

10th UNIT TASK-10: CRUD OPERATIONS IN DOCUMENT DATABASES

Aim: To perform Mongoose using NPM design on MongoDB designing document database and performing CRUD operations like creating, inserting, querying, finding and removing operations.

STEPS:

- * install Mongo db using following link
<https://www.mongodb.com/t/try/download/community>
- * install Mongosh using the below link
<https://www.mongodb.com/docs/mongodb-shell/#download-and-install-mongosh>.
- * To add the MongoDB Shell binary's location to your PATH environment variable:
Open the Control Panel
In the system and security category, click system
click advanced system settings. The system properties modal displays.
Click environment variables.
In the system variables section, select path and click edit. The edit environment variable modal displays.
Click New and add the filepath to your mongosh binary.

- * Open mongo shell v.0 from ~~c:\programfiles\mongodb\server\bin\mongod.exe~~
- * Type the CRUD(CREATE READ UPDATE DELETE) COMMANDS GIVEN IN TEXT FILE.

CRUD OPERATIONS:

```
db.createCollection("mylab")
```

```
{ "ok": 1 }
```

```
>db.mylab.insertOne({item: "canvas", aty: 100, tags: ["cotton"]}, {size: {h: 28, w: 355, uom: "cm"}})
```

```
{ "acknowledged": true }
```

```
"insertedId": ObjectId("627d13acc73990c074e6393")
```

```
>db.mylab.find({item: "canvas"})
```

Output :

- **Employee** (Employee_ID, Name, Department_ID, Job_Title, Hire_Date, Salary) ↗
- **Department** (Department_ID, Manager_ID) ↗
- **Manager** (Manager_ID, Name)

```
{ "id": ObjectId("627d1598c73990c074e6397d"),
  "item": "canvas", "qty": 100, "tags": [
    "cotton"], "size": { "h": 28, "w": 35.5, " uom": "cm" },
  "tags": "blank", "bed" }, size: { h: 14, w: 21, uom: "cm" },
  { item: "mat", qty: 85, tags: ["gray"], size: { h: 27, w: 35.5, uom: "cm" },
  item: "mousepad", qty: 15, tags: ["green", "blue"] }, size: { h: 19, w: 22.2, uom: "cm" },
  "acknowledged": true,
  "insertedIds": [
    ObjectId("627d1598c73990c074e6397d"),
    ObjectId("627d1598c73990c074e6397e"),
    ObjectId("627d1598c73990c074e6397f") ] }
```

```
3 db.mylab.find({ _id: 1, qty: 15 })
```

```
{ "id": ObjectId("627d13acc73990c074e6397c"),
  "item": "canvas",
  "qty": 100
  "id": ObjectId("627d1598c73990c074e6397d"),
  "item": "journal",
  "qty": 25
  "id": ObjectId("627d1598c73990c074e6397e"),
  "item": "mat",
  "qty": 85
  "id": ObjectId("627d1598c73990c074e6397f"),
  "item": "mousepad",
  "qty": 25 }
```

```
3 db.mylab.find({ item: "canvas" }).pretty().sort({ _id: 1 })
```

```
{ "id": ObjectId("627d13acc73990c074e6397c"),
  "item": "canvas",
  "qty": 100,
  "tags": [
```

Output:

- $\text{Employee_ID} \rightarrow \text{Name, Department, Job_Title, Manager_ID, Hire_Date, Salary}$
(The Employee's ID determines all other attributes).
- $\text{Department} \rightarrow \text{Manager_ID}$ (The department determines the manager's ID).
- $\text{Manager_ID} \rightarrow \text{Name}$ (The manager's ID determines the manager's name).

"Cotton"

3,

"size": {

"h": 28,

"w": 35.5

" uom": "1cm"

3

3

3

> db.mylab.dektonc({item: "journal"}
3

3

> db.mylab.find({3, item: 1, qty: 13}).pretty()
{
 "_id": ObjectId("627d13acc73990c074e6397"),
 "item": "canvas",
 "qty": 100
3

3

{
 "_id": ObjectId("627d1598c73990c074e6397"),
 "item": "journal",
 "qty": 25
3

3

{
 "_id": ObjectId("627d1598c73990c074e6397"),
 "item": "mat", "qty": 85
3

3

{
 "_id": ObjectId("627d1598c73990c074e6397"),
 "item": "mousepad", "qty": 25
3

Result: The implementation of CRUD operations like creating, inserting, finding and removing operations using MongoDB is successfully executed

VEL TECH-CSE	
EX NO.	10
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
FINAL (20)	20
DATE	16/10/23