

MongoDB Interview Q & A



Wasim Patwari

**Become Job-Ready in Data Analytics with
Our Paid Mentorship & Internship Programs**

WhatsApp:91- 9607157409

Presented By: Afnan& Nikhil

1. What is MongoDB?

Answer:

MongoDB is a NoSQL, document-oriented database that stores data in JSON-like documents (called BSON). It is schema-less, scalable, and flexible, making it good for handling unstructured or semi-structured data.

2. How is MongoDB different from traditional relational databases (RDBMS)?

Answer:

- MongoDB stores data in collections and documents.
- RDBMS stores data in tables and rows.
- MongoDB is schema-less, while RDBMS uses a fixed schema.
- MongoDB is horizontally scalable (easy to add servers), while RDBMS usually scales vertically (bigger servers).

3. What is a collection in MongoDB?

Answer:

A collection is like a table in RDBMS. It stores a group of documents. Unlike tables, collections don't enforce a schema, so each document can have different fields.

4. What is a document in MongoDB?

Answer:

A document is a record in a collection. It is stored as BSON (Binary JSON) and contains key-value pairs. Example:

```
{  
  "name": "Aryan",  
  "age": 22,  
  "city": "Mumbai"  
}
```

5. What is BSON?

Answer:

BSON stands for Binary JSON. It is a binary-encoded format used by MongoDB to store documents, allowing it to be faster and more efficient than plain JSON.

6. What is the default `_id` field in MongoDB?

Answer:

Every MongoDB document automatically gets an `_id` field which is a unique identifier for that document. If not provided, MongoDB generates an ObjectId automatically.

7. What are CRUD operations in MongoDB?

Answer:

Create → `insertOne()`, `insertMany()`

Read → `find()`, `findOne()`

Update → `updateOne()`, `updateMany()`

Delete → `deleteOne()`, `deleteMany()`

8. What is the use of `find()` in MongoDB?

Answer:

`find()` is used to retrieve documents from a collection. Example

```
db.users.find({ age: 22 })
```

This finds all documents where age is 22.

9. What is indexing in MongoDB?

Answer:

Indexes make search queries faster by creating a data structure that improves lookup speed. Without indexes, MongoDB must scan every document.

10. What is replication in MongoDB?

Answer:

Replication means copying data from one server to others (replica set). It provides data availability and fault tolerance — if one server goes down, another can take over.

11. What is a replica set in MongoDB?

Answer:

A replica set is a group of MongoDB servers that maintain the same data. One node acts as the primary (handles all writes), and others are secondaries (read-only and for backup). It ensures high availability.

12. What is the purpose of the ObjectId in MongoDB?

Answer:

ObjectId is the default unique identifier used in the `_id` field of documents. It is a 12-byte value containing a timestamp, machine ID, process ID, and counter. It helps uniquely identify each document.

13. Can MongoDB handle relationships between data like joins?

Answer:

MongoDB is not relational, but it can handle relationships using:

Embedded documents (for one-to-one or one-to-many)

References (manual linking using `_id`)

It also supports `$lookup` in aggregation to perform joins-like operations between collections.

14. What is the difference between `insertOne()` and `insertMany()`?

Answer:

`insertOne()` → Inserts a single document into a collection

`insertMany()` → Inserts multiple documents at once

Example:

```
db.users.insertOne({name:"Afnan"})
```

```
db.users.insertMany([{name:"Ali"},{name:"Sara"}])
```

15. What is the MongoDB shell?

Answer:

The MongoDB shell is a command-line interface that lets developers interact with the database by running commands, queries, and scripts directly.

16. What is the Aggregation Framework in MongoDB?

Answer:

The Aggregation Framework is used for data processing and transformation in MongoDB. It works like SQL's GROUP BY and allows:

- Filtering (\$match)
- Grouping (\$group)
- Sorting (\$sort)
- Joining collections (\$lookup)
- Reshaping documents (\$project)

17. What are capped collections?

Answer:

Capped collections are fixed-size collections that automatically overwrite the oldest documents when full.

- Useful for logging, caching, IoT, or sensor data.
- Support high-throughput inserts but no document deletions.

18. What are transactions in MongoDB, and when do you use them?

Answer:

- Transactions (introduced in MongoDB 4.0) allow multi-document ACID operations.
- Useful when updating multiple collections/documents that must succeed/fail together.

19. What is sharding in MongoDB?

Answer:

Sharding is horizontal partitioning of data across multiple servers for scalability.

- Data is split using a shard key.
- Balancer distributes data evenly across shards.
- Queries are routed via mongos router.

20. What is the difference between Replica Sets and Sharding?

Answer:

- Replica Set: Copies of the same data (high availability, fault tolerance).
- Sharding: Splits data into partitions (scalability, large datasets).
- In production, both are often combined.

21. What are covered queries in MongoDB?

Answer:

A covered query is when all required fields in a query are present in an index.

- MongoDB does not need to scan documents → faster performance.

22. How does MongoDB ensure high availability?

Answer:

- Through Replica Sets.
- If the primary fails, an automatic election promotes a secondary to primary.
- Clients automatically reconnect to the new primary.

23. What is TTL (Time-To-Live) index in MongoDB?

Answer:

A TTL index automatically removes documents after a certain period.

- Useful for sessions, logs, temporary data.

24. What is difference between \$lookup and \$graphLookup?

Answer:

- \$lookup: Performs joins between collections (similar to SQL JOIN).
- \$graphLookup: Traverses recursive relationships (hierarchical data like org charts, friends-of-friends).

25. How does MongoDB handle indexing for large-scale data?

Answer:

- Uses B-Tree indexes.
- Supports compound indexes, multikey indexes (for arrays), text indexes, and geospatial indexes.
- Trade-off: More indexes → faster reads but slower writes.

26. What is the difference between MongoDB and Cassandra?

Answer:

- MongoDB: Document-oriented, flexible schema, strong query language, good for ad-hoc queries.
- Cassandra: Wide-column store, optimized for high write throughput and time-series data.
- MongoDB is CP (Consistency + Partition Tolerance).
- Cassandra is AP (Availability + Partition Tolerance).

27. How does MongoDB handle concurrency?

Answer:

- Uses WiredTiger storage engine (default).
- Provides document-level locking (not collection-level).
- Supports multi-version concurrency control (MVCC) → allows multiple readers/writers.

28. What are MongoDB Atlas and its benefits?

Answer:

MongoDB Atlas is a cloud-managed MongoDB service.

- Automated backups, scaling, and monitoring.
- Multi-cloud availability (AWS, GCP, Azure).
- Security (encryption, access controls).
- Global data distribution.

29. What is the difference between MongoDB Compass and Mongo Shell?

Answer:

- Mongo Shell: CLI to run commands and queries.
- MongoDB Compass: GUI tool for visualizing collections, indexes, queries, and performance.

30. What are some limitations of MongoDB?

Answer:

- Joins are not as efficient as RDBMS.
- High memory usage for indexes.
- Not ideal for complex transactions-heavy workloads (like banking).
- Requires careful schema design to avoid performance issues.



Kickstart Your Data Analytics Journey Today!

Internship Program : Get real-world experience through hands-on projects, preparing you for the professional world.

Mentorship Program : Learn to upskill effectively with expert guidance and personalized self-learning strategies
Job Assistance Program : Connect with top employers and gain the tools to confidently secure your dream job.

If you want to become a Data Analyst with the help of free resources and without investing in expensive

Courses, Connect With Us.

Vision:

Solve India's employment challenges
Make tech education affordable & accessible
Build job-ready professionals with real-world exposure

Linked 