

## Assignment-9

Title:- MongoDB Queries

Problem Statement:-

Design and Develop MongoDB Queries using CRUD operations  
(Use CRUD operations, pretty function, etc)

Objectives:-

- Understand the concept of binary JSON format
- Understand the concept of MongoDB document-model.

S/w and H/w requirement:-

MongoDB, windows-10 64bit, 8GB RAM  
512GB SSD

Theory:-

MongoDB is a cross-platform, document oriented database that provides high performance, high availability and easy scalability. MongoDB works on concept of collection and documents.

Database:-

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collections:-

Collection is a group of MongoDB documents. It is equivalent of RDBMS table. A collection exists within a single database.

Teacher's Signature \_\_\_\_\_



Collections do not enforce a schema. Documents within a collection can have different fields. Typically all documents in a collection are of similar or related purpose.

Document:-

Document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in same collection can have different schema.

|             |                    |
|-------------|--------------------|
| RDBMS       | MongoDB            |
| Database    | Database           |
| Table       | collection         |
| Tuple/Row   | Document           |
| column      | field              |
| Table Join  | embedded documents |
| Primary Key | Primary key (-id)  |

CRUD operations:-

- 1) db.createcollection(name, options)  
 db.COLLECTION\_NAME.insertOne(document)  
 db.COLLECTION\_NAME.insertMany([...document])
- 2) db.COLLECTION\_NAME.find()
- 3) db.COLLECTION\_NAME.update(criteria, update data)
- 4) db.COLLECTION\_NAME.remove(criteria)
- 5) db.COLLECTION\_NAME.save({-id: new data})

Conclusion:- Successfully designed and developed MongoDB queries for CRUD operations.

31124

Page No.

Date / /

Conclusion: Successfully implemented concept of cursors to merge data from old to new table.

C:\Users\HP>mongo

MongoDB shell version v5.0.0

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod

Implicit session: session { "id" : UUID("d8205b7e-f84a-4a58-9992-50bb3417b691") }

MongoDB server version: 5.0.0

=====

Warning: the "mongo" shell has been superseded by "mongosh",  
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in  
an upcoming release.

We recommend you begin using "mongosh".

For installation instructions, see

<https://docs.mongodb.com/mongodb-shell/install/>

=====

---

The server generated these startup warnings when booting:

2021-11-11T22:29:15.870+05:30: Access control is not enabled for the database. Read and write access to data and configuration  
is unrestricted

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display  
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you  
and anyone you share the URL with. MongoDB may use this information to make product  
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: `db.enableFreeMonitoring()`

To permanently disable this reminder, run the following command: `db.disableFreeMonitoring()`

---

>





```
> show dbs
admin    0.000GB
config  0.000GB
crud     0.000GB
local    0.000GB
> use assignment09
switched to db assignment09
> db
assignment09
>
> db.places.insertOne({
... student:"Rupesh",
... city:"Akot"
... })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("618ea12711a01e44a2796284")
}
>
> db.places.insertMany([
... {
... student:"Vimal",
... city:"Nashik"
... },
... {
... student:"Dheeraj",
... city:"Pune"
... },
... {
... student:"Gayatri",
... city:"Mumbai"
... }
... ])
```



21°C Clear



ENG 10:45 PM



```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("618ea12811a01e44a2796285"),
    ObjectId("618ea12811a01e44a2796286"),
    ObjectId("618ea12811a01e44a2796287")
  ]
}
>
> db.places.find({student:"Rupesh"}).pretty()
{
  "_id" : ObjectId("618ea12711a01e44a2796284"),
  "student" : "Rupesh",
  "city" : "Akot"
}
> db.places.find({city:"Mumbai"}).pretty()
{
  "_id" : ObjectId("618ea12811a01e44a2796287"),
  "student" : "Gayatri",
  "city" : "Mumbai"
}
>
> db.places.update({
... student:"Dheeraj"
... },
... {
... $set:{city:"Bangaluru"}
... })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.places.find().pretty()
{
  "_id" : ObjectId("618ea12711a01e44a2796284"),
```



```
> db.places.find().pretty()
{
  "_id" : ObjectId("618ea12711a01e44a2796284"),
  "student" : "Rupesh",
  "city" : "Akot"
}
{
  "_id" : ObjectId("618ea12811a01e44a2796285"),
  "student" : "Vimal",
  "city" : "Nashik"
}
{
  "_id" : ObjectId("618ea12811a01e44a2796286"),
  "student" : "Dheeraj",
  "city" : "Bangaluru"
}
{
  "_id" : ObjectId("618ea12811a01e44a2796287"),
  "student" : "Gayatri",
  "city" : "Mumbai"
}
>
> db.places.remove({
... student:"Gayatri"
... })
WriteResult({ "nRemoved" : 1 })
>
> db.places.find().pretty()
{
  "_id" : ObjectId("618ea12711a01e44a2796284"),
  "student" : "Rupesh",
  "city" : "Akot"
}
```



```
>
> db.places.find().pretty()
{
  "_id" : ObjectId("618ea12711a01e44a2796284"),
  "student" : "Rupesh",
  "city" : "Akot"
}
{
  "_id" : ObjectId("618ea12811a01e44a2796285"),
  "student" : "Vimal",
  "city" : "Nashik"
}
{
  "_id" : ObjectId("618ea12811a01e44a2796286"),
  "student" : "Dheeraj",
  "city" : "Bangaluru"
}
>
> show collections
places
>
```



21°C Clear



ENG

10:45 PM

