Introduction to MongoDB

When dealing with data, there are two types of data as we know – (i) structured data and (ii) unstructured data. Structured data is usually stored in a tabular form whereas unstructured data is not. To manage huge sets of unstructured data like log or IoT data, a NoSQL database is used.

What is MongoDB ?

- MongoDB is an open-source NoSQL database written in C++ language. It uses JSON-like documents with optional schemas.
- It provides easy scalability and is a cross-platform, document-oriented database.
- MongoDB works on the concept of Collection and Document.
- It combines the ability to scale out with features such as secondary indexes, range queries, sorting, aggregations, and geospatial indexes.
- MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

MongoDB Basic Interview Questions

1. What are some of the advantages of MongoDB?

Some advantages of MongoDB are as follows:

- MongoDB supports field, range-based, string pattern matching type queries. for searching the data in the database
- MongoDB support primary and secondary index on any fields
- MongoDB basically uses JavaScript objects in place of procedures
- MongoDB uses a dynamic database schema
- MongoDB is very easy to scale up or down
- MongoDB has inbuilt support for data partitioning (Sharding).

2. 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}
```

3. 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"}
```

You can download a PDF version of Mongodb Interview Questions.

---

4. 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

5. What is the Mongo Shell?

It is a JavaScript shell that allows interaction with a MongoDB instance from the command line. With that one can perform administrative functions, inspecting an instance, or exploring MongoDB.

To start the shell, run the mongo executable:

```
$ mongod
$ mongo
MongoDB shell version: 4.2.0
connecting to: test
>
```

The shell is a full-featured JavaScript interpreter, capable of running arbitrary JavaScript programs. Let’s see how basic math works on this:

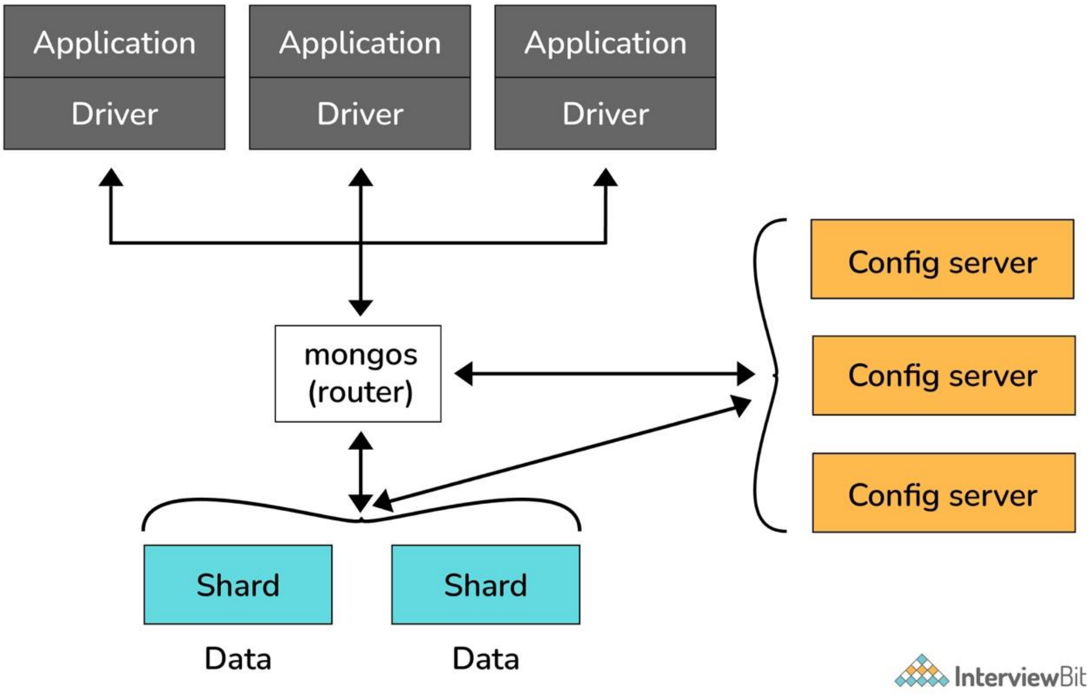
```
> x = 100;
200
> x / 5;
20
```

6. How does Scale-Out occur in MongoDB?

The document-oriented data model of MongoDB makes it easier to split data across multiple servers. Balancing and loading data across a cluster is done by MongoDB. It then redistributes documents automatically.

The mongos acts as a query router, providing an interface between client applications and the sharded cluster.

Config servers store metadata and configuration settings for the cluster. MongoDB uses the config servers to manage distributed locks. Each sharded cluster must have its own config servers.



7. What are some features of MongoDB?

- **Indexing:** It supports generic secondary indexes and provides unique, compound, geospatial, and full-text indexing capabilities as well.
- **Aggregation:** It provides an aggregation framework based on the concept of data processing pipelines.
- **Special collection and index types:** It supports time-to-live (TTL) collections for data that should expire at a certain time
- **File storage:** It supports an easy-to-use protocol for storing large files and file metadata.
- **Sharding:** Sharding is the process of splitting data up across machines.

8. 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.

9. 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
}
```

10. 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})
```

11. 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})
```

12. 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() { /* ... */ }}
  ```

13. 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.

MongoDB Intermediate Interview Questions

14. How is Querying done 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.

For example: If we have a string we want to match, such as a "username" key with the value "alice", we use that key/value pair instead:

```
> db.users.find({"username" : "alice"})
```

15. Explain the term "Indexing" in MongoDB.

In MongoDB, indexes help in efficiently resolving queries. What an Index does is that it stores a small part of the data set in a form that is easy to traverse. The index stores the value of the specific field or set of fields, ordered by the value of the field as specified in the index.
MongoDB's indexes work almost identically to typical relational database indexes.

Indexes look at an ordered list with references to the content. These in turn allow MongoDB to query orders of magnitude faster. To create an index, use the `createIndex` collection method.

For example:

```
> db.users.find({"username": "user101"}).explain("executionStats")
```

Here, `executionStats` mode helps us understand the effect of using an index to satisfy queries.

16. What are Geospatial Indexes in MongoDB?

MongoDB has two types of geospatial indexes: 2dsphere and 2d. 2dsphere indexes work with spherical geometries that model the surface of the earth based on the WGS84 datum. This datum model the surface of the earth as an oblate spheroid, meaning that there is some flattening at the poles. Distance calculations using 2sphere indexes, therefore, take the shape of the earth into account and provide a more accurate treatment of distance between, for example, two cities, than do 2d indexes. Use 2d indexes for points stored on a two-dimensional plane.

2dsphere allows you to specify geometries for points, lines, and polygons in the GeoJSON format. A point is given by a two-element array, representing [longitude, latitude]:

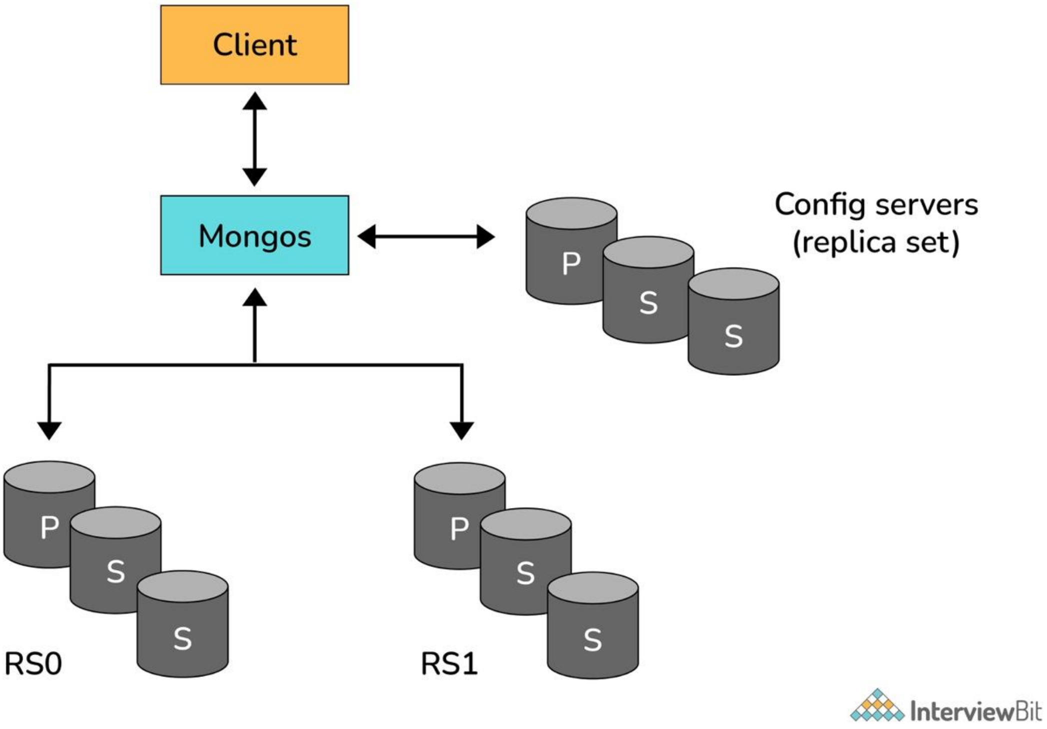
```
{
   "name" : "New York City",
   "loc" : {
       "type" : "Point",
       "coordinates" : [50, 2]
   }
}
```

A line is given by an array of points:

```
{
   "name" : "Hudson River",
   "loc" : {
       "type" : "LineString",
       "coordinates" : [[0,1], [0,2], [1,2]]
   }
}
```

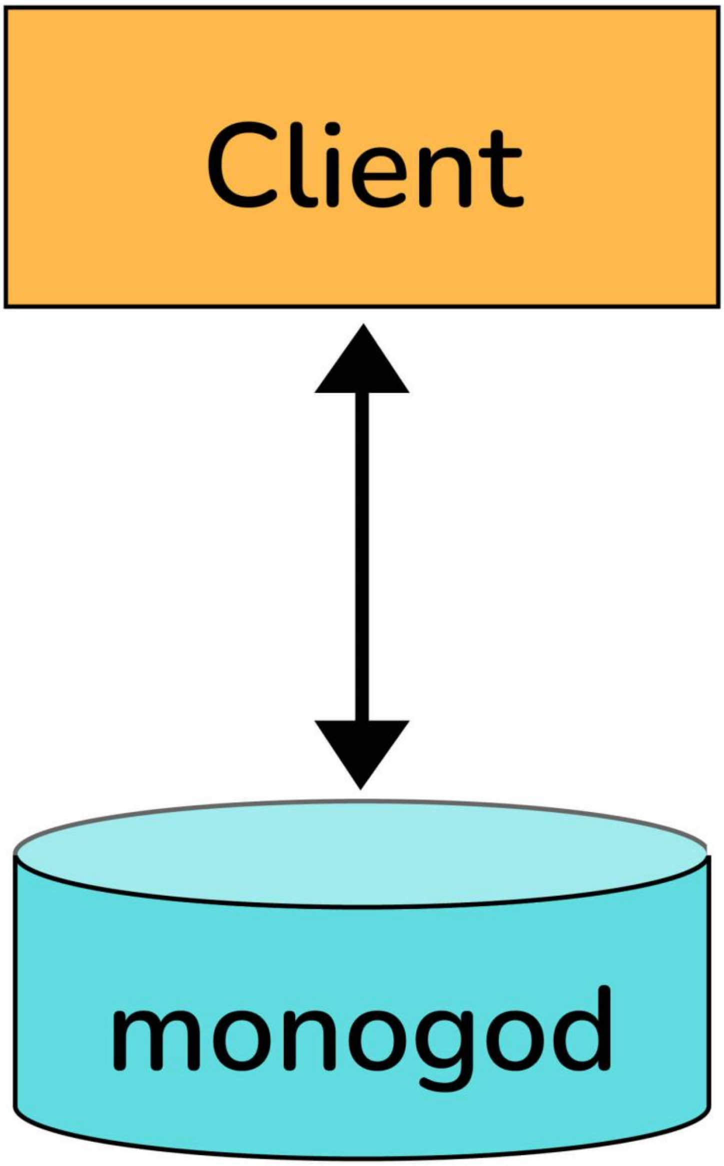
17. Explain the process of Sharding.

Sharding is the process of splitting data up across machines. We also use the term “partitioning” sometimes to describe this concept. We can store more data and handle more load without requiring larger or more powerful machines, by putting a subset of data on each machine.
In the figure below, RS0 and RS1 are shards. MongoDB’s sharding allows you to create a cluster of many machines (shards) and break up a collection across them, putting a subset of data on each shard. This allows your application to grow beyond the resource limits of a standalone server or replica set.



Sharded Client

Connection



Non Sharded Client Connection

18. Explain the SET Modifier in MongoDB?

If the value of a field does not yet exist, the "$set" sets the value. This can be useful for updating schemas or adding user-defined keys.

Example:

```
> db.users.findOne()
{
   "_id" : ObjectId("4b253b067525f35f94b60a31"),
   "name" : "alice",
   "age" : 23,
   "sex" : "female",
   "location" : "India"
}
```

To add a field to this, we use “$set”:

```
> db.users.updateOne({"_id" :
ObjectId("4b253b067525f35f94b60a31")},
... {"$set" : {"favorite book" : "Start with Why"}})
```

MongoDB Advanced Interview Questions

19. What do you mean by Transactions?

A transaction is a logical unit of processing in a database that includes one or more database operations, which can be read or write operations. Transactions provide a useful feature in MongoDB to ensure consistency.

MongoDB provides two APIs to use transactions.

- **Core API:** It is a similar syntax to relational databases (e.g., start_transaction and commit_transaction)
- **Call-back API:** This is the recommended approach to using transactions. It starts a transaction, executes the specified operations, and commits (or aborts on the error). It also automatically incorporates error handling logic for "TransientTransactionError" and"UnknownTransactionCommitResult".

20. What are MongoDB Charts?

MongoDB Charts is a new, integrated tool in MongoDB for data visualization.

MongoDB Charts offers the best way to create visualizations using data from a MongoDB database.
It allows users to perform quick data representation from a database without writing code in a programming language such as Java or Python.

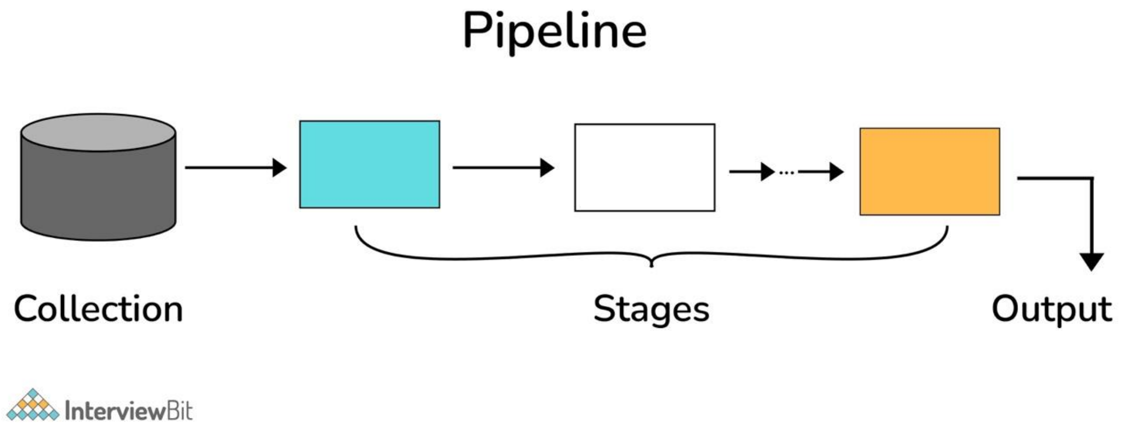
The two different implementations of MongoDB Charts are:

- MongoDB Charts PaaS (Platform as a Service)
- MongoDB Charts Server

21. What is the Aggregation Framework in MongoDB?

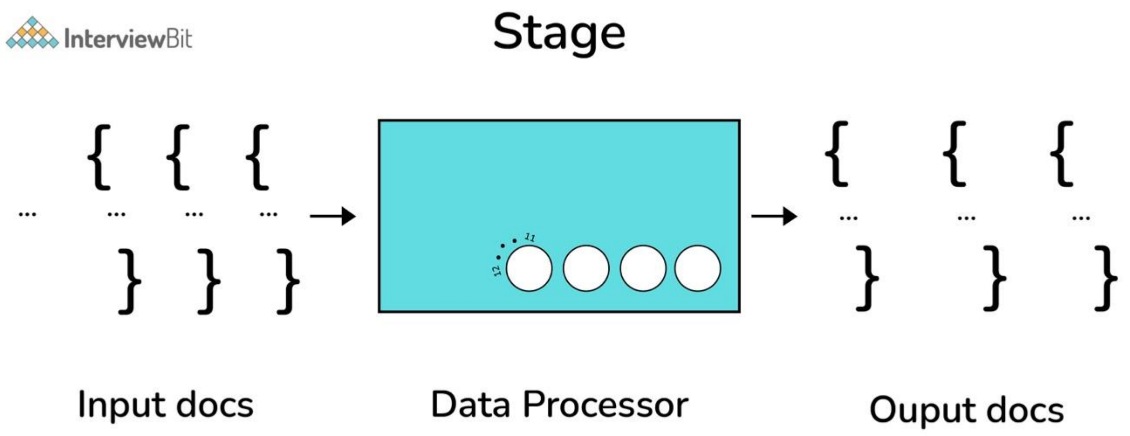
- The aggregation framework is a set of analytics tools within MongoDB that allow you to do analytics on documents in one or more collections.

- The aggregation framework is based on the concept of a pipeline. With an aggregation pipeline, we take input from a MongoDB collection and pass the documents from that collection through one or more stages, each of which performs a different operation on its inputs (See figure below). Each stage takes as input whatever the stage before it produced as output. The inputs and outputs for all stages are documents—a stream of documents.



22. Explain the concept of pipeline in the MongoDB aggregation framework.

An individual stage of an aggregation pipeline is a data processing unit. It takes in a stream of input documents one at a time, processes each document one at a time, and produces an output stream of documents one at a time (see figure below).



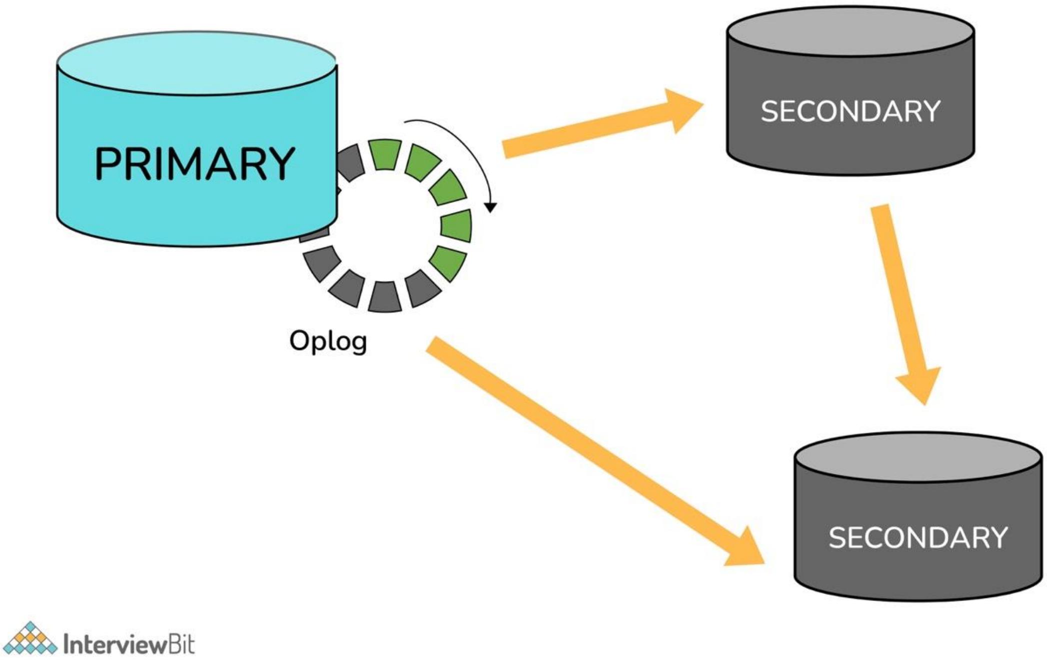
23. What is a Replica Set in MongoDB?

To keep identical copies of your data on multiple servers, we use replication. It is recommended for all production deployments. Use replication to keep your application running and your data safe, even if something happens to one or more of your servers.

Such replication can be created by a replica set with MongoDB. A replica set is a group of servers with one primary, the server taking writes, and multiple secondaries, servers that keep copies of the primary's data. If the primary crashes, the secondaries can elect a new primary from amongst themselves.

24. Explain the Replication Architecture in MongoDB.

The following diagram depicts the architecture diagram of a simple replica set cluster with only three server nodes – one primary node and two secondary nodes:



- In the preceding model, the PRIMARY database is the only active replica set member that receives write operations from database clients. The PRIMARY database saves data changes in the Oplog. Changes saved in the Oplog are sequential—that is, saved in the order that they are received and executed.
- The SECONDARY database is querying the PRIMARY database for new changes in the Oplog. If there are any changes, then Oplog entries are copied from PRIMARY to SECONDARY as soon as they are created on the PRIMARY node.
- Then, the SECONDARY database applies changes from the Oplog to its own datafiles. Oplog entries are applied in the same order they were inserted in the log. As a result, datafiles on SECONDARY are kept in sync with changes on PRIMARY.
- Usually, SECONDARY databases copy data changes directly from PRIMARY. Sometimes a SECONDARY database can replicate data from another SECONDARY. This type of replication is called Chained Replication because it is a two-step replication process. Chained replication is useful in certain replication topologies, and it is enabled by default in MongoDB.

25. What are some utilities for backup and restore in MongoDB?

The mongo shell does not include functions for exporting, importing, backup, or restore. However, MongoDB has created methods for accomplishing this, so that no scripting work or complex GUIs are needed. For this, several utility scripts are provided that can be used to get data in or out of the database in bulk. These utility scripts are:

- mongoimport
- mongoexport
- mongodump
- mongorestore

26. Conclusion

MongoDB is a powerful, flexible, and scalable general-purpose database. It combines the ability to scale out with features such as secondary indexes, range queries, sorting, aggregations, and geospatial indexes.
Thus, in conclusion, MongoDB is:

- Supports Indexing
- Designed to scale
- Rich with Features
- High Performance
- Load Balancing
- Supports sharding

Although MongoDB is powerful, incorporating many features from relational systems, it is not intended to do everything that a relational database does. For some functionality, the database server offloads processing and logic to the client-side (handled either by the drivers or by a user's application code). Its maintenance of this streamlined design is one of the reasons MongoDB can achieve such high performance.

Here are few References to understand MongoDB in-depth:

---

1. Compare MongoDB with Cassandra.

Learn about a brief comparison between MongoDB and Cassandra using this table.

Criteria	MongoDB	Cassandra
Data Model	Document	Bigtable like

Database scalability	Read	Write
Querying of data	Multi-indexed	Using Key or Scan

To have a detailed comparison between MongoDB and Cassandra, check out Cassandra Versus MongoDB!

2. What makes MongoDB the best?

MongoDB is considered to be the best NoSQL database because of its following features:

- Document-oriented (DO)
- High performance (HP)
- High availability (HA)
- Easy scalability
- Rich query language

3. How to do transactions/locking in MongoDB?

MongoDB does not use conventional locking with reduction as it is planned to be light, high-speed, and knowable in its presentation. It can be considered as parallel to the MySQL MyISAM auto entrust sculpt. With the simplest business sustain, performance is enhanced, particularly in a structure with numerous servers.

4. When and to what extent does data get extended to multi-slice?

MongoDB scrap stands on a collection. So, an album of all substances is kept in a lump or mass. Only when there is an additional time slot, there will be more than a few slice data achievement choices, but when there is more than one lump, data gets extended to a lot of slices and it can be extended to 64 MB.

5. Compare MongoDB with Couchbase and CouchbaseDB.

Although MongoDB, Couchbase, and Couchbase DB are common in many ways, still they are different in the case of necessities for the execution of the model, crossing points, storage, duplications, etc.

6. When do we use a namespace in MongoDB?

During the sequencing of the names of the database and the collection, the namespace is used.

7. If you remove an object attribute, is it deleted from the database?

Yes, it is deleted. Hence, it is better to eliminate the attribute and then save the object again.

Enroll now in Database Training Course to learn more.

8. How can we move an old file into the moveChunk directory?

Once the functions are done, the old files are converted to backup files and moved to the **moveChunk** directory at the time of balancing the slices.

9. Explain the situation when an index does not fit into RAM.

When an index is too huge to fit into RAM, then MongoDB reads the index, which is faster than reading RAM because the indexes easily fit into RAM if the server has got RAM for indexes, along with the remaining set.

10. How does MongoDB provide consistency?

MongoDB uses the **reader–writer locks**, allowing simultaneous readers to access any supply like a database or a collection but always offering private access to single writes.

11. Why is MongoDB not chosen for a 32-bit system?

Mongo DB is not considered as a 32-bit system because for running the 32-bit MongoDB, with the server, information and indexes require 2 GB. That is why it is not used in 32-bit devices.

12. How does Journaling work in MongoDB?

Write operations are saved in memory while journaling is going on. The on-disk journal files are really dependable for the reason that the journal writes are habitual. Inside **dbPath**, a journal subdirectory is designed by MongoDB.

13. How can you isolate the cursors from intervening with the write operations?

The **snapshot()** method is used to isolate the cursors from intervening with writes. This method negotiates the index and makes sure that each query comes to any article only once.

14. Define MongoDB.

It is a document-oriented database that is used for high availability, easy scalability, and high performance. It supports the dynamic schema design.

15. Explain the replica set.

It is a group of mongo instances that maintains the same dataset. Replica sets provide redundancy and high availability and are the basis for all production deployments.

Also, Checkout this interesting blog on Redis Vs MongoDB!

16. What are the key features of MongoDB?

There are three main features of MongoDB:

- Automatic scaling
- High performance

- High availability

17. What is CRUD?

MongoDB provides CRUD operations:

- **C**reate
- **R**ead
- **U**pdate
- **D**elete

Click here to learn more about [MongoDB CRUD Operations](#) in our MongoDB Tutorial!

18. What is Sharding?

In MongoDB, sharding means to store data on multiple machines.

19. What is Aggregation in MongoDB?

In MongoDB, aggregations are operations that process data records and return computed results.

20. Define Namespace in MongoDB.

It is the concatenation of the collection name and the name of the database.

Career Transition

Intermediate Interview Questions

21. Which syntax is used to create a Collection in MongoDB?

We can create a collection in MongoDB using the following syntax:

```
db.createCollection(name,options)
```

22. Which syntax is used to drop a Collection in MongoDB?

We can use the following syntax to drop a collection in MongoDB:

```
db.collection.drop()
```

23. Explain Replication.

Replication is the process of synchronizing data across multiple servers.

24. What is the use of an Index in MongoDB?

In MongoDB, indexes provide high-performance read operations for frequently used queries.

If you have any doubts or queries related to MongoDB, get them clarified from MongoDB Experts on our MongoDB Community!

25. Which command is used for inserting a document in MongoDB?

The following command is used for inserting a document in MongoDB:

```
database.collection.insert (document)
```

26. What is the use of GridFS in MongoDB?

GridFS is used for storing and retrieving large files, such as audio, image, and video files.

27. What is the use of Journaling in MongoDB?

Journaling is used for safe backups in MongoDB.

28. Which command is used to see a connection?

We can use the following command to see the connection:

```
db_adminCommand (“connPoolStats”)
```

29. Define the primary Replica set.

The primary replica set accepts all write operations from clients.

To have a detailed comparison between Firebase and MongoDB, check out Firebase Vs MongoDB!

30. Define the secondary Replica sets.

The secondaries replicate the primary replica set's oplog and apply the operations to their datasets such that the secondaries' datasets reflect the primary's dataset.

31. What is the use of Profiler?

Profiler is used to show the performance characteristics of every operation against the database.

32. What type of data is stored by MongoDB?

MongoDB stores data in the form of documents, which are JSON-like field and value pairs.

33. What is the purpose of Replication?

Replication provides redundancy, and it increases data availability.

34. What are Embedded documents?

Embedded documents capture relationships between data by storing related data in a single document structure.

35. Define the application-level Encryption.

The application-level encryption provides encryption on a per-field or per-document basis within the application layer.

36. What is Storage Encryption?

Storage encryption encrypts all MongoDB data on storage or on the operating system to ensure that only authorized processes can access the protected data.

37. Which method is used to create an index?

The createIndex() method is used to create an index.

38. What is Replica set oplog?

The oplog records all operations that modify the data in the replica set.

39. What is Vertical Scaling?

Vertical scaling adds more CPU and storage resources to increase capacity.

40. Define Horizontal Scaling.

Horizontal scaling divides the dataset and distributes data over multiple servers, or shards.

Advanced Interview Questions

41. What are the components of the Sharded cluster?

The sharded cluster has the following components:

- Shards
- Query routers
- Config servers

42. Which command is used to create a database?

To create a database, we can use the **Database_Name** command.

43. Which command is used to drop a database?

The **db.dropDatabse()** command is used to drop a database.

44. What is the use of the pretty() method?

The pretty() method is used to show the results in a formatted way.

45. Which method is used to remove a document from a collection?

The remove() method is used to remove a document from a collection.

46. Define MongoDB Projection.

Projection is used to select only the necessary data. It does not select the whole data of a document.

47. What is the use of the limit() method?

The limit() method is used to limit the records in the database.

48. What is the syntax of the limit() method?

The syntax of the limit() method is as follows:

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

49. What is the syntax of the sort() method?

In MongoDB, the following syntax is used for sorting documents:

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

50. Which command is used to create a backup of the database?

The mongodump command is used to create a backup of the database.

51. What is a Collection in MongoDB?

In MongoDB, a collection is a group of MongoDB documents.

52. What is the use of the db command?

The db command gives the name of the currently selected database.

53. Which method is used to update documents into a collection?

The update() and save() methods are used to update documents into a collection.

54. What is the syntax of the skip() method?

The syntax of the skip() methopd is as follows:

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

55. Which command is used to restore the backup?

The mongorestore command is used to restore the backup.

56. What is the use of the dot notation in MongoDB?

MongoDB uses the dot notation to access the elements of an array and the fields of an embedded document.

57. Define Auditing.

Auditing provides administrators with the ability to verify that the implemented security policies are controlling the activity in the system.

58. Define the Aggregation pipeline.

The aggregation pipeline is a framework for performing aggregation tasks. The pipeline is used to transform documents into aggregated results.

59. Define MapReduce.

MapReduce is a generic multi-phase data aggregation modality that is used for processing quantities of data.

60. What is Splitting in MongoDB?

Splitting is a background process that is used to keep chunks from growing too large.

61. Which language is used to write for MongoDB?

C++ is used for writing and implementing MongoDB.

62. In which format does MongoDB store data?

MongoDB uses collections to store data rather than tables.

63. What is the use of the save() method?

The save() method is used to replace the existing document with a new document.

64. What is MongoDB?

MongoDB (from humongous) is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format 'BSON'), making the integration of data in certain types of applications easier and faster. Released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is open-source.

MongoDB was first developed by the software company 10gen (now, MongoDB Inc.) in October 2007 as a component of a planned platform as a service product. Then, the company shifted to an open-source development model in 2009, with 10gen offering commercial support and other services. Since then, MongoDB has been adopted as backend software by a number of major websites and services, including Craigslist, eBay, Foursquare, SourceForge, Viacom, and the New York Times, among others. Currently, MongoDB is the most popular NoSQL database system.

For a better understanding of MongoDB, refer to this What is MongoDB? blog.

65. What is the use of MongoDB?

MongoDB is a relational database management system (RDBMS) replacement for web applications. So, when we have something close to RDBMS, MongoDB could be of good use.

It gives us the additional partition tolerance, which RDMBS doesn't offer, but it has problems with availability. Nonetheless, if we want more scalability, MongoDB would be the right choice for us. It's suitable for real-time analytics and high-speed logging, and it's highly scalable as well. Craigslist uses MongoDB for archived posts.

66. What do you understand by NoSQL databases? Is MongoDB a NoSQL database? Explain.

Presently, the Internet is loaded with big data, big users, and so on that are 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', and it is a type of database that can handle and sort all types of unstructured, messy, and complicated data. It is just a new way to think about databases.

Yes, MongoDB is a NoSQL database.

67. What type of a DBMS is MongoDB?

MongoDB is a document-oriented DBMS.

68. What is the difference between MongoDB and MySQL?

Although both MongoDB and MySQL are free and open-source databases, there is a lot of difference between them in terms of data representation, relationships, 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.

To have a detailed comparison between Firebase and MongoDB, check out MongoDB vs SQL!

69. What is the use of MongoDB?

- MongoDB is typically used as the primary data store for operational applications with real-time requirements (i.e., low latency, high availability, etc.). MongoDB is generally a good fit for 60–80 percent of the applications we build today. MongoDB is easy to operate and scale in the ways that are hard if not impossible with relational databases.
- MongoDB excels in many use cases where the relational databases aren't a good fit, like applications with unstructured, semi-structured, and polymorphic data, as well as those with large scalability requirements or multi-datacenter deployments.
- MongoDB may not be a good fit for some applications. For example, applications that require complex transactions (e.g., a double-entry bookkeeping system) and scan-oriented applications that access large subsets of the data mostly may not be a good fit for MongoDB. Also, MongoDB is not a drop-in replacement for legacy applications built around the relational data model and SQL.
- Some common use cases of MongoDB include mobile apps, product catalogs, real-time personalization, content management, and applications delivering a single view across multiple systems.

70. What kind of a database is MongoDB?

MongoDB is a document-oriented DBMS. We can think of it as MySQL but with JSON-like objects comprising the data model, rather than RDBMS tables. Significantly, MongoDB supports neither joins nor transactions. However, it features secondary indexes, an expressive query language, atomic writes on a per-document level, and fully-consistent reads. Operationally, MongoDB offers the master–slave replication with automated failover and built-in horizontal scaling via automated range-based partitioning.

To learn more about MongoDB, check out Intellipaat's [MongoDB Tutorial]!

71. Which language is MongoDB written in?

MongoDB is implemented in C++. However, drivers and client libraries are typically written in their own respective languages. Although, some drivers use C extensions for better performance.

Get a detailed comparison between Dynamodb and Mongodb. Read our blog on Dynamodb Vs Mongodb

72. What are the limitations of the 32-bit versions of MongoDB?

MongoDB uses memory-mapped files. When running a 32-bit build of MongoDB, the total storage size for the server, including data and indexes, is 2 GB. For this reason, we do not deploy MongoDB to production on 32-bit machines.

If we're running a 64-bit build of MongoDB, there's virtually no limit to the storage size. For production deployments, 64-bit builds and operating systems are strongly recommended.

73. While creating a schema in MongoDB, what are the points need to be taken into consideration?

While creating a schema in MongoDB, the points need to be taken care of are as follows:

- Design our schema according to the user requirements
- Combine objects into one document if we want to use them together; otherwise, separate them
- Do joins while on write, and not when it is on read
- For most frequent use cases, optimize the schema
- Do complex aggregation in the schema