**Collections Methods**

|  |  |
| --- | --- |
| **Name** | **Description** |
| [db.collection.analyzeShardKey()](https://www.mongodb.com/docs/manual/reference/method/db.collection.analyzeShardKey/#mongodb-method-db.collection.analyzeShardKey) | Calculates metrics for evaluating a shard key. |
| [db.collection.aggregate()](https://www.mongodb.com/docs/manual/reference/method/db.collection.aggregate/#mongodb-method-db.collection.aggregate) | Provides access to the [aggregation pipeline.](https://www.mongodb.com/docs/manual/core/aggregation-pipeline/#std-label-aggregation-pipeline) |
| [db.collection.bulkWrite()](https://www.mongodb.com/docs/manual/reference/method/db.collection.bulkWrite/#mongodb-method-db.collection.bulkWrite) | Provides bulk write operation functionality. |
| [db.collection.configureQueryAnalyzer()](https://www.mongodb.com/docs/manual/reference/method/db.collection.configureQueryAnalyzer/#mongodb-method-db.collection.configureQueryAnalyzer) | Configures query sampling for a collection. |
| [db.collection.count()](https://www.mongodb.com/docs/manual/reference/method/db.collection.count/#mongodb-method-db.collection.count) | Wraps [count](https://www.mongodb.com/docs/manual/reference/command/count/#mongodb-dbcommand-dbcmd.count) to return a count of the number of documents in a collection or a view. |
| [db.collection.countDocuments()](https://www.mongodb.com/docs/manual/reference/method/db.collection.countDocuments/#mongodb-method-db.collection.countDocuments) | Wraps the [$group](https://www.mongodb.com/docs/manual/reference/operator/aggregation/group/#mongodb-pipeline-pipe.-group) aggregation stage with a [$sum](https://www.mongodb.com/docs/manual/reference/operator/aggregation/sum/#mongodb-group-grp.-sum) expression to return a count of the number of documents in a collection or a view. |
| [db.collection.createIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.createIndex/#mongodb-method-db.collection.createIndex) | Builds an index on a collection. |
| [db.collection.createIndexes()](https://www.mongodb.com/docs/manual/reference/method/db.collection.createIndexes/#mongodb-method-db.collection.createIndexes) | Builds one or more indexes on a collection. |
| [db.collection.dataSize()](https://www.mongodb.com/docs/manual/reference/method/db.collection.dataSize/#mongodb-method-db.collection.dataSize) | Returns the size of the collection. Wraps the [size](https://www.mongodb.com/docs/manual/reference/command/collStats/#mongodb-data-collStats.size) field in the output of the [collStats.](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-dbcommand-dbcmd.collStats) |
| [db.collection.deleteOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteOne/#mongodb-method-db.collection.deleteOne) | Deletes a single document in a collection. |
| [db.collection.deleteMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteMany/#mongodb-method-db.collection.deleteMany) | Deletes multiple documents in a collection. |
| [db.collection.distinct()](https://www.mongodb.com/docs/manual/reference/method/db.collection.distinct/#mongodb-method-db.collection.distinct) | Returns an array of documents that have distinct values for the specified field. |
| [db.collection.drop()](https://www.mongodb.com/docs/manual/reference/method/db.collection.drop/#mongodb-method-db.collection.drop) | Removes the specified collection from the database. |
| [db.collection.dropIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.dropIndex/#mongodb-method-db.collection.dropIndex) | Removes a specified index on a collection. |
| [db.collection.dropIndexes()](https://www.mongodb.com/docs/manual/reference/method/db.collection.dropIndexes/#mongodb-method-db.collection.dropIndexes) | Removes all indexes on a collection. |
| [db.collection.ensureIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.ensureIndex/#mongodb-method-db.collection.ensureIndex) | Removed. Use [db.collection.createIndex().](https://www.mongodb.com/docs/manual/reference/method/db.collection.createIndex/" \l "mongodb-method-db.collection.createIndex) |
| [db.collection.estimatedDocumentCount()](https://www.mongodb.com/docs/manual/reference/method/db.collection.estimatedDocumentCount/#mongodb-method-db.collection.estimatedDocumentCount) | Wraps [count](https://www.mongodb.com/docs/manual/reference/command/count/#mongodb-dbcommand-dbcmd.count) to return an approximate count of the documents in a collection or a view. |
| [db.collection.explain()](https://www.mongodb.com/docs/manual/reference/method/db.collection.explain/#mongodb-method-db.collection.explain) | Returns information on the query execution of various methods. |
| [db.collection.find()](https://www.mongodb.com/docs/manual/reference/method/db.collection.find/#mongodb-method-db.collection.find) | Performs a query on a collection or a view and returns a cursor object. |
| [db.collection.findAndModify()](https://www.mongodb.com/docs/manual/reference/method/db.collection.findAndModify/#mongodb-method-db.collection.findAndModify) | Atomically modifies and returns a single document. |
| [db.collection.findOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.findOne/#mongodb-method-db.collection.findOne) | Performs a query and returns a single document. |
| [db.collection.findOneAndDelete()](https://www.mongodb.com/docs/manual/reference/method/db.collection.findOneAndDelete/#mongodb-method-db.collection.findOneAndDelete) | Finds a single document and deletes it. |
| [db.collection.findOneAndReplace()](https://www.mongodb.com/docs/manual/reference/method/db.collection.findOneAndReplace/#mongodb-method-db.collection.findOneAndReplace) | Finds a single document and replaces it. |
| [db.collection.findOneAndUpdate()](https://www.mongodb.com/docs/manual/reference/method/db.collection.findOneAndUpdate/#mongodb-method-db.collection.findOneAndUpdate) | Finds a single document and updates it. |
| [db.collection.getIndexes()](https://www.mongodb.com/docs/manual/reference/method/db.collection.getIndexes/#mongodb-method-db.collection.getIndexes) | Returns an array of documents that describe the existing indexes on a collection. |
| [db.collection.getShardDistribution()](https://www.mongodb.com/docs/manual/reference/method/db.collection.getShardDistribution/#mongodb-method-db.collection.getShardDistribution) | For collections in sharded clusters, [db.collection.getShardDistribution()](https://www.mongodb.com/docs/manual/reference/method/db.collection.getShardDistribution/" \l "mongodb-method-db.collection.getShardDistribution) reports data of [chunk](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-chunk) distribution. |
| [db.collection.getShardVersion()](https://www.mongodb.com/docs/manual/reference/method/db.collection.getShardVersion/#mongodb-method-db.collection.getShardVersion) | Internal diagnostic method for sharded cluster. |
| [db.collection.hideIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.hideIndex/#mongodb-method-db.collection.hideIndex) | Hides an index from the query planner. |
| [db.collection.insertOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.insertOne/#mongodb-method-db.collection.insertOne) | Inserts a new document in a collection. |
| [db.collection.insertMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.insertMany/#mongodb-method-db.collection.insertMany) | Inserts several new document in a collection. |
| [db.collection.isCapped()](https://www.mongodb.com/docs/manual/reference/method/db.collection.isCapped/#mongodb-method-db.collection.isCapped) | Reports if a collection is a [capped collection.](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-capped-collection) |
| [db.collection.latencyStats()](https://www.mongodb.com/docs/manual/reference/method/db.collection.latencyStats/#mongodb-method-db.collection.latencyStats) | Returns latency statistics for a collection. |
| [db.collection.mapReduce()](https://www.mongodb.com/docs/manual/reference/method/db.collection.mapReduce/#mongodb-method-db.collection.mapReduce) | Performs map-reduce style data aggregation. |
| [db.collection.reIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.reIndex/#mongodb-method-db.collection.reIndex) | Rebuilds all existing indexes on a collection. |
| [db.collection.remove()](https://www.mongodb.com/docs/manual/reference/method/db.collection.remove/#mongodb-method-db.collection.remove) | Deletes documents from a collection. |
| [db.collection.renameCollection()](https://www.mongodb.com/docs/manual/reference/method/db.collection.renameCollection/#mongodb-method-db.collection.renameCollection) | Changes the name of a collection. |
| [db.collection.replaceOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.replaceOne/#mongodb-method-db.collection.replaceOne) | Replaces a single document in a collection. |
| [db.collection.stats()](https://www.mongodb.com/docs/manual/reference/method/db.collection.stats/#mongodb-method-db.collection.stats) | Reports on the state of a collection. Provides a wrapper around the [collStats.](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-dbcommand-dbcmd.collStats) |
| [db.collection.storageSize()](https://www.mongodb.com/docs/manual/reference/method/db.collection.storageSize/#mongodb-method-db.collection.storageSize) | Reports the total size used by the collection in bytes. Provides a wrapper around the [storageSize](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-data-collStats.storageSize) field of the [collStats](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-dbcommand-dbcmd.collStats) output. |
| [db.collection.totalIndexSize()](https://www.mongodb.com/docs/manual/reference/method/db.collection.totalIndexSize/#mongodb-method-db.collection.totalIndexSize) | Reports the total size used by the indexes on a collection. Provides a wrapper around the [totalIndexSize](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-data-collStats.totalIndexSize) field of the [collStats](https://www.mongodb.com/docs/manual/reference/command/collStats/" \l "mongodb-dbcommand-dbcmd.collStats) output. |
| [db.collection.totalSize()](https://www.mongodb.com/docs/manual/reference/method/db.collection.totalSize/#mongodb-method-db.collection.totalSize) | Reports the total size of a collection, including the size of all documents and all indexes on a collection. |
| [db.collection.unhideIndex()](https://www.mongodb.com/docs/manual/reference/method/db.collection.unhideIndex/#mongodb-method-db.collection.unhideIndex) | Unhides an index from the query planner. |
| [db.collection.updateOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/#mongodb-method-db.collection.updateOne) | Modifies a single document in a collection. |
| [db.collection.updateMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.updateMany/#mongodb-method-db.collection.updateMany) | Modifies multiple documents in a collection. |
| [db.collection.watch()](https://www.mongodb.com/docs/manual/reference/method/db.collection.watch/#mongodb-method-db.collection.watch) | Establishes a Change Stream on a collection. |
| [db.collection.validate()](https://www.mongodb.com/docs/manual/reference/method/db.collection.validate/#mongodb-method-db.collection.validate) | Performs diagnostic operations on a collection. |

## MongoDB Query Operators

There are many query operators that can be used to compare and reference document fields.

### Comparison

The following operators can be used in queries to compare values:

* $eq: Values are equal
* $ne: Values are not equal
* $gt: Value is greater than another value
* $gte: Value is greater than or equal to another value
* $lt: Value is less than another value
* $lte: Value is less than or equal to another value
* $in: Value is matched within an array

### Logical

The following operators can logically compare multiple queries.

* $and: Returns documents where both queries match
* $or: Returns documents where either query matches
* $nor: Returns documents where both queries fail to match
* $not: Returns documents where the query does not match

### Evaluation

The following operators assist in evaluating documents.

* $regex: Allows the use of regular expressions when evaluating field values
* $text: Performs a text search
* $where: Uses a JavaScript expression to match documents

## Update Operators

There are many update operators that can be used during document updates.

### Fields

The following operators can be used to update fields:

* $currentDate: Sets the field value to the current date
* $inc: Increments the field value
* $rename: Renames the field
* $set: Sets the value of a field
* $unset: Removes the field from the document

### Array

The following operators assist with updating arrays.

* $addToSet: Adds distinct elements to an array
* $pop: Removes the first or last element of an array
* $pull: Removes all elements from an array that match the query
* $push: Adds an element to an array

## Aggregation Pipelines

Aggregation operations allow you to group, sort, perform calculations, analyze data, and much more.

Aggregation pipelines can have one or more "stages". The order of these stages are important. Each stage acts upon the results of the previous stage.

db.posts.aggregate([

// Stage 1: Only find documents that have more than 1 like

{

$match: { likes: { $gt: 1 } }

},

// Stage 2: Group documents by category and sum each categories likes

{

$group: { \_id: "$category", totalLikes: { $sum: "$likes" } }

}

])

## $group

## This aggregation stage groups documents by the unique \_id expression provided.

db.listingsAndReviews.aggregate(

[ { $group : { \_id : "$property\_type" } } ]

)

This will return the distinct values from the property\_type field.

## $limit

## This aggregation stage limits the number of documents passed to the next stage.

db.movies.aggregate([ { $limit: 1 } ])

## This will return the 1 movie from the collection.

## $project

## This aggregation stage passes only the specified fields along to the next aggregation stage.

db.restaurants.aggregate([

{

$project: {

"name": 1,

"cuisine": 1,

"address": 1

}

},

{

$limit: 5

}

])

## This will return the documents but only include the specified fields.

## We use a 1 to include a field and 0 to exclude a field.

## $sort

## This aggregation stage groups sorts all documents in the specified sort order.

db.listingsAndReviews.aggregate([

{

$sort: { "accommodates": -1 }

},

{

$project: {

"name": 1,

"accommodates": 1

}

},

{

$limit: 5

}

])

## This will return the documents sorted in descending order by the accommodates field.

## The sort order can be chosen by using 1 or -1. 1 is ascending and -1 is descending.

## $match

## This aggregation stage behaves like a find. It will filter documents that match the query provided.

db.listingsAndReviews.aggregate([

{ $match : { property\_type : "House" } },

{ $limit: 2 },

{ $project: {

"name": 1,

"bedrooms": 1,

"price": 1

}}

])

## This will only return documents that have the property\_type of "House".

## $addFields

## This aggregation stage adds new fields to documents.

db.restaurants.aggregate([

{

$addFields: {

avgGrade: { $avg: "$grades.score" }

}

},

{

$project: {

"name": 1,

"avgGrade": 1

}

},

{

$limit: 5

}

])

## This will return the documents along with a new field, avgGrade, which will contain the average of each restaurants grades.score.

## $count

## This aggregation stage counts the total amount of documents passed from the previous stage.

db.restaurants.aggregate([

{

$match: { "cuisine": "Chinese" }

},

{

$count: "totalChinese"

}

])

## This will return the number of documents at the $count stage as a field called "totalChinese".

## $lookup

This aggregation stage performs a left outer join to a collection in the same database.

There are four required fields:

* from: The collection to use for lookup in the same database
* localField: The field in the primary collection that can be used as a unique identifier in the from collection.
* foreignField: The field in the from collection that can be used as a unique identifier in the primary collection.
* as: The name of the new field that will contain the matching documents from the from collection.

db.comments.aggregate([

{

$lookup: {

from: "movies",

localField: "movie\_id",

foreignField: "\_id",

as: "movie\_details",

},

},

{

$limit: 1

}

])

## This will return the movie data along with each comment.

## $out

## This aggregation stage writes the returned documents from the aggregation pipeline to a collection.

db.listingsAndReviews.aggregate([

{

$group: {

\_id: "$property\_type",

properties: {

$push: {

name: "$name",

accommodates: "$accommodates",

price: "$price",

},

},

},

},

{ $out: "properties\_by\_type" },

])

## The first stage will group properties by the property\_type and include the name, accommodates, and price fields for each. The $out stage will create a new collection called properties\_by\_type in the current database and write the resulting documents into that collection.

## A screenshot of a computer Description automatically generated

## A screenshot of a computer program Description automatically generated