/\*

use:

load('script.js');

\*/

db = connect("mongodb://localhost/31380\_db");

// Creating a movies collection

db.createCollection("movies");

/\*

true

\*/

// Inserting data into collection

db.movies.insertMany([

    {

        title: "The Shawshank Redemption",

        director: "Frank Darabont",

        releaseYear: 1994,

        genre: ["Drama", "Crime"],

        rating: 9.3,

    },

    {

        title: "The Godfather",

        director: "Francis Ford Coppola",

        releaseYear: 1972,

        genre: ["Crime", "Drama"],

        rating: 9.2,

    },

    {

        title: "Pulp Fiction",

        director: "Quentin Tarantino",

        releaseYear: 1994,

        genre: ["Crime", "Drama"],

        rating: 8.9,

    },

    {

        title: "The Dark Knight",

        director: "Christopher Nolan",

        releaseYear: 2008,

        genre: ["Action", "Crime", "Drama"],

        rating: 9.0,

    },

    {

        title: "Inception",

        director: "Christopher Nolan",

        releaseYear: 2010,

        genre: ["Action", "Adventure", "Sci-Fi"],

        rating: 8.8,

    },

]);

// Selecting all movies

db.movies.find().pretty();

/\*

[

  {

    \_id: ObjectId("6549072b4f850d45854084bb"),

    title: 'The Shawshank Redemption',

    director: 'Frank Darabont',

    releaseYear: 1994,

    genre: [ 'Drama', 'Crime' ],

    rating: 9.3

  },

  {

    \_id: ObjectId("6549072b4f850d45854084bc"),

    title: 'The Godfather',

    director: 'Francis Ford Coppola',

    releaseYear: 1972,

    genre: [ 'Crime', 'Drama' ],

    rating: 9.2

  },

  {

    \_id: ObjectId("6549072b4f850d45854084bd"),

    title: 'Pulp Fiction',

    director: 'Quentin Tarantino',

    releaseYear: 1994,

    genre: [ 'Crime', 'Drama' ],

    rating: 8.9

  },

  {

    \_id: ObjectId("6549072b4f850d45854084be"),

    title: 'The Dark Knight',

    director: 'Christopher Nolan',

    releaseYear: 2008,

    genre: [ 'Action', 'Crime', 'Drama' ],

    rating: 9

  },

  {

    \_id: ObjectId("6549072b4f850d45854084bf"),

    title: 'Inception',

    director: 'Christopher Nolan',

    releaseYear: 2010,

    genre: [ 'Action', 'Adventure', 'Sci-Fi' ],

    rating: 8.8

  }

]

\*/

// Finding by name

db.movies.find({ title: "The Godfather" });

/\*

[

  {

    \_id: ObjectId("6549072b4f850d45854084bc"),

    title: 'The Godfather',

    director: 'Francis Ford Coppola',

    releaseYear: 1972,

    genre: [ 'Crime', 'Drama' ],

    rating: 9.2

  }

]

\*/

// Finding by rating

db.movies.find({ rating: { $gt: 9 } });

/\*

[

  {

    \_id: ObjectId("6549072b4f850d45854084bb"),

    title: 'The Shawshank Redemption',

    director: 'Frank Darabont',

    releaseYear: 1994,

    genre: [ 'Drama', 'Crime' ],

    rating: 9.3

  },

  {

    \_id: ObjectId("6549072b4f850d45854084bc"),

    title: 'The Godfather',

    director: 'Francis Ford Coppola',

    releaseYear: 1972,

    genre: [ 'Crime', 'Drama' ],

    rating: 9.2

  }

]

\*/

// Updating a document

db.movies.updateOne({ title: "The Godfather" }, { $set: { rating: 9.9 } });

/\*

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}

\*/

// Deprecated, use updateOne instead

db.movies.save(

    { \_id: ObjectId("6549072b4f850d45854084bb") },

    {

        title: "The Shawshank Redemption",

        director: "Frank Darabont",

        releaseYear: 1994,

        genre: ["Drama", "Crime", "Thriller"],

        rating: 9.5,

    }

);

// Deleting a document

db.movies.deleteOne({ title: "The Shawshank Redemption" });

/\*

{ acknowledged: true, deletedCount: 1 }

\*/

// AND operator

db.movies.find({

    title: "The Dark Knight",

    director: "Christopher Nolan",

});

/\*

[

  {

    \_id: ObjectId("6549072b4f850d45854084be"),

    title: 'The Dark Knight',

    director: 'Christopher Nolan',

    releaseYear: 2008,

    genre: [ 'Action', 'Crime', 'Drama' ],

    rating: 9

  }

]

\*/

// OR operator

db.movies.find({

    $or: [{ director: "Christopher Nolan" }, { genre: "Action" }],

});

/\*

[

  {

    \_id: ObjectId("6549072b4f850d45854084be"),

    title: 'The Dark Knight',

    director: 'Christopher Nolan',

    releaseYear: 2008,

    genre: [ 'Action', 'Crime', 'Drama' ],

    rating: 9

  },

  {

    \_id: ObjectId("6549072b4f850d45854084bf"),

    title: 'Inception',

    director: 'Christopher Nolan',

    releaseYear: 2010,

    genre: [ 'Action', 'Adventure', 'Sci-Fi' ],

    rating: 8.8

  }

]

\*/

mongosh

use mydatabase

show dbs

db.dropDatabase()

db.createCollection("students")

**Insert a Single Document:**

javascript

Copy code

db.students.insertOne({

  "name": "Alice Smith",

  "age": 21,

  "subjects": ["Math", "English"],

  "scores": { "Math": 90, "English": 85 }

})

**Insert Multiple Documents:**

javascript

Copy code

db.students.insertMany([

  {

    "name": "Bob Johnson",

    "age": 22,

    "subjects": ["Math", "Science"],

    "scores": { "Math": 80, "Science": 88 }

  },

  {

    "name": "Carol White",

    "age": 20,

    "subjects": ["History", "English"],

    "scores": { "History": 91, "English": 87 }

  }

])

**Retrieve All Documents:**

javascript

Copy code

db.students.find()

**Retrieve Documents with Specific Criteria:** For example, find all students aged 21 or older:

javascript

Copy code

db.students.find({ "age": { "$gte": 21 } })

**Update Documents in a Collection**

To modify existing documents, you can use updateOne or updateMany.

**Update a Single Document:** For example, update the age of "Alice Smith":

javascript

Copy code

db.students.updateOne(

  { "name": "Alice Smith" },

  { "$set": { "age": 22 } }

)

**Update Multiple Documents:** For example, set a new field "isActive" to true for all students aged over 20:

javascript

Copy code

db.students.updateMany(

  { "age": { "$gt": 20 } },

  { "$set": { "isActive": true } }

)

**7. Delete Documents**

To delete documents, use deleteOne or deleteMany.

**Delete a Single Document:**

javascript

Copy code

db.students.deleteOne({ "name": "Bob Johnson" })

**Delete Multiple Documents:**

javascript

Copy code

db.students.deleteMany({ "age": { "$lt": 21 } })

**To Drop a Collection:**

javascript

Copy code

db.students.drop()

**To Drop the Entire Database:**

javascript

Copy code

db.dropDatabase()

To list all databases:  
javascript  
Copy code  
show dbs

To list all collections in the current database:  
javascript  
Copy code  
show collections

To delete all the data in a MongoDB collection without deleting the collection itself, you can use the deleteMany method with an empty filter {}. Here’s how to do it:

javascript

Copy code

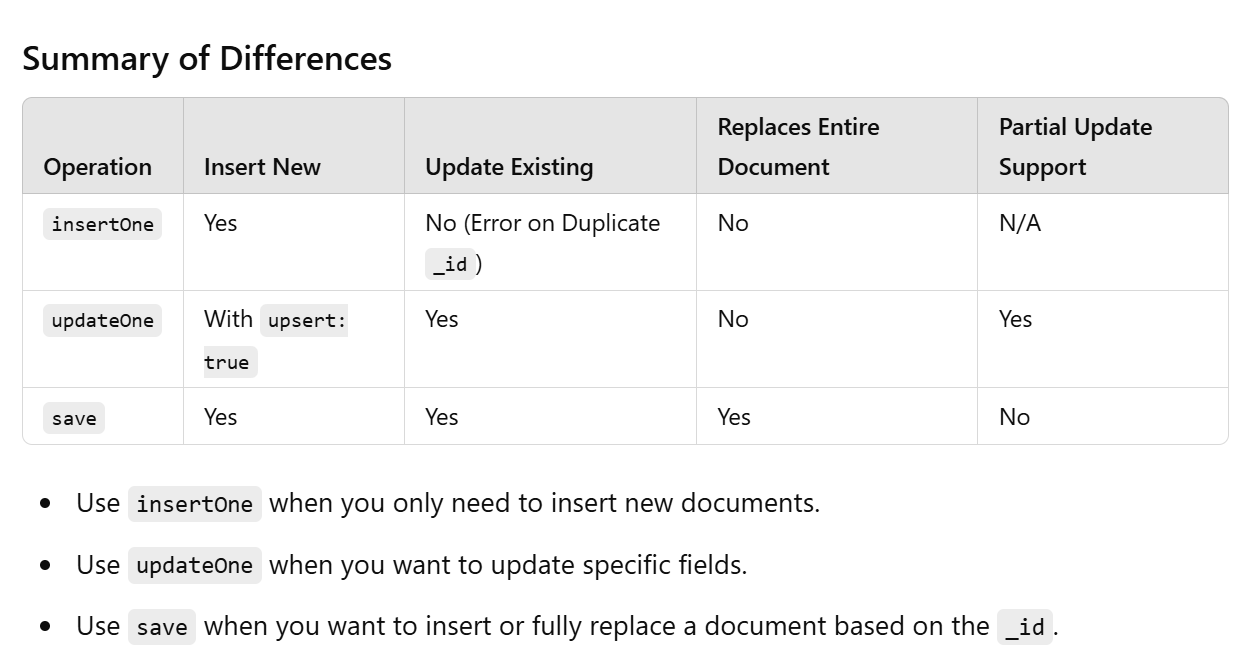
db.collectionName.deleteMany({})

**3. save**

The save method behaves differently depending on whether the document has an \_id:

* **If no \_id exists** in the document, save will insert it as a new document.
* **If \_id exists** and matches a document in the collection, it replaces the entire document.

db.students.save({ name: "Carol", age: 23, major: "English" })



[TE-Lab-Assignments/09/script.js at DBMSL · shxntanu/TE-Lab-Assignments · GitHub](https://github.com/shxntanu/TE-Lab-Assignments/blob/DBMSL/09/script.js)