MONGODB Q&A

Q1. What is MongoDB and Why we need MongoDB?

MongoDB is the most famous NoSQL open source database management system. It is written in C++ language and developed by MongoDB Inc.
MongoDB is a document oriented database which means it stores the data in BSON format which is a binary representation of JSON and it contains more data types than JSON.

In MongoDB, tables are called as collection, rows are called as document, columns are called as field, joins are called as linking.
MongoDB is widely supported by most of the languages like, JavaScript, Java, C# and runs on all the available OS.

Q2. What is difference between MongoDB and SQL or NoSQL between SQL?

In SQL, data is stored in Table format whereas in MongoDB, it stores in JSON format.
In MongoDB, development is simplified as its documents map naturally to modern, object-oriented programming languages. In SQL, we need object-relational mapping (ORM) layer that translates objects in code to relational tables.
The MongoDB provides high performance, high availability, easy scalability rather than SQL Server.
In MongoDB, each document can store data with different attributes from other documents. With JSON documents, we can add new attributes when we need to, without having to alter a centralized database schema. But in a relational database, this causes downtime and significant performance overhead
Having all the data for an object in one place also makes it easier for developers to understand and optimize query performance.

Q3. What is BSON in MongoDB?

MongoDB stores data as BSON documents. BSON is a binary representation of JSON documents, though it contains more data types than JSON. Some of the supported data types are: Double, String, Object, Array, Binary Data, RegEx, Boolean, Date, Null, JavaScript, Timestamp. ObjectId.
Some of the features of BSON:
BSON is lightweight and is an important feature for any data representation format, especially when used over the network.
BSON is designed to be traversed easily.
It is efficient. Encoding data to BSON and decoding from BSON can be performed very quickly in most languages

Q4. What is _id Field in MongoDB?

In MongoDB, each document stored in a collection requires a unique _id field that acts as a primary key. If an inserted document omits the _id field, the MongoDB driver automatically generates an ObjectId for the _id field.
The _id field has the following behavior and constraints:
By default, MongoDB creates a unique index on the _id field during the creation of a collection.
The _id field is always the first field in the documents. If the server receives a document that does not have the _id field first, then the server will move the field to the beginning.
The _id field may contain values of any BSON data type, other than an array.
While storing values for _id, we can follow these options: Use an ObjectId, Use a natural unique identifier, Generate an auto-incrementing number, Generate a UUID in our application code.

Q5. How do you create or select a database in MongoDB?

In MongoDB, databases hold collections of documents. To select a database to use issue the use <db> statement, as in the following example:

If a database does not exist, MongoDB creates the database when we first store data for that database. Some points to be noted when creating database in MongoDB are:
Database Name Case Sensitivity
For MongoDB deployments running on Windows, database names cannot contain any of the following characters: /\. "$*<>:|?
For MongoDB deployments running on Unix and Linux systems, database names cannot contain any of the following characters: /\. "$
Database names cannot be empty and must have fewer than 64 characters.

Q6. What is collection in MongoDB?

A collection is a group of documents. If a document is the MongoDB analog of a row in a relational database, then collection is the analog to a table.
Collections have dynamic schemas. This means that the documents within a single collection can have any number of different shapes. For example, both the following documents can be stored in a single collection.

Note that the previous documents not only have different types for their values (string versus integer) but also have entirely different keys. Because in MongoDB any document can be put into any collection.

Q7. What is writeConcern?

Whenever we query the database to insert(), insertOne() or insertMany(), we can pass an additional parameter which is writeConcern. It is a document describes the level of acknowledgement requested from MongoDB for write operations. Write concern can include the following fields:

the w option to request acknowledgement that the write operation has propagated to a specified number of mongod instances or to mongod instances with specified tags.
the j option to request acknowledgement that the write operation has been written to the journal, and
the wtimeout option to specify a time limit to prevent write operations from blocking indefinitely.

Q8. What is upsert?

upsert is one of the option that we can pass to the update operation as third parameter.
If upsert is true and no document matches the query criteria, update() inserts a single document. The update creates the new document with either:
The fields and values of the update parameter if the update parameter is a replacement document (i.e., contains only field and value pairs). If neither the query nor the update document specifies an _id field, MongoDB adds the _id field with an ObjectId value.
The fields and values of both the query and update parameters if the update parameter contains update operator expressions (such as $inc, $set, $min, $max). The update creates a base document from the equality clauses in the parameter, and then applies the update expressions from the updateparameter. Comparison operations from the query will not be included in the new document.
Also if we want to update multiple documents we can pass multi: true as another option.

Q9. Explain some of the Evaluation Query Operators.

$expr: $expr can build query expressions that compare fields from the same document in a $match stage. Example: db.users.find({ $expr: { $gt: ["$age", "$requiredAge"] } }). This query will finds documents where the age is greater than the requiredAge.
$jsonSchema: $jsonSchema can be used in a document validator, which enforces that inserted or updated documents are valid against the schema.
$mod: Select documents where the value of a field divided by a divisor has the specified remainder. It is performing a modulo operation. Example: db.users.find({ items: { $mod: [4, 0] } })
$regex: Provides regular expression capabilities for pattern matching strings in queries. MongoDB uses Perl compatible regular expressions. Example: db.users.find({ name: { $regex: /^John/ } })
$text: $text performs a text search on the content of the fields indexed with a text index. Example: db.users.find({ $text: { $search: "john" } })
$where: Use the $where operator to pass either a string containing a JavaScript expression or a full JavaScript function to the query system.

Q10. What is Storage Engine? What are all the Storage Engines available in MongoDB?

The storage engine is the component of the database that is responsible for managing how data is stored, both in memory and on disk. MongoDB supports multiple storage engines, as different engines perform better for specific workloads.
WiredTiger Storage Engine: This is the default engine. It is well-suited for most workloads and is recommended for new deployments. WiredTiger provides a document-level concurrency model, checkpointing, and compression, among other features.
In-Memory Storage Engine: In-Memory Storage Engine is available in MongoDB Enterprise. Rather than storing documents on-disk, it retains them in-memory for more predictable data latencies.
MMAPv1 Storage Engine: This is deprecated as of MongoDB v4.0 and is the default storage engine for MongoDB versions 3.0 and earlier.

Q11. Define a replica set?

The group of instances which host similar data set is known as a replica set. Two nodes are present in a replica set, one is secondary and the other is primary, where data is replicated from primary and sent to the secondary node.

Q12. What is the role of profiler in MongoDB?

The role of a MongoDB profiler is to show the performance and analyze the characteristics of every operation of the database. By using the profiler, you will find all the queries which are slower than usual.

Q13. What are the key features of MongoDB?

The following are the core features of MongoDB:
High performance
Automatic scaling
Rich query language
High availability

Q14. What is the advantage of MongoDB?

The following are a few advantages of MongoDB database:
Schemaless
Easy to scale-out
No complex joins
Structure of a single object is clear

Q15. What is an embedded document?

Embedded documents specify the relationship between data that is written inside the body. The documents are received while the related data body is small.

Q16.What is an embedded document?

Embedded documents specify the relationship between data that is written inside the body. The documents are received while the related data body is small.

Q16. In MongoDB what is ObjecId composed of?

The ObjectId is composed of the following parameters:
Client machine ID
Timestamp
3 byte incremented counter
Client process ID

Q17. What does sharding mean in MongoDB?

Sharding is a process of storing data records among one or more machines applied by MongoDB to meet the demands of data growth. It forms a horizontal partition in the database and each partition is known as database shard or a shard.

Q18. What is CRUD?

Mongodb offers best CRUD operations to perform better database operations. The following are the operations:
Create
Read
Update
Delete

Q19. what command is used to create a MongoDB collection?

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

Q20. What feature in MongoDB is used to do safe backups?

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

Q21. What's a good way to get a list of all unique tags for a collection of documents millions of items large when we doesn't have access to mongodb's new "distinct" command ?

The normal way of doing tagging seems to be indexing multikeys. Even if your MongoDB driver doesn't implement distinct, we can implement.
In JavaScript you can write something like this:
1
result = db.$cmd.findOne({"distinct" : "collection_name", "key" : "tags"})
You do a findOne on the "$cmd" collection of whatever database you're using. Pass it the collection name and the key you want to run distinct on.

Q22. Comparison between MongoDB and Cassandra.

MongoDB:

Data mode are the document
Database scalability is read-only
Query of the data is multi-index

Cassandra:

Data mode are a big table like
Database scalability is write only
Query of the data by using scan or key

Q23. Explain about replication and when should we use it?

Replication is the process of synchronizing data across multiple servers so that data is not lost in any condition. It gives redundancy and rises data availability with many copies of data on other database servers. We can use it for various purposes like for keeping the data safe, for high availability of data, disaster recovery and in no downtime for maintenance.

Q 24. How can MongoDB simulate join or subquery?

We are trying to figure out the best way to structure data in Mongo to simulate what would be a simple join or subquery in SQL.

Say we have the classic Users and Posts example, with Users in one collection and Posts in another. We want to find all posts by users who's city is "Mumbai".

We have simplified things in this question, in real-world scenario storing posts as an array in the user document won't work as we have 1,000's of "posts" per user constantly inserting.

Q25. What is a Namespace in MongoDB?

A Namespace is the concatenation of the database name and collection name. For e.g. school.students with school as the database and students as the collection.

Q26.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

Q27.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.

Q28. 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.

Q29. 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.

Q30. What is GridFS?

GridFS is a specification for storing and retrieving files that exceed the BSON-document size limit of 16MB. Instead of storing a file in a single document, GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.