**CSD Module4 Discussion: Advantages & Disadvantages of MongoDB (Notes)**

1. First and foremost, it is very easy to install and setup the MongoDB.
2. The very basic feature of MongoDB is that it is a schema-less database. No schema migrations anymore. Since MongoDB is schema-free, your code defines your schema.
3. The ability to derive a document-based data model is one of the most attractive advantages of MongoDB. Because, the way it stores the data in the form of BSON (Binary JSON), ruby hashes etc., helps to store the data in a very rich way while being capable of holding arrays and other documents.
4. The document query language supported by MongoDB plays a vital role in supporting dynamic queries.
5. Very easy to scale.
6. Due to the structuring (BSON format - key value pair) way of the data in MongoDB, no complex joins are needed.
7. Performance tuning is absolutely easy compared to any relational databases.
8. No need of mapping the application objects to the data objects.
9. Enables faster access of the data due to its nature of using the internal memory for the storage.
10. Since it is a NOSQL database, then it is obviously secure because no sql injection can be made.
11. MongoDB can also be used as a file system, which helps in easier way of load balancing.
12. MongoDB supports, the search by regex and fields as well.
13. MongoDB can be run as windows service as well.
14. Good amount of documentation is available.
15. MongoDB does not require a VM to be run.
16. MongoDB follows regular release cycle of its newer versions.
17. The support for Sharding is one of its key feature. Sharding is the process of storing the data in different machines and MongoDB's ability to process the data, as and when the size of the data grows. This results in the horizontal scaling. With sharding, more amount of data can be written and read back as and when there is an increase in the data growth.

Data Insert Consistency

Basically, MongoDB likes to have more data insertion rate over the safety concerns of doing inserts in a transaction. Hence the write consistency is low. If there is a need for huge load of data to be written, without the worry of losing some data, then MongoDB should be preferred and really it's best suited.

Data Corruption Recovery

When data recovery process needs to be faster, safe and automatic, MongoDB is preferred. In MySQL if a database (a few tables) become corrupt, you can repair them individually by deleting/updating the data. In MongoDB, you have to repair on a database level. But there is a command to do this automatically, but it reads all the data and re-writes it to a new set of files. So if your database is huge, it might take some time, and for that time your DB will be locked. But again, this is better than losing the complete dataset.

Load Balancing

When data grows infinitely and proper load balancing of the same is required, MongoDB is the best solution. Because, it supports, faster replica setting options and its built in Sharding feature.

Avoid JOINS

When the developers do not want to normalize their data and insist on not using of any JOINS, then they should really go for MongoDB. For Example : If there are 2 collections student and address (where a student can have more than one address). In a typical RDBMS, to fetch the addresses associated to a student from the **address** table, JOIN is used. But, in MongoDB, the address data can be embedded as a document in the **student** collection itself. Hence, without using any JOIN all the required details of student and address can be fetched with one simple query.

There are a few disadvantages of the MongoDB NoSQL database as well.

* MongoDB uses high memory for data storage.
* There is a limit for document size, i.e. 16mb.
* There is no transaction support in MongoDB.

Work Cited:

<https://acodez.in/mongodb-nosql-database/>