**Name – Shahane Akash Dilip Roll No – C32260 Class – TE Div:2**

**Batch – T8 Subject - DBMS**

**Assignment No – B1**

**Title:** Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators.

**Problem Statement:** Create an Articles Collection and perform basic operations (CRUD) like Insert, Display, Update and Delete operations. Eg. Articles (Tittle, Content, Author, Author\_age, Dop, Category, Comment). Comment is an embedded document and also an array. Comment may consist of the Name, Remarks keys

**----------------------------------------------------------------**

**QUERIES –**

**1. Insert couple of Documents.**

test> use b1

b1> db.article.insertMany([

... {"Title": "Introduction to MongoDB",

... "Content": "MongoDB is a NoSQL Database",

... "Author": "John Doe",

... "Author\_age": 30,

... "Dop": "17-10-2023",

... "Category": "Database",

... "Comments": [

... { "user": "Sam Brown", "comment": "I learned a lot!" },

... { "user": "Alice Green", "comment": "Thank you for sharing." }

... ]},

... {"Title": "Web Development",

... "Content": "Use of MERN",

... "Author": "Jane Doe",

... "Dop": "16-10-2023",

... "Category": "Web",

... "Comments": [

... { "user": "Jane Smith", "comment": "Great article!" },

... { "user": "Bob Johnson", "comment": "Very informative." }

... ]}

... ])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("652d9b8a07c463bba275588a"),

'1': ObjectId("652d9b8a07c463bba275588b")

}

}

**2. Display the first document found in database.**

b1> db.article.findOne()

{

\_id: ObjectId("652d9b8a07c463bba275588a"),

Title: 'Introduction to MongoDB',

Content: 'MongoDB is a NoSQL Database',

Author: 'John Doe',

Author\_age: 30,

Dop: '17-10-2023',

Category: 'Database',

"Comments": [

{ "user": "Sam Brown", "comment": "I learned a lot!" },

{ "user": "Alice Green", "comment": "Thank you for sharing." }

]}

}

**3. Display first document belonging to a certain Author say “Sharma” found in database.**

b1> db.article.findOne({Author:"Rohit Sharma"})

{

\_id: ObjectId("652d9ddc07c463bba275588c"),

Title: 'Cricket Strategies for Success',

Content: 'Effective strategies for winning cricket matches...',

Author: 'Rohit Sharma',

Author\_age: 34,

Dop: ISODate("2023-10-18T09:45:00.000Z"),

Category: 'Sports',

Comments: [

{ user: 'Sara Williams', comment: 'Fantastic article!' },

{ user: 'David Lee', comment: 'I learned a lot about cricket.' }

]

}

**4. Modify the comment made by certain person on a certain article.**

b1> db.article.update(

... { "Title": "Cricket Strategies for Success", "Comments.user": "David Lee" },

... { $set: { "Comments.$.comment": "I thoroughly enjoyed reading this article!" } })

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

b1> db.article.findOne({ "Title": "Cricket Strategies for Success" })

{

\_id: ObjectId("652d9ddc07c463bba275588c"),

Title: 'Cricket Strategies for Success',

Content: 'Effective strategies for winning cricket matches...',

Author: 'Rohit Sharma',

Author\_age: 34,

Dop: ISODate("2023-10-18T09:45:00.000Z"),

Category: 'Sports',

Comments: [

{ user: 'Sara Williams', comment: 'Fantastic article!' },

{

user: 'David Lee',

comment: 'I thoroughly enjoyed reading this article!'

}]}

**5. Insert record with save method with and without objectID**

b1> db.article.save ({

"Title": "New Article Without ID",

"Content": "This is a new article without an explicit \_id field.",

"Author": "John Doe",

"Author\_age": 30,

"Dop": new Date("2023-10-19T14:30:00Z"),

"Category": "Technology",

"Comments": [

{ "user": "Jane Smith", "comment": "Great article!" },

{ "user": "Bob Johnson", "comment": "Very informative." }

]

})

{

acknowledged: true,

insertedId: ObjectId("652da23d07c463bba275588d")

}

b1> db.article.save ({

"\_id": 7,

"Title": "New Article With Explicit ID",

"Content": "This is a new article with an explicit \_id field.",

"Author": "Jane Doe",

"Author\_age": 28,

"Dop": new Date("2023-10-18T12:00:00Z"),

"Category": "Science",

"Comments": [

{ "user": "Sam Brown", "comment": "I learned a lot!" },

{ "user": "Alice Green", "comment": "Thank you for sharing." }

]

})

{ acknowledged: true, insertedId: 7 }

**6. Update collection with save method.**

b1> db.article.save(

... { \_id: 7 },

... { $set: { "Author": "Virat Kohli" } }

... )

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

**7. Add one more comment for particular title.**

b1> db.article.updateOne({\_id: 7},{$push: {Comments:{user:"Hope", comment: "Bad"}}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

b1> db.article.findOne({\_id:7})

{

\_id: 7,

Title: 'New Article With Explicit ID',

Content: 'This is a new article with an explicit \_id field.',

Author: 'Virat Kohli',

Author\_age: 28,

Dop: ISODate("2023-10-18T00:00:00.000Z"),

Category: 'Science',

Comments: [

{ user: 'Sam Brown', comment: 'I learned a lot!' },

{ user: 'Alice Green', comment: 'Thank you for sharing.' },

{ user: 'Hope', comment: 'Bad' }

]

}

**8. Delete the documents whose author age is less than 18.**

b1> db.article.deleteOne( { Author\_age: { $lt: 18 } } )

{ acknowledged: true, deletedCount: 0 }

**9. Delete All Documents from a Collection**

b1> db.article.deleteMany({})

{ acknowledged: true, deletedCount: 5 }

**10. Delete the Article collection**

b1> db.article.drop()

true