Assignment 9.3

1. NoSQL DataBase:

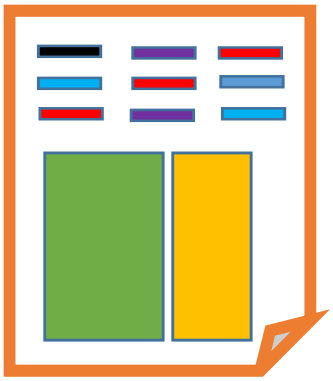
NoSQL database provides a mechanism for **storage** and **retrieval** of data that is modeled in means other than the tabular relations used in **Relational Database**. NoSQL databases are increasingly used in **Big Data** and **Real-Time Web Applications**. NoSQL systems are also sometimes called "**Not Only SQL**" to emphasize that they may support SQL-like query languages.

There are 4 types of NoSQL DataBase:

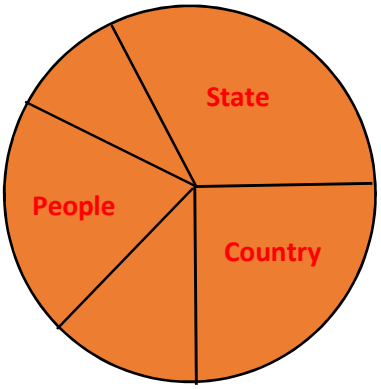
a. **Graph: N**eo4j, Titan
b. **Key-Value:** Cassandra, Dynamo DB
c. **Column Store:** HBase, Big Table
d. **Document:** Mono DataBase

2. Types of NoSQL database

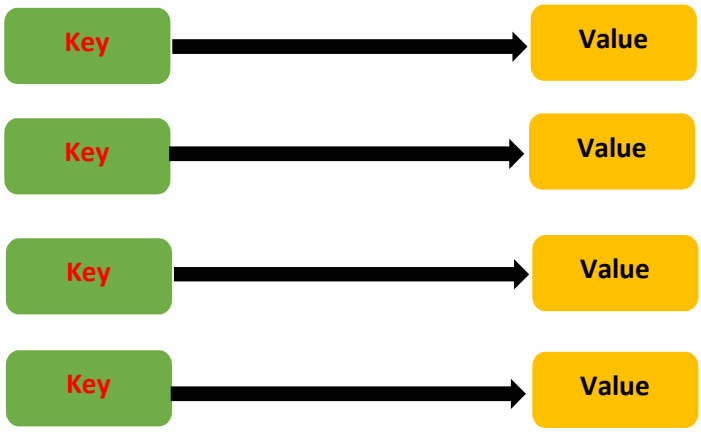
a. **Document databases** pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.



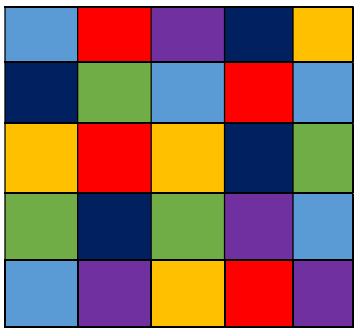
b. **Graph stores** are used to store information about network of data. Graph stores include Neo4J and Giraph.



c. **Key-value stores** are the simplest NoSQL database. Every single item in the database is stored as an attribute name (or "Key"), together with its value.



d. **Wide-Column stores** such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

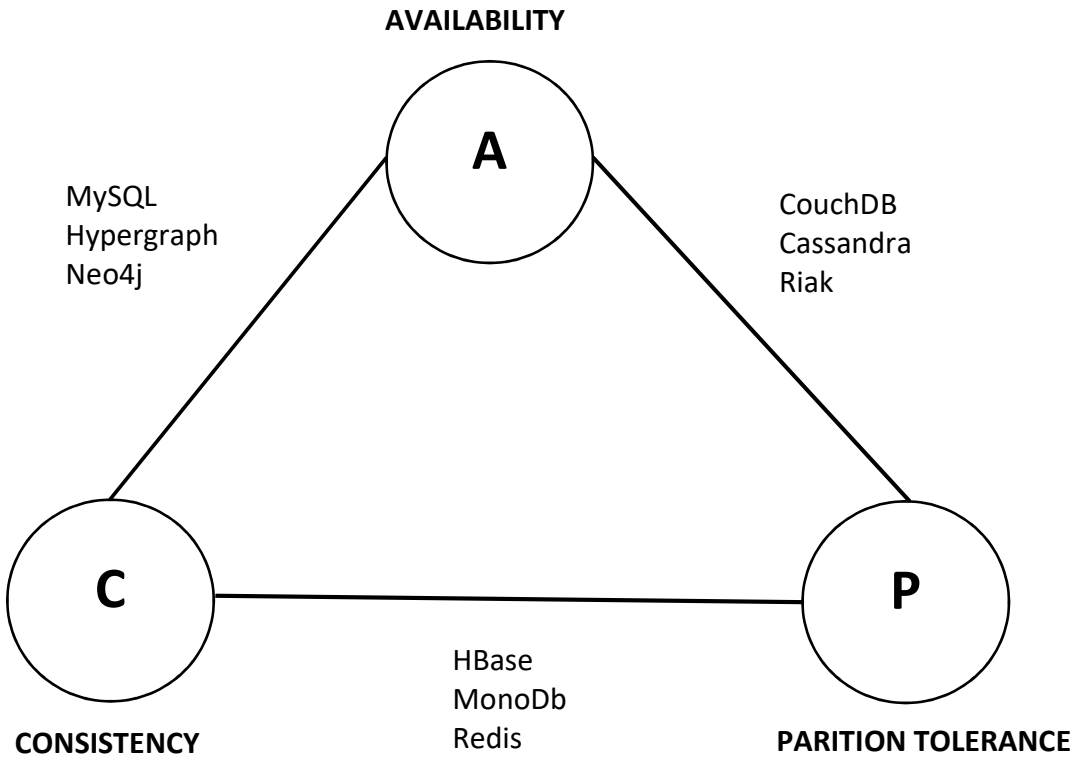


3. CAP Theorem:

Consistency: All the servers in the system will have the same data so anyone accessing the system will get the same copy regardless of which server answer's the request.

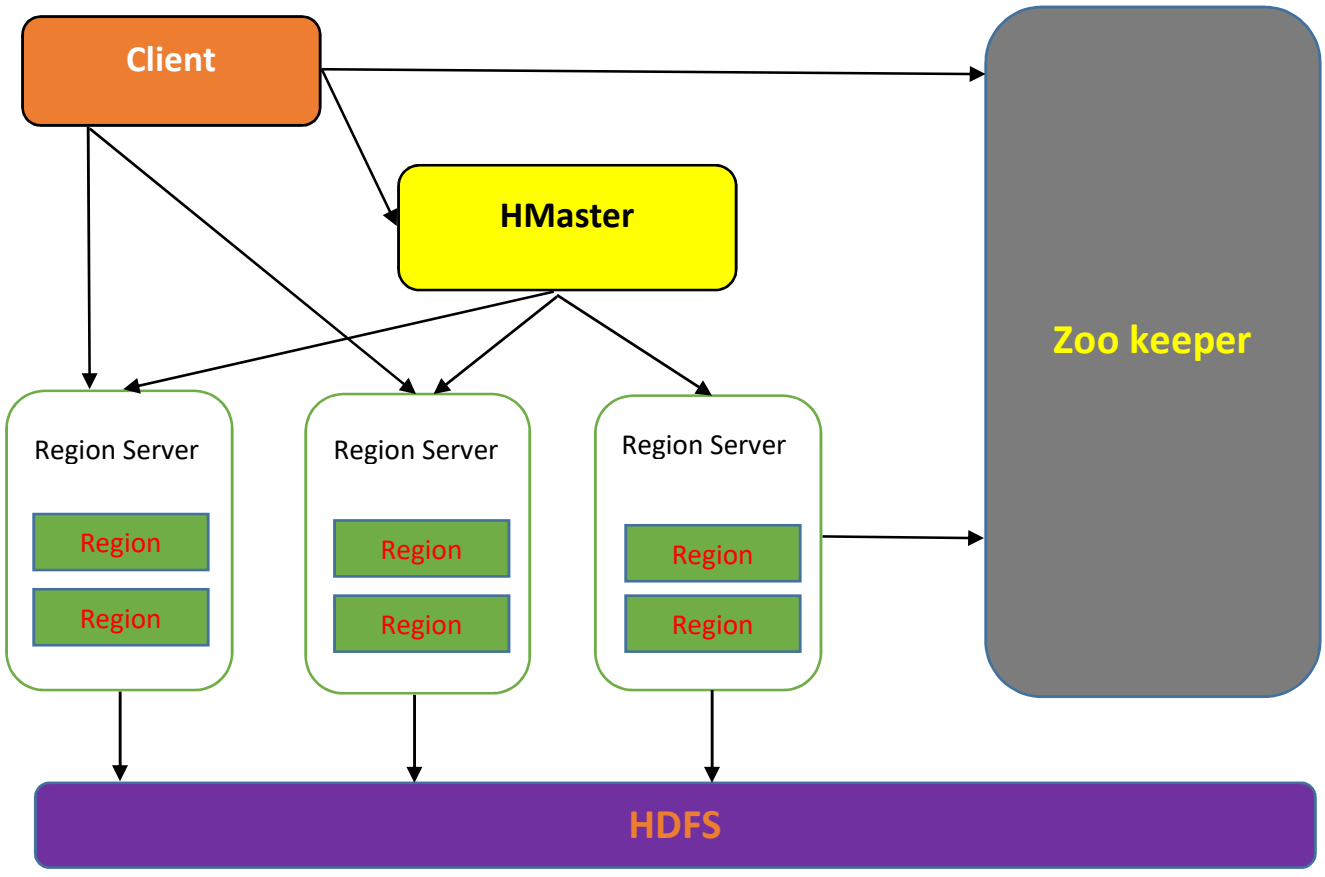
Availability: The system will always respond to a request.

Partition Tolerance: The system continues to operate as a whole even if individuals server fail or can't be reached.



4. HBase Architecture:

HBase architecture has a single HBase master node (HMaster) and several slaves i.e. region servers. Each region server (slave) serves a set of regions, and a region can be served only by a single region server. Whenever a client sends a write request, HMaster receives the request and forwards it to the corresponding region server.



5. HBase VS RDBMS

HBASE	RDBMS
Column Oriented	**Row Oriented** (mostly)
Flexible Schema, add columns on the fly	**Fixed Schema**
Good with sparse tables	**Not optimized for spares tables**
Joins using MR –not optimized	**Optimized for joins**
Tight integration with MR	**Not really**
Horizontal Scalability	**Hard to share and scale**
Good for Semi and Un-Structured Data	**Good for Structured Data**