MongoDB

INTRODUCTION:

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON (similar to JSON format).

ADVANTAGES OF MongoDB:

-Schema Not Required

-Simplified Performance Optimization

-Horizontal Scaling with Sharding

-Replication and Workload Distribution

-Ease of Maintenance

Features of MongoDB:

- Rich query language
- Support of multiple storage engine
- Select queries are faster due to faster indexing support
- Horizontal scalability due to sharding feature

SQL (Structured Query Language) Databases:

Structure: SQL databases are table-based. They store data in rows and columns, similar to a spreadsheet.

Schema: They use a fixed schema, meaning you must define the structure of your data (like columns and data types) before you can store anything.

Queries: They use SQL to interact with the database, which is a standardized language for managing and manipulating data.

Examples: Popular SQL databases include MySQL, PostgreSQL, Oracle, and SQL Server.

Best for: They are great for applications that require complex queries and transactions, such as financial systems and enterprise applications.

NoSQL (Not Only SQL) Databases:

Structure: NoSQL databases are more flexible and can store data in various formats like documents, key-value pairs, graphs, or wide-columns.

Schema: They use a dynamic schema, allowing you to store data without defining the structure beforehand. You can change the structure of your data without downtime.

Queries: They use different query languages depending on the database type. For example, MongoDB uses a JSON-like query language.

Examples: Popular NoSQL databases include MongoDB (document-based), Redis (key-value), Neo4j (graph), and Cassandra (wide-column).

Best for: They are ideal for applications that need to handle large volumes of unstructured data, such as big data applications, real-time web apps, and content management systems.

Organizations that use MongoDB:

- Adobe
- LinkedIn
- eBay
- SAP
- McAfee

MongoDB DATA TYPES:

- **String:** The string must be a valid UTF-8.
- **Integer:** We can store integer data type in two forms 32 -bit signed integer and 64 – bit signed integer.
- **Double:** The double data type is used to store the floating-point values.
- **Boolean:** The boolean data type is used to store either true or false.
- **Null:** The null data type is used to store the null value.
- **Array:** In MongoDB, the array is created using square brackets([]).
- **Object:** Object data type stores embedded documents.
- **Object Id:** There is an _id field in MongoDB for each document.
- **Undefined:** This data type stores the undefined values.

- **Min & Max key:** Min key compares the value of the lowest BSON element and Max key compares the value against the highest BSON element.
- **Date:** Date data type stores date. It is a 64-bit integer which represents the number of milliseconds.
- **Binary Data:** This datatype is used to store binary data.
- **Symbol:** It is generally not supported by a mongo shell, but if the shell gets a symbol from the database, then it converts this type into a string type.
- **Regular Expression:** This datatype is used to store regular expression

USES OF MongoDB:

1. **Web and Mobile Applications:**
- MongoDB's flexibility and performance make it an excellent choice for web and mobile applications that require rapid development cycles and scalability. It powers many e-commerce platforms, social networks, and content management systems.

2. **Big Data and Analytics:**
- MongoDB is capable of handling large volumes of data, making it suitable for big data applications. It integrates well with data processing frameworks like Apache Spark for real-time analytics and insights.

3. **Content Management Systems:**
- The schema-less nature of MongoDB is ideal for managing diverse content types, such as articles, videos, and images, without the need for predefined schemas.

4. **IoT and Sensor Data:**
- For Internet of Things (IoT) applications, MongoDB can efficiently store and process data from a multitude of sensors, allowing for real-time monitoring and analysis.

5. **Catalogs and Inventory Management:**
- E-commerce platforms and other businesses use MongoDB to manage product catalogs and inventory systems that require flexibility in data models and the ability to handle large datasets.

DATABASE:
Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

COLLECTIONS:
Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.Typically, all documents in a collection are of similar or related purpose.

DOCUMENT:
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same

set of fields or structure, and common fields in a collection's documents may hold different types of data.

Sample Document

```
{
  id : ObjectId("5099803df3f4948bd2f98391"),
  name : {first: "Alan", last: "Turing" },
  birth:new Date('Jun 23, 1912'),
  death : new Date('Jun 07, 1954'),
  contribs: [ "Turing machine", "Turing test", "Turingery" ],
  view : NumberLong(1250000)
}
```

MongoDB Installation Process
Download and Install MongoDB on windows using
https://www.mongodb.com/download-center/community
<div align="center">OR</div>
Refer :- https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/

Download Mongodb shell using https://www.mongodb.com/try/download/shell

MongoDB Shell Commands:

-To clear the screen:
    cls
-To view existing databases:
    show dbs
-To view collections:
    show collections
-To create or connect to an existing database:
    Use<db
- To check currently selected database:
    db
- To drop database, first select the database and then drop it by:
db.dropDatabase()
- To insert a document to students collection:
db.students.insert({name: "Viren" })
- To drop students collection:
  db.students.drop()
- To query data from collection:

```
db.students.find()
```
- To display the results in a formatted way:
```
db.students.find().pretty()
```