**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).

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today’s real-world highly growing applications. **Modern applications are more networked, social and interactive than ever**. Applications are storing more and more data and are accessing it at higher rates.

**Where do we use MongoDB?**

* Big Data: If you have huge amount of data to be stored in tables, think of MongoDB before RDBMS databases. MongoDB has built-in solution for partitioning and sharding your database.
* Unstable Schema: Adding a new column in RDBMS is hard whereas MongoDB is schema-less. Adding a new field does not effect old documents and will be very easy.
* Distributed data Since multiple copies of data are stored across different servers, recovery of data is instant and safe even if there is a hardware failure.

**Features of MongoDB:**

* Document Oriented: MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.
* Indexing: Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.
* Scalability: MongoDB scales horizontally using sharding (partitioning data across various servers). Data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. Also, new machines can be added to a running database.

# **Database & Collection**

Databases, collections, documents are important parts of MongoDB without them you are not able to store data on the MongoDB server. A Database contains a collection, and a collection contains documents and the documents contain data, they are related to each other.

To view a database we use the command:

**show dbs**

To use the database we use the command:

**use db**

**Collections:**

To show the collections stored in mongodb compass we use the command

**show collections**

# **DataTypes in MongoDB:**

**1. String:** This is the most commonly used data type in MongoDB to store data, BSON strings are of UTF-8. So, the drivers for each programming language convert from the string format of the language to UTF-8 while serializing and de-serializing BSON. The string must be a valid UTF-8.

**2. Integer:** In MongoDB, the integer data type is used to store an integer value. We can store integer data type in two forms 32 -bit signed integer and 64 – bit signed integer.

**3.Double:** The double data type is used to store the floating-point values.

**4. Boolean:** The boolean data type is used to store either true or false.

**5. Null**: The null data type is used to store the null value.

**6. Array:** The Array is the set of values. It can store the same or different data types values in it. In MongoDB, the array is created using square brackets([]).

**MongoDB Methods:**

# **1.Find() Method:**

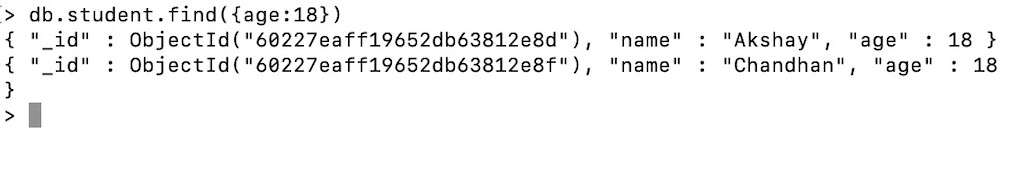
The **find ()** method in MongoDB **selects documents in a collection** that matches the specified conditions and returns a cursor to the selected documents

## **Syntax**

db.Collection\_name.find(selection\_criteria,projection,options)

Example:

db.student.find({age:18})



# **2.sort() Method:**

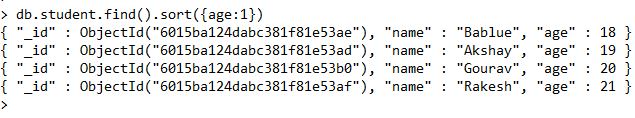
The **sort()** method specifies the order in which the query returns the matching documents from the given collection. You must apply this method to the cursor before retrieving any documents from the database. It takes a document as a parameter that contains a field: value pair that defines the sort order of the result set. The value is 1 or -1 specifying an ascending or descending sort respectively.

**Syntax:**

db.Collection\_Name.sort({field\_name:1 or -1})

**Example:**

db.student.find().sort({age:1})



**3.count() Method:**

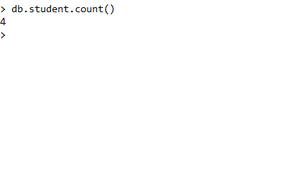
The count() method counts the number of documents that match the selection criteria. It returns the number of documents that match the selection criteria. It takes two arguments first one is the selection criteria and the other is optional.

**Syntax:**

db.Collection\_name.count()

**Example:**

**db.student.count()**

****