# What is NoSQL data base?

The term NoSQL was coined by Carlo Strozzi in the year 1998. He used this term to name his Open Source, Light Weight, DataBase which did not have an SQL interface.

NoSQL is a non-relational (generic data model) database management systems, designed for distributed data stores with very large scale of data storing needs. This type of data storing may not require fixed schema (non-relational), avoid join operations(de-normalized) and typically scale horizontally.

# How does data get stored in NoSQl database?

There are various NoSQL Databases. Each one uses a different method to store data. Some might use column store, some document, some graph, etc., Each database has its own unique characteristics.

## Document Oriented Databases

It’s a collection of documents that stores data. A document is a key value collection where the key allows access to its value. Documents are stored into collections in order to group different kinds of data and can contain many different key-value pairs, or key-array pairs, or even nested documents. E.g. MongoDB

## Key-value stores

Key-value stores are most basic types of NoSQL databases designed to handle huge amounts of data.In the key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBjec) etc. For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin". Key-Value stores follow the 'Availability' and 'Partition' aspects of CAP theorem and can be used as collections, dictionaries, associative arrays etc. e.g. Redis, Dynamo, Riak. Etc

## Column-oriented databases

Column-oriented databases primarily work on columns and values of a single column are stored contiguously.A ll data within each column datafile have the same type which makes it ideal for compression. Column stores can improve the performance of queries as it can access specific column data. e.g. HBase, BigTable, Cassandra, SimpleDB etc.

## Graph Databases

A graph database stores data in a graph structure which consists of a finite (and possibly mutable) set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices. Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes. Every node and edge are defined by a unique identifier. E.g. Neo4j

# What is a column family in HBase?

A **column family** is a NoSQL object that contains **columns** of related data. It is a tuple (pair) that consists of a key-value pair, where the key is mapped to a value that is a set of **columns**. Applications store **data** into an **HBase** table. Tables are made of rows and columns. All columns in **HBase** belong to a particular column family.

# How many maximum number of columns can be added to HBase table?

There is no hard limit

# Why columns are not defined at the time of table creation in HBase?

HBase doesn’t have a fixed schema and is designed to store semi-structured data. Column families are defined at the time of table creation while columns get assigned to a family while adding data.

# How does data get managed in HBase?

Just like in a Relational Database, data in HBase is stored in Tables and these Tables are stored in Regions. When a Table becomes too big, the Table is partitioned into multiple Regions. These Regions are assigned to Region Servers across the cluster. Each Region Server hosts roughly the same number of Regions and can host about 1000 regions at maximum.

The Data Model in HBase is designed to accommodate semi-structured data that could vary in field size, data type and columns. Additionally, the layout of the data model makes it easier to partition the data and distribute it across the cluster. The Data Model in HBase is made of different logical components such as Tables, Rows, Column Families, Columns, Cells and Versions.

What happens internally when new data gets inserted into HBase table?

When the client issues a Put request, the first step is to write the data to the write-ahead log, the WAL. Edits are appended to the end of the WAL file that is stored on disk. Once the data is written to the WAL, it is placed in the MemStore. Then, the put request acknowledgement returns to the client. The MemStore stores updates in memory as sorted KeyValues, the same as it would be stored in an HFile. There is one MemStore per column family. The updates are sorted per column family. When the MemStore accumulates enough data, the entire sorted set is written to a new HFile in HDFS. HBase uses multiple HFiles per column family, which contain the actual cells, or KeyValue instances. These files are created over time as KeyValue edits sorted in the MemStores are flushed as files to disk. This is a sequential write. It is very fast, as it avoids moving the disk drive head.