MongoDB Database and Collection Commands

# 📂 Database-Level Commands

show dbs

Lists all databases on the server.

use <databaseName>

Switches to or creates a new database.

db

Displays the current selected database.

db.dropDatabase()

Deletes the currently selected database.

db.stats()

Returns statistics about the current database.

the difference between db.dropDatabase() and db.stats() in MongoDB:

**🧨 db.dropDatabase()**

* **Purpose**: **Deletes the current database** entirely.
* **Effect**:
  + Removes **all collections**, **documents**, and **indexes** within that database.
  + Cannot be undone – **use with caution**.
* **Returns**: An object confirming success, e.g.

json

CopyEdit

{ "dropped": "yourDatabaseName", "ok": 1 }

✅ **Example**:

js

CopyEdit

use myDatabase

db.dropDatabase()

→ Deletes the myDatabase completely.

**📊 db.stats()**

* **Purpose**: Provides **statistics and information** about the current database.
* **Effect**: **Read-only** operation, safe to run anytime.
* **Returns**: Info like:
  + Number of collections
  + Storage size
  + Indexes
  + Data size
  + Object count

✅ **Example**:

js

CopyEdit

use myDatabase

db.stats()

→ Outputs something like:

json

CopyEdit

{

"db": "myDatabase",

"collections": 5,

"objects": 1200,

"dataSize": 204800,

"storageSize": 409600,

...

}

**🧾 Summary Table:**

| **Feature** | **db.dropDatabase()** | **db.stats()** |
| --- | --- | --- |
| Action | Deletes the current database | Shows stats for current database |
| Is it destructive? | ✅ Yes (cannot be undone) | ❌ No (safe/read-only) |
| Return Value | Confirmation of deletion | Stats object with database info |
| Common Use Case | Clean up/remove old or test DBs | Monitor DB usage, size, and collection count |

# 📄 Collection (Table) Commands

show collections

Lists all collections (equivalent to tables) in the current database.

db.createCollection("<name>")

Explicitly creates a new collection.

db.<collection>.drop()

Drops (deletes) a collection.

db.getCollectionNames()

Returns an array of all collection names.

db.getCollectionInfos()

Returns detailed info about collections.

db.<collection>.stats()

Returns statistics about a specific collection.

# 📑 Document (Row) Commands

db.<collection>.find()

Retrieves documents from a collection.

db.<collection>.findOne()

Retrieves a single document.

db.<collection>.insertOne({...})

Inserts a single document.

db.<collection>.insertMany([...])

Inserts multiple documents.

db.<collection>.updateOne({...}, {...})

Updates the first matching document.

db.<collection>.updateMany({...}, {...})

Updates multiple documents.

db.<collection>.deleteOne({...})

Deletes the first matching document.

db.<collection>.deleteMany({...})

Deletes all matching documents.

db.<collection>.countDocuments({...})

Counts documents matching a condition.

db.<collection>.aggregate([...])

Runs aggregation pipeline queries.

# ⚙️ Index and Schema Inspection

db.<collection>.getIndexes()

Lists all indexes on a collection.

db.<collection>.createIndex({ field: 1 })

Creates an index on a field.

db.<collection>.validate()

Checks the integrity of the collection.

# 🧪 Utility Commands

db.version()

Returns the version of MongoDB server.

db.serverStatus()

Displays server status and usage stats.

db.currentOp()

Shows currently running operations.

db.getMongo()

Returns the Mongo object representing the connection.

db.getSiblingDB("<otherDB>")

Access another DB without switching context.