MongoDB Interview Questions and Answers

mindmajix.com/mongodb-interview-questions

April 22, 2021

MongoDB is a document database that stores the data in JSON documents. It works over the documents and collections concept. MongoDB can store multiple databases and provides higher performance besides scalability and redundancy. This MongoDB interview question is mainly designed to provide you with basic ideas about the kind of interview questions you might face.

Normally, in interviews, recruiters start with basic questions, and slowly they will increase the difficulty level. So, in this **MongoDB Interview Questions** blog also, first, we will cover the basic questions, and then we will move to complex questions. Through these hand-picked MongoDB interview questions, you can prepare for your MongoDB job interview.

We have categorized MongoDB Interview Questions - 2023 (Updated) into 4 levels they are:

- **Top**
- **Basic**
- **Advanced**
- **Experienced**

Top 10 Frequently Asked MongoDB Interview Questions

Top MongoDB Interview Questions and Answers

1) What is MongoDB?

MongoDB is a cross-platform document-based database. Categorized as a NoSQL database, MongoDB avoids the conventional table-oriented relational database structure in support of the JSON-like documents with the dynamic schemas, making the data integration in specific kinds of applications quicker and simpler.

MongoDB was developed by a software company “10gen”, in October 2007 as an element of the planned platform as the service product. After that, the company was shifted to a freeware deployment model in 2009, providing sales assistance and other services.

2) What are the features of MongoDB?

Following are the important features of MongoDB:

- A compliant data model in the format of documents.
- Agile and extremely scalable database.
- Quicker than traditional databases.
- Demonstrative query language.

Check out MongoDB Tutorial

3)What type of NoSQL database MongoDB is?

MongoDB is a document-oriented database. It stores the data in the form of the BSON structure-oriented databases. We store these documents in a collection.

4) Explain Namespace?

A namespace is the series of the collection name and database name.

Basic MongoDB Interview Questions And Answers

5)Differentiate MongoDB and MySQL?

Despite MySQL and MongoDB being freeware and open source databases, there are several differences between them in terms of a data relationship, transaction, performance speed, querying data, schema design, normalization, etc. The comparison between MongoDB and MySQL is similar to the comparison between Non-relational and Relational databases.

Check out the Related Article MongoDB Vs MySQL

6) Explain Indexes in MongoDB?

In MongoDB, we use Indexes for executing the queries efficiently; without using Indexes, MongoDB should carry out a collection scan, i.e., scan all the documents of a collection, for selecting the documents which match the query statement. If a suitable index is available for a query, MongoDB will use an index for restricting the number of documents it should examine.

7) Why MongoDB is the best NoSQL database?

MongoDB is the best NoSQL database due to the following features:

- High Performance
- High Availability
- Easily Scalable

- Rich Query Language
- Document Oriented

8) Explain the significance of the covered query?

A covered query makes the query implementation quicker as we store the indexes in the RAM or consecutively located on the disk. It makes query execution quicker. The covered query covers all the fields in the index, MongoDB matches the query condition along with returning the result fields.

9) What is a replica set?

We can specify the replica as a set of the mongo instances which host a similar data set. In the replica set, one node will be primary, and another one will be secondary. We replicate all the data from the primary to the secondary nodes.

10) Differentiate MongoDB and Cassandra?

MongoDB	Cassandra
It is a cross-platform document-oriented database system	It is a high-performance distributed database system.
It is developed in C++	It is developed in Java
It is simple to administer in the failure case	It offers high availability

11) Explain the primary and secondary replica set?

In MongoDB, primary nodes are the nodes that accept writing. Primary nodes are also called master nodes. Replication in MongoDB is a single master. Therefore, only one node will accept the write operations at once.

12) Which languages can we use with MongoDB?

At Present, MongoDB offers driver support to C++, Java, PHP, Perl, Python, Go, Scala, and Ruby.

Check out **Cassandra vs MongoDB**

13) Explain Storage Encryption?

Storage encryption encodes all the MongoDB data over the storage or over the operating systems for assuring that only authenticated processes will access the safeguarded data.

14) Explain Primary and Secondary Replica Sets?

Primary Replica Set receives all the write operations from the clients. Secondary replica sets replicate the primary replica sets and implement the operations for their datasets so that secondary datasets affect the primary datasets.

15) What is the importance of GridFS and Journaling?

- GridFS: We use GridFS to retrieve and store large files like images, videos, and audio files.
- Journaling: We use Journaling for secure backups in MongoDB.

16) How to do locking or transactions in MongoDB?

MongoDB does not use traditional locking with the reduction because it is high-speed, knowable, and light in the presentation. We can consider it as the MyISAM, MySQL auto entrust script. Through the simpler business sustain, we can enhance the performance, specifically in the structure with various servers.

17) How to do Journaling in MongoDB?

We save the write operations in the memory while journaling is taking place. The on-disk journal files are dependable for the reason that journal writers are usual. In the DB path, MongoDB designs a journal subdirectory.

18) How does MongoDB provides concurrency?

MongoDB utilizes the reader-writer locks, enabling concurrent readers to access any supply such as collection or database though it provides private access to individual writers.

19) Explain Sharding and Aggregation in MongoDB?

- Aggregation: Aggregations are the activities that handle the data records and give the record results.
- Sharding: Sharding means storing the data on multiple machines.

20) What is the importance of profiler in MongoDB?

MongoDB contains the database profiler that shows the performance characteristics of every operation against the database. Through the profiler, we can identify the queries that are slower than they should be and use this data to determine when we require an index.

21) Define Collection?

The collection is a set of MongoDB documents.

22) Explain Aggregation Pipeline?

The aggregation Pipeline acts as a framework to perform aggregation tasks. We use this pipeline for transforming the documents into aggregated results.

23) Explain MapReduce?

MapReduce is a standard multi-phase data aggregation modality that we use to process the data quantities.

24) Explain Splitting?

Splitting is the background process that we use to store chunks from increasing too large.

25) What is the purpose of the save() method?

We use the save() method for replacing the existing documents with new documents.

26) What is the use of MongoDB?

- Generally, we use MongoDB as the main data store for the operational requirements with live needs. Generally, MongoDB is suitable for 80% of the applications which we develop today. MongoDB is simple to operate and extent in ways that are tough if they are not possible with the relational databases.
- MongoDB stands out in various use cases where the relational databases are not suitable, like applications with semi-structured, structured, along with the big scalability needs or the multi-datacenter deployments.
- MongoDB cannot be suitable for some applications. For instance, applications that need complex transactions and scan-based applications that access huge subsets of the data largely cannot be suitable for MongoDB.
- Some general uses of MongoDB comprise product catalogs, mobile apps, content management, real-time personalization, and applications providing individual views throughout several systems.

27) What is the purpose of the DB command?

We use the “DB” command to get the name of the presently selected database.

28) What are the restrictions of the MongoDB 32-bit versions?

When we run a 32-bit version of MongoDB, the total storage size of the server, containing indexes and data, is 2GB. Due to this reason, we will not deploy MongoDB to the production on the 32-bit machines. If we deploy a 64-bit version of MongoDB, there is no virtual restriction to the storage size. For the creation deployments, we strongly recommend 64-bit operating systems and builds.

29) When should we normalize the data in MongoDB?

It relies on our objectives. Normalization provides an updated effective data representation. Denormalisation makes data reading effective. Generally, we utilize embedded data models when:

- When we have “contains” relationships between the entities.
- When we have one-to-many relationships between the entities. In the relationships, “many” or the child documents display in the context of the parent documents.

Generally, we use normalized data models:

- When embedding results in duplication of the data yet they will not give enough read performance advantages to prevail the duplication implications.
- For representing more difficult many-to-many relationships.
- For modeling the big hierarchical data sets.

30) How do we perform sorting and Explain Project in MongoDB?

For finding any data in MongoDB, we use the find() method. The discovery () method returns the collection’s documents over which we invoked this method. We can use the “Where” clause in the MongoDB query in order to restrict the output by using MongoDB projection. Anytime we execute the find() method, MongoDB returns all the documents associated with a particular collection.

```
db.<collection_name>.find({ }, {<key_Name>:<Flag to display>})
```

31) How can MongoDB simulate subquery or join?

We have to find the best method for structuring the data in MongoDB for simulating what would be the simple subquery or join in SQL. For example, we have users and posts, with the users in one collection and posts in another collection. We have to find all the posts by the users whose city is “Hyderabad”.

32) Define oplog(operational log)?

An operational log (oplog) is a special kind of limited collection that stores a rolling record of all the operations which change the data we store in our databases. Primarily, it applies all the database operations over the primary and, after that, records these operations on the oplog of the primary. After that, the secondary members replicate and apply the operations in the asynchronous process.

33) How do we create a database in MongoDB?

When I want to create a database in MongoDB, I faced the following error:

```
 :~$mongo

MongoDB shell version:1.65

Connecting to: test

Error: Could not connect to the server

Exception: connect failed
```

The solution to the above error:

1. cd/var1/lib1/MongoDB
2. We remove the mongod. lock from the folder
3. Sudo start MongoDB
4. Mongo

34) What is the syntax of the skip() method?

skip() method syntax is:

```
db.COLLECTION_NAME.find().limit(NUMBER).skip(NUMBER)
```

35) How do we delete everything from the MongoDB database?

By using the following code, we can delete everything from the MongoDB database:

```
use [database];
db.dropDatabase();
Ruby code should be pretty similiar.
Also, from the command line:
mongo [Database] -eval "db.dropDatabase();"
use
[databaseName]
db.Drop+databasename();
drop colllection
use databaseName
db.collectionName.drop();
```

36) Which command do we use for creating the backup of the database?

We use the mongodump command for creating the database backup.

37) Which command do we use for restoring the backup?

We use mongorestore for restoring the backup.

38) Explain the importance of the dot notation?

In MongoDB, we use dot notation for accessing the array elements and the fields of an embedded document.

39) What is the syntax of the limit() and sort() method?

Syntax of the limit() method is:

```
>db.COLLECTION_NAME.find().limit(NUMBER)
```

Syntax of the sort() method is:

```
>db.COLLECTION_NAME.find().sort({KEY:1})
```

40) What do you know about NoSQL databases? What are the various types of NoSQL databases?

NoSQL refers to “Not Only SQL”. NoSQL is a kind of database that handles and sorts all kinds of structured, massive, and difficult data. It is a new method to think about databases. Kinds of NoSQL databases:

- Key-Value
- Graph
- Column Oriented
- Document Oriented

41) Which command do we use for dropping a database?

We use the “DB.drop database” command for dropping a database.

42) Explain MongoDB Projection

In MongoDB, we use Projection for selecting only the required data. It will not select the complete data of a document.

43) Why do we use the pretty() method?

We use the pretty() method for displaying the results in a formatted way.

44) How do we remove a document from the collection?

By using the remove() method, we remove a document from the collection.

45) What are the points we should consider while creating a schema in MongoDB?

We must consider the following points while creating a schema:

- Designing the Scheme based on the user requirements.
- Combining the objects into one document, if we have to use them jointly, or else, separate them.
- Perform joins while on write, and not while it is reading.
- For most general application scenarios, maximize the schema.
- Perform complex aggregations in the schema.

46) What does ObjectId contain?

ObjectId contains the following:

- Client machine ID
- Client process ID
- Byte incremented counter
- Timestamp

47) How do we use the select * group by MongoDB aggregation?

For instance, if we have to select all the attributes and groups by name throughout the records. For example:

```
{Name: George, x: 5, y: 3}
{Name: George, z: 9}
{Name: Rob, x: 12, y: 2}
```

We can do MongoDB aggregation as follows:

```
db.example.aggregate(
  {
    $group:{
      _id:'$name',
x: {$addToSet: "$x"    },
y: {$addToSet: "$y"    },
z: {$addToSet: "$z"    },
 }
}
)
```

48) Explain Vertical Scaling and Horizontal Scaling?

- **Vertical Scaling:** Vertical Scaling increases storage and CPU resources for expanding the capacity.
- **Horizontal Scaling:** Horizontal Scaling splits the datasets and circulates the data over multiple shards or servers.

49) What are the elements of the Sharded Cluster?

Following are the elements of the Sharded Cluster:

- Query routers
- Shards
- Config servers

50) What are the substitutes for MongoDB?

Following are the substitutes to MongoDB:

- Hbase

- CouchDB
- Cassandra
- Redis
- Riak

51) How can we old files in the moveChunk directory?

In the course of general shard balancing operations, we make the old files as backups, and we can delete them when those operations are completed.

52) What is a Storage Engine?

Storage Engine is a component of the database that is accountable to manage how we store on the disk. For instance, one storage engine may provide better performance for the read-heavy workloads, and another one may support a great throughput for the write operations.

53) Does MongoDB require plenty of RAM?

No, MongoDB does not require plenty of RAM. It can run on a small amount of memory. MongoDB dynamically assigns and unassigns RAM according to the needs of other processes.

Advanced MongoDB Interview Questions And Answers

54) Differentiate MongoDB and CouchDB?

MongoDB	CouchDB
MongoDB is quicker than CouchDB	CouchDB is more secure than MongoDB
Triggers do not exist in MongoDB.	Triggers exist in CouchDB
MongoDB serializes the JSON Data to the BSON	CouchDB does not store the data in JSON format

55) Explain Capped Collection?

In MongoDB, the Capped collection is a special kind of collection. This indicates that in this collection, we can restrict the collection size. Syntax of Capped Collection is as follows:

```
db.createCollection(<collection_name>, {capped: Boolean, autoIndexId: Boolean,
size: Number, max : Number})
```

In the Capped Collection syntax, we have the following fields:

- Collection_Name: This field is the collection name that we create as the capped collection.
- Capped: Capped is a boolean field; it is true if we create a capped collection. By default, its value is false.
- auto indexed: It is a boolean flag that we use for auto-indexing. If this flag is true, indexes will be created automatically. If the flag is false, indexes will not be created automatically.
- Size: Size is the parameter that represents the maximum amount of documents in bytes. It is the required field in the context of capped collections.
- Max: Max is the parameter that represents the highest number of documents that permit the collection.

56) How do we perform the Join operations in MongoDB?

From MongoDB3.2, we can perform the Join operation. The new $lookup operator included with the aggregation pipeline is the same as the left outer join. Example:

```
{
   $lookup:
     {
       from: <collection to join>,
       localField: <field from the input documents>,
       foreignField: <field from the documents of the "from" collection>,
       as: <output array field>
     }
}
```

57) What are the storage engines used by MongoDB?

WiredTiger and MMAPv1 are the two storage engines used by MongoDB.

58) How do we configure the cache size in MongoDB?

In MongoDB, we cannot configure the cache. MongoDB utilizes the free spaces over the system automatically by using memory-mapped files.

59) How do we control the MongoDB Performance?

We can control the MongoDB Performance by:

- Locking the Performance
- Identifying the number of connections
- Database Profiling
- Full-time Diagnostic Data Capture

60) What are the aggregate functions of MongoDB?

Following are the aggregate functions of MongoDB:

- AVG

- Sum
- Min
- Max
- First
- Push
- addTo Set
- Last

61) What are the CRUD operations of MongoDB?

Following are the CRUD operations of MongoDB:

```
Create-db.collection.insert();

Read-db.collection.find();

Update-db.collection.update();

Delete-db.collection.remove();
```

62) What are the datatypes of MongoDB?

Following are the datatypes of MongoDB:

- Integer
- String
- Boolean
- Array
- Double
- Date
- Timestamp
- Regular Expression

63) Is it required to invoke “get last error” for making a write durable?

No, it is not required to invoke “get last error”. The server acts as if it has been invoked. “get last error” enables us to acquire confirmation that a write operation is committed. You will get the confirmation, yet the durability and safety of the writer are independent.

64) What happens when the Shard is slow or down while querying?

When the Shard is slow, the query returns an error until partial query options are fixed. When the shard is reacting slowly, MongoDB waits for it.

65) How do we use a primary key in MongoDB?

“_id field” is reticent for a primary key in MongoDB. And it is a distinct value. If we do not set anything to the “_id”, it will systematically fill it with the “MongoDB Id Object”. Yet, we can store any distinct information in that field.

66) How do we see the connections utilized by MongoDB?

For seeing the connections utilized by MongoDB, we use db_adminCommand(”connPoolStats”).

67) When a “moveChunk” fails, is it required to clean up partly moved docs?

No, it is not required to clean up the partly moved docs because chunk moves are deterministic and consistent. The move will try again, and when finished, data will be on the latest Shard.

68) Explain how to start the MongoDB Instance or Server?

We have to follow the below steps for starting the MongoDB Server:

- First, open the command prompt and execute the “mongod.exe” file.
- On the other hand, we move to the path where we installed MongoDB.
- Go to the bin folder, find the “mongod.exe” file, and double click the file for executing it.
- We can go to the folder, for instance, “C: MongoDB/bin” and type mongo for connecting MongoDB by using the Shell.

69) Differences between MongoDB and RDBMS

Basis for Comparison	**MongoDB**	**RDBMS**
Definition	It is a non-relational database	It is a relational database management system
Working	It works over relationships among the tables, which use columns and rows	It is a document-oriented database system through fields and documents
Scalability	It is horizontally and vertically scalable	It is vertically scalable
Performance	Performance enhances with the rise in the processors	Performance enhances with the rise in the RAM capacity
Hierarchical Data Storage	It has a built-in provision to store the hierarchical data	It is hard to store the hierarchical data
Support to Joins	It does not support difficulty Joins	It supports complex joins
Query Language	It uses BSON for database querying	It uses SQL to query the database

Javascript Support	It provides support to javascript-based clients for querying the database	It does not provide support to the javascript-based clients to query the database

70) How do applications access the real-time data modifications in MongoDB?

Applications access the real-time data modifications through the Change streams that serve as the subscriber for every collection operation like delete, insert, and update.

71) What are the different kinds of Indexes in MongoDB?

Following are the different kinds of Indexes in MongoDB:

- Default: It is the "_id" that MongoDB creates.
- Compound: It is useful for multiple fields.
- Multi-key: It indexes the array data.
- Single field: It sorts and indexes over a single field.
- Geospatial: It is useful for querying the location data.
- Hashed: It indexes the hashes of the multiple fields.

MongoDB Interview Questions And Answers For Experienced

71) Define BSON?

Binary JSON or BSON is a binary-encoded format of the JSON. BSON extends the JSON and offers various data fields and types.

72) How does MongoDB store the data?

As it is a document-based database, MongoDB stores the documents in Binary Javascript Object Notation or BSON, which is a binary-encoded format of JSON.

73) Does MongoDB support ACID Transaction? Define ACID Transaction?

Yes, MongoDB supports ACID Transaction. ACID refers to Atomicity, Consistency, Isolation, and Durability. Transaction manager assures that we handle these attributes.

74) Explain Composing elements or Structure of ObjectID in MongoDB?

In MongoDB, ObjectID is associated with the "_id" field, and MongoDB uses it as the default value of the "_id" in the documents. For generating "ObjectID", we use the following Syntax:

```
ObjectId([SomeHexaDecimalValue])
```

Example:

```
ObjectId() = newObjectId
```

ObjectID has the following methods:

- Str: This method provides the string representation of the object id.
- valueOf()- This method returns hexadecimal representation of the ObjectId.
- getTimeStamp()- This method returns timestamp of the ObjectId.
- toString()- This method returns the string representation of the ObjectId in “ObjectId(haxstring)”.

75) How do we find array elements with multiple criteria?

For example, if we have the below documents:

```
{ _id: 1, numbers: [1000, -1000]]
{ _id: 2, numbers: [500]]
```

When we execute the following command:

```
db.example.find( { numbers: { $elemMatch: { $gt: -10, $lt: 10 } } } );
```

76) How can we sort the user-defined function? For example, x and y are integers, and how do we calculate “x-y”?

By executing the following code, we calculate x-y.

```
db.eval(function() {
return db.scratch.find().toArray().sort(function(doc1, doc2) {
return doc1.a – doc2.a
})
});

Versus the equivalent client-side sort:
db.scratch.find().toArray().sort(function(doc1, doc2) {
return doc1.a – doc2.b
});
```

By using the aggregation pipeline and “$orderby” operator, it is possible to sort.

77) Upto Which extent does the data expand to multi-slice?

MongoDB shred stands on the collection. Therefore, we store all the substances in a mass or a lump. When we have an additional time slot, then we will have few slice data achievement options, yet when we have multiple lumps, data will be extended to numerous slices.

78) How do we retrieve MongoDB databases in Javascript Array?

In the MongoDB terminal, we can run “Show DBS” to retrieve the existing databases. To get the MongoDB databases programmatically, we execute the following code:

```
use admin
dbs = db.runCommand({listDatabases: 1})
dbNames = []
for (var i in dbs.databases) { dbNames.push(dbs.databases[i].name) }
Hopefully this will help someone else.
The below will create an array of the names of the database:
var connection = new Mongo();
var dbNames = connection.getDBNames();
```

79) How do we update the object in the Nested Array?

By executing the following code, we update the object:

```
Skip code block
{
“_id” : ObjectId(“4faaba123412d654fe83hg876”),
“user_id” : 123456,
“total” : 100,
“items” : [
{
“item_name” : “my_item_one”,
“price” : 20
},
{
“item_name” : “my_item_two”,
“price” : 50
},
{
“item_name” : “my_item_three”,
“price” : 30
}
]
}
```

80) How do we retrieve a particular embedded document in a MongoDB collection?

I have a collection that has an embedded document known as notes.

```
Skip code block
{
“_id” : ObjectId(“4f7ee46e08403d063ab0b4f9”),
“name” : “MongoDB”,
“notes” : [
{
“title” : “Hello MongoDB”,
“content” : “Hello MongoDB”
},
{
“title” : “ReplicaSet MongoDB”,
“content” : “ReplicaSet MongoDB”
}
]
}
```

81) How do we query a nested Join?

To query the nested join, we use “tested”. For example:

```
{“_id” : ObjectId( “abcd” ),
“className” : “com.myUser”,
“reg” : 12345,
“test” : [
{ “className” : “com.abc”,
“testid” : “pqrs” } ] }
```

82) Can we run more than one Javascript Operation in one MongoDB instance?

Yes, we can run multiple javascript operations in one MongoDB instance.