E:\Web D (SIGMA)\Web-Dev\Backend\MongoDB\main.js

```

//to use any collection
// use db_name

//  => Insert Operation

//our data
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);


db.inventory.insertOne(
   { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }
)

db.inventory.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
   { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
   { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }
])


//  => Read operation

 //You can query documents in MongoDB by using the following methods:

 db.inventory.find( {} ) //To select all documents in the collection, pass an empty document as the query filter parameter to the find method.

 //Specify Equality Condition
//  { <field1>: <value1>, ... }
db.inventory.find( { status: "D" } )

//The following example retrieves all documents from the inventory collection where status equals either "A" or "D":
db.inventory.find( { status: { $in: [ "A", "D" ] } } ) //use the $in operator rather than the $or operator when performing equality checks on the same field.


//AND OPERATOR
// The following example retrieves all documents in the inventory collection where the status equals "A" and qty is less than ($lt) 30:
db.inventory.find( { status: "A", qty: { $lt: 30 } } )

//OR OPERATOR
// The following example retrieves all documents in the collection where the status equals "A" or qty is less than ($lt) 30:
db.inventory.find( { $or: [{ status: "A" },{ qty: { $lt: 30 } } ] } )


//Specify AND as well as OR Conditions

//In the following example, the compound query document selects all documents in the collection where the status equals "A" and either qty is less than ($lt) 30 or item starts with the character p:
db.inventory.find( {
```

```
    status: "A",
    $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]
} )

// => Update operation

//This page uses the following
// mongosh methods:
db.inventory.insertMany( [
    { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" },
    { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
    { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" },
    { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" },
    { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" },
    { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
    { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
    { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },
    { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
    { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" }
 ] );
// db.collection.updateOne(<filter>, <update>, <options>)
// db.collection.updateMany(<filter>, <update>, <options>)
// db.collection.replaceOne(<filter>, <update>, <options>)


//Update one file
//The following example uses the db.collection.updateOne() method on the inventory collection to update the first document where item equals "paper":
db.inventory.updateOne(
    { item: "paper" },
    {
      $set: { "size.uom": "cm", status: "P" },
      $currentDate: { lastModified: true }
    }
 )
/*The update operation:

uses the $set operator to update the value of the size.uom field to "cm" and the value of the status field to "P",

uses the $currentDate operator to update the value of the lastModified field to the current date. If lastModified field does not exist, $currentDate will create the field. See $currentDate for details. */

//Update Multiple file
// The following example uses the db.collection.updateMany() method on the inventory collection to update all documents where qty is less than 50:
db.inventory.updateMany(
    { "qty": { $lt: 50 } },
    {
      $set: { "size.uom": "in", status: "P" },
      $currentDate: { lastModified: true }
    }
 )


//Replace a Document
// To replace the entire content of a document except for the _id field, pass an entirely new document as the second argument to => db.collection.replaceOne().

db.inventory.replaceOne(
    { item: "paper" },
    { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 40 } ] }
```

```
)//The following example replaces the first document from the inventory collection where item: "paper":


//=> Delete Operation

//The following example removes all documents from the inventory collection where the status field equals "A":
db.inventory.deleteMany({ status : "A" })

//To delete at most a single document that matches a specified filter (even though multiple documents may match the specified filter) use the db.collection.deleteOne() method.

// The following example deletes the first document where status is "D":
db.inventory.deleteOne( { status: "D" } )


//  To delete all documents from a collection, pass an empty filter document {} to the db.collection.deleteMany() method.
db.inventory.deleteMany({})



//Other important operation
db.scores.insertMany([
   { "_id" : 1, "team" : "Fearful Mallards", "score" : 25000 },
   { "_id" : 2, "team" : "Tactful Mooses", "score" : 23500 },
   { "_id" : 3, "team" : "Aquatic Ponies", "score" : 19250 },
   { "_id" : 4, "team" : "Cuddly Zebras", "score" : 15235 },
   { "_id" : 5, "team" : "Garrulous Bears", "score" : 18000 }
]);

//  The following operation finds a document with score less than 20000 and replaces it:
db.scores.findOneAndReplace(
   { "score" : { $lt : 20000 } },
   { "team" : "Observant Badgers", "score" : 20000 }
)
```