Name : Shekar C
Reg.No: 18CS601025
Course Code : MCA504A51
Course Title : NoSQL DATABASES
No. of Pages : 7
College : Krista Jayanti College

(1) What is NoSQL :

NoSQL databases are different than relational databases like MQSQL. In relational database you need to Create the table, define schema, set the data types of fields etc. In NoSQL you don't have to worry about you can insert, update data on the fly. One advantages of NoSQL database is that they are really easy to scale and they are much faster in most types of operations that we perform on databases.

Types of NoSQL Databases :

NoSQL Databases are mainly Categorized into four types, Every Category has its unique attributes and limitations. User should select the databases based on their Product needs.

i) Key - Value Pair Based
ii) Column - oriented Graph
iii) Graphs based
iv) Document - oriented

1) Key Value Pair Data:

Data is stored in key/value pairs. It is designed in such a way to handle lots of data & heavy load.

Key - value pair storage database store data as a hash table where each key is unique, and the value can be a JSON, BLOB (Binary Large Object), string, etc.

Example: a key - value pair contain a key like "website" associated with a value like "Hello".

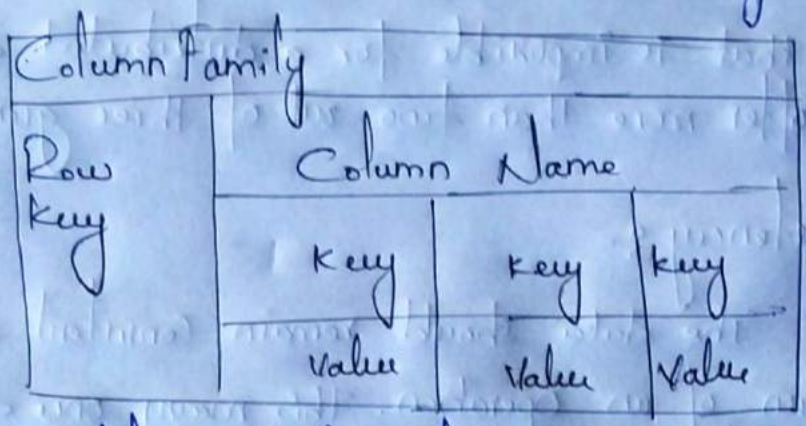
| Key | Value |
|---|---|
| Name | Shek |
| Age | 49 |

2) Column - based:

Column - oriented databases work on columns and are based on BigTable plan.

Graph

Every column is treated separately. Values of single column databases are stored contiguously.



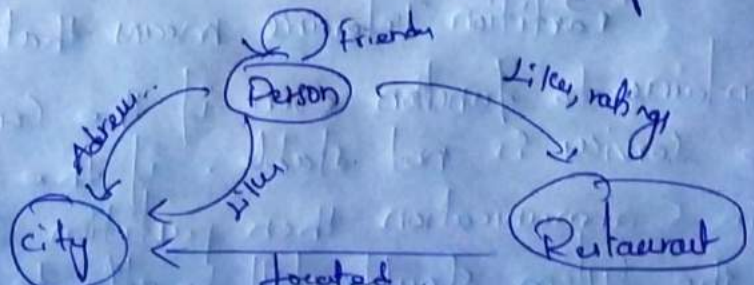

They delivers high performance on aggregation queries like SUM, COUNT, AVG ...

3) Document - Oriented :

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

4) A Graph - Based :

A Graph type database stores entities as well the relations amongst those entities. The entity is based stored as a node with relationship as edge.




Query C

(2) CAP Theorm :

CAP theorem is also called brewer's theorem. It states that is impossible for a distributed data store to offer more than two out of three guarantees.

1) Consistency :-

The data should remain consistent even after the execution of an operation. This means once data is written, any future read request should contain the data.

Consistency is of importance when it is required that all the clients or users view the same data.

2) Availability :-

Availability means that every request from the user should elicit a response from the system. Whether the user wants to read or write, the user should get a response even if the operation was unsuccessful.

3) Partition Tolerance :-

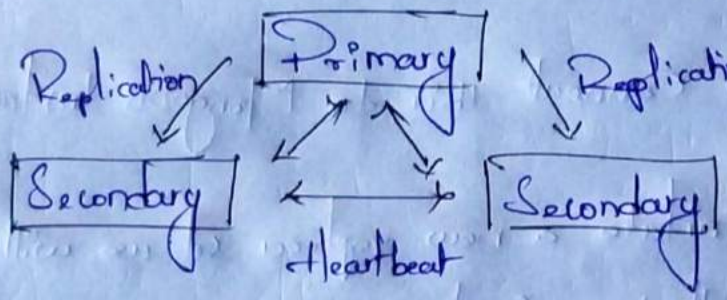
Partition Tolerance means that the system should continue to function even if the communication among the servers is not stable. If the partition means a break in communication then partition tolerance would mean that the system should still be able to work.

Eg. C

(3) Replication:

MongoDB replication means instance that maintain the same data set. It Contains several data bearing nodes optionally one arbiter node. Out of all the data bearing nodes, only one of them is primary node while the other are Secondary nodes.

A primary node can do all the write Operations. A replica set containing primary node is can Confirm writes with {w: "majority"}.




How to Set Up a Replica Set in MongoDB:

Here,

Converting a standlone MongoBD instance to the replica.

Following are the steps to Convert:

Step 1: Shutdown the already running server.

Step 2: Again start the server by writing the following syntax:

> mongod --port "PORT" --dbpath "c:/data/.." --replSet "Replic_Set_Instance_name"

Eg:
mongd --port 27017 -dbpat "c:\setup\.." --replset rs0,

Step 3: It will start an instance with name rs0, on port 27017.

Step 4: Now Connect this to MongoDB Instance

Step 5: To initate a new replica set we will use the following Syntax:

> rs.add(HOST_NAME:PORT).

The Secondary nodes replicate the primary one and apply to the Operations to their respective dataset.

Qus:c

(4) Collection :

Collection is a group of MongoDB documents. A Collection is the equivalent of an RDBMS table. A Collection exists within a single database. Collections do not enforce a schema.

The insert() Method :

To insert data into MongoDB Collection, Need to use MongoDB's insert() or save() method.

Syntax :

> db.Collection_name.insert(document)

Ex :

>db.CreateCollection
> db.user.insert({title : "MongoDB"});

The insertOne() Method :

If you need to insert only one document into a Collection you can use this method.

Syntax :

>db.Collection_name.insertOne(document).