**PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**Practice sheet –2 MongoDB**

**Practice Sheet :2 Topic coverage: CRUD operations-Insert,update,delete,remove,drop**

**1. Insert Multiple Documents in collection**

var beginners =

[

{

"StudentId" : 1001,

"StudentName" : "Steve",

"age": 30

},

{

"StudentId" : 1002,

"StudentName" : "Negan",

"age": 42

},

{

"StudentId" : 3333,

"StudentName" : "Rick",

"age": 35

},

];

**db.students.insert(beginners);**

**2.Update Operator Set**

**2.a.Create Product collection:**

Consider a collection  products with the following document:

{

\_id: 100,

sku: "abc123",

quantity: 250,

instock: true,

reorder: false,

details: { model: "14Q2", make: "xyz" },

tags: [ "apparel", "clothing" ],

ratings: [ { by: "ijk", rating: 4 } ]

}

**Step:2 Set Top-Level Fields**

For the document matching the criteria \_id equal to 100, the following operation uses the [$set](https://docs.mongodb.com/manual/reference/operator/update/set/#up._S_set) operator to update the value of the quantity field, details field, and the tags field.

db.products.update(

{ \_id: 100 },

{ $set:

{

quantity: 500,

details: { model: "14Q3", make: "xyz" },

tags: [ "coats", "outerwear", "clothing" ]

}

}

)

### Step:3 Set Fields in Embedded Documents

To specify a <field> in an embedded document or in an array, use [dot notation](https://docs.mongodb.com/manual/core/document/#document-dot-notation).

For the document matching the criteria \_id equal to 100, the following operation updates the make field in the details document:

db.products.update(

{ \_id: 100 },

{ $set: { "details.make": "zzz" } }

)

### Step:4 Set Elements in Arrays

To specify a <field> in an embedded document or in an array, use [dot notation](https://docs.mongodb.com/manual/core/document/#document-dot-notation).

For the document matching the criteria \_id equal to 100, the following operation updates the value second element (array index of 1) in the tags field and the rating field in the first element (array index of 0) of the ratings array.

db.products.update(

{ \_id: 100 },

{ $set:

{

"tags.1": "rain gear",

"ratings.0.rating": 2

}

}

)

**2.b.Create a collection Employee**

**Step 1)**Issue the update command

**Step 2)**Choose the condition which you want to use to decide which document needs to be updated. In our example, we want to update the document which has the Employee id 22.

**Step 3)** Use the set command to modify the Field Name

**Step 4)** Choose which Field Name you want to modify and enter the new value accordingly.

db.Employee.update(

{"Employeeid" : 1},

{$set: { "EmployeeName" : "NewMartin"}});

**3.Delete Documents**

To populate the inventory collection, run the following:

**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: "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" },**

**] );**

## **Delete All Documents**

To delete all documents from a collection, pass an empty [filter](https://docs.mongodb.com/manual/core/document/#document-query-filter) document {} to the [db.collection.deleteMany()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/#db.collection.deleteMany) method.

**db.inventory.deleteMany({})**

## **Delete All Documents that Match a Condition**

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

To specify equality conditions, use <field>:<value> expressions in the query filter document:

**db.inventory.deleteMany({ status : "A" })**

## **Delete Only One Document that Matches a Condition**

**db.inventory.deleteOne( { status: "D" } )**

**4.Remove**

MongoDB's **remove()** method is used to remove a document from the collection. remove() method accepts two parameters. One is deletion criteria and second is justOne flag.

* **deletion criteria** − (Optional) deletion criteria according to documents will be removed.
* **justOne** − (Optional) if set to true or 1, then remove only one document.

Basic syntax of **remove()** method is as follows −

>db.COLLECTION\_NAME.remove(DELLETION\_CRITTERIA)

db.mycol.remove({'title':'MongoDB Overview'})

db.mycol.remove({})

**5.Drop database:**

db.dropDatabase()

**Drop Collection**

MongoDB's **db.collection.drop()** is used to drop a collection from the database.

db.mycollection.drop()

**6.Practice:**

**Create database EMP and Make Collection With name "EMPL" and Follow Queries**  
  
**=>Created Database**  
> use emp  
 switched to DB emp  
  
**=>Create Collection**  
> db.createCollection("empl")  
 **=>Insert Records Into EMPL Collection**  
> db.empl.insert([  
 {no:1,name:"ST",salary:2000,role:"OB"},  
 {no:2,name:"MSD",salary:1500,role:"WK"},  
 {no:3,name:"YS",salary:1000,role:"ALR"},  
 {no:4,name:"RD",salary:1000,role:"MOB"},  
 {no:5,name:"RS",salary:500,role:"OB"},  
 {no:6,name:"BK",salary:500,role:"MOB"},  
 {no:7,name:"VK",salary:300,role:"BW"},  
 {no:8,name:"JB",salary:400,role:"BW"},  
 {no:9,name:"HP",salary:400,role:"ALR"},  
 {no:10,name:"VS",salary:300,role:"OB"}])  
 **=>Display Data in Proper Format**   
>db.empl.find().pretty()  
 **=>Update Salary Of Employee where Name is "ST" by +8000**  
>db.empl.update({name:"ST"},{[**$inc:**](https://codingonly4u.blogspot.com/2018/07/mongodb-exercises-practice-solution.html){salary:8000}})  
 **=>Update Salary Of All Employee by giving an increment of +4000 each**  
>db.empl.update({},{**$inc:**{salary:4000}},{multi:true})  
 **=>update role of "MSD" as "C and WK"**  
>db.empl.update({name:"MSD"},{**$set:**{role:"c and WK"}})  
 **=>Add a New Field remark to document with name "RS" set Remark as WC**  
>db.emp.update({name:"RS"},{**$set:**{remark:"WC"}})  
 **=>Add a New Field As Number 11,name AK,Salary 10000,role coch without using insert statement. But for Doing So You should have a Record Added woth number 11.**  
>db.empl.update({no:11},{**$set:**{no:11,name:"AK",salary:10000,role:"coch"}},{upsert:true})  
 **=>remove added New Field**  
>db.empl.update({name:"RD"},{**$unset:**{remark:"WC"}})  
 **=>Update the Field "RD" by multiplying with salary by 2**  
>db.empl.update({name:"RD"},{**$mul:**{salary:2}})  
  
**=>To Find Document From the empl collection where name begins with S**  
>db.empl.find({name:**/^**S**/**})  
  
**=>To Find Document From the empl collection where name begins with R**  
>db.empl.find({name:**/^**R**/**})  
  
**=>To Find Document From the empl collection where name ends with K**  
>db.empl.find({name:**/**K**$/**})  
  
**=>To Find Document From the empl collection where name ends with D**  
>db.empl.find({name:**/**S**$/**})  
  
**=>To Find Document From the empl collection where name has S in any position**  
>db.empl.find({name:**/**S**/**})

mongodb+srv://<username>:<password>@cluster0.kl80f.mongodb.net/<dbname>?retryWrites=true&w=majority