NoSQL DATABASE

DBMS-Database management system is mainly used for storage and retrieval of data. Firstly we have many database management systems.

RDBMS-Structured data, Tables

OLAP-Data warehousing, Cubes

NoSQL-Structured or unstructured data, Collections

Previously, we used to store our data in a file system. As time passed, it became a difficult to store in them as the data is not stored in a format . There is no predefined structure to store the data. Updating and retrieval operations were very much difficult to perform on that data.

Then in 1970s relational database management system came into existence. We can easily create, insert, update and delete the data . Fetching data from the database became an fastest task. The data in it is stored in the form of rows and columns due to which storing data was an easy process.

As years passed by digitization has grown up very fast and the data we are dealing daily becoming large day by day. The data produced in recent days is in unstructured form as it can be audio text, images, videos, images, text messages, log files etc. So storing that kind of data is not allowed in RDBMS as it has some pre-defined structure to store data in it. As RDBMS unable to handle big data, NoSQL came into the picture.

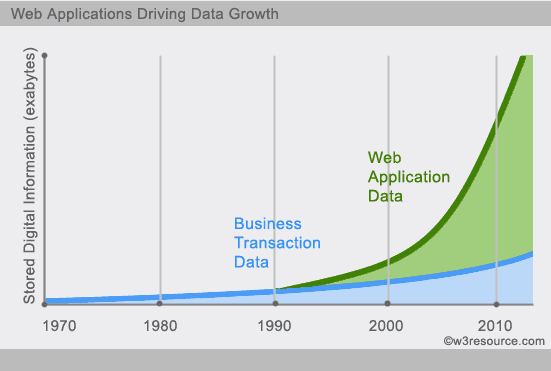
What is NoSQL?

The name stands for Not Only SQL which was introduced by Carl Strozzi in 1998 to name his file-based database. It was again re-introduced by Eric Evans when an event was organized to discuss oopen source distributed databases. NoSQL is a non-relational database management systems, different from traditional relational database management systems in some significant ways. It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users). These type of data storing may not require fixed schema, avoid join operations and typically scale horizontally.



Why NoSQL?

In today’s time data is becoming easier to access and capture through third parties such as Facebook, Google+ and others. Personal user information, social graphs, geo location data, user-generated content and machine logging data are just a few examples where the data has been increasing exponentially. To avail the above service properly, it is required to process huge amount of data. Which SQL databases were never designed. The evolution of NoSql databases is to handle these huge data properly.



**Example:-**

**Social-network graph :**

Each record:UserID1,UserID2

Separate records: UserID, first\_name, last\_name, age, gender,etc.

Task: Find all friends of friends of friends of ... friends of a given user.

**Wikipedia pages :**

Large collection of documents

Combination of structured and unstructured data

Task: Retrieve all pages regarding athletics of Summer Olympic before 1950.

Objective:-

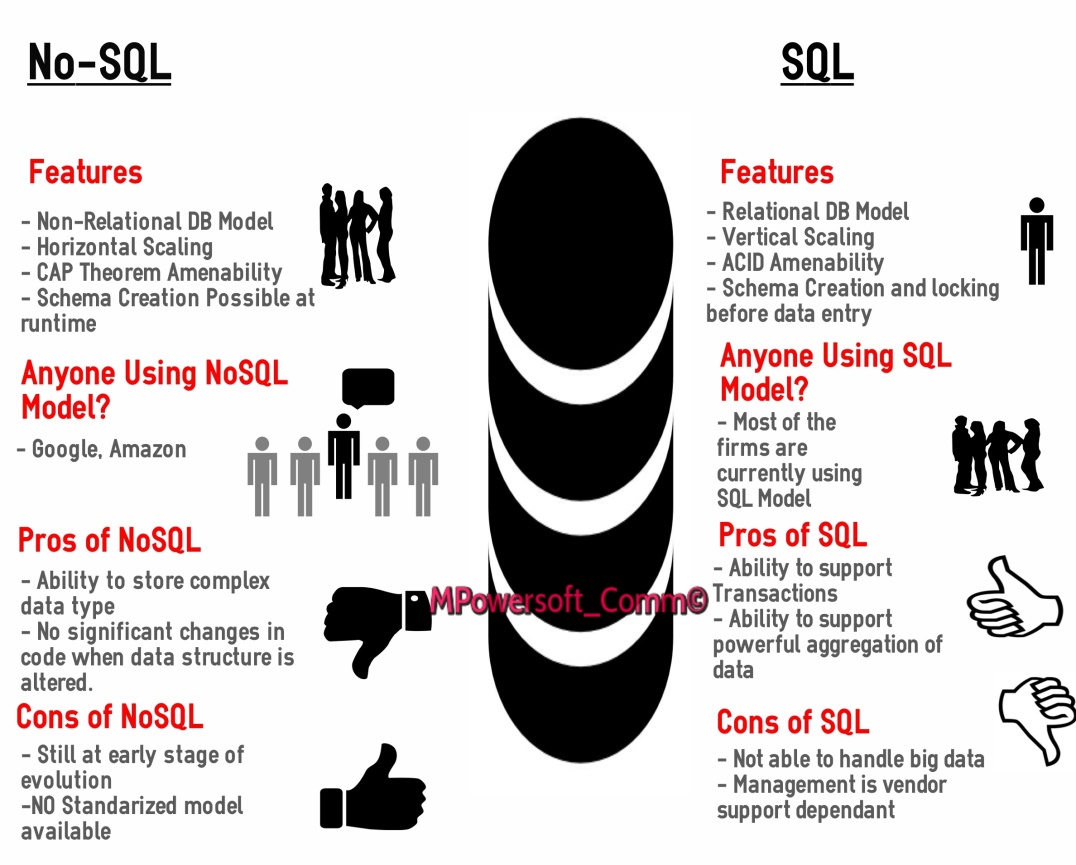
NoSQL is focused to provide:

1. Scalability

2. Performance

3. High availability

How NoSQL is different from RDBMS:-



NoSQL pros/cons:-

Advantages:

* Non-relational
* Don’t require schema
* Data are replicated to multiple nodes(identical, fault tolerance)
* Horizontal scalable
* Massive write performance
* Fast key-value access

Disadvantages:

* Don’t fully support relational features
* No join, group by, order by operations(except within partitions)
* No referential integrity constraints across partitions
* No declarative query language(e.g., SQL)
* Relaxed ACID properties
* No easy integration with other applications that support SQL

CAP Theorem:-

Suppose three properties of a distributed system(sharing data)

* Consistency: all copies have same value
* Availability: reads and writes always succeed
* Partition-tolerance: system properties(consistency &/or availability) hold even when network failures prevent some machines from communicating with others.



Types of NoSQL databases:

There are four categories of NoSQL databases where each of these has its own attributes and limitations.

1.Key-value store

2. Document-based

3. Column-based

4. Graph-based

1. Key-value store databases:-

* Key-value stores are most basic types of NoSQL databases.
* Designed to handle huge amounts of data.
* Based on Amazon’s Dynamo paper.
* Key value stores allow developer to store schema-less 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.
* A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.
* For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin".
* Key-Values stores would work well for shopping cart contents, or individual values like color schemes, a landing page URI, or a default account number.

**Example of Key-value store databases :**Redis, Dynamo, Riak. etc.

2. Document-based databases:-

* A collection of documents.
* Data in this model is stored inside documents.
* A document is a key value collection where the key allows access to its value.
* Documents are not typically forced to have a schema and therefore are flexible and easy to change.
* Documents are stored into collections in order to group different kinds of data.
* Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

**Example of Document Oriented databases :**MongoDB, CouchDB etc.

3. Column-based databases:-

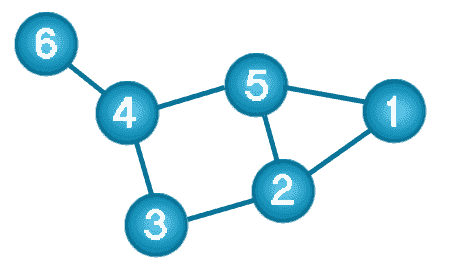
* Column-oriented databases primarily work on columns and every column is treated individually.
* Values of a single column are stored contiguously.
* Column stores data in column specific files.
* In Column stores, query processors work on columns too.
* All data within each column datafile have the same type which makes it ideal for compression.
* Column stores can improve the performance of queries as it can access specific column data.
* High performance on aggregation queries (e.g. COUNT, SUM, AVG, MIN, MAX).
* Works on data warehouses and business intelligence, customer relationship management (CRM), Library card catalogs etc.

**Example of Column-oriented databases :**BigTable, Cassandra, SimpleDB etc.

4. Graph-based databases:-

A graph data structure consists of a finite (and possibly mutable) set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices.

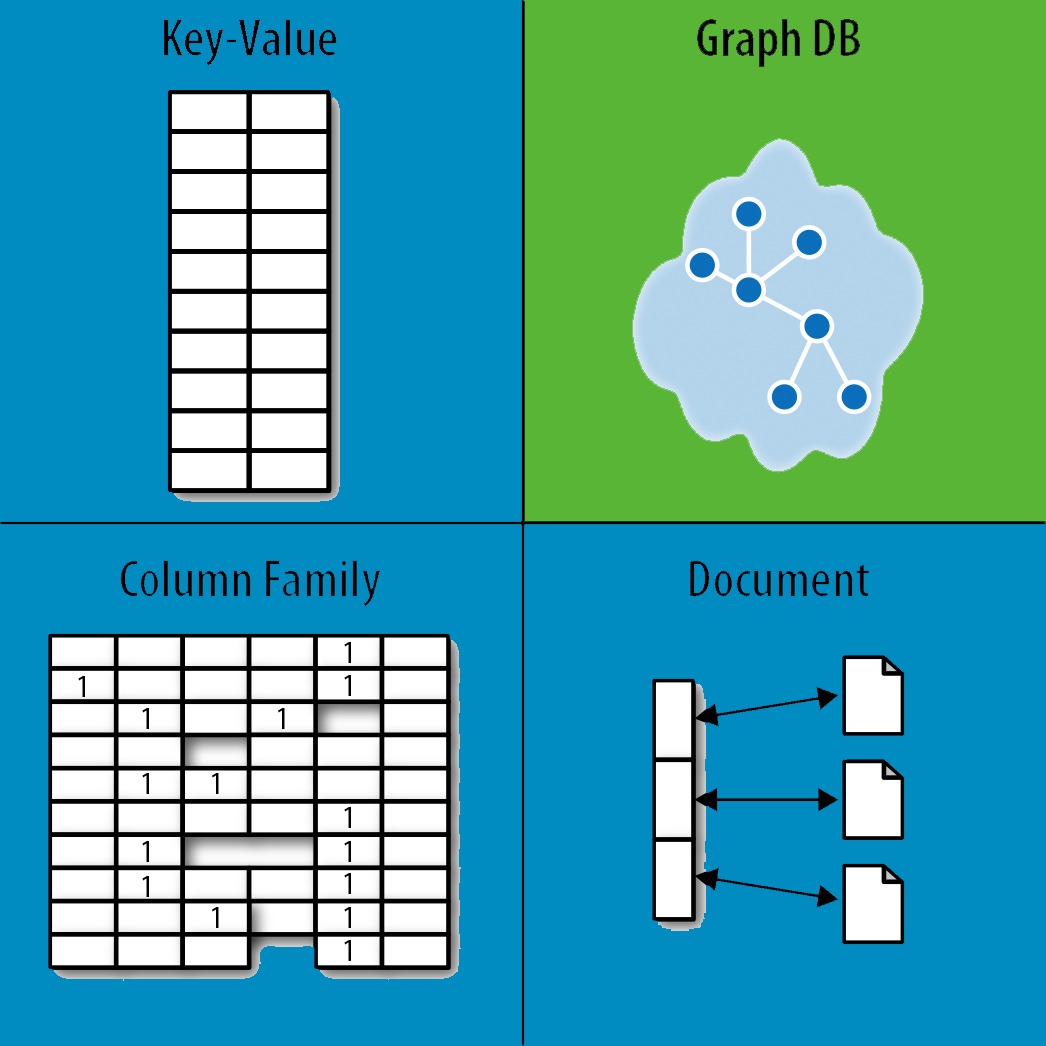
The following picture presents a labeled graph of 6 vertices and 7 edges.



What is a Graph Databases?

* A graph database stores data in a graph.
* It is capable of elegantly representing any kind of data in a highly accessible way.
* A graph database is a collection of nodes and edges.
* 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.
* Each node knows its adjacent nodes.

**Example of Graph databases** : OrientDB, Neo4J, Titan.etc.



Product Deployment:-

There are large number of companies using NoSQL. To name a few:

* Cassandra- Twitter, Facebook, Rackspace, Reddit, Digg
* HBase- TCS, Cloudera, Hortonworks
* BigTable- Google
* Dynamo- Amazon
* Voldemort-LinkedIn
* HyperTable- Baidu
* MongoDB- FourSquare,Craigslist