INTERVIEW QUESTIONS

FOR

MONGODB

Website: www.analytixlabs.co.in

Email: info@analytixlabs.co.in

MONGODB INTERVIEW QUESTIONS

Q. Compare MongoDB with Cassandra.

Criteria	MongoDB	Cassandra
Data Model	Document	Bigtable like
Database scalability	Read	Write
Querying of data	Multi-indexed	Using Key or Scan

Q. 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

Q. 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.

Q. What are the key features of mongodb?

- High perform
- High availability
- Automatic & Easy scaling,
- It is a Schema less.
- Structure of a single object is clear.
- No complex joins

Q. What is meant by CRUD?

Create, read update delete

Q. What is sharing in MongoDB?

The process of storing the record in multiple Machine is known as sharing.

Q. Explain what is a replica set?

A replica set is the group of magodb instance which is host in same data set

Q. How replication works in MongoDB?

Replication is the process of synchronization across the multiple servers.

Q. While creating Schema in MongoDB what are the points need to be taken in consideration?

Below are the following point needs to be take care while creating the schema

The combine objects into one document if you use them together for most frequent use cases optimize your schema

Q. Mention what is Object Id composed of?

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

Q. Does MongoDB need a lot of RAM?

It can run on small RAM because it is dynamic space allocated and re-allocated.

Q. What are the data types MongoDB supports?

String, Integer, Boolean, Double, Array, Timestamp, Date, Regular expression

Q. What is replication and why we need?

It is the process of synchronizing data across multiple servers. It provides redundancy and increases data availability with multiple copies of data on different database server.

WHY: To keep data safe, High availability of data, disaster recovery, and No downtime for maintenance.

Q. What is an embedded document?

Embedded documents are received while they were small, given the relations between the related data has been written in one body.

Q. What is the encryption storage?

MongoDB encryption encrypts data stored on any storage or processing by operating system may not be able to access protected data.

Q. How can we separate cursors connection with the write injustice?

Snapshot () method is used to separate them from coming in with cursors writes. Here is a list of query and make sure everyone comes to be, and the manner of his mother tongue, there has been only to look at the article.

Q. Define MongoDB.

It may be a document is to be oriented database of speech, with which the top of their minds to ease of scalability and easy to get high performance. The Scheme supports dynamic policy.

Q. Explain the image set.

I have to put the example of which is the group made up of Brunfelsia same knowledge of the set. To kill, and I need to provide high availability redundancy is the image of, and in the production of all of the deployments.

Q. What is the sum of the MongoDB?

What matters are aggregations process data records and return the results of counting.

Q. To define the nominal distance mongodb?

And what is the name of the collection database.

Q. Syntax is used to create a collection mongodb?

db.createCollection (name, options) is used to create a collection mongodb.

Syntax collection.

Q. That is to drop the mongodb?

db.collection.drop () is a collection to drop the mongodb.

Q. Explain Infection.

Infection by more synchronous processing of data servers.

Q. What is the use of index mongodb?

You can read a thing often used to provide a guideline for the high performance of queries.

Q. The government should be inserted here because it is used?

database.collection.insert (document) is used for inserting the document.

Q. What is the use of GridFS mongodb?

GridFS is used for storage and for the most retrieving files, images, video files.

Q. That the government used to bond?

db_adminCommand (“connPoolStats”); See the connection is not used.

Q. Define the image killing?

The first things to write replicas but embraced the culture.

Q. 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.

Q. 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.

Q. Compare MongoDB with Couchbase and CouchbaseDB.

Although MongoDB 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.

Q. When do we use a namespace in MongoDB?

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

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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)
```

Q. 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()
```

Q. Explain Replication.

Replication is the process of synchronizing data across multiple servers.

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

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

Q. 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)
```

Q. Which command is used to see a connection?

We can use the following command to see the connection:

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

Q. Define the primary Replica set.

The primary replica set accepts all write operations from clients.

Q. 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.

Q. What is the use of Profiler?

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

Q. What type of data is stored by MongoDB?

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

Q. What is the purpose of Replication?

Replication provides redundancy, and it increases data availability.

Q. Define the application-level Encryption.

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

Q. 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.

Q. Which method is used to create an index?

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

Q. What is Replica set oplog?

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

Q. What is Vertical Scaling?

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

Q. Define Horizontal Scaling.

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

Q. What are the components of the Sharded cluster?

The sharded cluster has the following components:

- Shards
- Query routers
- Config servers

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

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

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

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

Q. Define MongoDB Projection.

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

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

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

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

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

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

Q. 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})
```

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

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

Q. What is a Collection in MongoDB?

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

Q. What is the use of the db command?

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

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

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

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

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

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

Q. Which command is used to restore the backup?

The mongorestore command is used to restore the backup.

Q. 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.

Q. Define Auditing.

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

Q. Define the Aggregation pipeline.

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

Q. What is Splitting in MongoDB?

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

Q. Which language is used for developing MongoDB?

C++ is used for writing and implementing MongoDB.

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

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

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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.

Q. 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

Q. What are NoSQL databases? What are the different types of NoSQL databases?

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases (like SQL, Oracle, etc.).

Types of NoSQL databases:

- Document Oriented
- Key Value
- Graph
- Column Oriented

Q. What kind of NoSQL database MongoDB is?

MongoDB is a document oriented database. It stores data in the form of BSON structure based documents. These documents are stored in a collection.

Which are the most important features of MongoDB?

- Flexible data model in form of documents
- Agile and highly scalable database
- Faster than traditional databases
- Expressive query language

Q. Which all languages can be used with MongoDB?

Currently, MonggoDB provides official driver support for C, C++, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala, Go and Erlang. MongoDB can easily be used with any of these languages. There are some other community supported drivers too but the above mentioned ones are officially provided by MongoDB.

Q. Compare SQL databases and MongoDB at a high level.

SQL databases store data in form of tables, rows, columns and records. This data is stored in a pre-defined data model which is not very much flexible for today's real-world highly growing applications. MongoDB in contrast uses a flexible structure which can be easily modified and extended.

Q. How is MongoDB better than other SQL databases?

MongoDB allows a highly flexible and scalable document structure. For e.g. one data document in MongoDB can have five columns and the other one in the same collection can have ten columns. Also, MongoDB database are faster as compared to SQL databases due to efficient indexing and storage techniques.

Q. Compare MongoDB and CouchDB at high level.

Although both of these databases are document oriented, MongoDB is a better choice for applications which need dynamic queries and good performance on a very big database. On the other side, CouchDB is better used for applications with occasionally changing queries and pre-defined queries.

Q. Does MongoDB support foreign key constraints?

No. MongoDB does not support such relationships.

Q. Does MongoDB support ACID transaction management and locking functionalities?

No. MongoDB does not support default multi-document ACID transactions. However, MongoDB provides atomic operation on a single document.

Q. How can you achieve primary key - foreign key relationships in MongoDB?

By default MongoDB does not support such primary key - foreign key relationships. However, we can achieve this concept by embedding one document inside another. Foe e.g. an address document can be embedded inside customer document.

Q. Does MongoDB pushes the writes to disk immediately or lazily?

MongoDB pushes the data to disk lazily. It updates the immediately written to the journal but writing the data from journal to disk happens lazily.

Q. MongoDB uses BSON to represent document structures. True or False?

True

Q. If you remove a document from database, does MongoDB remove it from disk?

Yes. Removing a document from database removes it from disk too.

Q. Mention the command to insert a document in a database called school and collection called persons.

```
use school;  db.persons.insert( { name: "kadhir", dept: "CSE" } )
```

Q. What are Indexes in MongoDB?

Indexes support the efficient execution of queries in MongoDB. 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.

Q. How many indexes does MongoDB create by default for a new collection?

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

Q. Can you create an index on an array field in MongoDB? If yes, what happens in this case?

Yes. An array field can be indexed in MongoDB. In this case, MongoDB would index each value of the array.

Q. What is a covered query in MongoDB?

A covered query is the one in which:

- fields used in the query are part of an index used in the query, and

- the fields returned in the results are in the same index

Q. Why is a covered query important?

Since all the fields are covered in the index itself, MongoDB can match the query condition as well as return the result fields using the same index without looking inside the documents. Since indexes are stored in RAM or sequentially located on disk, such access is a lot faster.

Q. Does MongoDB provide a facility to do text searches? How?

Yes. MongoDB supports creating text indexes to support text search inside string content. This was a new feature which can introduced in version 2.6.

Q. Mention the command to list all the indexes on a particular collection.

```
db.collection.getIndexes()
```

Q. At what interval does MongoDB write updates to the disk?

By default configuration, MongoDB writes updates to the disk every 60 seconds. However, this is configurable with the commitIntervalMs and syncPeriodSecs options.

Q. How can you achieve transaction and locking in MongoDB?

To achieve concepts of transaction and locking in MongoDB, we can use the nesting of documents, also called embedded documents. MongoDB supports atomic operations within a single document.

Q. What is Aggregation in MongoDB?

Aggregations operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result. MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single purpose aggregation methods and commands.

Q. What is Sharding in MongoDB? Explain.

Sharding is a method for storing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

Q. What is Replication in MongoDB? Explain.

Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions.

Q. What are Primary and Secondary Replica sets?

Primary and master nodes are the nodes that can accept writes. MongoDB's replication is 'single-master:' only one node can accept write operations at a time.

Secondary and slave nodes are read-only nodes that replicate from the primary.

Q. By default, MongoDB writes and reads data from both primary and secondary replica sets. True or False.

False. MongoDB writes data only to the primary replica set.

Q. When should we embed one document within another in MongoDB?

You should consider embedding documents for:

- 'contains' relationships between entities
- One-to-many relationships
- Performance reasons

Q. Why MongoDB is not preferred over a 32-bit system?

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

If you're running a 64-bit build of MongoDB, there's virtually no limit to storage size.

Q. What is a Storage Engine in MongoDB

A storage engine is the part of a database that is responsible for managing how data is stored on disk. For example, one storage engine might offer better performance for read-heavy workloads, and another might support a higher-throughput for write operations.

Q. Which are the two storage engines used by MongoDB?

MongoDB uses MMAPv1 and WiredTiger.

Q. What is the role of a profiler in MongoDB? Where does the writes all the data?

The database profiler collects fine grained data about MongoDB write operations, cursors, database commands on a running mongod instance. You can enable profiling on a per-database or per-instance basis.

The database profiler writes all the data it collects to the system.profile collection, which is a capped collection.

Q. How does Journaling work in MongoDB?

When running with journaling, MongoDB stores and applies write operations in memory and in the on-disk journal before the changes are present in the data files on disk. Writes to the journal are atomic, ensuring the consistency of the on-disk journal files. With journaling enabled, MongoDB creates a journal subdirectory within the directory defined by dbPath, which is /data/db by default.

Mention the command to check whether you are on the master server or not.

db.isMaster()

Q. Can you configure the cache size for MMAPv1? How?

No. MMAPv1 does not allow configuring the cache size.

Q. Can you configure the cache size for WiredTiger? How?

For the WiredTiger storage engine, you can specify the maximum size of the cache that WiredTiger will use for all data. This can be done using storage.wiredTiger.engineConfig.cacheSizeGB option.

Q. How does MongoDB provide concurrency?

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

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

You can use the snapshot() method on a cursor to isolate the operation for a very specific case. snapshot() traverses the index on the _id field and guarantees that the query will return each document no more than once.

Q. Can one MongoDB operation lock more than one databases? If yes, how?

Yes. Operations like copyDatabase(), repairDatabase(), etc. can lock more than onne databases involved.

Q. How can concurrency affect replica sets primary?

In replication, when MongoDB writes to a collection on the primary, MongoDB also writes to the primary's oplog, which is a special collection in the local database. Therefore, MongoDB must lock both the collection's database and the local database.

Q. Can you run multiple Javascript operations in a single mongod instance?

Yes. The V8 JavaScript engine added in 2.4 allows multiple JavaScript operations to run at the same time.

Q. Which command can be used to provide various information on the query plans used by a MongoDB query?

The explain() command can be used for this information. The possible modes are: 'queryPlanner', 'executionStats', and 'allPlansExecution'.

Q. What are the best features of MongoDB?

MongoDB Features

Feature	Description
Indexing	It indexes are created in order to improve the search performance.
Replication	MongoDB distributes the data across different machines.
Ad-hoc Queries	It supports ad-hoc queries by indexing the BSON documents & using a unique query language.
Schemaless	It is very flexible because of its schema-less database that is written in C++.
Sharding	MongoDB uses sharding to enable deployments with very large data sets and high throughput operations.

Q. Can journaling feature be used to perform safe hot backups?

Yes!

Q. Will there be journal replay programs in case of incomplete entries (if there is a failure in the middle of one)?

Each journal (group) write is consistent and won't be replayed during recovery unless it is complete.

Q. What is the role of profiler in MongoDB?

MongoDB includes a database profiler which shows performance characteristics of each operation against the database. With this profiler you can find queries (and write operations) which are slower than they should be and use this information for determining when an index is needed.

Q. When an object attribute is removed, is it deleted from the store?

Yes, you can remove the attribute and then re-save() the object.

Q. Are null values allowed?

Yes, but only for the members of an object. A null cannot be added to the database collection as it isn't an object. But {}can be added.

Q. Does an update fsync to disk immediately?

No. Writes to disk are lazy by default. A write may only hit the disk a couple of seconds later. For example, if the database receives thousand increments to an object within one second, it will only be flushed to disk once. *(Note: fsync options are available both at the command line and via getLastError_old.)*

Q. How long does replica set failover take?

It may take 10-30 seconds for the primary to be declared down by the other members and a new primary to be elected. During this window of time, the cluster is down for primary operations i.e writes and strong consistent reads. However, eventually consistent queries may be executed to secondaries at any time (in slaveOk mode), including during this window.

Q. What's a Master or Primary?

This is a node/member which is currently the primary and processes all writes for the replica set. During a failover event in a replica set, a different member can become primary.

Q. What's a Secondary or Slave?

A secondary is a node/member which applies operations from the current primary. This is done by tailing the replication oplog (local.oplog.rs). Replication from primary to secondary is asynchronous, however, the secondary will try to stay as close to current as possible (often this is just a few milliseconds on a LAN).

Q. Is it required to call 'getLastError' to make a write durable?

No. If 'getLastError' (aka 'Safe Mode') is not called, the server does exactly behave the way as if it has been called. The 'getLastError' call simply allows one to get a confirmation that the write operation was successfully committed. Of course, often you will want that confirmation, but the safety of the write and its durability is independent.

Q. Should you start out with Sharded or with a Non-Sharded MongoDB environment?

We suggest starting with Non-Sharded for simplicity and quick startup, unless your initial data set will not fit on single servers. Upgrading to Sharded from Non-sharded is easy and seamless, so there is not a lot of advantage in setting up Sharding before your data set is large.

Q. How does Sharding work with replication?

Each Shard is a logical collection of partitioned data. The shard could consist of a single server or a cluster of replicas. Using a replica set for each Shard is highly recommended.

Q. When will data be on more than one Shard?

MongoDB Sharding is range-based. So all the objects in a collection lie into a chunk. Only when there is more than 1 chunk there is an option for multiple Shards to get data. Right now, the default chunk size is 64mb, so you need at least 64mb for migration.

Q. What happens when a document is updated on a chunk that is being migrated?

The update will go through immediately on the old Shard and then the change will be replicated to the new Shard before ownership transfers.

Q. What happens when a Shard is down or slow when querying?

If a Shard is down, the query will return an error unless the 'Partial' query options is set. If a shard is responding slowly, Mongos will wait for it.

Q. Can the old files in the 'moveChunk' directory be removed?

Yes, these files are made as backups during normal Shard balancing operations. Once the operations are done then they can be deleted. The clean-up process is currently manual so this needs to be taken care of to free up space.

Q. How do you see the connections used by Mongos?

The following command needs to be used: db._adminCommand("connPoolStats");

Q. If a 'moveChunk' fails, is it necessary to cleanup the partially moved docs?

No, chunk moves are consistent and deterministic. The move will retry and when completed, the data will be only on the new Shard.

Q. What are the disadvantages of MongoDB?

- A 32-bit edition has 2GB data limit. After that it will corrupt the entire DB, including the existing data. A 64-bit edition won't suffer from this bug/feature.
- Default installation of MongoDB has asynchronous and batch commits turned on. Meaning, it lies when asked to store something in DB and commits all changes in a batch at a later time in future. If there is a server crash or power failure, all those commits buffered in memory will be lost. This functionality can be disabled, but then it will perform as good as or worse than MySQL.
- MongoDB is only ideal for implementing things like analytics/caching where impact of small data loss is negligible.
- In MongoDB, it's difficult to represent relationships between data so you end up doing that manually by creating another table to represent the relationship between rows in two or more tables.

Q. Does MongoDB support primary-key, foreign-key relationship?

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

Q. Is it true that MongoDB uses BSON to represent document structure?

Yes.

Q. By default, which index is created by MongoDB for every collection?

By default, the_id collection is created for every collection by MongoDB.

Q. Can journaling features be used to perform safe hot backups?

Yes.

Q. Does MongoDB database have tables for storing records?

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

Q. Do the MongoDB databases have schema?

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

Q. 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.

Q. What will have to do if a shard is down or slow and you do a query?

If a shard is down and you even do query then your query will be returned with an error unless you set a partial query option. But if a shard is slow them Mongos will wait for them till response.

Q. Explain the covered query in MongoDB.

A query is called covered query if satisfies the following two conditions:

- The fields used in the query are part of an index used in the query.
- The fields returned in the results are in the same index.

Q. What is the importance of covered query?

Covered query makes the execution of the query faster because indexes are stored in RAM or sequentially located on disk. It makes the execution of the query faster.

Covered query makes the fields are covered in the index itself, MongoDB can match the query condition as well as return the result fields using the same index without looking inside the documents.

Q. By default, which replica sets are used to write data?

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

Q. 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.

Q. What is the difference between MongoDB and Redis database?

Difference between MongoDB and Redis:

- Redis is faster than MongoDB.
- Redis has a key-value storage whereas MongoDB has a document type storage.
- Redis is hard to code but MongoDB is easy.

Q. What is the difference between MongoDB and CouchDB?

Difference between MongoDB and CouchDB:

- MongoDB is faster than CouchDB while CouchDB is safer than MongoDB.
- Triggers are not available in MongoDB while triggers are available in CouchDB.
- MongoDB serializes JSON data to BSON while CouchDB doesn't store data in JSON format.

Q. What is the difference between MongoDB and Cassandra?

Difference between MongoDB and Cassandra:

- MongoDB is cross-platform document-oriented database system while Cassandra is high performance distributed database system.
- MongoDB is written in C++ while Cassandra is written in Java.
- MongoDB is easy to administer in the case of failure while Cassandra provides high availability with no single point of failure.

Q. Is there any need to create database command in MongoDB?

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

MongoDB Vs ElasticSearch

Name	MongoDB	Elastic Search
Database Models	Database and Key-value Storage	Search Analytics
Latest Version	3.6.3	6.2.3
Developed By	MongoDB Inc	ElasticSearch
Implementation LanguageLanguage Used	C++	Java
APIs and Access Methods	JSON Prorocol	Java API
Extra features	MapReduce and inbuilt memory capability	Triggers and map reducing

Q. In MongoDB, which command can be used to provide all information of a query plan?

The explain() command is used to provide information of all the query plans. The possible models are as follows:

- 'queryPlanner',
- 'executionStats'
- 'allPlansExecution'.

Q. What feature in MongoDB is used to do safe backups?

To save backups of the old files “Journaling” feature is used in MongoDB databases.