

MONGO DB INTRODUCTION



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)



[This Photo](#) by Unknown Author is licensed under [CC BY-NC-ND](#)

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).

A simple MongoDB document Structure:

```
{  
  title: 'Data Science',  
  by: 'Supreeth VS',  
  url: 'https://www.datascience.org',  
  type: 'NoSQL'  
}
```

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.

Relational Database Management System(RDBMS) is **not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable**. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

RDBMS vs MongoDB:

- RDBMS has a typical schema design that shows number of tables and the relationship between these tables whereas MongoDB is document-oriented. There is no concept of schema or relationship.
- Complex transactions are not supported in MongoDB because complex join operations are not available.
- MongoDB allows a highly flexible and scalable document structure. For example, one data document of a collection in MongoDB can have two fields whereas the other document in the same collection can have four.
- MongoDB is faster as compared to RDBMS due to efficient indexing and storage techniques.
- There are a few terms that are related in both databases. What's called Table in RDBMS is called a Collection in MongoDB. Similarly, a Row is called a Document and a Column is called a Field. MongoDB provides a default '_id' (if not provided explicitly) which is a 12-byte hexadecimal number that assures the uniqueness of every document. It is similar to the Primary key in RDBMS.

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.
- **Replication and High Availability:** MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.
- **Aggregation:** Aggregation operations process data records and return the computed results. It is similar

to the GROUPBY clause in SQL. A few aggregation expressions are sum, avg, min, max, etc

Where do we use MongoDB?

MongoDB is preferred over RDBMS in the following scenarios:

- **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.

Installing MongoDB:

Just go to <http://www.mongodb.org/downloads> and select your operating system out of [Windows](#), [Linux](#), [Mac OS X](#) and Solaris. A detailed explanation about the installation of MongoDB is given on their site.

For Windows, a few options for the 64-bit operating systems drops down. When you're running on Windows 7, 8 or newer versions, select **Windows 64-bit 2008 R2+**. When you're using Windows XP or Vista then select **Windows 64-bit 2008 R2+ legacy**.

RDBMS	MongoDB
It is a relational database .	It is a non-relational and document-oriented database.
Not suitable for hierarchical data storage.	Suitable for hierarchical data storage .
It is vertically scalable i.e increasing RAM.	It is horizontally scalable i.e we can add more servers.
It has a predefined schema.	It has a dynamic schema.
It is quite vulnerable to SQL injection.	It is not affected by SQL injection .
It centers around ACID properties (Atomicity, Consistency, Isolation, and Durability).	It centers around the CAP theorem (Consistency, Availability, and Partition tolerance).
It is row-based.	It is document-based.
It is slower in comparison with MongoDB.	It is almost 100 times faster than RDBMS.

RDBMS	MongoDB
Supports complex joins.	No support for complex joins.
It is column-based.	It is field-based.
It does not provide JavaScript client for querying.	It provides a JavaScript client for querying.
It supports SQL query language only.	It supports JSON query language along with SQL .

What is ObjectId in MongoDB

Every document in the collection has an “_id” field that is used to uniquely identify the document in a particular collection it acts as the primary key for the documents in the collection. “_id” field can be used in any format and the default format is **ObjectId** of the document.

An ObjectId is a 12-byte Field Of BSON type

- The first 4 bytes representing the Unix Timestamp of the document

- The next 3 bytes are the machine Id on which the MongoDB server is running.
- The next 2 bytes are of process id
- The last Field is 3 bytes used for increment the objectid.

Timestamp(4)	Machine ID(3)	Process.Id (2)	Increment(3)
--------------	---------------	----------------	--------------

What is a MongoDB Query?

MongoDB Query is a fundamental aspect of MongoDB that allows users to fetch data from the database. Similar to **SQL** queries in traditional databases, MongoDB queries provide simplicity and flexibility in retrieving specific data based on certain criteria or conditions.

In this article, We will learn about the MongoDB Query in detail by performing various **Queries on document** of MongoDB collection and so on.

MongoDB Query

- [MongoDB Query](#) allows retrieving data from the MongoDB database. MongoDB Query provides simplicity in the process of fetching data from the [database](#), it's similar to SQL queries in [SQL](#) Database language.

- While performing a query operation, one can also use criteria or conditions that can be used to retrieve specific data from the database.
- MongoDB provides the function names as **db.collection_name.find()** to operate query operations on the database.