# MODULE 04

## NODE.JS AND MONGODB

## ASSIGNMENT 02

### ◆ Introduction to MongoDB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. This document-oriented approach allows for dynamic schemes, making it ideal for applications that require scalability and adaptability to changing data structures. Unlike traditional relational databases, MongoDB can handle large volumes of diverse data efficiency.

### ◆ Key Features

- **Document Oriented**: Stores data in BSON (Binary JSON) format, allowing for complex data structures.
- **Scalability**: Easily scales horizontally by adding more servers.
- **Powerful Query Language**: Supports rich queries for data retrieval and manipulation.
- **Flexible Schema**: No fixed schema allows for easy updates and changes to the data model.

## 1 Data Modeling

Overview

Data modeling in MongoDB involves organizing data into collections and documents.

Key Components

- **Collection**: A group of related documents (similar to a table).
- **Document**: A single record in a collection (similar to a row), represented in BSON format.

Example Document:

```json
{
  "_id": "12345",
  "name": "John Doe",
  "age": 30,
  "address": { "city": "Anytown", "state": "CA" },
  "hobbies": ["reading", "traveling"]
}
```

- Best Practices
  - Use embedded documents for related data.
  - Use references for large datasets or loosely coupled data.
  - Design schema based on read and write patterns.

## 2. CRUD Operations

- **Create**

  Insert a new document:

  ```javascript
  db.collectionName.insertOne({ name: "Jane Doe", age: 25 });
  ```

- **Read**

  Retrieve documents:

  ```javascript
  db.collectionName.find({ age: { $gt: 20 } });
  ```

- **Update**

  Modify existing documents:

  ```javascript
  db.collectionName.updateOne({name: "Jane Doe"}, {$set: {age: 26}});
  ```

- **Delete**

  Remove documents:

  ```javascript
  db.collectionName.deleteOne({ name: "Jane Doe" });
  ```

## 3. Indexing

### Overview

Indexing improves query performance by allowing faster data retrieval.

### Creating an Index

To create an index on a field:

```javascript
db.collectionName.createIndex({ name: 1 }); // Ascending index
```

### Types of Indexes

- Single Field Index
- Compound Index
- Text Index
- Geospatial Index

# 4 Aggregation

Overview

Aggregation processes multiple documents and returns computed results

Aggregation Pipeline Example

[javascript]

```
db.collectionName.aggregate([
  { $match: { age: { $gte: 20 } } },
  { $group: { _id: "$hobby", totalCount: { $sum: 1 } } }
]);
```

# 5. Query Optimization

Overview

Optimization queries ensures efficient performance when accessing large datasets.

- ## Strategies for Optimization
  - Use indexes effectively.
  - Limit returned data using projections.
  - Analyze performance with the 'explain()' method.

- ## Common Use Cases
  1. Content Management Systems: Flexible schemas for diverse content types.
  2. Real-Time Analytics: High-speed data ingestion and querying.
  3. IoT Applications: Efficient handling of time-series data.
  4. Mobile Apps: Fast read/write operations for dynamic interactions

- ## Best Practices for Implementation and Management
  1. Schema Design:
     - Plan based on application requirements.
     - Regularly review and update schema.
  2. Data Backup:
     - Implement regular backups using 'mongodump' and 'mongorestore'.

3. Performance Monitoring
   - Use monitoring tools like MongoDB Atlas
4. Security Measures
   - Enable authentification and authorization
   - Use SSL/TLS for encrypted connections
5. Scalability Planning
   - Design for scalability; consider sharding for large datasets

## ◆ Conclusion

MongoDB provides a flexible, scalable solution for modern applications requiring dynamic data handling capabilities. Understanding its core concepts - data modeling, CRUD operations, indexing, aggregation, and query optimization - enables effective use of the database. This structured outline is designed to provide detailed notes on MongoDB concepts while being straightforward enough to facilitate easy understanding when handwritten. Each section is clearly defined with actionable examples and best practices, making it an effective study guide.