MongoDB Interview Questions

1) What do you understand by NoSQL databases? Is MongoDB a NoSQL database? explain.

At the present time, the internet is loaded with big data, big users, big complexity etc. and also becoming more complex day by day. NoSQL is the answer to all these problems, it is not a traditional database management system, not even a relational database management system (RDBMS). NoSQL stands for "Not Only SQL". NoSQL is a type of database that can handle and sort all type of unstructured, messy and complicated data. It is just a new way to think about the database.

Yes. MongoDB is a NoSQL database.

2) Which are the different languages supported by MongoDB?

MongoDB provides official driver support for C, C++, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala, Go and Erlang.

You can use MongoDB with any of the above languages. There are some other community supported drivers too, but the above-mentioned ones are officially provided by MongoDB.

3. What is Mongo shell?

Mongo shell is a JavaScript interface to MongoDB that can be used to query and update data. It is interactive and can also be used to execute administrative operations.

4) What are the different types of NoSQL databases? Give some examples?

NoSQL database can be classified as 4 basic types:

1. Key value store NoSQL database
2. Document store NoSQL database
3. Column store NoSQL database
4. Graph base NoSQL database

There are many NoSQL databases. MongoDB, Cassandra, CouchBD, Hypertable, Redis, Riak, Neo4j, HBASE, Couchbase, MemcacheDB, Voldemort, RevenDB etc. are examples of NoSQL databases.

5) Is MongoDB better than other SQL databases? If yes then how?

MongoDB is better than other SQL databases because it allows a highly flexible and scalable document structure.

For example:

- One data document in MongoDB can have five columns and the other one in the same collection can have ten columns.
- MongoDB databases are faster than SQL databases due to efficient indexing and storage techniques.

6) What type of DBMS is MongoDB?

MongoDB is a document-oriented DBMS

7. List out the important features of MongoDB.

Some of the important features of MongoDB are:

- Uses a schema-less database
- No complex joins
- Faster access to data because of the presence of the working set (internal memory)
- Features like aggregation, sharding, and replication make it easy to use
- Cross-platform and document-based
- Automatic fail-over and high-availability

8) What is the difference between MongoDB and MySQL?

Although MongoDB and MySQL both are free and open-source databases, there is a lot of difference between them in the term of data representation, relationship, transaction, querying data, schema design and definition, performance speed, normalization and many more. To compare MySQL with MongoDB is like a comparison between Relational and Non-relational databases.

9) Why MongoDB is known as the best NoSQL database?

MongoDb is the best NoSQL database because, it is:
Document Oriented
Rich Query language
High Performance
Highly Available
Easily Scalable

10) Does MongoDB support primary-key, foreign-key relationships?

No. By Default, MongoDB doesn't support primary key-foreign key relationship.

11) Does MongoDB need a lot of RAM?

No. There is no need for a lot of RAM to run MongoDB. It can be run even on a small amount of RAM because it dynamically allocates and de-allocates RAM according to the requirement of the processes.

12. MongoDB is called a schema-less database. If yes, how do you create the schema in MongoDB?

It would be more correct to say that MongoDB has a dynamically typed schema because it relies on JSON, which is a schema-free data structure. To create a schema, create and insert a document. Once a document is inserted, a corresponding collection will be created in the database.

13. What is a namespace?

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

Example: students.the subject, where students are the database and subject, is the collection.

14. How do you perform CRUD operations in MongoDB?

- **C** – Create: db.collection.insert();
- **R** – Read: db.collection.find();
- **U** – Update: db.collection.update();
- **D** – Delete: db.collection.remove({“fieldname” : ”value”});

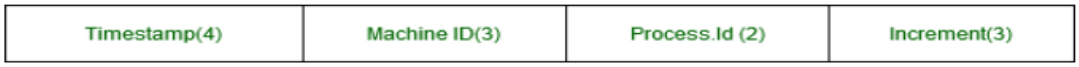
15) Explain the structure of ObjectID in MongoDB.

Every document in the collection has an “_id” field that is used to uniquely identify the document in a particular collection it acts as the primary key for the documents in the collection. “_id” field can be used in any format and the default format is **ObjectId** of the document.

An ObjectID is a 12-byte Field Of BSON type

- The first 4 bytes representing the Unix Timestamp of the document

- The next 3 bytes are the machine Id on which the MongoDB server is running.
- The next 2 bytes are of process id
- The last Field is 3 bytes used for increment the objectid.



16) Is it true that MongoDB uses BSON to represent document structure?

Yes.

17) What are Indexes in MongoDB?

In MondoDB, Indexes are used to execute query efficiently. Without indexes, MongoDB must perform a collection scan, i.e. scan every document in a collection, to select those documents that match the query statement. If an appropriate index exists for a query, MongoDB can use the index to limit the number of documents it must inspect.

18) By default, which index is created by MongoDB for every collection?

By default, the _id collection is created for every collection by MongoDB.

19) In which language MongoDB is written?

MongoDB is written and implemented in C++.

20. How do you create and drop a collection in MongoDB?

- **Create collection:** db.createCollection();
- **Drop collection:** db.collection.drop();

21. How can you store images, videos and other large files in MongoDB?

Large files are stored in MongoDB using the GridFS specification.

22) What language can you use with MongoDB?

MongoDB client drivers supports all the popular programming languages so there is no issue of language, you can use any language that you want.

23) Does MongoDB database have tables for storing records?

No. Instead of tables, MongoDB uses "Collections" to store data.

24) Do the MongoDB databases have schema?

Yes. MongoDB databases have dynamic schema. There is no need to define the structure to create collections.

25) What is the method to configure the cache size in MongoDB?

MongoDB's cache is not configurable. Actually MongoDB uses all the free spaces on the system automatically by way of memory mapped files.

26. What is the command to list all the indexes in a collection?

The command is db.collection.getIndexes();

27. How does MongoDB perform text search?

Text search can be done using text index. Here's an example:

db.collection_name.ensureIndex();

28. What is the default interval to write updates to the disk?

The default interval is 60 seconds.

29. List some of the data types supported by MongoDB.

Some data types are numbers, string, arrays, binary data, booleans, date, regular expressions, ObjectId, etc.

30. How can applications access real-time data changes?

Applications can access real-time data changes using Change streams which acts as a subscriber to all the collection operations like insert, delete and update.

31) What is sharding in MongoDB?

In MongoDB, Sharding is a procedure of storing data records across multiple machines. It is a MongoDB approach to meet the demands of data growth. It creates horizontal partition of data in a database or search engine. Each partition is referred as shard or database shard.

32) What is replica set in MongoDB?

A replica can be specified as a group of mongo instances that host the same data set. In a replica set, one node is primary, and another is secondary. All data is replicated from primary to secondary nodes.

33) By default, which replica sets are used to write data?

By default, MongoDB writes data only to the primary replica set.

34) In which format MongoDB represents document structure?

MongoDB uses BSON to represent document structures.

35) What will happen when you remove a document from database in MongoDB? Does MongoDB remove it from disk?

Yes. If you remove a document from database, MongoDB will remove it from disk too.

36) Why are MongoDB data files large in size?

MongoDB doesn't follow file system fragmentation and pre allocates data files to reserve space while setting up the server. That's why MongoDB data files are large in size.

37) How does MongoDB provide concurrency?

MongoDB uses reader-writer locks for concurrency. Reader-writer locks allow concurrent readers shared access to a resource, such as a database or collection, but give exclusive access to a single write operation.

38) Is there any need to create database command in MongoDB?

You don't need to create a database manually in MongoDB because it creates automatically when you save the value into the defined collection at first time.

39) Explain what is MongoDB?

Mongo-DB is a document database which provides high performance, high availability and easy scalability.

40. What is a Document in MongoDB?

A Document in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects:

```
{"greeting" : "Hello world!"}
```

Complex documents will contain multiple key/value pairs:

```
{"greeting" : "Hello world!", "views" : 3}
```

41. What is a Collection in MongoDB?

A collection in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

Documents within a single collection can have any number of different “shapes.”, i.e. collections have dynamic schemas.

For example, both of the following documents could be stored in a single collection:

{"greeting" : "Hello world!", "views": 3}

{"signoff": "Good bye"}

42. What are Databases in MongoDB?

MongoDB groups collections into databases. MongoDB can host several databases, each grouping together collections.

Some reserved database names are as follows:

admin

local

config

43. How to add data in MongoDB?

The basic method for adding data to MongoDB is “inserts”. To insert a single document, use the collection’s insertOne method:

> db.books.insertOne({"title" : "Start With Why"})

For inserting multiple documents into a collection, we use insertMany. This method enables passing an array of documents to the database.

44. How do you Update a Document?

Once a document is stored in the database, it can be changed using one of several update methods: updateOne, updateMany, and replaceOne. updateOne and updateMany each takes a filter document as their first parameter and a modifier document, which describes changes to make, as the second parameter. replaceOne also takes a filter as the first parameter, but as the second parameter replaceOne expects a document with which it will replace the document matching the filter.

For example, in order to replace a document:

```
{
  "_id" : ObjectId("4b2b9f67a1f631733d917a7a"),
  "name" : "alice",
  "friends" : 24,
  "enemies" : 2
}
```

45. How do you Delete a Document?

The CRUD API in MongoDB provides deleteOne and deleteMany for this purpose. Both of these methods take a filter document as their first parameter. The filter specifies a set of criteria to match against in removing documents.

For example:

```
> db.books.deleteOne({"_id" : 3})
```

46. How to perform queries in MongoDB?

The **find** method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to find, which is a document specifying the query criteria.

Example:

```
> db.users.find({"age" : 24})
```

47. What are the data types in MongoDB?

MongoDB supports a wide range of data types as values in documents. Documents in MongoDB are similar to objects in JavaScript. Along with JSON's essential key/value–pair nature, MongoDB adds support for a number of additional data types. The common data types in MongoDB are:

- Null
  {"x" : null}
- Boolean
  {"x" : true}
- Number
  {"x" : 4}
- String
  {"x" : "foobar"}

- Date
  {"x" : new Date()}
- Regular expression
  {"x" : /foobar/i}
- Array
  {"x" : ["a", "b", "c"]}
- Embedded document
  {"x" : {"foo" : "bar"}}
- Object ID
  {"x" : ObjectId()}
- Binary Data
  Binary data is a string of arbitrary bytes.
- Code
  {"x" : function() { /* ... */ }}

48. When to use MongoDB?

You should use MongoDB when you are building internet and business applications that need to evolve quickly and scale elegantly. MongoDB is popular with developers of all kinds who are building scalable applications using agile methodologies.

MongoDB is a great choice if one needs to:

- Support a rapid iterative development.
- Scale to high levels of read and write traffic - MongoDB supports horizontal scaling through Sharding, distributing data across several machines, and facilitating high throughput operations with large sets of data.
- Scale your data repository to a massive size.
- Evolve the type of deployment as the business changes.
- Store, manage and search data with text, geospatial, or time-series dimensions.

49. Define BSON.

BSON stands for Binary JSON, which is a binary-encoded serialization of JSON-like documents used in MongoDB. BSON extends the JSON model by adding additional data types and optimizing for efficient data storage and retrieval.

50. Does MongoDB support ACID Transaction? Define ACID Transaction?

Yes, MongoDB supports ACID transactions. ACID stands for Atomicity, Consistency, Isolation, and Durability. ACID transactions are a set of properties that guarantee that database transactions are processed reliably.