1. **NoSQL Databases:**

A NoSQL (originally referring to "non SQL", "non relational" or "not only SQL")[1] database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases.

Features of NoSQL:

*Generic data model:* Heterogeneous containers, including sets, maps, and arrays

*Dynamic type discovery and conversion:* NoSQL analytics systems support runtime type identification and conversion so that custom business logic can be used to dictate analytic treatment of variation.

*Non-relational and De-normalised:* Data is stored in single tables as compared to joining multiple tables.

*Commodity hardware:* Adding more of the economical servers allows NoSQL databases to scale to handle more data.

*Highly distributable:* Distributed databases can store and process a set of information on more than one device.

e.g. MongoDB, HBase.

1. **Types of NoSQL Databases**

*Document Oriented Databases (MongoDB):* Stores documents as a data and allows indexing of documents on the basis of not only its primary identifier but also its properties

*Graph Based Databases (Neo4j):* A graph database uses graph structures with nodes, edges, and properties to represent and store data.

*Column Based Databases (HBase):* The column-oriented storage allows data to be stored effectively. It avoids consuming space when storing nulls by simply not storing a column when a value doesn’t exist for that column.

*Key Value Databases (Membase):* The key of a key/value pair is a unique value in the set and can be easily looked up to access the data.

1. **CAP Theorem**

*Consistency* - This means that the data in the database remains consistent after the execution of an operation. For example after an update operation, all clients see the same data.

*Availability* - This means that the system is always on (service guarantee availability), no downtime.

*Partition Tolerance* - This means that the system continues to function even if the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

***Case1:*** Duplicate Copy of same data is maintained on Multiple Machines. This increases availability, however decreases consistency. (A^ C)

***Case2:*** If data on one machine changes, the update propagates to the other machine, system is inconsistent, but will become eventually consistent. (A C)

***Case3:*** If duplicate copy of same data is not maintained, consistency is superior BUT availability decreases. (C^ A)

**If we are dealing with distributed systems then it is assumed to be Partition Tolerant and game is between Availability and Consistency while choosing NoSQL database.**

1. **HBase Architecture**

HBase is composed of three types of servers in a master slave type of architecture.

* Region servers serve data for reads and writes.
* HBase Master process handles the Region assignment, DDL (create, delete tables) operations
* Zookeeper maintains a live cluster state.

***4.1 Region:***

* HBase Tables are divided horizontally by row key range into “Regions.”
* A region contains all rows in the table between the region’s start key and end key.
* Regions are assigned to the nodes in the cluster, called “Region Servers,” and these serve
* data for reads and writes.
* A region server can serve about 1,000 regions.
  1. ***HBase Master***
* Region assignment, DDL (create, delete tables) operations are handled by the HBase Master.
* A master is responsible for:
  + Coordinating the region servers
  + Assigning regions on startup
  + Re-assigning regions for recovery or load balancing
  + Monitoring all RegionServer instances in the cluster (listens for notifications from zookeeper)
* Admin functions
  + Interface for creating, deleting, updating tables
  1. ***Zookeeper:***
* HBase uses ZooKeeper as a distributed coordination service to maintain server state in the cluster.
* Zookeeper maintains which servers are alive and available, and provides server failure notification.
* Zookeeper uses consensus to guarantee common shared state. Note that there should be three or five machines for consensus.

1. **HBase vs RDBMS**

