Crack Your MongoDB Interview – Top 50 Questions with Answers

mongoDB

Basics of MongoDB

1. What is MongoDB?
MongoDB is a NoSQL, document-oriented database. It stores data in flexible, JSON-like documents (BSON), which makes it scalable and easy to work with.

2. What is a NoSQL database?
NoSQL databases don't use tables and rows like traditional relational databases. Instead, they use flexible formats like key-value pairs, documents, graphs, or wide-columns.

3. What is BSON?
BSON (Binary JSON) is a binary representation of JSON documents used to store data in MongoDB. It supports more data types than JSON.

4. What are collections and documents in MongoDB?

- **Document:** A single record in MongoDB, stored as BSON
- **Collection:** A group of documents, similar to a table in relational DBs

5. What are the advantages of MongoDB?

- Schema-less (flexible schema)
- Horizontal scaling
- High performance
- Built-in replication and sharding
- JSON-style documents

CRUD Operations

6. How do you insert data in MongoDB?

Using `insertOne()` or `insertMany()`:

```
db.users.insertOne({ name: "John", age: 30 });
```

7. How do you read/fetch data?

Use `find()`:

```
db.users.find({ name: "John" });
```

8. How do you update data in MongoDB?

Use `updateOne()` or `updateMany()`:

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

9. How do you delete data?

Use `deleteOne()` or `deleteMany()`:

```
db.users.deleteOne({ name: "John" });
```

10. What does the $set operator do?
It updates specific fields of a document without replacing the entire document.

Querying & Indexing

11. What are MongoDB query operators?
Operators like $eq, $ne, $gt, $lt, $in, $nin, $and, $or help filter documents.

12. How does indexing work in MongoDB?
Indexes improve query performance by allowing MongoDB to quickly locate data.
Example: db.users.createIndex({ name: 1 })

13. What is the default index in MongoDB?
Every document has a unique _id field which is indexed by default.

14. What is a compound index?
An index on multiple fields:

```
db.users.createIndex({ name: 1, age: -1 });
```

15. How do you perform a text search?
Use a text index:

```
db.posts.createIndex({ content: "text" });
db.posts.find({ $text: { $search: "mongodb" } });
```

Schema Design & Relationships

16. Is MongoDB schema-less?
Yes, but you can enforce structure using **schema validation** (e.g., with Mongoose or JSON Schema).

17. How do you model relationships in MongoDB?

- **Embedding** (storing documents inside documents)
- **Referencing** (storing references using ObjectId)

18. When to embed vs reference in MongoDB?

- Embed when data is tightly coupled and queried together
- Reference when data is accessed independently or grows large

19. What is ObjectId in MongoDB?
A 12-byte unique identifier for documents generated by MongoDB.

20. Can you enforce foreign key constraints?
No. MongoDB doesn't support traditional foreign keys — it's up to the application logic.

Aggregation Framework

21. What is aggregation in MongoDB?
Aggregation processes data records and returns computed results. It's like SQL's GROUP BY.

22. What is the aggregation pipeline?
A series of stages ($match, $group, $sort, $project, etc.) that process data.

23. Example of a simple aggregation:

```
db.orders.aggregate([
  { $match: { status: "active" } },
  { $group: { _id: "$userId", total: { $sum: "$amount" } } }
]);
```

24. What is the $project stage used for?
It reshapes each document, including or excluding fields.

25. What does $lookup do?
Performs a **left outer join** between collections.

Performance & Scaling

26. What is sharding in MongoDB?
Sharding splits data across multiple servers (shards) to handle large datasets and high traffic.

27. What is replication in MongoDB?
Replication copies data across multiple servers (replica set) to ensure high availability.

28. What is a replica set?
A group of MongoDB servers that maintain the same data. One is primary, others are secondary.

29. What is a primary and secondary in replica set?
Primary handles all write operations. Secondaries replicate data from the primary.

30. What is a write concern in MongoDB?
Specifies the level of acknowledgment requested from MongoDB for write operations.

Indexes & Performance Tuning

31. How do indexes affect performance?
Indexes speed up reads but slow down writes and use more memory.

32. What is a hashed index?
Used for sharding by hashing the value of the indexed field.

33. What are geospatial indexes?
Indexes used for location-based queries using 2d or 2dsphere.

34. How do you analyze query performance?
Use .explain():

```
db.users.find({ age: 30 }).explain("executionStats");
```

35. How to avoid performance issues in MongoDB?

- Use indexes wisely
- Avoid large documents
- Optimize aggregation pipelines
- Use projections to fetch only needed fields

Transactions & ACID

36. Does MongoDB support ACID transactions?
Yes, since version 4.0 (for multi-document transactions).

37. How do you start a transaction in MongoDB?
Using sessions in code:

```
const session = client.startSession();
session.startTransaction();
```

38. When should you use transactions in MongoDB?
When you need consistency across multiple operations or collections.

39. What is the performance impact of transactions?
They add overhead and can reduce performance — use only when needed.

40. Can MongoDB rollback a transaction?
Yes, if an error occurs before commit, the transaction can be aborted and rolled back.

Tools & Integration

41. What is MongoDB Atlas?
A fully-managed cloud database service by MongoDB, supporting automation, backups, and scaling.

42. What is Mongoose?

An ODM (Object Data Modeling) library for MongoDB and Node.js that provides schema enforcement and validation.

43. How do you define a schema in Mongoose?

```
const UserSchema = new mongoose.Schema({
  name: String,
  email: { type: String, required: true }
});
```

44. How do you connect MongoDB in Node.js?

```
const { MongoClient } = require("mongodb");
const client = new MongoClient(uri);
await client.connect();
```

45. What are some MongoDB GUI tools?

- MongoDB Compass
- Robo 3T
- Studio 3T
- NoSQLBooster

Security & Best Practices

46. How do you secure a MongoDB instance?

- Enable authentication
- Use TLS/SSL
- Restrict network access
- Enable IP whitelisting (in Atlas)

47. What is role-based access control (RBAC)?

MongoDB allows creating users with specific roles and access levels.

48. How do you backup MongoDB?

Using `mongodump` or Atlas backups for managed services.

49. What is the size limit of a document in MongoDB?

16MB per document.

50. What are some common MongoDB use cases?

- Real-time analytics
- Content management
- IoT apps
- Product catalogs
- Mobile apps with flexible schemas