

# SQL ASSIGNMENT – 15

- Q1. Install MongoDB database server
- Q2. Create a directory/folder to store MongoDB datafiles
- Q3. Set necessary path environment variable
- Q4. Startup MongoDB database
- Q5. Connect to MongoDB database through MongoDB shell command line interface
- Q6. View list of available collections in CDAC database

```
tejas@tejas-IdeaPad-Gaming-3-15IHU6: ~$ sudo systemctl start mongod
[sudo] password for tejas:
tejas@tejas-IdeaPad-Gaming-3-15IHU6: ~$ mongosh
Current Mongosh Log ID: 66092b72a7d0676e7b7b2da8
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.2
Using MongoDB:      7.0.7
Using Mongosh:      2.2.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-03-31T14:51:43.381+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-03-31T14:51:43.669+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-03-31T14:51:43.670+05:30: vm.max_map_count is too low
-----
test> 
```

- Q7. Create a new database named CDAC and connect to it

```
test> use cdac
switched to db cdac
cdac>
```

- Q8. View list of available collections in CDAC database
- Q9. Create a new collection by the name of LIBRARY

```
cdac> show collections
library
cdac> db.library.drop()
true
cdac> show collections

cdac>
```

Q10. Insert the following document in the LIBRARY collection:-

title:'MongoDB programming'

author:'Sameer', likes:100

```
cdac> db.library.insert({title: "MongoDB programming", author: "Sameer", likes : 100})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('66093022a7d0676e7b7b2da9') }
}
cdac> █
```

Q11. View the recently inserted document and note the \_id field

```
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('66093022a7d0676e7b7b2da9'),
    title: 'MongoDB programming',
    author: 'Sameer',
    likes: 100
  }
]
cdac> █
```

Q12. Insert another document in the LIBRARY collection as follows:-

title:'MySQL programming',

authors:['Jack','Jill'], likes:200

```
cdac> db.library.insert({title: "MySQL programming", author: ["Jack","Jill"], likes: 200})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660931cda7d0676e7b7b2daa') }
}
cdac> █
```

Q13. View the inserted documents

```
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('66093022a7d0676e7b7b2da9'),
    title: 'MongoDB programming',
    author: 'Sameer',
    likes: 100
  },
  {
    _id: ObjectId('660931cda7d0676e7b7b2daa'),
    title: 'MySQL programming',
    author: [ 'JACK', 'Jill' ],
    likes: 200
  }
]
cdac> █
```

Q14. View only the first inserted document

```
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('66093022a7d0676e7b7b2da9'),
    title: 'MongoDB programming',
    author: 'Sameer',
    likes: 100
  }
]
cdac> █
```

Q15. View the documents using the pretty() method

```
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('66093022a7d0676e7b7b2da9'),
    title: 'MongoDB programming',
    author: 'Sameer',
    likes: 100
  },
  {
    _id: ObjectId('660931cda7d0676e7b7b2daa'),
    title: 'MySQL programming',
    author: [ 'JACK', 'Jill' ],
    likes: 200
  }
]
cdac> █
```

Q16. Update the document where author name Sameer and change it to Sameer Dehadrai

```
cdac> db.library.update({author: "Sameer"},{$set: {author:"Sameer Dehadrai"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('66093022a7d0676e7b7b2da9'),
    title: 'MongoDB programming',
    author: 'Sameer Dehadrai',
    likes: 100
  },
  {
    _id: ObjectId('660931cda7d0676e7b7b2daa'),
    title: 'MySQL programming',
    author: [ 'JACK', 'Jill' ],
    likes: 200
  }
]
cdac>
```

Q17. Delete all documents that have 100 likes

```
cdac> db.library.remove({likes: 100})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
cdac> db.library.find().pretty()
[
  {
    _id: ObjectId('660931cda7d0676e7b7b2daa'),
    title: 'MySQL programming',
    author: [ 'JACK', 'Jill' ],
    likes: 200
  }
]
cdac>
```

Q18. Drop the LIBRARY collection

```
cdac> db.library.drop()  
true  
cdac>
```

Q19. Drop the CDAC database

```
cdac> db.dropDatabase()  
{ ok: 1, dropped: 'cdac' }  
cdac>
```

Q20. Exit from MongoDB shell

Q21. Stop MongoDB server

```
cdac> exit;  
tejas@tejas-IdeaPad-Gaming-3-15IHU6:~$
```