MongoDB

Introduction and Features

1. What is MongoDB?

MongoDB is a **NoSQL database** that stores data in a flexible, **JSON-like format called BSON**. It's different from traditional SQL databases, which use tables. Instead, MongoDB uses **collections** and **documents**, making it easier to handle unstructured or semi-structured data.

2. Why use MongoDB?

- Ideal for **big data**, **real-time analytics**, and **applications that evolve quickly**.
- Schemaless structure allows storing data without a fixed model.
- Highly **scalable**, **fast**, and **easy to integrate** with modern applications.

3.Key Features:

- **Document-Oriented**: Stores data in key-value pairs like JSON.
- **Flexible Schema**: No need to define the structure before inserting data.
- **Scalability**: Supports horizontal scaling (sharding).
- **High Performance**: Handles large volumes of data efficiently.

Installation Steps

1. Install MongoDB Community Server

- Go to the MongoDB Download Center.
- Choose OS (Windows/macOS/Linux).
- Download the **MongoDB Community Server**.
- Install it using the default options.
- After installation, MongoDB will run as a **background service**.

2. Install MongoDB Compass

- From the same page, download **MongoDB Compass**.
- It's a GUI for managing MongoDB databases.

- After installation, open Compass and connect using:

```
mongodb://localhost:27017
```

This connects to local MongoDB server.

3. How it works

- When MongoDB runs, it listens on **port 27017** by default.
- Data is stored in **collections** inside **databases**.
- We can insert, read, update, or delete documents using the MongoDB shell or Compass.

Databases, Collections, and Documents in MongoDB

1.What is a Database?

A database is like a big folder that stores related data. In MongoDB, a database contains collections, which hold our actual data.

2. What is a Collection?

A collection is like a table in SQL. It's a group of documents. Collections do not enforce any structure, so documents can look different from one another.

3. What is a Document?

A document is a single piece of data stored in JSON-like format (BSON). It’s like a row in SQL but more flexible.

Example

```
{
  "name": "Shahla",
  "email": "sha@example.com",


}
```

Main Difference Between SQL and NoSQL

- **SQL (Relational Database)** stores data in **tables** with **fixed rows and columns**. We must define the structure (schema) before adding data.
- **NoSQL (MongoDB)** stores data in **collections of documents** (like JSON). The structure is **flexible**, and each document can be different.