ASSIGNMENT 2

Aim

Execute any 10 queries on a suitable sample database to demonstrate various query criteria.

Objectives

- To study & execute queries.
- To study & execute aggregate queries.

Problem Statement

Problem statement no. 5 :

Institute database w/ student collection.

Theory

1] Create database institute

A. use institute

2] Create collection students

A. db.createCollection ("students")

or

db.students.insert ({ })

or

db["students"].insert ({ })

3] Insert 5 documents

A. db.students.insertMany ([

{ doc1 },

{ doc2 },

.

.

.

{ doc5 }

])

we can also use .insert

4] Display all student information

A. db.students.find().pretty()

5] Update student branch from IT to Computer for student ID 3

A.
```
db.students.updateOne(
    {"studentId": 3},
    {$set: {"Branch": "Computer"}}
)
```

6] Add interest python in ~~set~~ studentId 5

A.
```
db.students.updateOne(
    {"studentId": 5},
    {$addToSet: {"AreaOfInterest": "Python"}}
)
```

7] Add one subject name & it's score for studentId 8.

A.
```
db.students.updateOne(
    {"studentId": 8},
    {$set: {"subjects.BDA": 80}}
)
```

8] Change city name from Pune to Delhi

A.
```
db.students.updateMany(
    {"Address.city": "Pune"},
    {$set: {"Address.city": "Delhi"}}
)
```

9] Remove record with studentId 3

A. db.students.deleteOne ({"studentId" : 3})

10] Drop collection

A. db.students.drop()

- Aggregation Framework

Aggregation framework operations process data records & return computed results. It is used to group values from multiple documents.
Mongodb provides three ways for aggregation:

- Aggregation Pipeline
- Map Reduce
- Single purpose aggregation operations

* Input

Sample collection.

* Output

Execution of all commands.

* Platform

Linux - 5.8.5 (Arch Linux)

* FAQ

Q. Explain in brief aggregation pipelines framework of MongoDB.

A. Documents enter a multistage pipeline that transforms documents into an aggregated result.

```
db.collection.aggregate ( [
    { stage 1 },
    { stage 2 },
      :
    { stage N }
] )
```

The pipeline stages provide filters that operate like queries & document transformations that transform into output document.

Q. How do we know which indexes are defined over a collection ?

A. db.collection.getIndexes()

Q. How to define / alter the collection schema ?

A. db.createCollection ("Name", options)
options is a JSON used to define the schema via the value of validator key.

```
db.createCollection ( "Name", { validator : {
    $jsonSchema : { ... }
} } )
```