# mongoDB

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

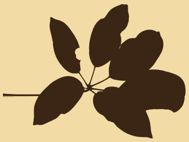
Documents are JSON-like data structures stored in a format called BSON (bsonspec.org). Documents are stored in Collections, each of which resides in its own Database. Collections can be thought of as the equivalent of a table in an RDBMS. There are **no fixed schemas** in MongoDB, so documents with different “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 any indexes you've defined. MongoDB also provides complex atomic update modifiers (see reverse) to keep code contention-free.

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

## Queries and What They Match

| | |
|---|---|
| `{a: 10}` | **docs where “a” is 10, or an array containing the value 10** |
| `{a: 10, b: "hello"}` | **docs where “a” is 10 and “b” is “hello”** |
| `{a: {$gt: 10}}` | **docs where “a” is greater than 10** |
| | also $lt (<), $gte (>=), $lte (<=) and $ne (!=) |
| `{a: {$in: [10, "hello"]}}` | **docs where “a” is either 10 or “hello”** |
| `{a: {$nin: [10, "hello"]}}` | **docs where “a” is anything but 10 or “hello”** |
| `{a: {$all: [10, "hello"]}}` | **docs where “a” is an array containing both 10 and “hello”** |
| `{a: {$mod: [10, 1]}}` | **docs where “a” % 10 is 1** |
| `{a: {$size: 3}}` | **docs where “a” is an array with exactly 3 elements** |
| `{a: {$exists: true}}` | **docs containing an “a” field** |
| `{a: {$exists: false}}` | **docs not containing an “a” field** |
| `{a: {$type: 2}}` | **docs where “a” is a string (see bsonspec.org for more types)** |
| `{a: /foo.*bar/}` | **docs where “a” matches the regular expression “foo.*bar”** |
| `{"a.b": 10}` | **docs where “a” is an embedded document with “b” equal to 10** |
| `{a: {$elemMatch: {b: 1, c: 2}}}` | **docs where “a” is an array containing a single item** |
| | **with both “b” equal to 1 and “c” equal to 2** |
| `{a: {$not: {$type: 2}}}` | **docs where “a” is not a string** |
| | $not can negate any of the other queries |
| `{$where: "this.a == this.b"}` | **docs where the value of “a” equals the value of “b”** |
| | $where can be any JavaScript expression |
| `{$or: [{a: 1}, {b: 2}]}` | **docs where “a” is 1 or “b” is 2** |



## Update Modifiers

| | |
|---|---|
| `{$inc: {a: 2}}` | **increment “a” by 2** |
| `{$set: {a: 5}}` | **set “a” to the value 5** |
| `{$unset: {a: 1}}` | **deletes the “a” key** |
| `{$push: {a: 1}}` | **append the value 1 to the array “a”** |
| `{$pushAll: {a: [1, 2]}}` | **append both 1 and 2 to the array “a”** |
| `{$addToSet: {a: 1}}` | **append the value 1 to the array “a” if it doesn’t already exist** |
| `{$addToSet: {a: {$each: [1, 2]}}}` | **append both 1 and 2 to the array “a” if they don’t already exist** |
| `{$pop: {a: 1}}` | **remove the last element in the array “a”** |
| `{$pop: {a: -1}}` | **remove the first element in the array “a”** |
| `{$pull: {a: 5}}` | **remove all occurrences of 5 from the array “a”** |
| `{$pullAll: {a: [5, 6]}}` | **remove all occurrences of 5 or 6 from the array “a”** |