

## Assignment No. 10

**Assignment Title:** Implement the following MongoDB Query

Create a collection named books.

Insert 5 records with field TITLE, DESCRIPTION, BY, URL, TAGS AND LIKES

1. Insert 1 more document in collection with additional field of user name and comments.
2. Display all the documents whose title is 'mongodb'.
3. Display all the documents written by 'john' or whose title is 'mongodb'.
4. Display all the documents whose title is 'mongodb' and written by 'john'.
5. Display all the documents whose like is greater than 10.
6. Display all the documents whose like is greater than 100 and whose title is either 'mongodb' or written by 'john'.
7. Update the title of 'mongodb' document to 'mongodb overview'.
8. Delete the document titled 'nosql overview'.
9. Display exactly two documents written by 'john'.
10. Display the second document published by 'john'.
11. Display all the books in the sorted fashion.
12. Insert a document using save method.

### Queries:

```
> use TE_B_15
switched to db TE_B_15
> db
TE_B_15
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
> db.movie.insert({"name":"tutorials point"})
WriteResult({ "nInserted" : 1 })
> show dbs
TE_B_15 0.000GB
admin 0.000GB
config 0.000GB
local 0.000GB
> ^C
bye
unix@unix-HP-280-G1-MT:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("19f10663-19d1-422d-9cd0-96d782d1bb4e") }
MongoDB server version: 3.6.8
```

Server has startup warnings:

2023-10-09T14:08:10.030+0530 I STORAGE [initandlisten]

2023-10-09T14:08:10.030+0530 I STORAGE [initandlisten] \*\* WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine

2023-10-09T14:08:10.030+0530 I STORAGE [initandlisten] \*\* See

<http://dochub.mongodb.org/core/prodnotes-filesystem>

2023-10-09T14:08:11.190+0530 I CONTROL [initandlisten]

2023-10-09T14:08:11.190+0530 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.

2023-10-09T14:08:11.190+0530 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.

2023-10-09T14:08:11.190+0530 I CONTROL [initandlisten]

> use TE\_B\_15

switched to db TE\_B\_15

> db

TE\_B\_15

> db.createCollection("Books")

{ "ok" : 1 }

> show collections

Books

movie

> db.Books.insert({title: "MongoDB Overview", description: "MongoDB is NoSQL Database", by: "Shraddha", url: "http://www.mongodb.com", tags: ['mongodb', 'database', 'NoSQL'], likes:100})

WriteResult({ "nInserted" : 1 })

> show Books

2023-10-09T14:37:45.997+0530 E QUERY [thread1] Error: don't know how to show [Books] :

shellHelper.show@src/mongo/shell/utils.js:997:11

shellHelper@src/mongo/shell/utils.js:750:15

@(shellhelp2):1:1

> db.Books.find()

{ "\_id" : ObjectId("6523c2c2050ea82209c3ec22"), "title" : "MongoDB Overview", "description" : "MongoDB is NoSQL Database", "by" : "Shraddha", "url" : "http://www.mongodb.com", "tags" : [ "mongodb", "database", "NoSQL" ], "likes" : 100 }

> db.Books.find().pretty()

```
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
```

```

        "database",
        "NoSQL"
    ],
    "likes" : 100
}
>

```

unix@unix-HP-280-G1-MT:~\$ mongo

MongoDB shell version v3.6.8

connecting to: mongodb://127.0.0.1:27017

Implicit session: session { "id" : UUID("05eb7ff4-6a60-4752-a454-a975e26b8dd0") }

MongoDB server version: 3.6.8

Server has startup warnings:

2023-10-12T13:10:57.532+0530 I STORAGE [initandlisten]

2023-10-12T13:10:57.532+0530 I STORAGE [initandlisten] \*\* WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine

2023-10-12T13:10:57.532+0530 I STORAGE [initandlisten] \*\* See

<http://dochub.mongodb.org/core/prodnotes-filesystem>

2023-10-12T13:11:00.315+0530 I CONTROL [initandlisten]

2023-10-12T13:11:00.315+0530 I CONTROL [initandlisten] \*\* WARNING: Access control is not enabled for the database.

2023-10-12T13:11:00.315+0530 I CONTROL [initandlisten] \*\* Read and write access to data and configuration is unrestricted.

2023-10-12T13:11:00.315+0530 I CONTROL [initandlisten]

> use TE\_B\_15;

switched to db TE\_B\_15

> db

TE\_B\_15

> db.Books.find().pretty()

```

{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}

```

```

> db.Books.insert({title: "NoSQL Database",description: "NoSQL database doesn't have
tables",by: "XYZ",url: "http://www.nosql.com",tags: ["mongodb", "database",
"NoSQL"],likes: 20,comments: [{user:"user1",message: "My first comment",dateCreated:
new Date(2013,11,10,2,35),like: 0}]});
WriteResult({ "nInserted" : 1 })
> db.Books.find().pretty()
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
{
  "_id" : ObjectId("6527abec9891a0125192d1a0"),
  "title" : "NoSQL Database",
  "description" : "NoSQL database doesn't have tables",
  "by" : "XYZ",
  "url" : "http://www.nosql.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 20,
  "comments" : [
    {
      "user" : "user1",
      "message" : "My first comment",
      "dateCreated" : ISODate("2013-12-09T21:05:00Z"),
      "like" : 0
    }
  ]
}
> db.books.find({title:"mongodb"}).pretty()
> ;
> db.books.find({title:"mongodb"}).pretty();
> db.Books.find({title:"mongodb"}).pretty();

```

```

> db.Books.find({title:"MongoDB Overview"}).pretty();
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
> db.Books.find({$or:[{title:"MongoDB Overview"},{by:"Shraddha"}]}).pretty();
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
> db.Books.find({likes:{$gt:10}}).pretty();
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
{
  "_id" : ObjectId("6527abec9891a0125192d1a0"),

```

```

    "title" : "NoSQL Database",
    "description" : "NoSQL database doesn't have tables",
    "by" : "XYZ",
    "url" : "http://www.nosql.com",
    "tags" : [
        "mongodb",
        "database",
        "NoSQL"
    ],
    "likes" : 20,
    "comments" : [
        {
            "user" : "user1",
            "message" : "My first comment",
            "dateCreated" : ISODate("2013-12-09T21:05:00Z"),
            "like" : 0
        }
    ]
}
> db.books.find({$and:[{likes:{$gt:100}},{$or:[{title:"mongodb"},{by:"john"}]}]}).pretty()
> db.Books.find({$and:[{likes:{$gt:100}},{$or:[{title:"MongoDB
Overview"},{by:"Shraddha"}]}]}).pretty()
> db.Books.find({$and:[{likes:{$gt:100}},{$or:[{title:"MongoDB
Overview"},{by:"Shraddha"}]}]}).pretty();
> db.Books.find({$and:[{likes:{$gt:100}},{$or:[{title:"MongoDB
Overview"},{by:"Shraddha"}]}]}).pretty();
> db.Books.find({$and:[{likes:{$gt:10}},{$or:[{title:"MongoDB
Overview"},{by:"Shraddha"}]}]}).pretty();
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
> db.Books.update({title:"MongoDB Overview"},{$set:{title:"mongodb"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Books.find().pretty();

```

```

{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "mongodb",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 100
}
{
  "_id" : ObjectId("6527abec9891a0125192d1a0"),
  "title" : "NoSQL Database",
  "description" : "NoSQL database doesn't have tables",
  "by" : "XYZ",
  "url" : "http://www.nosql.com",
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "likes" : 20,
  "comments" : [
    {
      "user" : "user1",
      "message" : "My first comment",
      "dateCreated" : ISODate("2013-12-09T21:05:00Z"),
      "like" : 0
    }
  ]
}
}
> db.Books.remove({title:"NoSQL Database"});
WriteResult({ "nRemoved" : 1 })
> db.Books.find({by:"Shraddha"}).pretty().limit(1)
{
  "_id" : ObjectId("6523c2c2050ea82209c3ec22"),
  "title" : "mongodb",
  "description" : "MongoDB is NoSQL Database",
  "by" : "Shraddha",
  "url" : "http://www.mongodb.com",
  "tags" : [

```

```
        "mongodb",
        "database",
        "NoSQL"
    ],
    "likes" : 100
}
> ^C
bye
```