



MongoDB Interview Questions

Advertisements



⬅ Previous Page

Next Page ➡

Dear readers, these **MongoDB Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **MongoDB**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer:

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

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

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

Which all languages can be used with MongoDB? ▼

Currently, MongoDB 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.

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.

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.

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.

Does MongoDB support foreign key constraints? ▼

No. MongoDB does not support such relationships.

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.

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. For e.g. an address document can be embedded inside customer document.

Does MongoDB need a lot of RAM? ▼

No. MongoDB can be run even on a small amount of RAM. MongoDB dynamically allocates and de-allocates RAM based on the requirements of other processes.

Does MongoDB push 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.

Explain the structure of ObjectID in MongoDB. ▼

ObjectID is a 12-byte BSON type with:

- 4 bytes value representing seconds
- 3 byte machine identifier
- 2 byte process id
- 3 byte counter

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

True

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

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

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.

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

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

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.

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

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.

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.

What happens if an index does not fit into RAM? ▼

If the indexes do not fit into RAM, MongoDB reads data from disk which is relatively very much slower than reading from RAM.

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

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

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.

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.

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.

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.

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.

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.

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.

Why are MongoDB data files large in size? ▼

MongoDB preallocates data files to reserve space and avoid file system fragmentation when you setup the server.

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

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.

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.

Which are the two storage engines used by MongoDB? ▼

MongoDB uses MMAPv1 and WiredTiger.

What is the role of a profiler in MongoDB? Where does it write 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.

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()

Can you configure the cache size for MMAPv1? How? ▼

No. MMAPv1 does not allow configuring the cache size.

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.

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.

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.

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

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

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.

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.

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.

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'.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

⬅ Previous Page

Next Page ➡

Advertisements

Cloud Computing Explained

Cloud computing is not that complicated. Here's why
Yahoo Search



[Privacy Policy](#) [Cookies Policy](#) [Contact](#)

© Copyright 2019. All Rights Reserved.