities?

Let's discuss why we need the complicated, formal, and academic constructs we have learned till now.

**We'll cover the following**

- The need for formal models

The previous chapters spent a significant amount of time going through many different formal models.

## The need for formal models#

All these complicated, formal, and academic constructs help us define different types of properties in a more precise way. As a result, when we design a system, we can easily reason about the properties the system needs to satisfy, and know which of these models are sufficient to provide the required guarantees.



In many cases, applications are built on top of pre-existing datastores. They derive most of their properties from these datastores because most of the data management is delegated to them. Consequently, necessary research needs to be done to identify datastores that can provide the guarantees the application needs.

Unfortunately, the terminology and their associated models presented here are not used consistently across the industry. This makes decision-making and comparison of systems a lot harder.

For example, some data stores do not precisely state what kind of consistency guarantees their system can provide—or, at least, these statements are well hidden. In fact, these statements should be highlighted as some of the most important things in their documentation. In some other cases, such documentation exists, but the various levels we discussed earlier are misused and lead to a lot of confusion. As we learned earlier, one source of this confusion was the initial ANSI-SQL standard. For example, the serializable level provided by Oracle 11g, MySQL 5.7, and PostgreSQL 9.0 was not truly serializable and was susceptible to some anomalies.

Understanding the models presented here is a good first step towards being more careful when we design systems, so we reduce the risk for errors. We should be willing to search the documentation of systems we want to use to understand what kind of guarantees they provide. Ideally, we should also be able to read between the lines and identify mistakes or incorrect usages of terms. This will help us make more informed decisions. Hopefully, it will also help raise awareness across the industry and encourage vendors of distributed systems to specify the guarantees their systems can provide.

**Back**

Hierarchy of Models

**Next**

Quiz

Mark as Completed

[Report an Issue](#)