THE CAP THEOREM :

The CAP theorem is also called Brewer’s Theorem, because it was first advanced by Professor Eric A. Brewer during a talk he gave on distributed computing in 2000. It is a technique that system designers use to help them understand the trade-offs involved in creating networked shared-data systems.

CAP refers o three desirable properties of distributed systems with replication of data :

CONSISTENCY (among replicated copies)

AVAILABILITY (for read and write operations)

PARTITION TOLERENCE (nodes in the system being partitioned by network fault)

According to the CAP theorem, in a distributed system with data replication, it is impossible to simultaneously ensure all three of the desired properties: consistency, availability, and partition tolerance.

The theorem states that networked shared-data systems can only support any two of the given three properties :

CONSISTENCY : When a replicated data item is consistent, all nodes will see the same copies of it throughout different transactions. a promise that each node in a dispersed cluster would always return the most recent, identical write that was successful. Every client should see the data in the same way, according to consistency. Different kinds of consistency models exist. Sequential consistency, a particularly powerful type of consistency, is what is meant by consistency in CAP.

AVAILABILITY : When a data item is available, it indicates that every read or write request will either be fulfilled without error or will result in an error message. Within a reasonable timeframe, each non-failing node responds to all read and write requests. Here, the term "every" is crucial. To put it plainly, each node (on both sides of a network split) needs to be able to react within a respectable time frame.

PARTITION TOLERENCE : Partition tolerance refers to the ability of the system to function even in the event of a defect in the network that links the nodes, resulting in two or more partitions, each of which can only communicate with the nodes in its own partition. That means that even in the face of network partitions, the system keeps running and maintains its consistency guarantees. Partitions inside networks happen. Once a partition heals, distributed systems that ensure partition tolerance can smoothly recover from partitions.



* CA(Consistency and Availability) :

 The system gives its priority over availability and consistency and can respond with data

Eg: RDBMS

* AP (availability and partition tolerance ):

The system gives its priority over availability and partition tolerance and can respond with data

Eg: Cassandra , couche DB,Riak

* CP (consistency and partition tolerance ):

The system gives its priority over consistency and partition tolerance and can respond with data

Eg: HBase,Mongo DB

It's crucial to remember that these database systems could have various configurations and settings, which could alter how they behave in terms of availability, consistency, and partition tolerance. Consequently, a database system's precise behaviour may vary depending on how it is configured and used.

HOW CAP THEOREM IS USED IN CASSANDRA :



 Often referred to as a "AP" system, Apache Cassandra on the side of guaranteeing data availability even at the expense of consistency. This is a little oversimplified, but Cassandra may be set up to function similarly to a "CP" database and aims to meet all three requirements at once.