Introduction

mongoDB is an open-source, NoSQL database that uses a flexible, JSON-like document model to store data.

Database : A database is essentially a big organized collection of information stored electronically on a computer system.

SQL v/s NoSQL:

SQL and NoSQL are two different approaches to managing data in databases. Here's a breakdown of their key differences:

SQL (Structured Query Language)

Structure: Relational - Data is stored in tables with rows and columns, and tables are linked together based on relationships between data points.

Schema: Predefined - Requires a defined structure for data upfront (like specifying data types for columns).

Query Language: Uses SQL, a standardized language for querying and manipulating data.

Scalability: Primarily vertically scalable (adding more resources to a single server).

Use Cases: Excellent for well-defined data structures, complex queries involving joins across tables, and maintaining data consistency (ACID properties). Common in business applications, inventory management, and traditional enterprise systems.

NoSQL (Not Only SQL or Non-Relational)

Structure: Non-relational - Data can be stored in various formats like documents, key-value pairs, graphs, or wide-column stores. Offers more flexibility for unstructured or evolving data.

Schema: Dynamic - Schema can evolve as new data is added without strict upfront definition.

Query Language: Varies depending on the database type (e.g., Javascript for MongoDB).

Scalability: Primarily horizontally scalable (adding more servers to distribute the load).

Use Cases: Ideal for big data, large amounts of unstructured data (like social media posts), frequent data changes, and applications requiring high availability and performance.

Here's an analogy: Imagine a library.

SQL database: Like a traditional library with books categorized and shelved in a specific order. Easy to find specific information but requires the books to fit predefined categories.

NoSQL database: Like a modern library with various sections for different media (books, movies, audiobooks). More flexible for storing different types of content but browsing might require searching across sections.

Ultimately, the choice between SQL and NoSQL depends on your specific data needs and application requirements.

Instructions to download MongoDB

There are two main ways to download MongoDB depending on your preference:

1.Using an Installer: This is the recommended method for most users, especially beginners. Here's how to do it:

- Go to the official MongoDB download page [download mongodb].
- Select the desired version of MongoDB.
- Choose "Windows" from the Platform dropdown menu.
- Pick "msi" from the Package options (this is the installer type).
- Click the "Download" button.
- Run the downloaded installer and follow the on-screen instructions to complete the installation.

2.Downloading the Zip Archive: This method offers more flexibility but requires some manual configuration. Here's how to download the archive:

- Follow steps 1 and 2 from the previous method to access the download page [download mongodb].
- Keep the version selected but choose "Zip archive" from the Package options.
- Click "Download" and extract the downloaded zip file to a convenient location on your computer.

Once you have downloaded MongoDB using either method, you can proceed with the installation process which involves starting the MongoDB service. There might be slight variations depending on the chosen download method, so you can refer to the official documentation for detailed instructions:

https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/

Installation of Mongo Shell OR Studio3t

- Mongo Shell download link
- All the work is expected to do it in mongo shell not in mongo compass OR
- You can also install Studio3T
- Connect to mongodb://localhost:27017

Add,Update and Delete:

Commands

Command	Expected Output	Notes
show dbs	admin 40.00 KiB config 72.00 KiB db 128.00 KiB local 40.00 KiB	All Databases are shown
use db	switched to db db	Connect and use db
show collections	Students	Show all tables
db.foo.insert({"bar" : "baz"})		Insert a record to collection. Create Collection if not exists

Command	Notes
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows
db.foo.remove()	Remove foo table

Documents:

In MongoDB, documents are the fundamental unit of data storage. They are similar to JSON objects and act as containers for your data. Here's a closer look at documents in MongoDB:

Structure:

- Documents are composed of field-value pairs. Each field acts like a key with a corresponding value.
- Values can be of various data types, including strings, numbers, booleans, arrays, and even nested documents (documents within documents).
- This flexible structure allows you to model complex data relationships without rigid table structures.

Benefits of Documents:

- Flexibility: Documents can accommodate diverse data structures, making them ideal for modeling complex data.
- Scalability: MongoDB's document-oriented storage scales well horizontally by adding more servers.
- Ease of Use: The JSON-like structure of documents makes them easy to understand and work with, especially for developers familiar with JSON.

Collections

- **Function:** Collections are the fundamental units of data organization in MongoDB. They act like containers that hold documents (similar to JSON objects) representing your data.
- **Structure:** Each collection has a unique name within a database. Documents within a collection can vary in structure, but they typically share a common set of fields or properties.
- **Relationships:** Collections can be linked through references, but MongoDB doesn't enforce strict relationships like relational databases. This flexibility allows for more schema-less data modeling.

Database

In MongoDB, a database is a container that holds multiple collections of documents. It's like a filing cabinet that keeps your information organized.

Creating a Simple Database:

1. **Connect to MongoDB:** You can use a graphical tool like MongoDB Compass or the command-line interface (CLI).
2. **Create a Database:** In Compass, click "Create Database" and give it a name (e.g., "my_bookstore"). The CLI uses the use database_name command.
3. **Create a Collection:** In Compass, right-click the database and choose "Create Collection". Give it a name (e.g., "books"). The CLI uses the db.createCollection("collection_name") command.
4. **Add Documents:** In Compass, click the collection and start adding documents with fields and values (e.g., title, author, genre). The CLI uses the db.collection.insertOne({...}) or db.collection.insertMany([...]) commands.

Datatype

- String. One of the most basic and widely used data types is the string. ...
- Integer. Numeric values are stored using the integer data type. ...
- Double. Numeric numbers containing 8 bytes (64-bit IEEE 754 floating point) floating-point are stored using the double data type. ...
- Boolean. ...
- Array. ...
- Object. ...
- Date. ...
- Timestamp.

MongoDB Use Cases

- Artificial Intelligence. Streamline building AI-enriched applications. ...
- Edge Computing. Unlock the benefits of edge computing by simplifying data management. ...
- Internet of Things. Analyze and act on data from the physical world. ...
- Mobile. ...
- Payments. ...
- Serverless Development. ...
- Single View. ...
- Personalization.