MongoDB Kya Hai?

MongoDB ek NoSQL database hai jo document-oriented storage use karta hai. Iska data JSON-like format me store hota hai, jise BSON (Binary JSON) kehte hain. Iska mukhya fayda hai ki ye flexible schema allow karta hai, jiska matlab hai ki aap alag-alag documents me alag-alag fields rakh sakte hain.

Key Features:

1. **Schema-less**: Data structure ko pehle se define karne ki zaroorat nahi.
2. **Scalability**: Horizontal scaling support karta hai, jisse aap easily database ko grow kar sakte hain.
3. **High Availability**: Replica sets ke zariye data redundancy aur availability ensure karta hai.

Mongoose Kya Hai?

Mongoose ek ODM (Object Data Modeling) library hai jo MongoDB ke sath Node.js me use hoti hai. Ye MongoDB ke sath interaction ko aasaan banata hai aur schema validation, type casting, aur other features provide karta hai.

Key Features:

1. **Schema Validation**: Data ko validate karne ki capability.
2. **Middleware Support**: Pre aur post hooks ke zariye complex workflows create karna.
3. **Built-in Methods**: CRUD operations ke liye convenient methods.

Setup

1. MongoDB Install Karna

MongoDB ko install karne ke liye MongoDB ki official website se installer download karein aur instructions follow karein.

2. Node.js Aur Mongoose Install Karna

Niche diye gaye commands ke zariye apne project me Node.js aur Mongoose install karein:

bash

Copy code

```
npm init -y
npm install mongoose
```

Mongoose Schema aur Model

Mongoose me schema define karne ke liye, pehle aapko ek schema create karna hota hai, jo aapke data structure ko define karta hai. Phir is schema se ek model create hota hai, jisse aap database me operations karte hain.

Schema Example

javascript
Copy code

```
const mongoose = require('mongoose');

// Database se connect karna
mongoose.connect('mongodb://localhost:27017/mydatabase', {
useNewUrlParser: true, useUnifiedTopology: true });

// Schema define karna
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: { type: Number, required: true },
    email: { type: String, required: true, unique: true },
});

// Model create karna
const User = mongoose.model('User', userSchema);
```

CRUD Operations

1. Create (Document Insert Karna)

javascript
Copy code

```
const createUser = async () => {
    const newUser = new User({
        name: 'Ali',
        age: 25,
        email: 'ali@example.com',
    });

    await newUser.save();
    console.log('User created:', newUser);
};

createUser();
```

2. Read (Documents Padhna)

javascript

Copy code

```
const readUsers = async () => {
    const users = await User.find();
    console.log('All Users:', users);
};

readUsers();
```

3. Update (Document Update Karna)

javascript

Copy code

```
const updateUser = async (userId) => {
    const updatedUser = await User.findByIdAndUpdate(
        userId,
        { age: 26 },
        { new: true } // New data return karne ke liye
    );
    console.log('Updated User:', updatedUser);
};

updateUser('60d5f4844b50d27f4c23b9ef'); // Replace with an actual user
ID
```

4. Delete (Document Delete Karna)

javascript

Copy code

```
const deleteUser = async (userId) => {
    await User.findByIdAndDelete(userId);
    console.log('User deleted:', userId);
};
```

```
deleteUser('60d5f4844b50d27f4c23b9ef'); // Replace with an actual user
ID
```

Conclusion

MongoDB aur Mongoose ka istemal karne se aapko flexible aur scalable data storage milta hai. Mongoose aapko Node.js ke sath MongoDB ko efficiently manage karne ki facility deta hai. Is guide ke zariye aap CRUD operations aur basic schema creation samajh gaye hain. Agar aapko aur advanced topics jaise middleware ya population ke baare me jaana ho, to zaroor batayein!

4o mini

You said:
crud ka mujha theory mil sakta hai code k saath

ChatGPT said:
ChatGPT

CRUD operations ka matlab hai Create, Read, Update, aur Delete. Ye operations kisi bhi database ke liye basic functionality hain. MongoDB aur Mongoose ke saath in operations ko kaise implement kiya jaata hai, uski theory aur code snippets niche diye gaye hain.

1. Create (Document Insert Karna)

Theory: Create operation ka matlab hai naya document database me insert karna. Is operation ke liye Mongoose ka `save()` method istemal hota hai.

Code Example:

javascript
Copy code

```
const mongoose = require('mongoose');

// Database se connect karna
mongoose.connect('mongodb://localhost:27017/mydatabase', {
useNewUrlParser: true, useUnifiedTopology: true });

// Schema define karna
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: { type: Number, required: true },
    email: { type: String, required: true, unique: true },
});

// Model create karna
const User = mongoose.model('User', userSchema);

// Naya user create karna
const createUser = async () => {
    const newUser = new User({
        name: 'Sara',
        age: 30,
        email: 'sara@example.com',
    });

    await newUser.save();
    console.log('User created:', newUser);
};

createUser();
```

2. Read (Documents Padhna)

Theory: Read operation ka matlab hai database se existing documents ko retrieve karna. Mongoose me `find()` aur `findById()` methods ka istemal hota hai.

Code Example:

javascript
Copy code

```
// Sabhi users ko padhna
const readUsers = async () => {
    const users = await User.find();
    console.log('All Users:', users);
};

readUsers();

// Kisi specific user ko padhna (ID se)
const readUserById = async (userId) => {
    const user = await User.findById(userId);
    console.log('User Found:', user);
};

readUserById('60d5f4844b50d27f4c23b9ef'); // Actual user ID se replace
karein
```

3. Update (Document Update Karna)

Theory: Update operation ka matlab hai existing document me changes karna. Mongoose me `findByIdAndUpdate()` method ka istemal hota hai.

Code Example:

javascript
Copy code

```
// User ko update karna
const updateUser = async (userId) => {
    const updatedUser = await User.findByIdAndUpdate(
        userId,
        { age: 32 }, // New data
        { new: true } // New document return karne ke liye
    );
    console.log('Updated User:', updatedUser);
};

updateUser('60d5f4844b50d27f4c23b9ef'); // Actual user ID se replace
karein
```

4. Delete (Document Delete Karna)

Theory: Delete operation ka matlab hai kisi document ko database se remove karna. Mongoose me `findByIdAndDelete()` method ka istemal hota hai.

Code Example:

javascript
Copy code

```
// User ko delete karna
const deleteUser = async (userId) => {
    await User.findByIdAndDelete(userId);
    console.log('User deleted:', userId);
};

deleteUser('60d5f4844b50d27f4c23b9ef'); // Actual user ID se replace
karein
```

Conclusion

CRUD operations ka istemal karte waqt, aap MongoDB aur Mongoose ki powerful functionalities ka fayda utha sakte hain. Ye operations aapko data management ke liye basic tools provide karte hain. Iske alawa, aap validation, middleware aur advanced querying features ka bhi istemal kar sakte hain jisse aapka application aur bhi robust ban sake. Agar aapko aur details chahiye ya kisi specific operation ke baare me samajhna hai, to batayein!

4o mini

You said:

mongodb me aur bhee bahut sare topic hota hai explain kar sakte ho

ChatGPT said:

ChatGPT

MongoDB ke andar kaafi saare advanced topics aur features hote hain jo aapko apne database ko effectively manage karne aur optimize karne me madad karte hain. Niche kuch important topics diye gaye hain jo MongoDB me aapko samajhne chahiye:

1. Data Modeling

Theory: Data modeling ka matlab hai aapke data ko kaise structure karna hai. MongoDB me aapko apne data ka schema design karna hota hai, jo data relationships aur storage ko define karta hai.

Example: Aap user aur post ka model create kar sakte hain, jisme user ki details aur uske posts ka reference hota hai.

2. Indexes

Theory: Indexes data retrieval ko fast karne ke liye istemal hote hain. Ye database me specific fields par pointers create karte hain, jisse queries ki speed improve hoti hai.

Example: Agar aap aksar user ke email se search karte hain, to aap email field par index create kar sakte hain:

javascript

Copy code

```
userSchema.index({ email: 1 });
```

3. Aggregation Framework

Theory: Aggregation framework aapko data ko transform aur analyze karne ki capability deta hai. Ye powerful queries ko perform karne ke liye pipelines ka istemal karta hai.

Example: Aap total posts count karne ke liye aggregation pipeline istemal kar sakte hain:

javascript
Copy code

```
const result = await Post.aggregate([
    { $group: { _id: '$userId', totalPosts: { $sum: 1 } } }
]);
```

4. Transactions

Theory: Transactions ek set of operations ko ek atomic unit me execute karne ki suvidha dete hain, yaani ya to saare operations successful hote hain ya phir none.

Example: Aap multiple collections me data insert karne ke liye transactions ka istemal kar sakte hain.

javascript
Copy code

```
const session = await mongoose.startSession();
session.startTransaction();
try {
    // Multiple operations
    await user.save({ session });
    await post.save({ session });
    await session.commitTransaction();
} catch (error) {
    await session.abortTransaction();
} finally {
    session.endSession();
}
```

5. Sharding

Theory: Sharding large datasets ko horizontally partition karne ki technique hai. Ye data ko multiple servers par distribute karta hai, jisse performance aur scalability improve hoti hai.

Example: Agar aapke paas bahut zyada data hai, to aap sharding enable kar sakte hain jisse data distribute hota hai.

6. Replication

Theory: Replication data redundancy aur high availability ke liye hota hai. Ye data ko multiple servers par copy karta hai, jisse agar ek server fail ho jaye, to dusra server available ho.

Example: MongoDB me replica set create karne se aapko automatic failover aur backup milta hai.

7. Data Validation

Theory: Data validation aapko ensure karta hai ki database me valid aur consistent data hi store ho. Aap Mongoose me schema validation use kar sakte hain.

Example: Aap schema me required fields aur types specify kar sakte hain:

javascript
Copy code

```
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: { type: Number, min: 0, required: true },
});
```

8. Change Streams

Theory: Change streams aapko real-time updates sunne ki suvidha dete hain jab database me koi changes hote hain. Iska istemal notifications ya updates ke liye hota hai.

Example: Aap kisi collection me changes sun sakte hain:

javascript
Copy code

```
const changeStream = User.watch();
changeStream.on('change', (change) => {
    console.log('Change detected:', change);
});
```

9. GridFS

Theory: GridFS MongoDB ka file storage system hai jo large files (jaise images, videos) ko store karne ke liye use hota hai. Ye files ko chunks me divide karke store karta hai.

Example: Aap ek file ko upload karne ke liye GridFS ka istemal kar sakte hain:

javascript
Copy code

```
const { GridFSBucket } = require('mongodb');
const bucket = new GridFSBucket(mongoose.connection.db);

const uploadStream = bucket.openUploadStream('myfile.txt');
uploadStream.end(fileBuffer);
```

Conclusion

MongoDB ke ye advanced topics aapko apne applications me better performance, scalability, aur data management features ka istemal karne me madad karte hain. Agar aap in topics me se kisi specific topic par aur details chahte hain, to zaroor batayein!

4o mini

You said:
aur bhee notes ho ga

ChatGPT said:
ChatGPT

Bilkul! MongoDB aur Mongoose ke aur bhi kuch important topics aur concepts hain, jinke baare me aapko pata hona chahiye. Niche kuch additional topics aur unki explanations diye gaye hain:

10. Data Import/Export

Theory: Data import/export ka matlab hai database me data ko import karna ya database se data ko export karna. Ye aapko data transfer karne me madad karta hai.

Commands:

Import:
bash

Copy code

```
mongoimport --db mydatabase --collection users --file users.json
```

-

Export:
bash
Copy code

```
mongoexport --db mydatabase --collection users --out users.json
```

-

11. Full-Text Search

Theory: MongoDB me full-text search features aapko text data me complex queries karne ki suvidha dete hain. Ye indexes ke through implement hota hai.

Example: Aap text index create kar sakte hain:

javascript
Copy code

```
userSchema.index({ name: 'text' });
```

Uske baad aap search kar sakte hain:

javascript
Copy code

```
const results = await User.find({ $text: { $search: 'Sara' } });
```

12. Schema Design Patterns

Theory: Schema design patterns aapko efficient data structures banane me madad karte hain. Common patterns me "Embedding" aur "Referencing" shamil hain.

Embedding: Related data ko ek document me store karna.
javascript
Copy code

```
const blogSchema = new mongoose.Schema({
    title: String,
    comments: [{ body: String, date: Date }]
});
```

-

Referencing: Alag documents me data ko link karna.
javascript
Copy code

```
const postSchema = new mongoose.Schema({
    title: String,
    author: { type: mongoose.Schema.Types.ObjectId, ref: 'User' }
});
```

-

13. Backup and Restore

Theory: Backup aur restore strategies aapko apne data ko loss se bachane me madad karte hain. Ye aapke data ki periodic copies create karte hain.

Commands:

Backup:
bash
Copy code

```
mongodump --db mydatabase --out /path/to/backup
```

-

Restore:
bash
Copy code

```
mongorestore --db mydatabase /path/to/backup/mydatabase
```

-

14. Monitoring and Performance Tuning

Theory: Monitoring tools aapko database performance track karne aur optimize karne me madad karte hain. Aap MongoDB ke built-in tools ya third-party tools jaise **MongoDB Compass** ka istemal kar sakte hain.

- **Performance Tuning**: Queries aur indexes ko optimize karke performance improve karna.

15. Security

Theory: Security measures aapko unauthorized access se bachane me madad karte hain. MongoDB me authentication, authorization, aur encryption shamil hote hain.

- **Authentication**: Users ko authenticate karna.
- **Authorization**: Users ko specific roles aur permissions dena.
- **Encryption**: Data ko encrypt karna, chahe wo at rest ho ya in transit.

16. Geospatial Queries

Theory: MongoDB geospatial data ko store aur query karne ki capability deta hai, jo location-based applications me useful hota hai.

Example: Aap geospatial index create kar sakte hain aur location-based queries perform kar sakte hain:

javascript
Copy code

```
const locationSchema = new mongoose.Schema({
    name: String,
    location: {
        type: { type: String, enum: ['Point'], required: true },
        coordinates: { type: [Number], required: true }
    }
});
locationSchema.index({ location: '2dsphere' });
```

17. Using MongoDB with Aggregation Framework

Theory: Aggregation framework ko complex queries ke liye istemal kiya jata hai, jisme multiple stages hoti hain jisse data ko process kiya ja sakta hai.

Example: Aap data ko group aur sum karne ke liye aggregation pipeline ka istemal kar sakte hain:

javascript
Copy code

```
const totalAge = await User.aggregate([
    { $group: { _id: null, totalAge: { $sum: '$age' } } }
]);
```

Conclusion

Ye additional topics aapko MongoDB ki capabilities ko aur achhe se samajhne me madad karte hain. MongoDB ek powerful NoSQL database hai jo aapke applications ko flexible aur scalable banane ki suvidha deta hai. Agar aap kisi specific topic par aur detail me jaana chahte hain ya aur kuch questions hain, to mujhe zaroor batayein!

4o mini

You said:

biggner se le kar advance topics cover karo

ChatGPT said:

ChatGPT

MongoDB ko samajhne ke liye, beginners se lekar advanced topics tak step-by-step approach ka istemal karna zaroori hai. Niche diye gaye notes me har level ke topics cover kiye gaye hain, jisse aap gradually seekh sakein.

Beginner Topics

1. **MongoDB Introduction**
    - **Kya hai**: MongoDB ek NoSQL database hai, jo document-oriented hai aur JSON-like format (BSON) me data store karta hai.
    - **Features**: Schema-less design, horizontal scalability, high availability.
2. **Installation**
    - MongoDB ko apne system par install karne ke liye [MongoDB ki official website]() se download karein.
3. **Basic Commands**
    - **Mongo Shell**: MongoDB shell ke zariye commands execute karna.

Common Commands:

bash

Copy code

```
use mydatabase      # Database select karna
db.createCollection('users')   # Collection create karna
```

    -
4. **CRUD Operations (Beginner Level)**
    - **Create**: Documents ko insert karna.
    - **Read**: Documents ko read karna.
    - **Update**: Documents ko update karna.
    - **Delete**: Documents ko delete karna.
5. **Schema Design Basics**
    - Data ko kaise structure karna hai: Embedded documents vs. References.

Simple schema create karna:

javascript

Copy code

```
const userSchema = new mongoose.Schema({
    name: String,
    age: Number
});
```

-

Intermediate Topics

6. **Mongoose Introduction**
    - Mongoose ek ODM (Object Data Modeling) library hai jo MongoDB ke sath use hoti hai.
    - **Basic Usage**: Mongoose se database connect karna aur schema aur models define karna.
7. **Data Validation**
    - Data ko validate karne ke liye schema me constraints set karna.

Example:
javascript
Copy code

```
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: { type: Number, min: 0 }
});
```

-

8. **Indexes**
    - Data retrieval ko speed up karne ke liye indexes create karna.

Example:
javascript
Copy code

```
userSchema.index({ email: 1 });
```

-

9. **Aggregation Framework**
    - Complex data analysis ke liye aggregation pipeline ka istemal.

Example:
javascript
Copy code

```
const result = await User.aggregate([
    { $group: { _id: '$age', count: { $sum: 1 } } }
]);
```

-

10. **Middleware**
    - Pre aur post hooks ka istemal karke operations me additional functionality add karna.

Example:
javascript
Copy code

```
userSchema.pre('save', function(next) {
    this.age = Math.abs(this.age); // Age ko positive banana
    next();
});
```

-

Advanced Topics

11. **Transactions**
    - Multiple operations ko atomic way me perform karna.

Example:
javascript
Copy code

```
const session = await mongoose.startSession();
session.startTransaction();
try {
    await user.save({ session });
    await post.save({ session });
    await session.commitTransaction();
} catch (error) {
    await session.abortTransaction();
} finally {
    session.endSession();
}
```

-

12. **Sharding**
    - Large datasets ko horizontally partition karna.
    - **Example**: Sharding enable karne ke liye configurations set karna.
13. **Replication**
    - Data redundancy aur availability ke liye replica sets ka istemal.

    - **Example**: Replica set configure karna.
14. **Geospatial Queries**
    - Location-based data ko query karne ki capability.

Example:
javascript
Copy code

```
const locationSchema = new mongoose.Schema({
    name: String,
    location: { type: { type: String, enum: ['Point'] }, coordinates:
[Number] }
});
locationSchema.index({ location: '2dsphere' });
```

    -
15. **Change Streams**
    - Real-time notifications ke liye database changes ko track karna.

Example:
javascript
Copy code

```
const changeStream = User.watch();
changeStream.on('change', (change) => {
    console.log('Change detected:', change);
});
```

    -
16. **Backup and Restore**
    - Data ko backup aur restore karne ki techniques.

Commands:
bash
Copy code

```
mongodump --db mydatabase --out /path/to/backup
mongorestore --db mydatabase /path/to/backup/mydatabase
```

    -
17. **Monitoring and Performance Tuning**
    - Database ki performance monitor karna aur optimize karna.
    - Tools jaise **MongoDB Compass** ka istemal.
18. **Security Best Practices**
    - Database ko secure karne ke liye authentication, authorization, aur encryption implement karna.

    - **Example**: User roles aur permissions set karna.

Conclusion

Ye structured approach aapko MongoDB ke basic se lekar advanced topics tak samajhne me madad karega. Har level par concepts ko samajhne se aapka knowledge enhance hoga aur aap effectively MongoDB ka istemal kar paenge. Agar aap kisi specific topic par aur detail chahte hain, to zaroor batayein!

4o mini

You said:

keya mujha iss ka interview se realate mil sakta hai biggner se le kar advance k leya ho

ChatGPT said:

ChatGPT

MongoDB ke interview ke liye kuch common questions diye gaye hain, jo aapko beginner se lekar advanced level tak ke concepts cover karne me madad karenge. Ye questions aapko prepare karne aur apni knowledge ko evaluate karne me help karte hain.

Beginner Level Questions

1. **What is MongoDB?**
    - Answer: MongoDB ek NoSQL document-oriented database hai jo JSON-like format (BSON) me data store karta hai. Iska schema-less design flexibility provide karta hai.
2. **Explain the difference between SQL and NoSQL databases.**
    - Answer: SQL databases relational hote hain aur fixed schemas ka istemal karte hain, jabki NoSQL databases non-relational hote hain aur schema-less ya dynamic schemas ka istemal karte hain.
3. **What is a document in MongoDB?**
    - Answer: Document ek basic unit hai MongoDB me, jo key-value pairs ke form me hota hai. Ye BSON format me store hota hai.
4. **How do you insert a document in MongoDB?**
    - Answer: Document insert karne ke liye `insertOne()` ya `insertMany()` methods ka istemal karte hain.

javascript
Copy code
```
db.users.insertOne({ name: 'Sara', age: 30 });
```

5.
6. **How do you query documents in MongoDB?**

    - Answer: Documents ko query karne ke liye `find()` method ka istemal karte hain.

javascript
Copy code

```javascript
db.users.find({ age: { $gt: 25 } });
```

7.

Intermediate Level Questions

6. **What is Mongoose, and why is it used?**
    - Answer: Mongoose ek ODM (Object Data Modeling) library hai jo MongoDB ke sath use hoti hai, jisse schema definitions, validation aur relationships manage karna aasan hota hai.
7. **Explain the concept of indexing in MongoDB.**
    - Answer: Indexing data retrieval ko speed up karne ke liye use hota hai. Indexes specific fields par pointers create karte hain, jisse queries ki performance improve hoti hai.
8. **What is the Aggregation Framework?**
    - Answer: Aggregation Framework MongoDB ka ek feature hai jo complex data analysis ke liye multiple stages me data ko process karta hai. Isme `$group`, `$match`, aur `$sort` stages shamil hain.
9. **How do you update documents in MongoDB?**
    - Answer: Documents ko update karne ke liye `updateOne()`, `updateMany()`, ya `findByIdAndUpdate()` methods ka istemal karte hain.

javascript
Copy code

```javascript
db.users.updateOne({ name: 'Sara' }, { $set: { age: 31 } });
```

10.
11. **What are the differences between embedded documents and referenced documents?**
    - Answer: Embedded documents related data ko ek document me store karte hain, jabki referenced documents alag documents ko link karte hain. Embedding data denormalized hota hai, jabki referencing normalization ko promote karta hai.

Advanced Level Questions

11. **What are transactions in MongoDB?**
    - Answer: Transactions multiple operations ko atomic way me perform karne ki suvidha dete hain, jisse ya to saare operations successful hote hain ya phir none. Ye MongoDB 4.0 se available hain.

12. **How does sharding work in MongoDB?**
    - Answer: Sharding large datasets ko horizontally partition karne ka process hai. Ye data ko multiple servers par distribute karta hai, jisse performance aur scalability improve hoti hai.
13. **What is replication in MongoDB?**
    - Answer: Replication data redundancy aur high availability ke liye use hota hai. Ye data ko multiple servers par copy karta hai, jisse agar ek server fail ho jaye to dusra server available hota hai.
14. **Explain the concept of Change Streams.**
    - Answer: Change Streams real-time notifications ke liye MongoDB me changes ko track karne ki suvidha dete hain. Iska istemal applications me real-time updates dene ke liye hota hai.
15. **How do you secure a MongoDB database?**
    - Answer: MongoDB me security ke liye authentication, authorization, aur encryption implement karna hota hai. Aap users ko roles assign kar sakte hain aur data ko encrypt kar sakte hain.

Conclusion

Ye questions aapko MongoDB ke concepts ko samajhne me madad karte hain aur interview preparation ke liye useful hote hain. In questions par practice karne se aap apne knowledge ko enhance kar sakte hain. Agar aapko aur specific questions ya topics chahiye, to mujhe zaroor batayein!

4o mini

You said:

isse theory aur code dono k saath explain karo

ChatGPT said:

ChatGPT

Bilkul! MongoDB ke interview questions ko theory aur code dono ke saath explain karte hain, taaki aap in concepts ko achhe se samajh sakein. Niche kuch common questions diye gaye hain, unke answers ke sath code examples bhi shamil hain.

Beginner Level Questions

1. **What is MongoDB?**
    - **Theory**: MongoDB ek NoSQL document-oriented database hai jo JSON-like format (BSON) me data store karta hai. Iska schema-less design flexibility provide karta hai, yani aap easily data ko add aur modify kar sakte hain bina kisi strict schema ke.

Code Example: MongoDB ko start karne ke liye, aap `mongod` command run karte hain aur shell ko access karte hain.

bash

Copy code

```
mongod
```

-

2. **Explain the difference between SQL and NoSQL databases.**
    - **Theory**: SQL databases relational hote hain, fixed schemas aur tables me data store karte hain, jabki NoSQL databases non-relational hote hain aur schema-less ya dynamic schemas ka istemal karte hain. NoSQL databases data ko document, key-value, columnar, ya graph formats me store karte hain.
    - **Code Example**: SQL aur NoSQL ka comparison:

SQL:

sql

Copy code

```
SELECT * FROM users WHERE age > 25;
```

■

MongoDB:

javascript

Copy code

```
db.users.find({ age: { $gt: 25 } });
```

■

3. **What is a document in MongoDB?**
    - **Theory**: Document ek basic unit hai MongoDB me, jo key-value pairs ke form me hota hai. Ye BSON format me store hota hai, jo JSON ka binary representation hai.

Code Example: Document insert karna:

javascript

Copy code

```
db.users.insertOne({
    name: 'Sara',
    age: 30,
    email: 'sara@example.com'
});
```

-

4. **How do you insert a document in MongoDB?**

   - **Theory**: MongoDB me documents ko insert karne ke liye `insertOne()` ya `insertMany()` methods ka istemal karte hain.

Code Example:
javascript
Copy code
```
// Single document insertion
db.users.insertOne({ name: 'Sara', age: 30 });

// Multiple documents insertion
db.users.insertMany([
    { name: 'Ali', age: 25 },
    { name: 'Zara', age: 28 }
]);
```

   -

5. **How do you query documents in MongoDB?**
   - **Theory**: MongoDB me documents ko query karne ke liye `find()` method ka istemal karte hain. Is method me aap query criteria define karte hain.

Code Example:
javascript
Copy code
```
// Fetch all users
db.users.find();

// Fetch users older than 25
db.users.find({ age: { $gt: 25 } });
```

   -

Intermediate Level Questions

6. **What is Mongoose, and why is it used?**
   - **Theory**: Mongoose ek ODM (Object Data Modeling) library hai jo MongoDB ke sath use hoti hai. Ye schema definitions, validation, aur relationships manage karne ke liye madadgar hoti hai.

Code Example:
javascript
Copy code
```
const mongoose = require('mongoose');
```

```
// Connect to MongoDB
mongoose.connect('mongodb://localhost/mydatabase', { useNewUrlParser:
true, useUnifiedTopology: true });

// Define a schema
const userSchema = new mongoose.Schema({
    name: String,
    age: Number,
    email: String
});

// Create a model
const User = mongoose.model('User', userSchema);
```

    -

7. **Explain the concept of indexing in MongoDB.**
    - **Theory**: Indexing data retrieval ko speed up karne ke liye use hota hai. Indexes specific fields par pointers create karte hain, jisse queries ki performance improve hoti hai.

Code Example:
javascript
Copy code

```
// Create an index on the email field
userSchema.index({ email: 1 });

// Ensure index is created
User.createIndexes();
```

    -

8. **What is the Aggregation Framework?**
    - **Theory**: Aggregation Framework MongoDB ka ek feature hai jo complex data analysis ke liye multiple stages me data ko process karta hai. Isme `$group`, `$match`, aur `$sort` stages shamil hain.

Code Example:
javascript
Copy code

```
const result = await User.aggregate([
    { $match: { age: { $gt: 25 } } }, // Filter users older than 25
    { $group: { _id: '$age', count: { $sum: 1 } } } // Group by age
and count
]);
```

  -

9. **How do you update documents in MongoDB?**
   - **Theory**: Documents ko update karne ke liye `updateOne()`, `updateMany()`, ya `findByIdAndUpdate()` methods ka istemal karte hain.

Code Example:
javascript
Copy code

```
// Update a single document
db.users.updateOne({ name: 'Sara' }, { $set: { age: 31 } });

// Update multiple documents
db.users.updateMany({ age: { $lt: 30 } }, { $set: { status: 'young' }
});
```

  -

10. **What are the differences between embedded documents and referenced documents?**
    - **Theory**: Embedded documents related data ko ek document me store karte hain, jabki referenced documents alag documents ko link karte hain. Embedding data denormalized hota hai, jabki referencing normalization ko promote karta hai.

Code Example:
javascript
Copy code

```
// Embedded documents example
const blogSchema = new mongoose.Schema({
    title: String,
    comments: [{ body: String, date: Date }]
});

// Referenced documents example
const postSchema = new mongoose.Schema({
    title: String,
    author: { type: mongoose.Schema.Types.ObjectId, ref: 'User' }
});
```

  -

Advanced Level Questions

11. **What are transactions in MongoDB?**
    - ○ **Theory**: Transactions multiple operations ko atomic way me perform karne ki suvidha dete hain, jisse ya to saare operations successful hote hain ya phir none. Ye MongoDB 4.0 se available hain.

Code Example:
javascript
Copy code

```
const session = await mongoose.startSession();
session.startTransaction();
try {
    const user = new User({ name: 'Sara', age: 30 });
    await user.save({ session });

    const post = new Post({ title: 'My First Post', author: user._id
});
    await post.save({ session });

    await session.commitTransaction();
} catch (error) {
    await session.abortTransaction();
} finally {
    session.endSession();
}
```

○

12. **How does sharding work in MongoDB?**
    - ○ **Theory**: Sharding large datasets ko horizontally partition karne ka process hai. Ye data ko multiple servers par distribute karta hai, jisse performance aur scalability improve hoti hai.

Code Example: Sharding enable karne ke liye configurations set karte hain, jo command line se kiye jaate hain. Example:
bash
Copy code

```
sh.addShard("shard1/localhost:27017");
sh.enableSharding("mydatabase");
sh.shardCollection("mydatabase.users", { "userId": 1 });
```

○

13. **What is replication in MongoDB?**
    - **Theory**: Replication data redundancy aur high availability ke liye use hota hai. Ye data ko multiple servers par copy karta hai, jisse agar ek server fail ho jaye to dusra server available hota hai.

Code Example: Replica set configure karne ke liye MongoDB shell me commands run ki jaati hain.
bash
Copy code

```
rs.initiate();
rs.add("localhost:27018");
rs.add("localhost:27019");
```

○

14. **Explain the concept of Change Streams.**
    - **Theory**: Change Streams real-time notifications ke liye MongoDB me changes ko track karne ki suvidha dete hain. Iska istemal applications me real-time updates dene ke liye hota hai.

Code Example:
javascript
Copy code

```
const changeStream = User.watch();
changeStream.on('change', (change) => {
    console.log('Change detected:', change);
});
```

○

15. **How do you secure a MongoDB database?**
    - **Theory**: MongoDB me security ke liye authentication, authorization, aur encryption implement karna hota hai. Aap users ko roles assign kar sakte hain aur data ko encrypt kar sakte hain.

Code Example: User authentication aur roles setup karne ka example:
javascript
Copy code

```
db.createUser({
    user: "appUser",
    pwd: "securePassword",
    roles: [{ role: "readWrite", db: "mydatabase" }]
});
```

○

Conclusion

Ye questions aur unke answers aapko MongoDB ke concepts ko achhe se samajhne me madad karte hain. Theory ke sath code examples se aap easily practice kar sakte hain. Agar aapko kisi specific topic par aur detail chahiye, to mujhe zaroor batayein!

4o mini

You said:

aur questios hai keya

ChatGPT said:

ChatGPT

Bilkul! MongoDB ke interview ke liye aur bhi kaafi saare questions hain jo aapki knowledge aur understanding ko test karte hain. Niche kuch aur common questions diye gaye hain, jo beginner se lekar advanced level tak ke concepts cover karte hain.

Additional Beginner Level Questions

1. **What is BSON?**
    - Answer: BSON (Binary JSON) MongoDB ka data format hai jo JSON se derived hai. Ye data ko binary form me store karta hai aur isme additional data types jaise date aur binary data shamil hote hain.
2. **How do you delete a document in MongoDB?**

Answer: Documents ko delete karne ke liye `deleteOne()` ya `deleteMany()` methods ka istemal karte hain.

javascript

Copy code

```
db.users.deleteOne({ name: 'Sara' });
```

   -

3. **What are the different data types supported in MongoDB?**
    - Answer: MongoDB me supported data types me String, Number, Boolean, Date, ObjectId, Array, aur Null shamil hain.

Additional Intermediate Level Questions

4. **Explain the concept of a replica set.**
    - Answer: Replica set MongoDB ka ek feature hai jo data redundancy aur high availability ke liye use hota hai. Ye ek primary node aur multiple secondary nodes ka group hota hai. Primary node write operations handle karta hai, jabki secondary nodes read operations ke liye data replicate karte hain.
5. **What is the purpose of the `$lookup` operator?**

Answer: `$lookup` operator aggregation framework ka part hai jo two collections ke beech join operation perform karta hai.
javascript
Copy code

```
db.orders.aggregate([
    {
        $lookup: {
            from: 'products',
            localField: 'product_id',
            foreignField: '_id',
            as: 'product_details'
        }
    }
]);
```

    ○

6. **What are the different ways to index a collection in MongoDB?**
    - Answer: MongoDB me indexes create karne ke liye aap single field indexes, compound indexes (multiple fields), unique indexes, geospatial indexes, aur text indexes ka istemal kar sakte hain.

Additional Advanced Level Questions

7. **What is the purpose of the `$group` operator in aggregation?**

Answer: `$group` operator aggregation pipeline me documents ko group karne ke liye use hota hai. Ye grouped data par aggregate functions jaise `sum`, `avg`, `max`, `min` apply kar sakta hai.
javascript
Copy code

```
db.sales.aggregate([
    {
        $group: {
            _id: '$item',
            totalSales: { $sum: '$amount' }
        }
    }
]);
```

    ○

8. **What is an aggregation pipeline?**

    - Answer: Aggregation pipeline MongoDB ka ek powerful feature hai jo data ko multiple stages me process karta hai. Har stage output ko next stage me input provide karta hai, jisse complex data transformations possible hote hain.

9. **How do you implement a many-to-many relationship in MongoDB?**

Answer: Many-to-many relationships ko reference documents ya embedded documents ka istemal karke implement kiya ja sakta hai. References ke liye aap ObjectId ka istemal karte hain.
javascript
Copy code

```javascript
// Example for referencing
const studentSchema = new mongoose.Schema({
    name: String,
    courses: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Course' }]
});
```

    -

10. **How do you handle errors in MongoDB?**

Answer: MongoDB operations me error handling ke liye try-catch blocks ka istemal kiya jata hai. Mongoose me promise-based error handling bhi hota hai.
javascript
Copy code

```javascript
try {
    await user.save();
} catch (error) {
    console.error('Error saving user:', error);
}
```

    -

Conclusion

Ye additional questions aapki MongoDB ki understanding ko aur gehra karne me madad karenge. Har question ka theory aur code example se aapki preparation aur bhi behtar ho jayegi. Agar aapko kisi specific area me aur questions ya topics chahiye, to mujhe zaroor batayein!

4o mini

You said:
aur bhee hai kuch

ChatGPT said:

ChatGPT

Bilkul! MongoDB ke interview preparation ke liye aur bhi questions hain jo aapke knowledge ko test karte hain. Niche kuch aur advanced aur specialized topics ke questions diye gaye hain:

Additional Beginner Level Questions

1. **What is a database in MongoDB?**
    - ○ **Answer**: MongoDB me, database ek container hota hai jisme collections (tables in SQL) hote hain. Har database me multiple collections ho sakti hain.
2. **What are collections in MongoDB?**
    - ○ **Answer**: Collection ek group hai documents ka, jo MongoDB me tables ke barabar hota hai. Collections schema-less hote hain, yani aap different types ke documents ek collection me rakh sakte hain.
3. **What are the advantages of using MongoDB?**
    - ○ **Answer**: MongoDB ki advantages me flexibility, scalability, high performance, horizontal scalability (sharding), aur rich query capabilities shamil hain.

Additional Intermediate Level Questions

4. **What is the difference between `find()` and `findOne()`?**

Answer: `find()` method multiple documents ko return karta hai jo query criteria se match karte hain, jabki `findOne()` method sirf pehla document return karta hai jo criteria se match karta hai.

javascript
Copy code

```
const allUsers = db.users.find(); // returns all matching documents
const singleUser = db.users.findOne({ name: 'Sara' }); // returns the
first matching document
```

○

5. **Explain how to use the `countDocuments()` method.**

Answer: `countDocuments()` method collection me matching documents ki total count return karta hai, jo criteria se match karte hain.

javascript
Copy code

```
const count = await db.users.countDocuments({ age: { $gt: 25 } });
console.log(count);
```

○

6. **What is the `$set` operator?**

Answer: `$set` operator ko update operations me use kiya jata hai taaki specific fields ko modify kiya ja sake. Ye existing fields ko update karta hai ya naye fields add karta hai.
javascript
Copy code

```
db.users.updateOne({ name: 'Sara' }, { $set: { age: 31 } });
```

- ○

Additional Advanced Level Questions

7. **What is data denormalization?**
   - **Answer**: Data denormalization ek technique hai jisme related data ko ek hi document me store kiya jata hai taaki read operations fast ho sakein. Iska istemal MongoDB me aksar hota hai.
8. **How does MongoDB handle data consistency?**
   - **Answer**: MongoDB me data consistency ko achieve karne ke liye read and write concerns ka istemal kiya jata hai. Ye specify karte hain ki data kahan se read karna hai aur data ko write karne me kitni durability chahiye.
9. **What are the differences between `update()` and `save()`?**

Answer: `update()` method existing documents ko update karta hai bina document ko completely replace kiye. `save()` method document ko completely save karta hai aur agar document already exist karta hai to overwrite kar deta hai.
javascript
Copy code

```
// update
db.users.updateOne({ name: 'Sara' }, { $set: { age: 31 } });

// save
const user = new User({ name: 'Sara', age: 30 });
await user.save(); // saves new or updates existing
```

- ○

10. **Explain the `$project` stage in aggregation.**

Answer: `$project` stage aggregation pipeline me specific fields ko include ya exclude karne ke liye use hota hai. Ye output document ka structure customize karta hai.
javascript
Copy code

```
db.users.aggregate([
    {
        $project: {
            name: 1, // include name
            age: 1,  // include age
            _id: 0   // exclude _id
        }
    }
]);
```

-

Additional Specialized Topics Questions

11. **What is GridFS?**
    - **Answer**: GridFS MongoDB ka ek specification hai jo large files ko store aur manage karne ke liye use hota hai. Ye files ko smaller chunks me break karta hai aur unhe documents ke roop me store karta hai.
12. **How do you handle schema migrations in MongoDB?**
    - **Answer**: Schema migrations MongoDB me aise handle kiye jaate hain ki aap application logic me changes karte hain, aur Mongoose ka `update()` ya `save()` methods ka istemal karke existing documents ko migrate karte hain.
13. **What are the implications of using large documents in MongoDB?**
    - **Answer**: MongoDB me documents ka size 16 MB tak hota hai. Agar aap large documents ka istemal karte hain, to read/write operations slow ho sakte hain, aur network overhead badh sakta hai.
14. **How do you ensure data integrity in MongoDB?**
    - **Answer**: Data integrity MongoDB me ensure karne ke liye aap transactions, validation rules, aur proper indexing ka istemal karte hain. Ye techniques data consistency aur correctness maintain karte hain.
15. **Explain the difference between a primary and secondary node in a replica set.**
    - **Answer**: Primary node write operations ko handle karta hai aur data ko store karta hai, jabki secondary nodes primary node se data ko replicate karte hain aur read operations ke liye use ho sakte hain.

Conclusion

Ye additional questions MongoDB ke concepts ko aur gehra karne me madad karenge. In questions ko samajhne aur un par practice karne se aap interview me behtar perform kar sakte hain. Agar aapko kisi specific area me aur questions ya topics chahiye, to mujhe zaroor batayein!

4o mini

You said:
aur bhee kuch topic hai

ChatGPT said:

ChatGPT

Bilkul! MongoDB ke baare me aur bhi kai topics hain jo aapko interview preparation ke liye madadgar sabit ho sakte hain. Niche kuch aur specialized topics aur unse jude questions diye gaye hain:

Additional Topics and Questions

1. **MongoDB Performance Tuning**
   - **What are some common performance issues in MongoDB?**
     - Answer: Common performance issues me inefficient queries, unoptimized indexes, large document sizes, aur high disk I/O operations shamil hain.
   - **How can you improve query performance in MongoDB?**
     - Answer: Query performance improve karne ke liye aap proper indexing, aggregation framework ka istemal, aur query optimization techniques ka sahara le sakte hain.
2. **Backup and Restore**
   - **How do you backup a MongoDB database?**

Answer: MongoDB database ka backup lene ke liye `mongodump` command ka istemal hota hai.
bash
Copy code

```
mongodump --db mydatabase --out /path/to/backup
```

-
   - **How do you restore a MongoDB database?**

Answer: MongoDB database ko restore karne ke liye `mongorestore` command ka istemal hota hai.
bash
Copy code

```
mongorestore --db mydatabase /path/to/backup/mydatabase
```

-

3. **Data Modeling**
   - **What are the best practices for data modeling in MongoDB?**
     - Answer: Best practices me schema design ke waqt denormalization, embedding versus referencing ka sahara lena, aur proper indexing shamil hain.
   - **How do you design a schema for a blogging application in MongoDB?**
     - Answer: Aap ek `Users` collection aur ek `Posts` collection bana sakte hain. Posts collection me ek reference field ho sakta hai jo user ki ID point karta ho.

4. **Security and Authentication**
    - **How do you implement role-based access control in MongoDB?**
        - Answer: MongoDB me role-based access control implement karne ke liye aap roles create karte hain aur unhe specific users ya user groups ko assign karte hain.
    - **What is TLS/SSL in MongoDB?**
        - Answer: TLS/SSL MongoDB me data in transit ko encrypt karne ke liye use hota hai, jisse data ki security badhti hai.
5. **Monitoring and Management**
    - **What tools can be used to monitor MongoDB performance?**
        - Answer: MongoDB ke performance ko monitor karne ke liye `mongostat`, `mongotop`, aur MongoDB Atlas jaise tools ka istemal kiya ja sakta hai.
    - **What is the role of the MongoDB profiler?**
        - Answer: MongoDB profiler queries ki performance metrics collect karta hai aur slow queries ko identify karne me madad karta hai.
6. **Schema Validation**
    - **What is schema validation in MongoDB?**
        - Answer: Schema validation MongoDB me documents ki structure ko enforce karne ka process hai. Ye ensure karta hai ki documents defined schema se match karte hain.
    - **How do you set up validation rules in Mongoose?**
        - Answer: Mongoose me validation rules schema definitions me define kiye jaate hain.

javascript
Copy code

```javascript
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: { type: Number, min: 0 }
});
```

    -
7. **Replication and High Availability**
    - **How does MongoDB handle failover in a replica set?**
        - Answer: Agar primary node fail ho jata hai, to secondary node automatic primary node ban jata hai aur is process ko election kehte hain.
    - **What are the configurations needed for setting up a replica set?**
        - Answer: Replica set configure karne ke liye aapko `rs.initiate()` command se replica set ko start karna hota hai aur secondary nodes ko add karna hota hai.
8. **Sharding**
    - **What is the difference between horizontal and vertical scaling?**

        - Answer: Horizontal scaling me aap multiple servers (nodes) ko add karte hain, jabki vertical scaling me aap ek server ki capacity ko badhate hain.
    - **How do you shard a collection in MongoDB?**
        - Answer: Collection ko shard karne ke liye `shardCollection()` command ka istemal karte hain.

javascript
Copy code

```
sh.shardCollection("mydatabase.users", { "userId": 1 });
```

    -

9. **Data Migration**
    - **How do you migrate data from SQL to MongoDB?**
        - Answer: Data migration SQL se MongoDB me karne ke liye aap tools jaise `mongoimport`, ETL tools, ya custom scripts ka istemal kar sakte hain.
    - **What challenges can arise during data migration?**
        - Answer: Challenges me data type mismatch, schema differences, aur data consistency issues shamil ho sakte hain.
10. **Using Aggregation Framework**
    - **What is the `$unwind` operator in aggregation?**
        - Answer: `$unwind` operator array ko flatten karta hai, yani har element ke liye alag document create karta hai.

javascript
Copy code

```
db.users.aggregate([
    { $unwind: "$hobbies" }
]);
```

    -
    - **What are pipelines in aggregation?**
        - Answer: Aggregation pipelines multiple stages ka sequence hai jisse data ko process kiya jata hai. Har stage data ko transform ya filter karne ke liye use hoti hai.

Conclusion

In additional topics aur questions ke through, aap MongoDB ki deeper understanding hasil kar sakte hain. Ye questions aapko interview me aane wale diverse topics ke liye prepare karne me madad karenge. Agar aapko kisi specific topic par aur information chahiye, to mujhe bataiye!

4o mini

You said:

aur kuch sab se jyada puche jane wale questions hai keya

ChatGPT said:

ChatGPT

Bilkul! MongoDB ke interview me sabse jyada puche jane wale kuch questions niche diye gaye hain, jo aapki knowledge ko test karte hain. Ye questions aapko fundamental concepts se lekar advanced topics tak cover karte hain:

Frequently Asked Questions

1. **What is MongoDB?**
   - **Answer**: MongoDB ek NoSQL database hai jo documents ko JSON-like format me store karta hai, jise BSON (Binary JSON) kehte hain. Ye schema-less hai, yani aap alag-alag structures ke documents ek hi collection me rakh sakte hain.
2. **What are the differences between SQL and NoSQL databases?**
   - **Answer**: SQL databases structured data ko tabular format me store karte hain aur fixed schemas rakhte hain, jabki NoSQL databases unstructured ya semi-structured data ko store karte hain aur schema-less hote hain.
3. **Explain the difference between a document and a collection in MongoDB.**
   - **Answer**: Document ek individual record hai jo key-value pairs me hota hai, jabki collection ek group hai documents ka. Collection SQL me tables ke barabar hota hai.
4. **What are indexes in MongoDB? Why are they used?**
   - **Answer**: Indexes data retrieval ko faster banate hain. Ye specific fields par search operations ko optimize karte hain, jisse queries ki performance improve hoti hai.
5. **What is the purpose of `ObjectId` in MongoDB?**
   - **Answer**: `ObjectId` MongoDB ka default unique identifier hai jo har document ke liye generate hota hai. Ye 12-byte hexadecimal value hoti hai jo document ko uniquely identify karti hai.
6. **How do you create a new database in MongoDB?**

Answer: Naya database create karne ke liye aap `use` command ka istemal karte hain. Agar database exist nahi karta to MongoDB use automatically create kar deta hai.
javascript
Copy code

```
use myNewDatabase;
```

   -

7. **What is the aggregation framework in MongoDB?**

- **Answer**: Aggregation framework MongoDB me data ko process aur analyze karne ke liye use hota hai. Isme aap multiple stages me data ko transform kar sakte hain, jaise filtering, grouping, aur sorting.

8. **Explain the role of primary and secondary nodes in a replica set.**
   - **Answer**: Primary node write operations handle karta hai aur data ko store karta hai, jabki secondary nodes primary se data ko replicate karte hain aur read operations me help karte hain.
9. **What are the different types of joins in MongoDB?**
   - **Answer**: MongoDB me joins ke liye `$lookup` operator ka istemal hota hai, jo aggregation framework ka part hai. Ye do collections ke beech join operation perform karta hai.
10. **How do you perform data validation in MongoDB?**
    - **Answer**: Data validation MongoDB me schema validation rules ke through ki ja sakti hai, jo documents ki structure ko enforce karte hain. Mongoose me validation rules schema definitions me define kiye jaate hain.

Additional Common Questions

11. **What is sharding in MongoDB?**
    - **Answer**: Sharding ek technique hai jo large datasets ko multiple servers (shards) me distribute karta hai, jisse data scalability aur performance improve hoti hai.
12. **How do you handle transactions in MongoDB?**

Answer: MongoDB 4.0 se transactions ko support karta hai. Aap multi-document transactions ka istemal karke multiple operations ko atomic manner me execute kar sakte hain.
javascript
Copy code

```
const session = client.startSession();
session.startTransaction();
// Perform operations here
await session.commitTransaction();
session.endSession();
```

-

13. **What is the difference between `update` and `replace`?**

Answer: `update` method existing fields ko modify karta hai, jabki `replace` method poore document ko replace karta hai.
javascript
Copy code

```
// Update
db.collection.updateOne({ _id: id }, { $set: { field: value } });
```

```
// Replace
db.collection.replaceOne({ _id: id }, { newField: value });
```

    -

14. **What are the benefits of using MongoDB?**
    - **Answer**: Benefits me flexibility, horizontal scalability, rich query capabilities, aur high performance shamil hain. MongoDB schema-less design data modeling me zyada flexibility deta hai.
15. **How do you implement security in MongoDB?**
    - **Answer**: Security MongoDB me authentication, authorization, TLS/SSL encryption, aur auditing ke through implement ki jati hai. Role-based access control (RBAC) se users ko specific permissions assign kiye ja sakte hain.

Conclusion

Ye questions MongoDB ke fundamentals aur core concepts ko cover karte hain. Inka accha knowledge aapki interview preparation me madad karega. Agar aapko kisi specific question ya topic par aur detail chahiye, to mujhe zaroor batayein!

4o mini