Glossary: Working with Distributed Data

Welcome! This alphabetized glossary contains many of the terms you'll find within this course. This comprehensive glossary also includes additional industry-recognized terms not used in course videos. These terms are important for you to recognize when working in the industry, participating in user groups, and participating in other certificate programs.

Term	Definition
ACID	This term is an acronym for Atomicity, Consistency, Isolation, and Durability, which is a set of properties that guarantee reliable processing of database transactions in traditional relational databases.
Atomic	In the context of database transactions, atomic means that an operation is indivisible and either completed fully or is completely rolled back. It ensures that the database remains in a consistent state.
Availability	In the context of CAP, availability means that the distributed system remains operational and responsive, even in the presence of failures or network partitions. Availablity is a fundamental aspect of distributed systems..
BASE	An alternative to ACID. Stands for Basically Available, Soft state, Eventually consistent. BASE allows for greater system availability and scalability, sacrificing strict consistency in favor of performance.
Basically available	A basically available system remains operational even in the presence of failures or faults.
CAP	CAP is a theorem that highlights the trade-offs in distributed systems, including NoSQL databases. CAP theorem states that in the event of a network partition (P), a distributed system can choose to prioritize either consistency (C) or availability (A). Achieving both consistency and availability simultaneously during network partitions is challenging.
Consistency	In the context of CAP, consistency refers to the guarantee that all nodes in a distributed system have the same data at the same time.
Consistent	The action of being consistent ensures that a database transaction transforms the database from one consistent state to another.
Denormalized	Denormalization is a database design technique used in NoSQL databases (and sometimes in traditional relational databases) where redundant or duplicate data is intentionally introduced into one or more tables to improve query performance and reduce the need for complex joins.
Durable	Guarantees that once a transaction is committed, its changes are permanent and will survive any system failures.
Eventually consistent	An eventually consistent system reaches a consistent state, where all nodes have the same data given that there are no new updates.
Isolated	Isolation refers to the property that multiple transactions can run concurrently without affecting each other.
Joins	Combining data from two or more database tables based on a related column between them.
Normalized	A database design practice where data is organized to minimize redundancy and maintain data integrity by breaking it into separate tables and forming relationships between them.
Partition tolerance	In the context of CAP, partition tolerance is the ability of a distributed system to continue functioning even when network partitions or communication failures occur.
Replication	Replication involves creating and maintaining copies of data on multiple nodes to ensure data availability, reduce data loss, fault tolerance (improve system resilience), and provide read scalability.
Sharding	Refers to the practice of partitioning a database into smaller, more manageable pieces called shards to distribute data across multiple servers. Sharding helps with horizontal scaling.
Soft state	A soft state acknowledges that the system's state might be transiently inconsistent due to factors like network partitions or concurrent updates. And it's willing to accept a certain level of inconsistency or uncertainty in the data temporarily.
Transactions	Transactions are sequences of database operations (such as reading and writing data) that are treated as a single, indivisible unit.

Skills Network

`