



mongoDB[®]

MongoDB is an open-source, high-performance, document-oriented database.

Documents are JSON-like data structures stored in a format called BSON (bsonspec.org). Documents are stored in **collections**, each of which resides in its own **database**. Collections can be thought of as the equivalent of a table in an RDBMS. There are **no fixed schemas** in MongoDB, so documents with different “shapes” can be stored in the same collection.

MongoDB features **full index support** (including secondary and compound indexes); indexes are specified per collection. There is a rich, **document-based query language** (see reverse) that leverages these indexes. MongoDB also provides sophisticated atomic update modifiers (see reverse) to keep code contention-free.

Clustered setups are supported, including easy replication for high availability, as well as auto-sharding for write-scaling and large data-set sizes.

Queries and What They Match

<code>{a: 10}</code>	Docs where a is 10, or an array containing the value 10.
<code>{a: 10, b: "hello"}</code>	Docs where a is 10 and b is "hello."
<code>{a: {\$gt: 10}}</code>	Docs where a is greater than 10. Also \$lt (<), \$gte (>=), \$lte (<=), and \$ne (!=).
<code>{a: {\$in: [10, "hello"]}</code>	Docs where a is either 10 or "hello."
<code>{a: {\$all: [10, "hello"]}</code>	Docs where a is an array containing both 10 and "hello".
<code>{"a.b": 10}</code>	Docs where a is an embedded document with b equal to 10.
<code>{a: {\$elemMatch: {b: 1, c: 2}}}</code>	Docs where a is an array containing a single item with both b equal to 1 and c equal to 2.
<code>{\$or: [{a: 1}, {b: 2}]}</code>	Docs where a is 1 or b is 2.
<code>db.foo.find({a: /^m/})</code>	Docs where a begins with the letter "m".

The following queries cannot use indexes as of MongoDB v2.0. These query forms should normally be accompanied by at least one other query term which *does* use an index:

<code>{a: {\$nin: [10, "hello"]}</code>	Docs where a is anything but 10 or "hello."
<code>{a: {\$mod: [10, 1]}</code>	Docs where a mod 10 is 1.
<code>{a: {\$size: 3}}</code>	Docs where a is an array with exactly 3 elements.
<code>{a: {\$exists: true}}</code>	Docs containing an a field.
<code>{a: {\$type: 2}}</code>	Docs where a is a string (see bsonspec.org for more types).
<code>{a: /foo.*bar/}</code>	Docs where a matches the regular expression "foo.*bar".
<code>{a: {\$not: {\$type: 2}}}</code>	Docs where a is not a string. \$not negates any of the other query operators.

Update Modifiers

<code>{\$inc: {a: 2}}</code>	Increment a by 2.
<code>{\$set: {a: 5}}</code>	Set a to the value 5.
<code>{\$unset: {a: 1}}</code>	Delete the a key.
<code>{\$push: {a: 1}}</code>	Append the value 1 to the array a .
<code>{\$pushAll: {a: [1, 2]}}</code>	Append both 1 and 2 to the array a .
<code>{\$addToSet: {a: 1}}</code>	Append the value 1 to the array a (if it doesn't already exist).
<code>{\$addToSet: {a: {\$each: [1, 2]}}</code>	Append both 1 and 2 to the array a (if they don't already exist).
<code>{\$pop: {a: 1}}</code>	Remove the last element from the array a .
<code>{\$pop: {a: -1}}</code>	Remove the first element from the array a .
<code>{\$pull: {a: 5}}</code>	Remove all occurrences of 5 from the array a .
<code>{\$pullAll: {a: [5, 6]}}</code>	Remove all occurrences of 5 or 6 from the array a .