

Table of Contents



- [1. Connecting to MongoDB](#)
 - [Start MongoDB](#)
 - [Connect to MongoDB Shell](#)
 - [Connect to a Specific Database](#)
 - [Show Current Database](#)
- [2. Database Management](#)
 - [List All Databases](#)
 - [Create or Switch Database](#)
 - [Delete a Database](#)
- [3. Collection Management](#)
 - [List Collections](#)
 - [Create a Collection](#)
 - [Delete a Collection](#)
- [4. Insert Documents](#)
 - [Insert One Document](#)
 - [Insert Multiple Documents](#)
- [5. Querying Data](#)
 - [Find All Documents](#)
 - [Find with Condition](#)
 - [Find Specific Fields](#)
 - [Find One Document](#)
 - [Count Documents](#)
- [6. Update Documents](#)
 - [Update One Document](#)
 - [Update Multiple Documents](#)
 - [Replace Document](#)
- [7. Delete Documents](#)
 - [Delete One Document](#)
 - [Delete Multiple Documents](#)
- [8. Sorting and Limiting](#)
 - [Sort Results](#)
 - [Limit Results](#)
 - [Skip Documents](#)
- [9. Aggregation](#)
 - [Basic Aggregation](#)

- [Match and Project](#)
- [10. Indexing](#)
 - [Create an Index](#)
 - [List Indexes](#)
 - [Drop Index](#)
- [11. Export and Import Data](#)
 - [Export Data](#)
 - [Import Data](#)
- [12. Backup and Restore](#)
 - [Backup Database](#)
 - [Restore Database](#)
- [13. Admin Commands](#)
 - [Show Server Status](#)
 - [Show Current Operations](#)
 - [Kill Operation](#)
 - [Compact Database](#)
- [14. User Management](#)
 - [Create Admin User](#)
 - [Create Database User](#)
 - [Show Users](#)
 - [Delete User](#)
- [15. Common Query Operators](#)
 - [Tips for MongoDB](#)

1. Connecting to MongoDB

Start MongoDB

```
mongod
```

Connect to MongoDB Shell

```
mongosh
```

Connect to a Specific Database

```
mongosh <database_name>
```

Show Current Database

```
db
```

2. Database Management

List All Databases

```
show dbs
```

Create or Switch Database

```
use my_database
```

Delete a Database

```
db.dropDatabase()
```

3. Collection Management

List Collections

```
show collections
```

Create a Collection

```
db.createCollection("employees")
```

Delete a Collection

```
db.employees.drop()
```

4. Insert Documents

Insert One Document

```
db.employees.insertOne({  
  name: "Alice",  
  position: "Manager",  
  salary: 75000  
})
```

Insert Multiple Documents

```
db.employees.insertMany([  
  { name: "Bob", position: "Developer", salary: 60000 },  
  { name: "Carol", position: "Analyst", salary: 55000 }  
])
```

5. Querying Data

Find All Documents

```
db.employees.find()
```

Find with Condition

```
db.employees.find({ salary: { $gt: 60000 } })
```

Find Specific Fields

```
db.employees.find({}, { name: 1, salary: 1, _id: 0 })
```

Find One Document

```
db.employees.findOne({ name: "Alice" })
```

Count Documents

```
db.employees.countDocuments({ position: "Developer" })
```

6. Update Documents

Update One Document

```
db.employees.updateOne(  
  { name: "Alice" },  
  { $set: { salary: 80000 } }  
)
```

Update Multiple Documents

```
db.employees.updateMany(  
  { position: "Developer" },  
  { $inc: { salary: 1000 } }
```

```
{ position: "Developer" },  
{ $set: { position: "Senior Developer" } }  
)
```

Replace Document

```
db.employees.replaceOne(  
  { name: "Bob" },  
  { name: "Bob", position: "Lead Developer", salary: 70000 }  
)
```

7. Delete Documents

Delete One Document

```
db.employees.deleteOne({ name: "Carol" })
```

Delete Multiple Documents

```
db.employees.deleteMany({ position: "Analyst" })
```

8. Sorting and Limiting

Sort Results

```
db.employees.find().sort({ salary: -1 }) // -1 for DESC, 1 for ASC
```

Limit Results

```
db.employees.find().limit(5)
```

Skip Documents

```
db.employees.find().skip(2)
```

9. Aggregation

Basic Aggregation

```
db.employees.aggregate([
  { $group: { _id: "$position", avgSalary: { $avg: "$salary" } } }
])
```

Match and Project

```
db.employees.aggregate([
  { $match: { salary: { $gt: 60000 } } },
  { $project: { name: 1, salary: 1, _id: 0 } }
])
```

10. Indexing

Create an Index

```
db.employees.createIndex({ name: 1 })
```

List Indexes

```
db.employees.getIndexes()
```

Drop Index

```
db.employees.dropIndex("name_1")
```

11. Export and Import Data

Export Data

```
mongoexport --db=my_database --collection=employees --  
out=employees.json
```

Import Data

```
mongoimport --db=my_database --collection=employees --  
file=employees.json
```

12. Backup and Restore

Backup Database

```
mongodump --db=my_database --out=/backup/path
```

Restore Database

```
mongorestore /backup/path
```

13. Admin Commands

Show Server Status

```
db.serverStatus()
```

Show Current Operations

```
db.currentOp()
```

Kill Operation

```
db.killOp(<operation_id>)
```

Compact Database

```
db.runCommand({ compact: 'employees' })
```

14. User Management

Create Admin User

```
use admin
db.createUser({
  user: "admin",
  pwd: "password123",
  roles: [{ role: "userAdminAnyDatabase", db: "admin" }]
})
```

Create Database User

```
db.createUser({
  user: "user1",
  pwd: "pass123",
  roles: [{ role: "readWrite", db: "my_database" }]
})
```

Show Users

```
show users
```

Delete User

```
db.dropUser("user1")
```

15. Common Query Operators

Operator	Description	Example
\$eq	Equal to	{ salary: { \$eq: 50000 } }
\$ne	Not equal to	{ salary: { \$ne: 50000 } }
\$gt	Greater than	{ salary: { \$gt: 50000 } }
\$lt	Less than	{ salary: { \$lt: 50000 } }
\$gte	Greater than or equal to	{ salary: { \$gte: 50000 } }
\$lte	Less than or equal to	{ salary: { \$lte: 50000 } }
\$in	Matches any value in array	{ position: { \$in: ['Manager', 'CEO'] } }
\$nin	Not in array	{ position: { \$nin: ['Analyst'] } }
\$regex	Matches a pattern	{ name: { \$regex: '^A' } }

Tips for MongoDB

- Backup regularly to avoid data loss.
- Use indexes to improve query performance.
- Prefer embedded documents for better performance.
- Practice aggregation to handle complex queries efficiently.