**Mongodb documentation**

Pipeline

The MongoDB aggregation pipeline consists of [stages](https://docs.mongodb.com/manual/reference/operator/aggregation-pipeline/#aggregation-pipeline-operator-reference). Each stage transforms the documents as they pass through the pipeline. Pipeline stages do not need to produce one output document for every input document. For example, some stages may generate new documents or filter out documents.

To create or define a view:

* Use the [db.createCollection()](https://docs.mongodb.com/manual/reference/method/db.createCollection/" \l "db.createCollection" \o "db.createCollection()) method or the [create](https://docs.mongodb.com/manual/reference/command/create/#dbcmd.create) command:

copy

copied

db.createCollection(

"<viewName>",

{

"viewOn" : "<source>",

"pipeline" : [<pipeline>],

"collation" : { <collation> }

}

)

* Use the [db.createView()](https://docs.mongodb.com/manual/reference/method/db.createView/" \l "db.createView" \o "db.createView()) method:

copy

copied

db.createView(

"<viewName>",

"<source>",

[<pipeline>],

{

"collation" : { <collation> }

}

)

Views exhibit the following behavior:

### Read Only

Views are read-only; write operations on views will error.

The following read operations can support views:

* [db.collection.find()](https://docs.mongodb.com/manual/reference/method/db.collection.find/#db.collection.find)
* [db.collection.findOne()](https://docs.mongodb.com/manual/reference/method/db.collection.findOne/#db.collection.findOne)
* [db.collection.aggregate()](https://docs.mongodb.com/manual/reference/method/db.collection.aggregate/#db.collection.aggregate)
* [db.collection.countDocuments()](https://docs.mongodb.com/manual/reference/method/db.collection.countDocuments/#db.collection.countDocuments)
* [db.collection.estimatedDocumentCount()](https://docs.mongodb.com/manual/reference/method/db.collection.estimatedDocumentCount/#db.collection.estimatedDocumentCount)
* [db.collection.count()](https://docs.mongodb.com/manual/reference/method/db.collection.count/#db.collection.count)
* [db.collection.distinct()](https://docs.mongodb.com/manual/reference/method/db.collection.distinct/#db.collection.distinct)

For example, consider a players collection with the following document:

copy

copied

db.players.insert( {

name: "player1",

games: [ { game: "abc", score: 8 }, { game: "xyz", score: 5 } ],

joined: **new** Date("2020-01-01"),

lastLogin: **new** Date("2020-05-01")

} )

## Drop a View

To remove a view, use the [db.collection.drop()](https://docs.mongodb.com/manual/reference/method/db.collection.drop/" \l "db.collection.drop" \o "db.collection.drop()) method on the view.

## Modify a View

You can modify a view either by dropping and recreating the view or using the [collMod](https://docs.mongodb.com/manual/reference/command/collMod/" \l "dbcmd.collMod" \o "collMod) command.

The following example uses [db.getCollection()](https://docs.mongodb.com/manual/reference/method/db.getCollection/" \l "db.getCollection" \o "db.getCollection()) to access the auth collection and insert a document into it.

copy

copied

**var** authColl = db.getCollection("auth")

authColl.insertOne(

{

usrName : "John Doe",

usrDept : "Sales",

usrTitle : "Executive Account Manager",

authLevel : 4,

authDept : [ "Sales", "Customers"]

}

)

The previous example requires the use of [db.getCollection("auth")](https://docs.mongodb.com/manual/reference/method/db.getCollection/" \l "db.getCollection" \o "db.getCollection()) because of a name conflict with the database method [db.auth()](https://docs.mongodb.com/manual/reference/method/db.auth/" \l "db.auth" \o "db.auth()). Calling db.auth directly to perform an insert operation would reference the [db.auth()](https://docs.mongodb.com/manual/reference/method/db.auth/" \l "db.auth" \o "db.auth()) method and would error.

The following example attempts the same operation, but without using the [db.getCollection()](https://docs.mongodb.com/manual/reference/method/db.getCollection/" \l "db.getCollection" \o "db.getCollection()) method:

copy

copied

db.auth.insertOne(

{

usrName : "John Doe",

usrDept : "Sales",

usrTitle : "Executive Account Manager",

authLevel : 4,

authDept : [ "Sales", "Customers"]

}

)

The operation errors as db.auth() method has no insertOne method.

**Collection names**

The following returns the names of all collections in the records database:

copy

copied

use records

db.getCollectionNames()

The method returns the names of the collections in an array:

copy

copied

[ "employees", "products", "mylogs", "system.indexes" ]

The following aggregation operation finds movies created in 1995 and includes the comment option to provide tracking information in the logs, the db.system.profile collection, and db.currentOp.

db.movies.aggregate( [ { $match: { year : 1995 } } ], { comment : "match\_all\_movies\_from\_1995" } ).pretty()

**find**

{ "\_id": "apples", "qty": 5 }

{ "\_id": "bananas", "qty": 7 }

{ "\_id": "oranges", "qty": { "in stock": 8, "ordered": 12 } }

{ "\_id": "avocados", "qty": "fourteen" }

The following query uses [$gt](https://docs.mongodb.com/manual/reference/operator/query/gt/#op._S_gt) to return documents where the value of qty is greater than 4.

copy

copied

db.collection.find( { qty: { $gt: 4 } } )

The query returns the following documents:

copy

copied

{ "\_id": "apples", "qty": 5 }

{ "\_id": "bananas", "qty": 7 }

db.bios.find( { "name.last": "Hopper" } )

* The following operation uses the [$in](https://docs.mongodb.com/manual/reference/operator/query/in/#op._S_in) operator to return documents in the [bios collection](https://docs.mongodb.com/manual/reference/bios-example-collection/) where \_id equals either 5 or ObjectId("507c35dd8fada716c89d0013"):

copy

copied

db.bios.find(

{ \_id: { $in: [ 5, ObjectId("507c35dd8fada716c89d0013") ] } }

)

* The following operation uses the [$gt](https://docs.mongodb.com/manual/reference/operator/query/gt/#op._S_gt) operator returns all the documents from the bios collection where birth is greater than new Date('1950-01-01'):

copy

copied

db.bios.find( { birth: { $gt: **new** Date('1950-01-01') } } )

* The following operation uses the [$regex](https://docs.mongodb.com/manual/reference/operator/query/regex/#op._S_regex) operator to return documents in the [bios collection](https://docs.mongodb.com/manual/reference/bios-example-collection/) where name.last field starts with the letter N (or is "LIKE N%")

copy

copied

db.bios.find(

{ "name.last": { $regex: /^N/ } }

)

For a list of the query operators, see [Query Selectors](https://docs.mongodb.com/manual/reference/operator/query/#query-selectors).

db.bios.find( { birth: { $gt: **new** Date('1940-01-01'), $lt: **new** Date('1960-01-01') } } )

# **Query and Projection Operators**

**On this page**

* [Query Selectors](https://docs.mongodb.com/manual/reference/operator/query/#query-selectors)
* [Projection Operators](https://docs.mongodb.com/manual/reference/operator/query/#projection-operators)
* [Miscellaneous Operators](https://docs.mongodb.com/manual/reference/operator/query/#miscellaneous-operators)

**NOTE**

For details on specific operator, including syntax and examples, click on the specific operator to go to its reference page.

## Query Selectors

### Comparison

For comparison of different BSON type values, see the [specified BSON comparison order](https://docs.mongodb.com/manual/reference/bson-type-comparison-order/#bson-types-comparison-order).

| **Name** | **Description** |
| --- | --- |
| [$eq](https://docs.mongodb.com/manual/reference/operator/query/eq/#op._S_eq) | Matches values that are equal to a specified value. |
| [$gt](https://docs.mongodb.com/manual/reference/operator/query/gt/#op._S_gt) | Matches values that are greater than a specified value. |
| [$gte](https://docs.mongodb.com/manual/reference/operator/query/gte/#op._S_gte) | Matches values that are greater than or equal to a specified value. |
| [$in](https://docs.mongodb.com/manual/reference/operator/query/in/#op._S_in) | Matches any of the values specified in an array. |
| [$lt](https://docs.mongodb.com/manual/reference/operator/query/lt/#op._S_lt) | Matches values that are less than a specified value. |
| [$lte](https://docs.mongodb.com/manual/reference/operator/query/lte/#op._S_lte) | Matches values that are less than or equal to a specified value. |
| [$ne](https://docs.mongodb.com/manual/reference/operator/query/ne/#op._S_ne) | Matches all values that are not equal to a specified value. |
| [$nin](https://docs.mongodb.com/manual/reference/operator/query/nin/#op._S_nin) | Matches none of the values specified in an array. |

### Logical

| **Name** | **Description** |
| --- | --- |
| [$and](https://docs.mongodb.com/manual/reference/operator/query/and/#op._S_and) | Joins query clauses with a logical AND returns all documents that match the conditions of both clauses. |
| [$not](https://docs.mongodb.com/manual/reference/operator/query/not/#op._S_not) | Inverts the effect of a query expression and returns documents that do not match the query expression. |
| [$nor](https://docs.mongodb.com/manual/reference/operator/query/nor/#op._S_nor) | Joins query clauses with a logical NOR returns all documents that fail to match both clauses. |
| [$or](https://docs.mongodb.com/manual/reference/operator/query/or/#op._S_or) | Joins query clauses with a logical OR returns all documents that match the conditions of either clause. |

### Element

| **Name** | **Description** |
| --- | --- |
| [$exists](https://docs.mongodb.com/manual/reference/operator/query/exists/#op._S_exists) | Matches documents that have the specified field. |
| [$type](https://docs.mongodb.com/manual/reference/operator/query/type/#op._S_type) | Selects documents if a field is of the specified type. |

### Evaluation

| **Name** | **Description** |
| --- | --- |
| [$expr](https://docs.mongodb.com/manual/reference/operator/query/expr/#op._S_expr) | Allows use of aggregation expressions within the query language. |
| [$jsonSchema](https://docs.mongodb.com/manual/reference/operator/query/jsonSchema/#op._S_jsonSchema) | Validate documents against the given JSON Schema. |
| [$mod](https://docs.mongodb.com/manual/reference/operator/query/mod/#op._S_mod) | Performs a modulo operation on the value of a field and selects documents with a specified result. |
| [$regex](https://docs.mongodb.com/manual/reference/operator/query/regex/#op._S_regex) | Selects documents where values match a specified regular expression. |
| [$text](https://docs.mongodb.com/manual/reference/operator/query/text/#op._S_text) | Performs text search. |
| [$where](https://docs.mongodb.com/manual/reference/operator/query/where/#op._S_where) | Matches documents that satisfy a JavaScript expression. |

### Geospatial

| **Name** | **Description** |
| --- | --- |
| [$geoIntersects](https://docs.mongodb.com/manual/reference/operator/query/geoIntersects/#op._S_geoIntersects) | Selects geometries that intersect with a [GeoJSON](https://docs.mongodb.com/manual/reference/glossary/" \l "term-geojson) geometry. The [2dsphere](https://docs.mongodb.com/manual/core/2dsphere/) index supports [$geoIntersects](https://docs.mongodb.com/manual/reference/operator/query/geoIntersects/#op._S_geoIntersects). |
| [$geoWithin](https://docs.mongodb.com/manual/reference/operator/query/geoWithin/#op._S_geoWithin) | Selects geometries within a bounding [GeoJSON geometry](https://docs.mongodb.com/manual/reference/geojson/" \l "geospatial-indexes-store-geojson). The [2dsphere](https://docs.mongodb.com/manual/core/2dsphere/) and [2d](https://docs.mongodb.com/manual/core/2d/) indexes support [$geoWithin](https://docs.mongodb.com/manual/reference/operator/query/geoWithin/#op._S_geoWithin). |
| [$near](https://docs.mongodb.com/manual/reference/operator/query/near/#op._S_near) | Returns geospatial objects in proximity to a point. Requires a geospatial index. The [2dsphere](https://docs.mongodb.com/manual/core/2dsphere/) and [2d](https://docs.mongodb.com/manual/core/2d/) indexes support [$near](https://docs.mongodb.com/manual/reference/operator/query/near/#op._S_near). |
| [$nearSphere](https://docs.mongodb.com/manual/reference/operator/query/nearSphere/#op._S_nearSphere) | Returns geospatial objects in proximity to a point on a sphere. Requires a geospatial index. The [2dsphere](https://docs.mongodb.com/manual/core/2dsphere/) and [2d](https://docs.mongodb.com/manual/core/2d/) indexes support [$nearSphere](https://docs.mongodb.com/manual/reference/operator/query/nearSphere/#op._S_nearSphere). |

### Array

| **Name** | **Description** |
| --- | --- |
| [$all](https://docs.mongodb.com/manual/reference/operator/query/all/#op._S_all) | Matches arrays that contain all elements specified in the query. |
| [$elemMatch](https://docs.mongodb.com/manual/reference/operator/query/elemMatch/#op._S_elemMatch) | Selects documents if element in the array field matches all the specified [$elemMatch](https://docs.mongodb.com/manual/reference/operator/query/elemMatch/#op._S_elemMatch) conditions. |
| [$size](https://docs.mongodb.com/manual/reference/operator/query/size/#op._S_size) | Selects documents if the array field is a specified size. |

### Bitwise

| **Name** | **Description** |
| --- | --- |
| [$bitsAllClear](https://docs.mongodb.com/manual/reference/operator/query/bitsAllClear/#op._S_bitsAllClear) | Matches numeric or binary values in which a set of bit positions all have a value of 0. |
| [$bitsAllSet](https://docs.mongodb.com/manual/reference/operator/query/bitsAllSet/#op._S_bitsAllSet) | Matches numeric or binary values in which a set of bit positions all have a value of 1. |
| [$bitsAnyClear](https://docs.mongodb.com/manual/reference/operator/query/bitsAnyClear/#op._S_bitsAnyClear) | Matches numeric or binary values in which any bit from a set of bit positions has a value of 0. |
| [$bitsAnySet](https://docs.mongodb.com/manual/reference/operator/query/bitsAnySet/#op._S_bitsAnySet) | Matches numeric or binary values in which any bit from a set of bit positions has a value of 1. |

## Projection Operators

| **Name** | **Description** |
| --- | --- |
| [$](https://docs.mongodb.com/manual/reference/operator/projection/positional/#proj._S_) | Projects the first element in an array that matches the query condition. |
| [$elemMatch](https://docs.mongodb.com/manual/reference/operator/projection/elemMatch/#proj._S_elemMatch) | Projects the first element in an array that matches the specified [$elemMatch](https://docs.mongodb.com/manual/reference/operator/projection/elemMatch/#proj._S_elemMatch) condition. |
| [$meta](https://docs.mongodb.com/manual/reference/operator/aggregation/meta/#proj._S_meta) | Projects the document’s score assigned during [$text](https://docs.mongodb.com/manual/reference/operator/query/text/#op._S_text) operation. |
| [$slice](https://docs.mongodb.com/manual/reference/operator/projection/slice/#proj._S_slice) | Limits the number of elements projected from an array. Supports skip and limit slices. |

## Miscellaneous Operators

| **Name** | **Description** |
| --- | --- |
| [$comment](https://docs.mongodb.com/manual/reference/operator/query/comment/#op._S_comment) | Adds a comment to a query predicate. |
| [$rand](https://docs.mongodb.com/manual/reference/operator/query/rand/#op._S_rand) | Generates a random float between 0 and 1. |

**Findone**

#### Specify the Fields to Return

The following operation finds a document in the [bios collection](https://docs.mongodb.com/manual/reference/bios-example-collection/) and returns only the name, contribs and \_id fields:

copy

copied

db.bios.findOne(

{ },

{ name: 1, contribs: 1 }

)

#### Return All but the Excluded Fields

The following operation returns a document in the [bios collection](https://docs.mongodb.com/manual/reference/bios-example-collection/) where the contribs field contains the element OOP and returns all fields except the \_id field, the first field in the name embedded document, and the birth field:

copy

copied

db.bios.findOne(

{ contribs: 'OOP' },

{ \_id: 0, 'name.first': 0, birth: 0 }

)

### . Define the On-Demand Materialized View

updateMonthlySales = **function**(startDate) {

db.bakesales.aggregate( [

{ $match: { date: { $gte: startDate } } },

{ $group: { \_id: { $dateToString: { format: "%Y-%m", date: "$date" } }, sales\_quantity: { $sum: "$quantity"}, sales\_amount: { $sum: "$amount" } } },

{ $merge: { into: "monthlybakesales", whenMatched: "replace" } }

] );

};

* The [$match](https://docs.mongodb.com/manual/reference/operator/aggregation/match/#pipe._S_match) stage filters the data to process only those sales greater than or equal to the startDate.
* The [$group](https://docs.mongodb.com/manual/reference/operator/aggregation/group/#pipe._S_group) stage groups the sales information by the year-month. The documents output by this stage have the form:
* { "\_id" : "<YYYY-mm>", "sales\_quantity" : <num>, "sales\_amount" : <NumberDecimal> }
* The [$merge](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#pipe._S_merge) stage writes the output to the monthlybakesales collection.

Based [on](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#merge-on) the \_id field (the default for unsharded output collections), the stage checks if the document in the aggregation results [matches](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#merge-whenmatched) an existing document in the collection:

* + [When there is a match](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#merge-whenmatched) (i.e. a document with the same year-month already exists in the collection), the stage [replaces the existing document](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#merge-whenmatched-replace) with the document from the aggregation results.
  + [When there is not a match](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#merge-whennotmatched), the stage inserts the document from the aggregation results into the collection (the default behavior when not matched).

The [$merge](https://docs.mongodb.com/manual/reference/operator/aggregation/merge/#pipe._S_merge) stage:

* Can output to a collection in the same or different database.
* Creates a new collection if the output collection does not already exist.
* Can incorporate results (insert new documents, merge documents, replace documents, keep existing documents, fail the operation, process documents with a custom update pipeline) into an existing collection.
* Can output to a sharded collection. Input collection can also be sharded.

# **Capped Collections**

## Overview

[Capped collections](https://docs.mongodb.com/manual/reference/glossary/#term-capped-collection) are fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. Capped collections work in a way similar to circular buffers: once a collection fills its allocated space, it makes room for new documents by overwriting the oldest documents in the collection.

# **Documents**

MongoDB stores data records as BSON documents. BSON is a binary representation of [JSON](https://docs.mongodb.com/manual/reference/glossary/#term-json) documents, though it contains more data types than JSON.

## Date

**var** mydate1 = **new** Date()

**var** mydate2 = ISODate()

The following example retrieves all documents from the inventory collection where status equals either "A" or "D":

Copy the following filter into the Compass query bar and click **Find**:

copy

copied

{ status: { $in: [ "A", "D" ] } }

Specify AND Conditions

A compound query can specify conditions for more than one field in the collection’s documents. Implicitly, a logical AND conjunction connects the clauses of a compound query so that the query selects the documents in the collection that match all the conditions.

The following example retrieves all documents in the inventory collection where the status equals "A" **and** qty is less than ([$lt](https://docs.mongodb.com/manual/reference/operator/query/lt/#op._S_lt)) 30:

Copy the following filter into the Compass query bar and click **Find**:

copy

copied

{ status: "A", qty: { $lt: 30 } }

The following example retrieves all documents in the collection where the status equals "A" **or** qty is less than ([$lt](https://docs.mongodb.com/manual/reference/operator/query/lt/#op._S_lt)) 30:

Copy the following filter into the Compass query bar and click **Find**:

copy

copied

{ $or: [ { status: "A" }, { qty: { $lt: 30 } } ] }

For example, the following query selects all documents where the field size equals the document { h: 14, w: 21, uom: "cm" }:

Copy the following filter into the Compass query bar and click **Find**:

copy

copied

{ size: { h: 14, w: 21, uom: "cm" } }

{ "size.h": { $lt: 15 } }

{ "size.h": { $lt: 15 }, "size.uom": "in", status: "D" }

## Match an Array

{ tags: ["red", "blank"] }

{ tags: { $all: ["red", "blank"] } }

{ tags: "red" }

{ dim\_cm: { $gt: 15, $lt: 20 } }