## MongoDB

### What are NoSQL? 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, MongoDB is type of this.
* Key Value
* Graph
* Column Oriented

### Key features of MongoDB, compare with other SQL databases.

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

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.

MongoDB does not support such relationships like foreign key constraints. By embedding one document inside another, we can achieve this concept.

MongoDB does not support default multi-document ACID transactions. However, MongoDB provides atomic operation on a single document.

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. Compare to Redis, Redis support the memory mode which keep all data in memory.

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.

### 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.

### What's the MongoDB version you have used?

Current stable version is MongoDB 3.0

* 2015-12-08, V3.2.0
* 2015-04-28, V2.4.14
* 2014-01-16, V2.2.7

### Does MongoDB support SQL?

No.

MongoDB does support a rich, ad-hoc query language of its own.

### What are typical uses for MongoDB?

MongoDB has a general-purpose design, making it appropriate for a large number of use cases. Examples include content management systems, mobile applications, gaming, e-commerce, analytics, archiving, and logging.

Do not use MongoDB for systems that require SQL, joins, and multi-object transactions.

### Does MongoDB support transactions?

MongoDB does not support multi-document transactions. However, MongoDB does provide atomic operations on a single document.

### How do I configure the cache size?

MongoDB has no configurable cache for the [MMAPv1 storage engine](https://docs.mongodb.org/v3.0/core/mmapv1/). MongoDB uses all free memory on the system automatically by way of memory-mapped files. Operating systems use the same approach with their file system caches.

### Does MongoDB handle caching?

Yes. MongoDB keeps all of the most recently used data in RAM. If you have created indexes for your queries and your working data set fits in RAM, MongoDB serves all queries from memory.

MongoDB does not implement a query cache; i.e. MongoDB does not cache the query results in order to return the cached results for identical queries.

### Are writes written to disk immediately or lazily?

In the default configuration for the [MMAPv1 storage engine](https://docs.mongodb.org/v3.0/core/mmapv1/), MongoDB writes to the data files on disk every 60 seconds and writes to the [journal](https://docs.mongodb.org/v3.0/reference/glossary/#term-journal) files roughly every 100 milliseconds.

To change the interval for writing to the data files, use the [storage.syncPeriodSecs](https://docs.mongodb.org/v3.0/reference/configuration-options/#storage.syncPeriodSecs) setting. For the journal files, see storage.journal.commitIntervalMs setting.

### What language is MongoDB written in?

MongoDB is implemented in C++. [Drivers](https://docs.mongodb.org/v3.0/reference/glossary/#term-driver) and client libraries are typically written in their respective languages, although some drivers use C extensions for better performance.

### What is Namespace/database/collection name 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

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

True

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

True

### When does MongoDB write updates to disk?

MongoDB flushes writes to disk on a regular interval.

### How do you aggregate data with MongoDB?

In version 2.1 and later, you can use the new [aggregation framework](https://docs.mongodb.org/v3.0/core/aggregation/), with the [aggregate](https://docs.mongodb.org/v3.0/reference/command/aggregate/#dbcmd.aggregate) command.

MongoDB also supports [map-reduce](https://docs.mongodb.org/v3.0/reference/glossary/#term-map-reduce) with the [mapReduce](https://docs.mongodb.org/v3.0/reference/command/mapReduce/#dbcmd.mapReduce) command, as well as basic aggregation with the [group](https://docs.mongodb.org/v3.0/reference/command/group/#dbcmd.group), [count](https://docs.mongodb.org/v3.0/reference/command/count/#dbcmd.count), and [distinct](https://docs.mongodb.org/v3.0/reference/command/distinct/#dbcmd.distinct). commands.

### How to optimize storage use for small documents?

* Use the \_id field explicitly.
* Use shorter field names.
* Embed documents.

### JavaScript in MongoDB

All of the following MongoDB operations permit you to run arbitrary JavaScript expressions directly on the server:

* [$where](https://docs.mongodb.org/v3.0/reference/operator/query/where/#op._S_where)
* [mapReduce](https://docs.mongodb.org/v3.0/reference/command/mapReduce/#dbcmd.mapReduce)
* [group](https://docs.mongodb.org/v3.0/reference/command/group/#dbcmd.group)

*You can disable all server-side execution of JavaScript, by passing the --noscripting option on the command line or setting*[*security.javascriptEnabled*](https://docs.mongodb.org/v3.0/reference/configuration-options/#security.javascriptEnabled)*in a configuration file.*

### What is the command to list index?

1. db.collection.getIndexes()

### 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
* 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

### 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'.

## MongoDB.js

### What is the version you use or have used?

Current Version V2.1, previous stable version is V1.4.9

* 2015-12-15, V2.1.1, V2.0.52, V1 is no longer maintained.
* 2014-10-07, V2.0.0 and V1.4.18
* 2013-04-26, V1.3.0
* 2012-11-27, V1.1.0
* 2012-04-26, V1.0.0
* 2011-11-10, V0.9.7