**What is MongoDB?**

MongoDB is a **NoSQL database** that uses a **document-oriented model** to store data. Unlike traditional relational databases that use tables and rows, MongoDB stores data in **JSON-like documents** with flexible schemas, making it easy to work with complex and evolving data structures.

**Key Characteristics of MongoDB:**

* **NoSQL Database:** It is a non-relational database designed to handle unstructured or semi-structured data.
* **Document-Oriented:** Stores data as documents in collections instead of rows and tables.
* **Schema Flexibility:** Documents can have varying structures, which allows for more flexible data modeling.
* **High Scalability:** Supports horizontal scaling through sharding, making it suitable for large-scale applications.
* **High Performance:** Designed for fast read and write operations.

**Key Features MongoDB Provides**

1. **Document Model:**
   * Stores data as **BSON** (Binary JSON) documents.
   * Supports complex nested structures and arrays, which are not easily achievable in relational databases.
2. **Schema Flexibility:**
   * Unlike relational databases, MongoDB allows documents in the same collection to have different structures.
   * This makes it easier to evolve the data model without requiring schema migrations.
3. **Indexing:**
   * Supports various types of indexes like single-field, compound, multi-key, geospatial, text, and hashed indexes.
   * Improves query performance by allowing fast lookups.
4. **Replication:**
   * Provides high availability through **replica sets**, which consist of a primary node and secondary nodes.
   * Automatic failover and recovery are supported.
5. **Sharding:**
   * MongoDB supports horizontal scaling by partitioning data across multiple servers using sharding.
   * Ensures high availability and load balancing.
6. **Aggregation Framework:**
   * Allows for advanced data processing and analytics.
   * Supports operations like filtering, grouping, and transforming data.
7. **Ad-hoc Queries:**
   * MongoDB supports dynamic queries using rich query syntax, allowing filtering, sorting, and projection of documents.
8. **File Storage:**
   * Supports storing large binary data and files using **GridFS**.

**Data Model Design in MongoDB**

**1. Document Model:**

* MongoDB stores data as documents in BSON format, which is similar to JSON.
* Example Document:
* {
* "\_id": ObjectId("507f191e810c19729de860ea"),
* "name": "John Doe",
* "email": "john@example.com",
* "address": {
* "street": "123 Main St",
* "city": "New York"
* },
* "hobbies": ["reading", "gaming", "cooking"]
* }

**2. Collections:**

* Documents are grouped into **collections**. A collection is analogous to a table in relational databases.
* Collections are **schema-less**, meaning documents in a collection do not need to have the same fields.

**3. Data Modeling Approaches:**

* **Embedded Documents:**
  + Stores related data within a single document.
  + Useful when related data is frequently read together.
  + Example:
  + {
  + "order\_id": 12345,
  + "customer": {
  + "name": "Jane Doe",
  + "email": "jane@example.com"
  + },
  + "items": [
  + {"product": "Laptop", "quantity": 1},
  + {"product": "Mouse", "quantity": 2}
  + ]
  + }
* **References (Normalization):**
  + Stores references to other documents using ObjectIds.
  + Useful when related data is accessed separately or to avoid data duplication.
  + Example:
  + {
  + "product\_id": ObjectId("5f1f4b7c2b5e6b2a2c3e9f1e"),
  + "name": "Laptop",
  + "price": 1200
  + }

In another collection:

{

"order\_id": 12345,

"product\_ids": [

ObjectId("5f1f4b7c2b5e6b2a2c3e9f1e"),

ObjectId("5f1f4b7c2b5e6b2a2c3e9f1f2")

]

}

**4. Best Practices:**

* **Embed data** when data is frequently accessed together.
* **Reference data** when data is accessed separately or frequently updated.
* **Avoid deep nesting** as it can complicate queries and indexing.
* **Design for the application’s query patterns** to optimize performance.

**Installation Options for MongoDB**

1. **MongoDB Community Edition:**
   * Free and open-source version.
   * Suitable for learning, development, and small to medium-scale applications.
2. **MongoDB Enterprise Edition:**
   * Commercial version with advanced security, monitoring, and support.
   * Suitable for enterprise applications requiring compliance and support.
3. **MongoDB Atlas:**
   * Cloud-based managed database service.
   * Eliminates the need for server maintenance and scaling.
4. **Docker Installation:**
   * MongoDB can be deployed using Docker containers for portability and consistency across environments.

**Packages and Considerations for Installing MongoDB on Linux**

**Supported Linux Distributions:**

* Ubuntu
* Debian
* CentOS
* Red Hat
* Amazon Linux

**Prerequisites:**

* **System Requirements:**
  + 64-bit architecture
  + Sufficient RAM and disk space
* **Dependency Packages:**
  + libcurl, openssl, libssl, and other system libraries.
* **Firewall Configuration:**
  + Ensure the required ports (default is **27017**) are open.

**Recommended Packages:**

* **mongodb-org:** Meta-package for MongoDB that includes:
  + mongodb-org-server: MongoDB server daemon.
  + mongodb-org-mongos: Sharding router.
  + mongodb-org-shell: Command-line client.
  + mongodb-org-tools: Database tools for backup, restore, and data manipulation.

**Installing MongoDB on Linux (Example: Ubuntu)**

1. **Import MongoDB Public Key:**
2. curl -fsSL https://pgp.mongodb.com/server-6.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg
3. **Create Source List File:**
4. echo "deb [signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
5. **Update Packages and Install MongoDB:**
6. sudo apt update
7. sudo apt install -y mongodb-org
8. **Start and Enable MongoDB Service:**
9. sudo systemctl start mongod
10. sudo systemctl enable mongod
11. **Verify Installation:**
12. mongod --version
13. sudo systemctl status mongod

**Connect to a MongoDB Database**

1. **Using MongoDB Shell:**
2. mongosh
   * Connect to a local MongoDB instance.
   * Default connection is on localhost:27017.
3. **Connecting to a Remote MongoDB Instance:**
4. mongosh "mongodb://username:password@hostname:port/dbname"
5. **Using MongoDB URI:**
6. mongosh "mongodb+srv://username:password@cluster0.mongodb.net/dbname"

**Using the MongoDB Shell (mongosh)**

1. **List Databases:**
2. show dbs
3. **Select Database:**
4. use myDatabase
5. **List Collections:**
6. show collections
7. **Insert Document:**
8. db.users.insertOne({ name: "Alice", age: 25 })
9. **Query Documents:**
10. db.users.find({ age: { $gt: 20 } })
11. **Update Document:**
12. db.users.updateOne({ name: "Alice" }, { $set: { age: 26 } })
13. **Delete Document:**
14. db.users.deleteOne({ name: "Alice" })
15. **Exit MongoDB Shell:**
16. exit

These are the foundational concepts and commands for working with MongoDB in a NoSQL environment.