

MONGO DB

INTRODUCTION:

MongoDB is a popular open-source NoSQL (non-relational) database system designed for storing and retrieving data in a flexible and scalable manner. It's a powerful choice for modern applications that deal with:

- Large and complex data structures: MongoDB stores data in flexible documents, similar to JSON objects. Documents can contain various data types, including nested structures and arrays, making them ideal for representing real-world entities and their relationships.
- Evolving data models: Unlike relational databases with rigid schemas, MongoDB allows documents within a collection (analogous to a table) to have different structures. This flexibility is crucial for applications where data models are likely to change over time.
- High performance and scalability: MongoDB is built for horizontal scaling, meaning you can easily add more servers to handle growing data volumes or increased query load. It offers features like sharding to distribute data efficiently across a cluster.
- Data Model and Schema Analysis: This report delves into the design of your MongoDB data model. It explains how collections and documents are structured, how effectively they represent your data, and potential areas for optimization.
- Migration Planning: If you're considering migrating data to MongoDB, this report would assess the feasibility and potential challenges. It might compare data models, analyze query performance implications, and outline the migration process.
- Security Assessment: This report focuses on the security posture of your MongoDB deployment. It would evaluate access controls, encryption strategies, and any potential security vulnerabilities.
- Evaluating Performance: This type of report focuses on measuring the query performance, throughput, and scalability of MongoDB for a specific application. It might analyze factors like average query time, latency, and how well MongoDB handles increasing data volumes or workloads.
- Comparing MongoDB to Relational Databases: This report pits MongoDB against a relational database for a particular use case. It would highlight the strengths and weaknesses of each approach, considering factors like data model complexity, query patterns, and scalability requirements.

HOW MONGO DB WORKS:

Now, we will see how actually thing happens behind the scene. As we know that MongoDB is a database server and the data is stored in these databases. Or in other words, MongoDB environment gives you a server that you can start and then create multiple databases on it using MongoDB.

Because of its NoSQL database, the data is stored in the collections and documents. Hence the database, collection, and documents are related to each other.

- The MongoDB database contains collections just like the MYSQL database contains tables. You are allowed to create multiple databases and multiple collections.
- Now inside of the collection we have documents. These documents contain the data we want to store in the MongoDB database and a single collection can contain multiple documents and you are schema-less means it is not necessary that one document is similar to another.
- The documents are created using the fields. Fields are key-value pairs in the documents, it is just like columns in the relation database. The value of the fields can be of any BSON data types like double, string, boolean, etc.
- The data stored in the MongoDB is in the format of BSON documents. Here, BSON stands for Binary representation of JSON documents. Or in other words, in the backend, the MongoDB server converts the JSON data into a binary form that is known as BSON and this BSON is stored and queried more efficiently.
- In MongoDB documents, you are allowed to store nested data. This nesting of data allows you to create complex relations between data and store them in the same document which makes the working and fetching of data extremely efficient as compared to SQL. In SQL, you need to write complex joins to get the data from table 1 and table 2. The maximum size of the BSON document is 16MB.

ADVANTAGES OF MONGO DB:

- It is a schema-less NoSQL database. You need not to design the schema of the database when you are working with MongoDB.
- It does not support join operation.
- It provides great flexibility to the fields in the documents.
- It contains heterogeneous data.
- It provides high performance, availability, scalability.
- It supports Geospatial efficiently.
- It is a document oriented database and the data is stored in BSON documents.
- It also supports multiple document ACID transition(string from MongoDB 4.0).
- It does not require any SQL injection.
- It is easily integrated with Big Data Hadoop

INSTALLING MONGO DB:

- Visit the MongoDB download page on the official website.
- Click the “On-Premises” tab (for MongoDB on your own infrastructure).
- Select “MongoDB Community Server.”
- Choose your desired version and platform (Windows).
- Click the download button to get the installation file.
- Once downloaded, extract the ZIP archive to a location with write permissions.
- Navigate to the “bin” folder within the extracted files.
- Run the MongoDB server using the command: .
 - `mongod.exe --dbpath=pathtodatadirectory`

Replace `pathtodatadirectory` with the actual path where you want to store MongoDB data1

What is Data?

DATA is statically raw and unprocessed information. For example – name, class, marks, etc. In computer language, a piece of information that can be translated into a form for efficient movement and processing is called data. Data is interchangeable information.

What is a Database?

DATABASE is a collection of data that is organized, which is also called structured data. It can be accessed or stored in a computer system.

- Database is a container for collections.
- Each database gets its own set of files.
- A single MongoDB server has multiple databases.

DATA TYPE:

Databases can hold various kinds of information, including text, numbers, images, videos and more.

SQL and NO-SQL:

SQL: SQL stands for Structured Query Languages. To access a database, we use SQL statements.

NO-SQL: NO-SQL databases (also known as “not only SQL”) are non tabular databases that store data differently than traditional relational databases. SQL

SQL vs NoSQL IN MONGO DB:

Feature	SQL (Relational Databases)	MongoDB (NoSQL)
Data Model	Structured tables with rows and columns (fixed schema)	Flexible documents (JSON-like) with dynamic schema
Schema	Predefined schema enforced for all data	Flexible schema, documents within a collection can have different structures
Relationships	Defined through foreign keys and joins between tables	Embedded documents or references within documents
Scaling	Primarily vertical scaling (upgrades)	Horizontal scaling (adding more servers)
Queries	Uses SQL for querying data	Uses its own query language specific to document structure
ACID Properties	Typically enforces ACID properties (Atomicity, Consistency, Isolation, Durability)	Often prioritizes Availability over Consistency (following the CAP theorem)

Choosing Between SQL and MongoDB

The choice between SQL and MongoDB depends on your specific needs:

Use SQL if:

You have a well-defined and stable data model.

Your application relies on complex queries involving joins across multiple tables.

Maintaining strict data consistency is critical.

Use MongoDB if:

You have a flexible or evolving data model.

You need high performance for large datasets or complex documents.

Horizontal scalability is a major requirement.

