1.Nosql Databases:

Document-Oriented Databases

* [Couchbase](http://www.couchbase.com/)
* [CouchDB](http://couchdb.apache.org/)
* [MongoDB](http://www.mongodb.org/)
* [Riak](http://sendgrid.com/blog/what-is-nosql/basho.com/riak/)

In-Memory Databases

* [Memcached](http://memcached.org/)
* [Redis](http://redis.io/)
* Riak
* [VoltDB](http://voltdb.com/)

Graph Databases

* [InifiniteGraph](http://www.objectivity.com/infinitegraph#.U3Y0GS_rmVI)
* [Neo4J](http://www.neo4j.org/)
* [OrientDB](http://www.orientechnologies.com/orientdb/)

Column Store Databases

* [Apache HBase](http://hbase.apache.org/)
* [Cassandra](http://sendgrid.com/blog/what-is-nosql/cassandra.apache.org)
* [Google’s BigTable](http://research.google.com/archive/bigtable.html)

2.TYPES OF NOSQL DATABASES

Several different varieties of NoSQL databases have been created to support specific needs and use cases. These fall into four main categories:

Key-value data stores: [Key-value NoSQL databases](http://basho.com/resources/nosql-databases/) emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned and replicated across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deal with high-velocity, non-transactional data.

Ex:Hbase,cassandra

Document stores: [Document databases](http://basho.com/resources/document-databases/) typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure.

Ex:Mongo DB

Wide-column stores: Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval.

Graph stores: A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

3. CAP Theorem

* C.Consistency: every read would get you the most recent write
* A.Availability: every node (if not failed) always executes queries
* P.Partition-tolerance: even if the connections between nodes are down, the other two (A & C) promises, are kept.

### 4.Components of Apache HBase Architecture

HBase architecture has 3 important components- HMaster, Region Server and ZooKeeper.

#### HMaster

HBase HMaster is a lightweight process that assigns regions to region servers in the Hadoop cluster for load balancing. Responsibilities of HMaster –

* Manages and Monitors the Hadoop Cluster
* Performs Administration (Interface for creating, updating and deleting tables.)
* Controlling the failover
* DDL operations are handled by the HMaster
* Whenever a client wants to change the schema and change any of the metadata operations, HMaster is responsible for all these operations.

#### Region Server

These are the worker nodes which handle read, write, update, and delete requests from clients. Region Server process, runs on every node in the hadoop cluster. Region Server runs on HDFS DataNode and consists of the following components –

* Block Cache – This is the read cache. Most frequently read data is stored in the read cache and whenever the block cache is full, recently used data is evicted.
* MemStore- This is the write cache and stores new data that is not yet written to the disk. Every column family in a region has a MemStore.
* Write Ahead Log (WAL) is a file that stores new data that is not persisted to permanent storage.
* HFile is the actual storage file that stores the rows as sorted key values on a disk.

#### Zookeeper

HBase uses ZooKeeper as a distributed coordination service for region assignments and to recover any region server crashes by loading them onto other region servers that are functioning. ZooKeeper is a centralized monitoring server that maintains configuration information and provides distributed synchronization. Whenever a client wants to communicate with regions, they have to approach Zookeeper first. HMaster and Region servers are registered with ZooKeeper service, client needs to access ZooKeeper quorum in order to connect with region servers and HMaster. In case of node failure within an HBase cluster, ZKquoram will trigger error messages and start repairing failed nodes.

ZooKeeper service keeps track of all the region servers that are there in an HBase cluster- tracking information about how many region servers are there and which region servers are holding which DataNode. HMaster contacts ZooKeeper to get the details of region servers. Various services that Zookeeper provides include –

* Establishing client communication with region servers.
* Tracking server failure and network partitions.
* Maintain Configuration Information
* Provides ephemeral nodes, which represent different region servers.

# RDBMS vs HBase

There differences between RDBMS and HBase are given below.

* Schema/Database in RDBMS can be compared to namespace in Hbase.
* A table in RDBMS can be compared to column family in Hbase.
* A record (after table joins) in RDBMS can be compared to a record in Hbase.
* A collection of tables in RDBMS can be compared to a table in Hbase..