Definition of nosql

****

NoSQL is not a relational database.

The reality is that a relational database model may not be the best solution for all situations.it is called not only SQL or non sql or non-relational.Early NoSQL databases used for web and cloud applications. NoSQL databases are document based, key-value pairs, graph databases or wide-column stores.

List of nosql databases:
****
NoSQL database examples: Mongo DB, Big Table, Redis, Raven DB, Cassandra, Hbase, and Couch DB

Characteristics of NOSQL Systems (or) advantages of NoSql

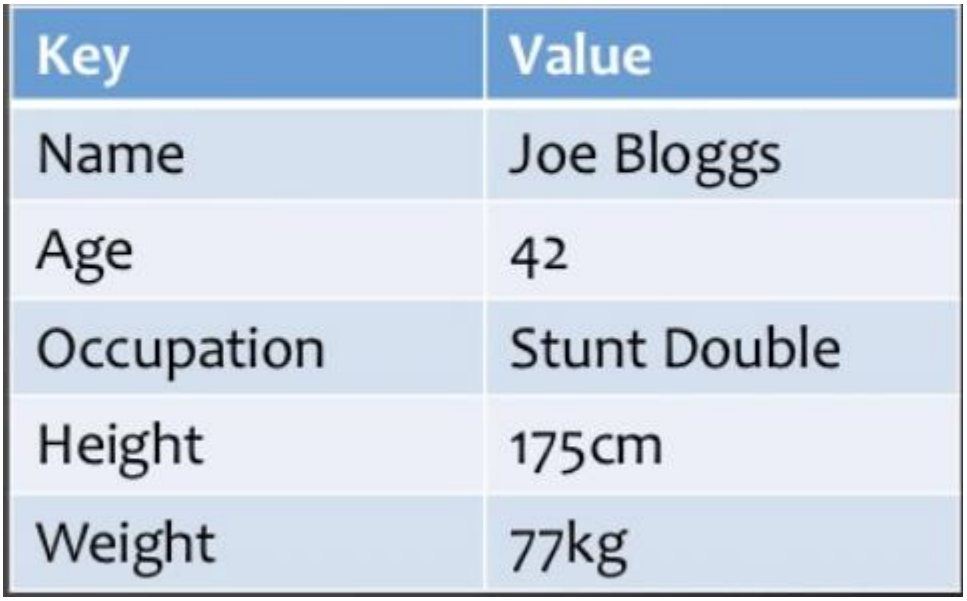
- **It's more than rows in tables—NoSQL systems store and retrieve data from many formats: key-value stores, graph databases, column-family (Big table) stores and document stores**
- **It's free of joins—NoSQL systems allow you to extract your data using simple interfaces without joins.**
- **It's schema-free—NoSQL systems allow you to drag-and-drop your data into a folder and then query it without creating an entity-relational model.**
- **It works on many processors—NoSQL systems allow you to store your database on multiple processors and maintain high-speed performance.**
- **It uses shared-nothing commodity computers—most (but not all) NoSQL systems leverage low-cost commodity processors that have separate RAM and disk.**
- **It supports linear scalability—When you add more processors, you get a consistent increase in performance.**
- **It's innovative—NoSQL offers options to a single way of storing, retrieving, and manipulating data. NoSQL supporters (also known as NoSQLers) have an inclusive**
- **Attitude about NoSQL and recognize SQL solutions as viable options. To the NoSQL community, NoSQL means "Not only SQL."**
- **High availability –**
  Auto replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.
- **High scalability – NoSQL database use sharding for horizontal scaling. Vertical scaling means adding more resources to the existing machine**
- **They can process both unstructured and semi-structured data.**

Categories of NOSQL Systems (OR) TYPES OF NOSQL Databases

NoSQL Databases **are mainly categorized into four types:**

- **Key-value Pair Based**
- **Column-oriented Graph**
- **Graphs based**
- **Document-oriented**

Key Value Pair Based: Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load. Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB (Binary Large Objects), string, etc. For example, a key-value pair may contain a key like "Website" associated with a value like "Guru99".



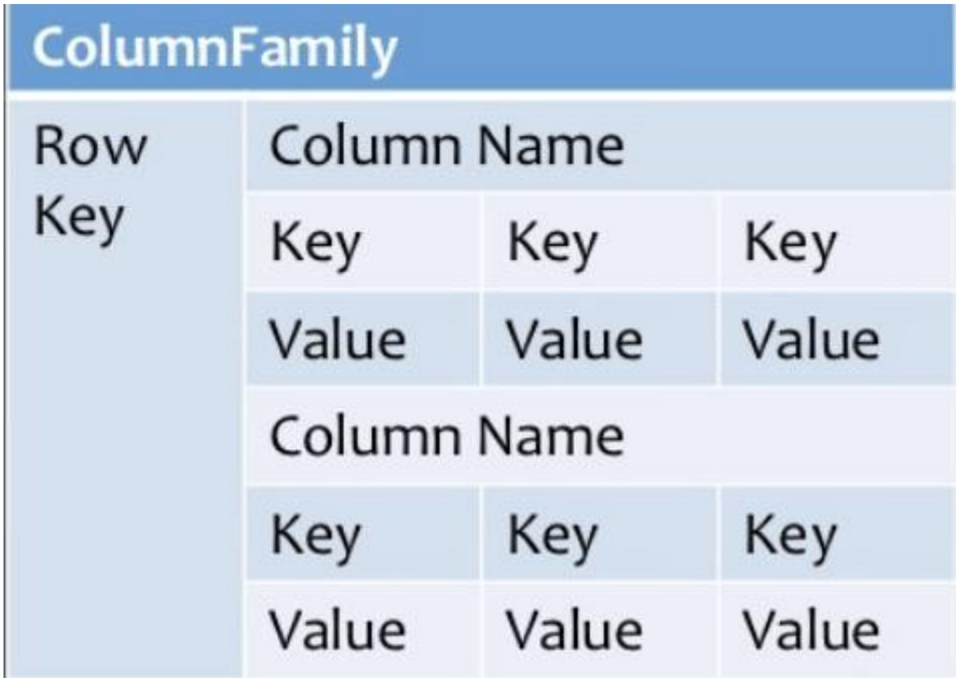
Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

It is one of the most basic NoSQL database example. This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc. Key value stores help the developer to store schema-less data. They work best for shopping cart contents.

Redis, Dynamo are some NoSQL examples of key-value store Databases. They are all based on Amazon's Dynamo paper.

Column-based

Column-oriented databases work on columns and are based on Big Table paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.



ColumnFamily			
Row Key	Column Name		
	Key	Key	Key
	Value	Value	Value
	Column Name		
	Key	Key	Key
	Value	Value	Value

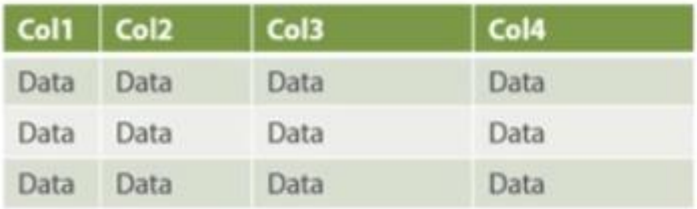
They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.

Column-based NoSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs,

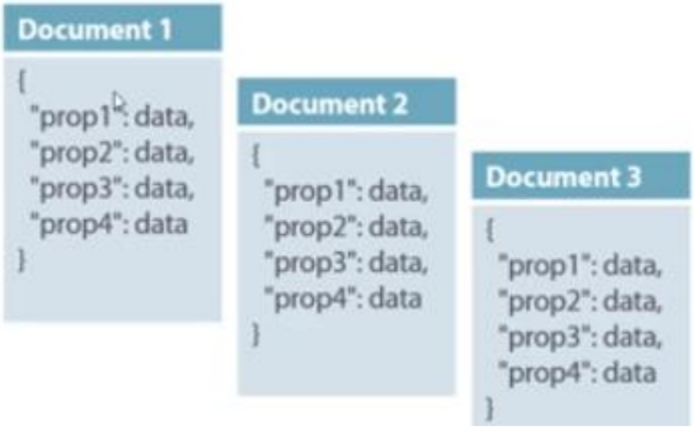
HBase, Cassandra, HBase, Hypertable are NoSQL query examples of column based database.

Document-Oriented:

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.



Col1	Col2	Col3	Col4
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data	Data	Data



Document 1

```
{
 "prop1": data,
 "prop2": data,
 "prop3": data,
 "prop4": data
}
```

Document 2

```
{
 "prop1": data,
 "prop2": data,
 "prop3": data,
 "prop4": data
}
```

Document 3

```
{
 "prop1": data,
 "prop2": data,
 "prop3": data,
 "prop4": data
}
```

Relational Vs. Document

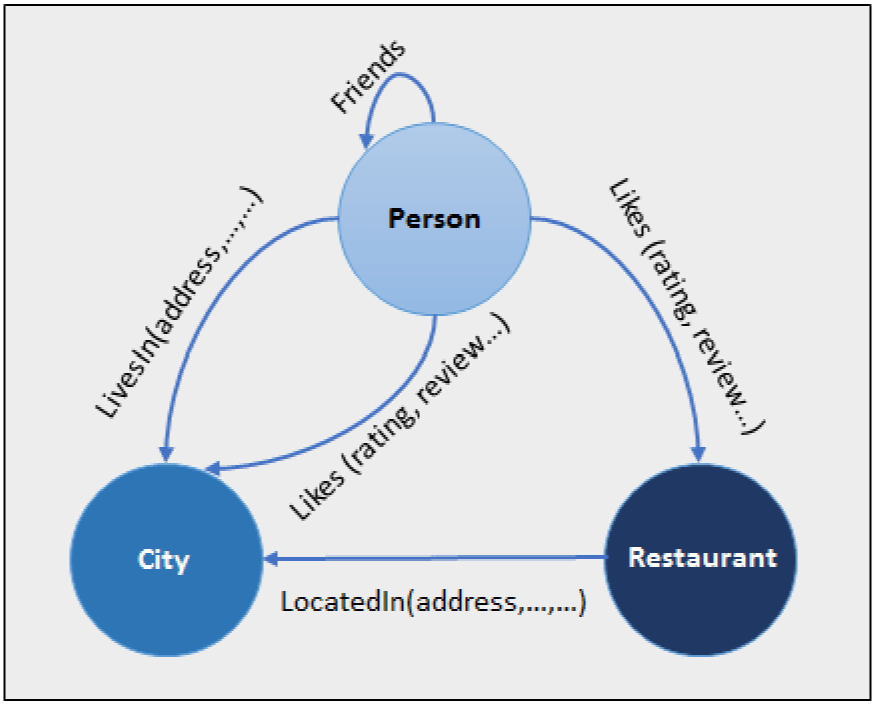
In this diagram on your left you can see we have rows and columns, and in the right, we have a document database which has a similar structure to JSON. Now for the relational database, you have to know what columns you have and so on. However, for a document database, you have data store like JSON object. You do not require to define which make it flexible.

The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.

Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, are popular Document originated DBMS systems.

Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.



Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.

Graph base database mostly used for social networks, logistics, spatial data.

Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.