

MongoDB CRUD Operations

On this page

- Create Operations
- Read Operations
- Update Operations
- Delete Operations
- Bulk Write

CRUD operations *create*, *read*, *update*, and *delete* documents.

Create Operations

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

- `db.collection.insert()`
- `db.collection.insertOne()` *New in version 3.2*
- `db.collection.insertMany()` *New in version 3.2*

In MongoDB, insert operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

```
db.users.insert (  ← collection
{
  name: "sue",      ← field: value
  age: 26,          ← field: value
  status: "A"       ← field: value
}                  } document
)
```

For examples, see Insert Documents.

Read Operations

Read operations retrieves documents from a collection; i.e. queries a collection for documents. MongoDB provides the following methods to read documents from a collection:

- `db.collection.find()`

You can specify query filters or criteria that identify the documents to return.

```
db.users.find(
  { age: { $gt: 18 } }, ← collection
  { name: 1, address: 1 } ← query criteria
).limit(5)              ← projection
                        ← cursor modifier
```

For examples, see:

- Query Documents
- Query on Embedded/Nested Documents
- Query an Array
- Query an Array of Embedded Documents

Update Operations

Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

- `db.collection.update()`

- `db.collection.updateOne()` *New in version 3.2*
- `db.collection.updateMany()` *New in version 3.2*
- `db.collection.replaceOne()` *New in version 3.2*

In MongoDB, update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

You can specify criteria, or filters, that identify the documents to update. These filters use the same syntax as read operations.

```
db.users.update(  
  { age: { $gt: 18 } },  
  { $set: { status: "A" } },  
  { multi: true }  
)
```

← collection
← update criteria
← update action
← update option

For examples, see Update Documents.

Delete Operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

- `db.collection.remove()`
- `db.collection.deleteOne()` *New in version 3.2*
- `db.collection.deleteMany()` *New in version 3.2*

In MongoDB, delete operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

You can specify criteria, or filters, that identify the documents to remove. These filters use the same syntax as read operations.

```
db.users.remove(  
  { status: "D" }  
)
```

← collection
← remove criteria

For examples, see [Delete Documents](#).

Bulk Write

MongoDB provides the ability to perform write operations in bulk. For details, see [Bulk Write Operations](#).