CL-2005: Database Systems
Lab # 12: Introduction To Mongo DB

Objective:

To learn Mongo DB.

Scope:

The student shall know the following:

- SQL Commands.
- SQL Queries
- SQL schema
- Hands-on experience with the above-mentioned concepts.

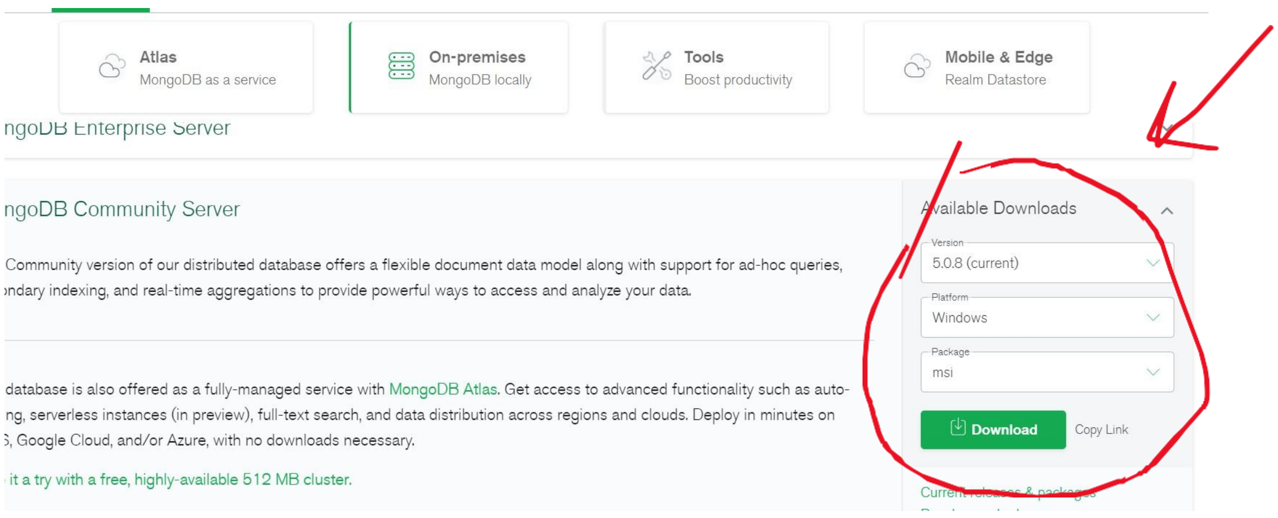
Discussion:

Installation:

Click on the below link to download Mongo DB community server.

https://www.mongodb.com/try/download/community

Now from below screen shot, select following requirements as shown below.



ngoDB Enterprise Server

ngoDB Community Server

Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries,
ondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-
ng, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on
5, Google Cloud, and/or Azure, with no downloads necessary.

it a try with a free, highly-available 512 MB cluster.

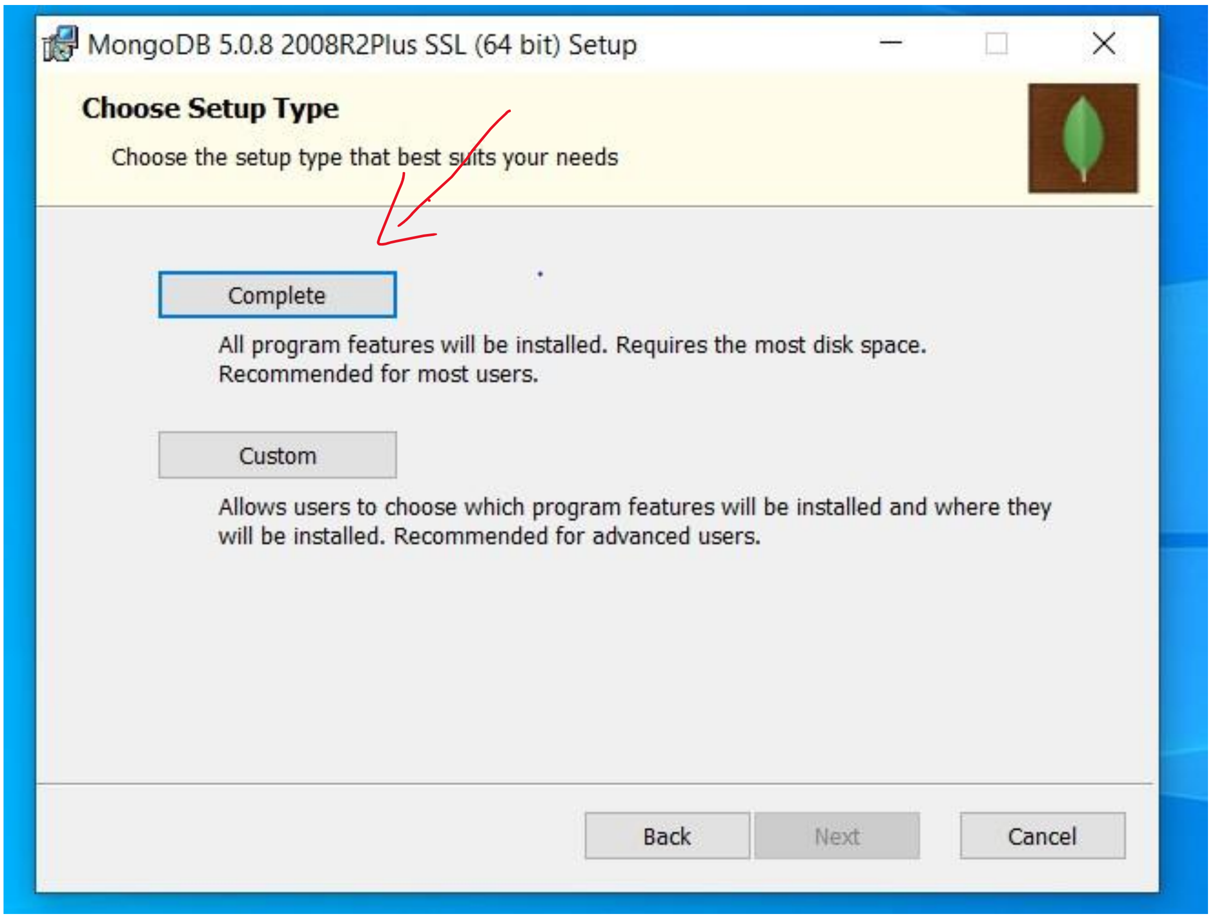
Available Downloads

Version: 5.0.8 (current)

Platform: Windows

Package: msi

Now after download, open installer and select **Complete** and install the software


MongoDB 5.0.8 2008R2Plus SSL (64 bit) Setup

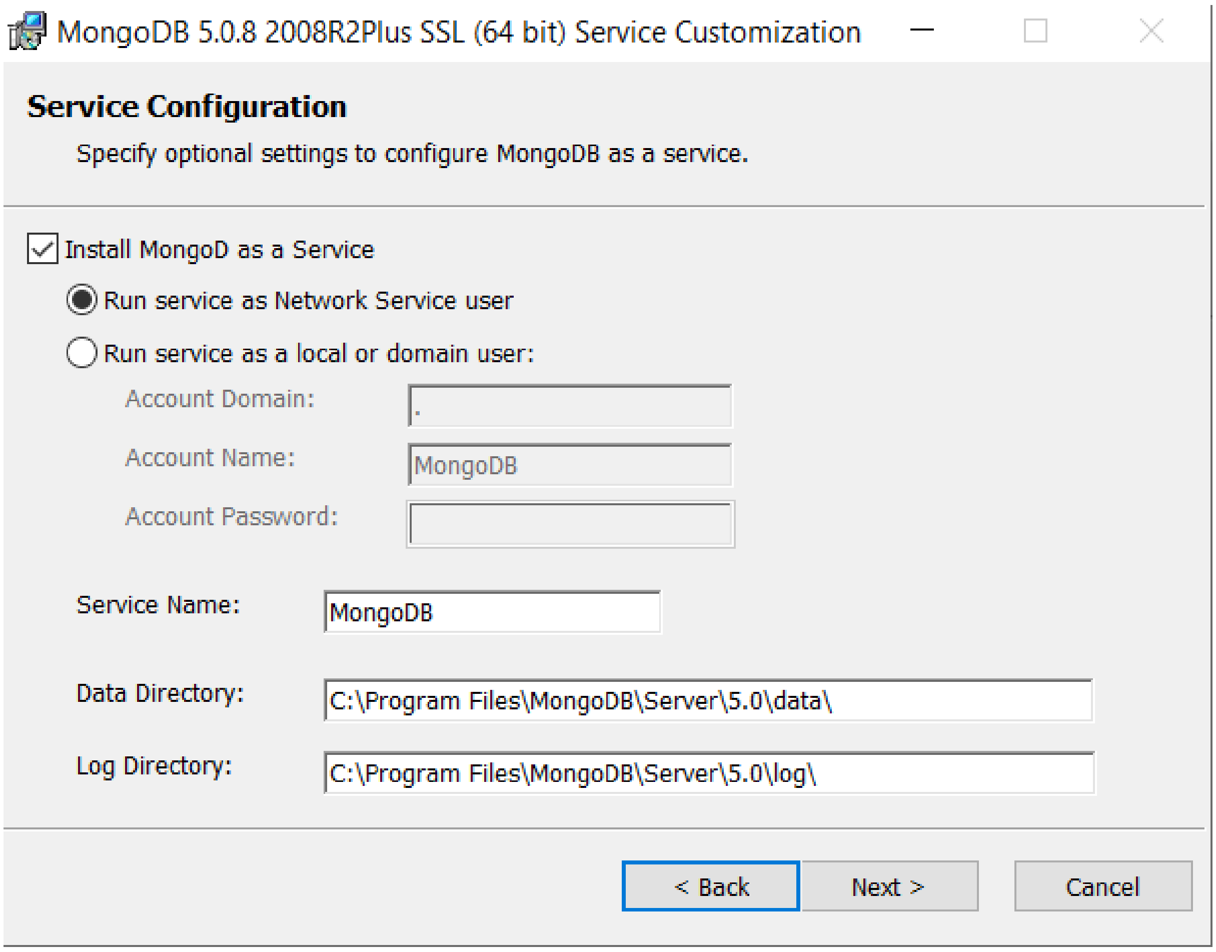
Choose Setup Type

Choose the setup type that best suits your needs

All program features will be installed. Requires the most disk space. Recommended for most users.

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

Now click Next in below screenshot



MongoDB 5.0.8 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

[✓] Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

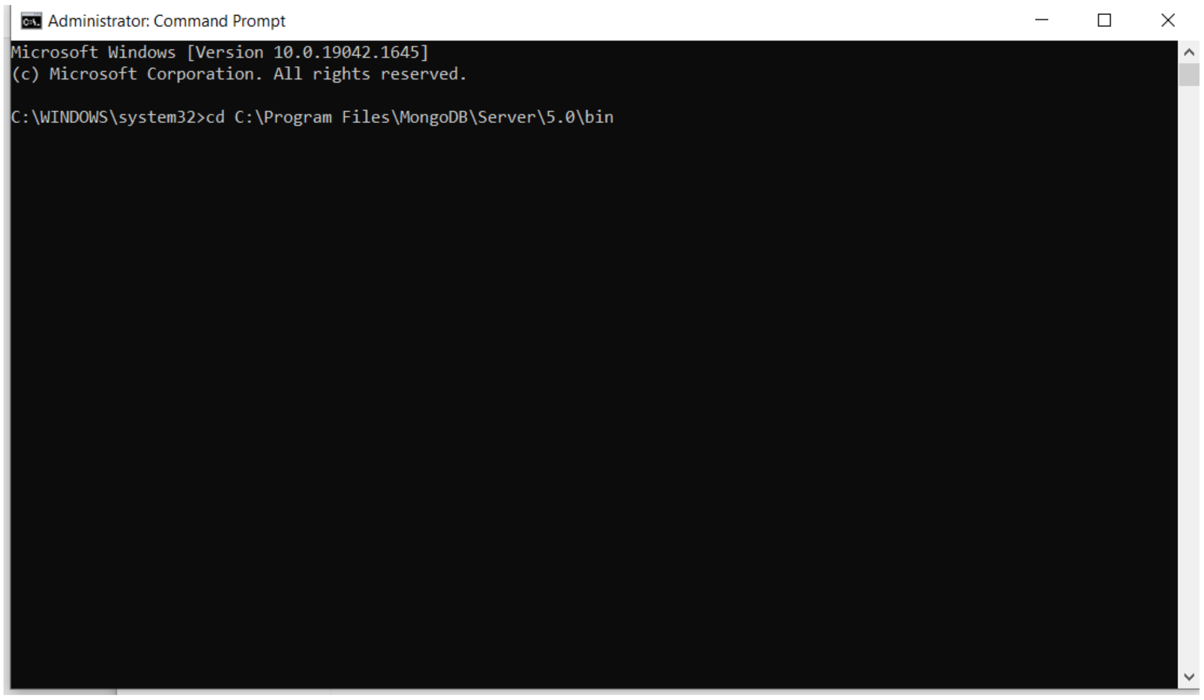
Account Domain:	.
Account Name:	MongoDB
Account Password:	
Service Name:	MongoDB
Data Directory:	C:\Program Files\MongoDB\Server\5.0\data\
Log Directory:	C:\Program Files\MongoDB\Server\5.0\log\

Now after installation head towards the following directory of Mongo DB installation:

C:\Program Files\MongoDB\Server\5.0\bin

Above path is the default installation path. If you have installed manually somewhere else then your path can be different.

Now open command prompt and change directory to the above path by following command.

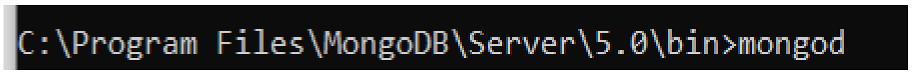


```
Microsoft Windows [Version 10.0.19042.1645]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\MongoDB\Server\5.0\bin
```

Now create directory **C:\data\db**

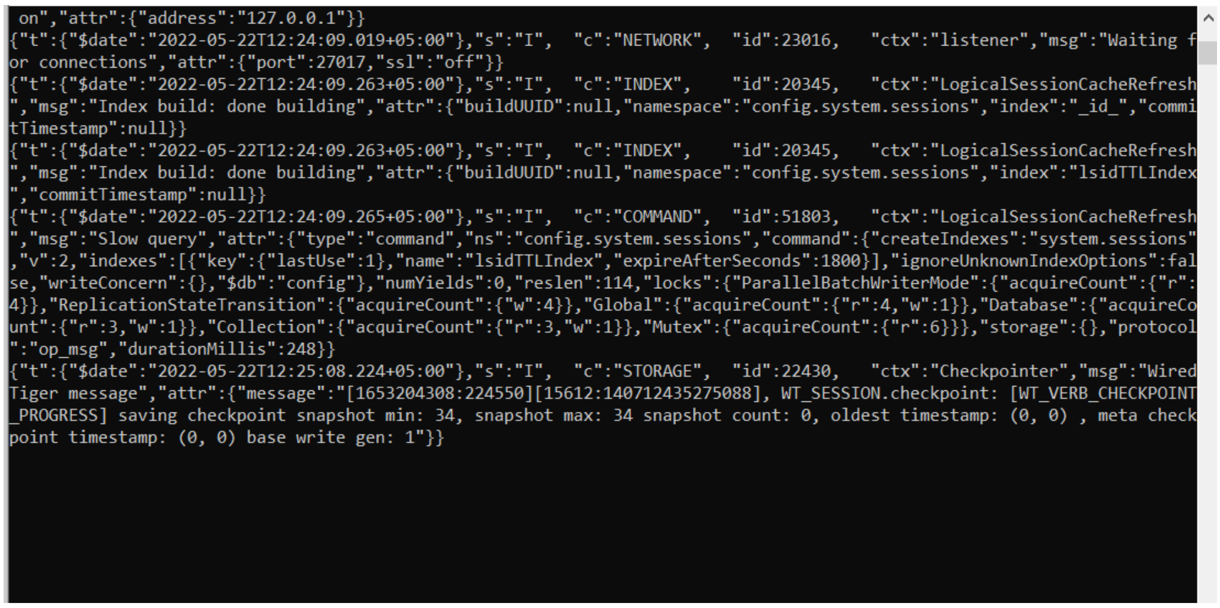
For this you can simply first create new folder in C drive and rename it “data”. After that open that folder and create “db” folder in it.

Now in command prompt run command ‘mongod’ as shown below.



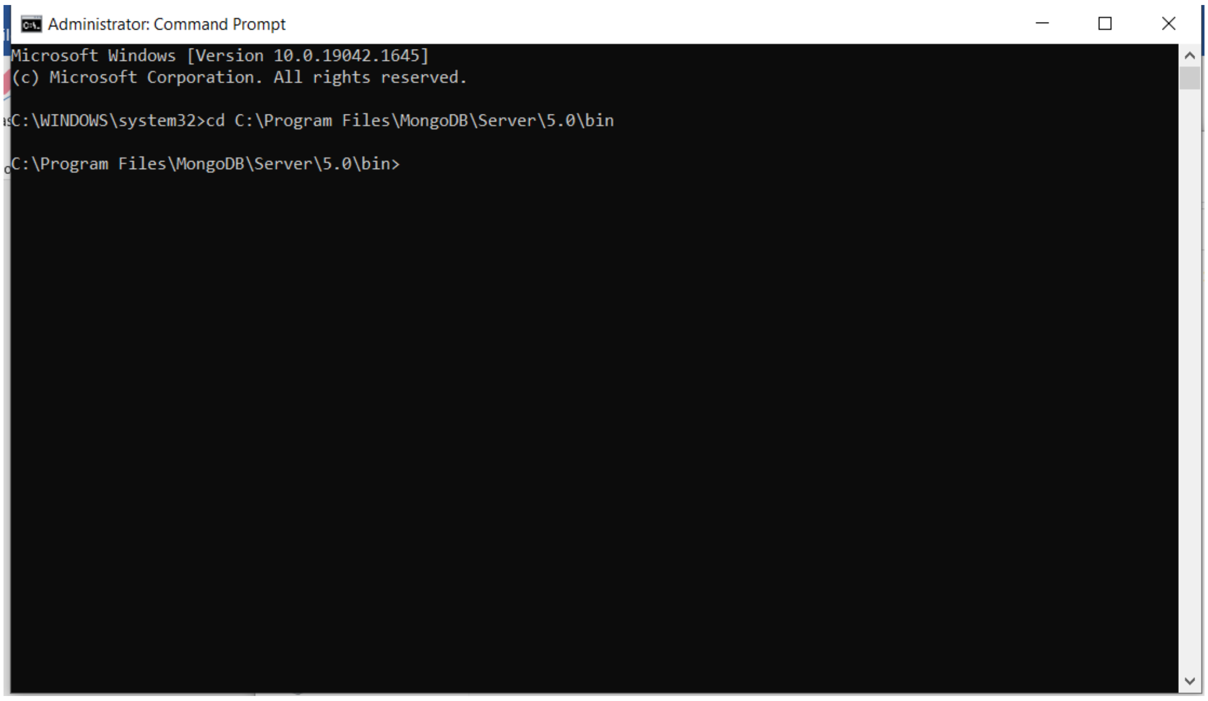
```
C:\Program Files\MongoDB\Server\5.0\bin>mongod
```

Following will be the result of this command



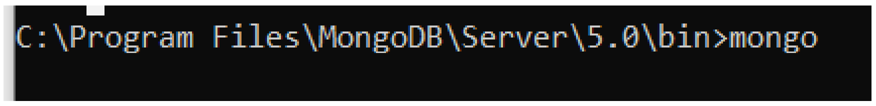
```
 on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2022-05-22T12:24:09.019+05:00"},"s":"I",  "c":"NETWORK",  "id":23016,   "ctx":"listener","msg":"Waiting f
or connections","attr":{"port":27017,"ssl":"off"}}
{"t":{"$date":"2022-05-22T12:24:09.263+05:00"},"s":"I",  "c":"INDEX",    "id":20345,   "ctx":"LogicalSessionCacheRefresh
","msg":"Index build: done building","attr":{"buildUUID":null,"namespace":"config.system.sessions","index":"_id_","commi
tTimestamp":null}}
{"t":{"$date":"2022-05-22T12:24:09.263+05:00"},"s":"I",  "c":"INDEX",    "id":20345,   "ctx":"LogicalSessionCacheRefresh
","msg":"Index build: done building","attr":{"buildUUID":null,"namespace":"config.system.sessions","index":"lsidTTLIndex
","commitTimestamp":null}}
{"t":{"$date":"2022-05-22T12:24:09.265+05:00"},"s":"I",  "c":"COMMAND",  "id":51803,   "ctx":"LogicalSessionCacheRefresh
","msg":"Slow query","attr":{"type":"command","ns":"config.system.sessions","command":{"createIndexes":"system.sessions"
,"v":2,"indexes":[{"key":{"lastUse":1},"name":"lsidTTLIndex","expireAfterSeconds":1800}],"ignoreUnknownIndexOptions":fal
se,"writeConcern":{},"$db":"config"},"numYields":0,"reslen":114,"locks":{"ParallelBatchWriterMode":{"acquireCount":{"r":
4}},"ReplicationStateTransition":{"acquireCount":{"w":4}},"Global":{"acquireCount":{"r":4,"w":1}},"Database":{"acquireCo
unt":{"r":3,"w":1}},"Collection":{"acquireCount":{"r":3,"w":1}},"Mutex":{"acquireCount":{"r":6}}},"storage":{},"protocol
":"op_msg","durationMillis":248}}
{"t":{"$date":"2022-05-22T12:25:08.224+05:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"Checkpointer","msg":"Wired
Tiger message","attr":{"message":"[1653204308:224550][15612:140712435275088], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT
_PROGRESS] saving checkpoint snapshot min: 34, snapshot max: 34 snapshot count: 0, oldest timestamp: (0, 0) , meta check
point timestamp: (0, 0) base write gen: 1"}}
```

Now your MongoDB server is running. Don’t close this command prompt. Minimize it and open another terminal in same directory.

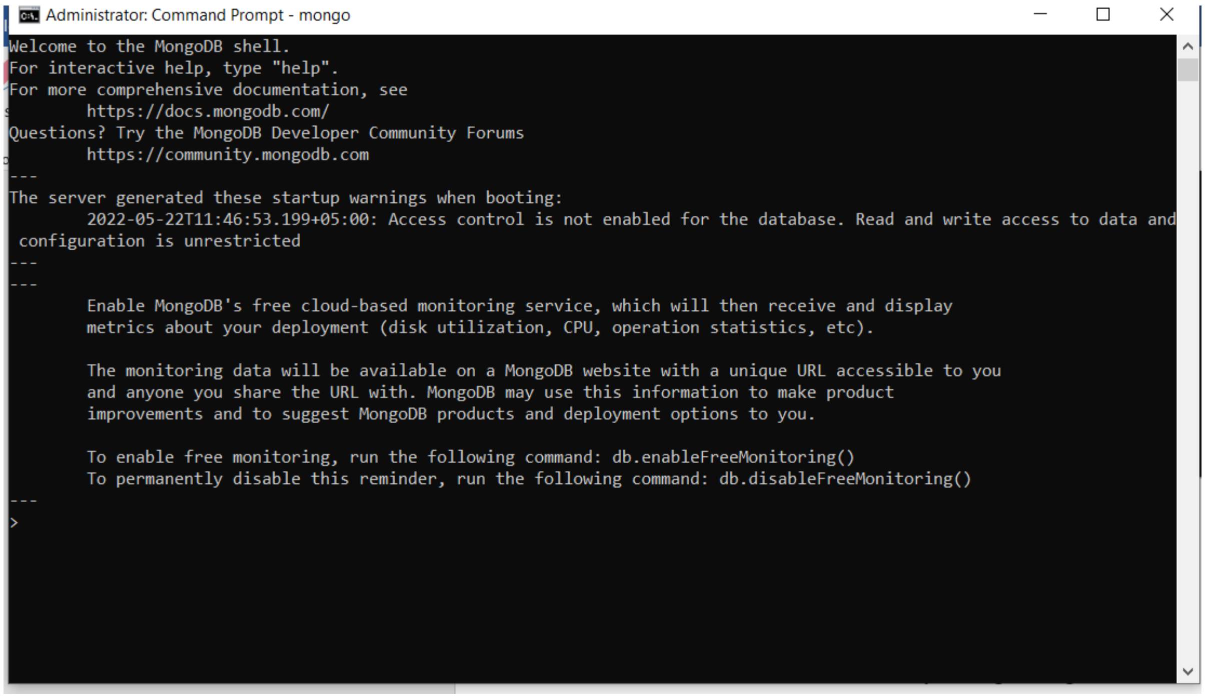


```
Microsoft Windows [Version 10.0.19042.1645]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\MongoDB\Server\5.0\bin

C:\Program Files\MongoDB\Server\5.0\bin>
```

Now run the command ‘mongo’



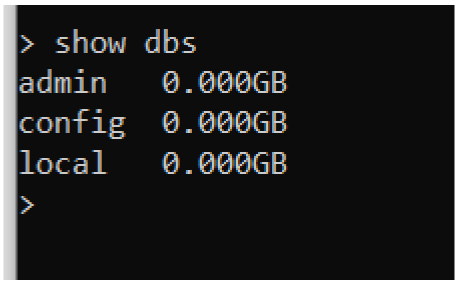
```
C:\Program Files\MongoDB\Server\5.0\bin>mongo
```



```
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2022-05-22T11:46:53.199+05:00: 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()
---
>
```

Now your mongo DB commands can be executed in this terminal. Now first lets try some basic commands to check list of databases we have.

You can runn **show dbs** to see your storage. As we have no database our result is following:



```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

The USE Command:

MongoDB **use DATABASE_NAME** is used to create database. The command will create a new database, if it doesn't exist otherwise, it will return the existing database

Basic syntax of use DATABASE statement is as follows:

```
use DATABASE_NAME
```

```
>use mydb

switched to db mydb
```

The DROP Command:

MongoDB **db.dropDatabase ()** command is used to drop a existing database.

Basic syn tax of dropDatabase () command is as follows:

```
db.dropDatabase()
```

Example:

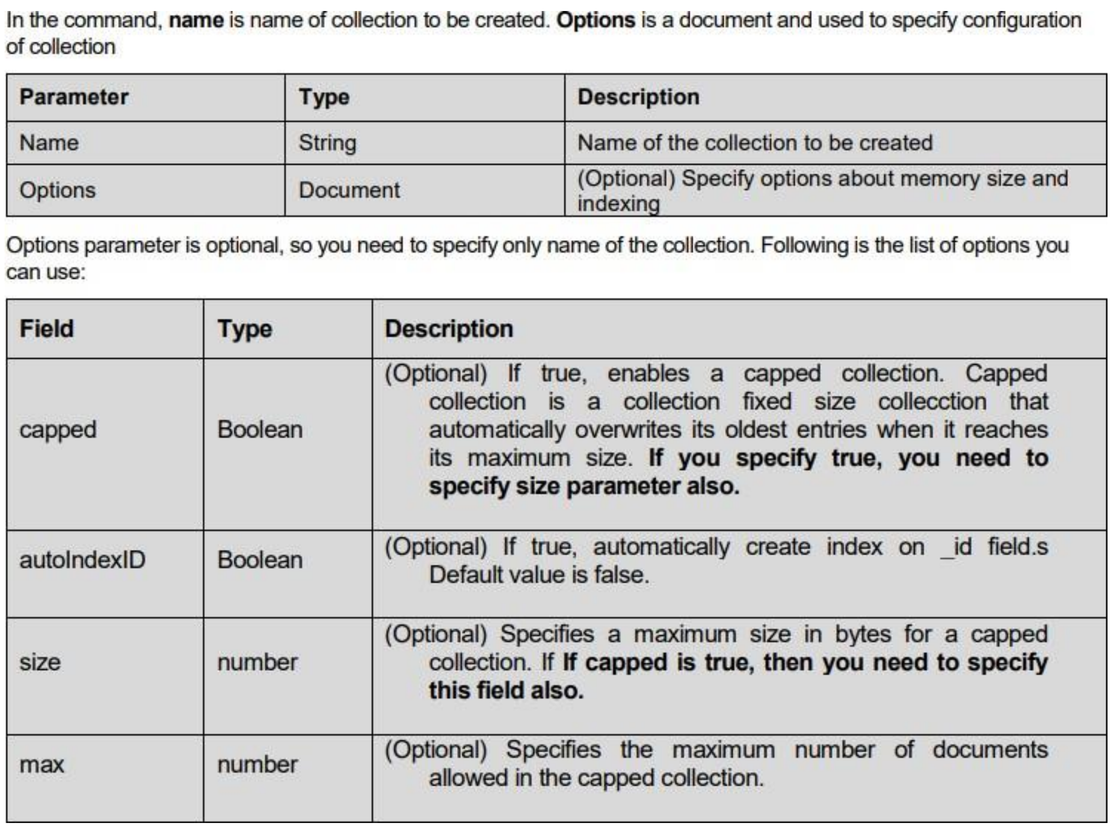
```
>use mydb
switched to db mydb
>db.dropDatabase()
>{ "dropped" : "mydb", "ok" : 1 }
>
```

Creating Collection in Mongo DB:

MongoDB **db.createCollection(name, options)** is used to create collection.

Basic syntax of createCollection() command is as follows

```
db.createCollection(name, options)
```


In the command, **name** is name of collection to be created. **Options** is a document and used to specify configuration of collection

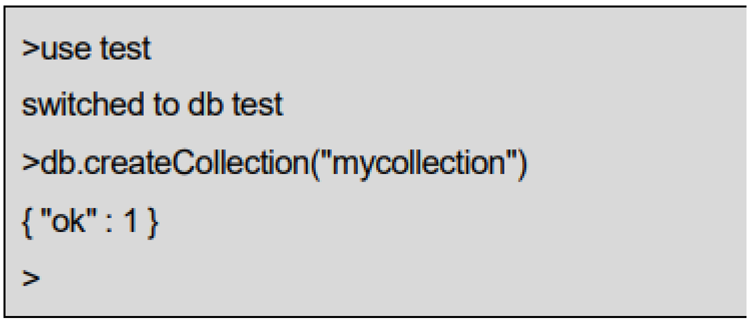
Parameter	Type	Description
Name	String	Name of the collection to be created
Options	Document	(Optional) Specify options about memory size and indexing

Options parameter is optional, so you need to specify only name of the collection. Following is the list of options you can use:

Field	Type	Description
capped	Boolean	(Optional) If true, enables a capped collection. Capped collection is a collection fixed size collecction that automatically overwrites its oldest entries when it reaches its maximum size. **If you specify true, you need to specify size parameter also.**
autoIndexID	Boolean	(Optional) If true, automatically create index on _id field.s Default value is false.
size	number	(Optional) Specifies a maximum size in bytes for a capped collection. If **If capped is true, then you need to specify this field also.**
max	number	(Optional) Specifies the maximum number of documents allowed in the capped collection.

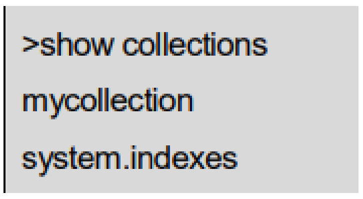
Example:

Below we have created collection “mycollection”



```
>use test
switched to db test
>db.createCollection("mycollection")
{ "ok" : 1 }
>
```

We can also see list of all collections in our database by using ‘show collections’.



```
>show collections
mycollection
system.indexes
```

Dropping Collection in Mongo DB:

You can drop any collection by the following command:

```
db.COLLECTION_NAME.drop()
```

Mongo DB – Insert Document:

To insert data into MongoDB collection, you need to use MongoDB's insert() or save()method.

Basic syntax of insert() command is as follows:

```
>db.COLLECTION_NAME.insert(document)
```

Example:

```
>db.mycol.insert({
  _id: ObjectId(7df78ad8902c),
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100
})
```

If the collection doesn't exist in the database, then MongoDB will create this collection and then insert document into it.

Mongo DB – Find Method:

To query data from MongoDB collection, you need to use MongoDB's find() method.

Basic syntax of find() method is as follows

```
>db.COLLECTION_NAME.find()
```

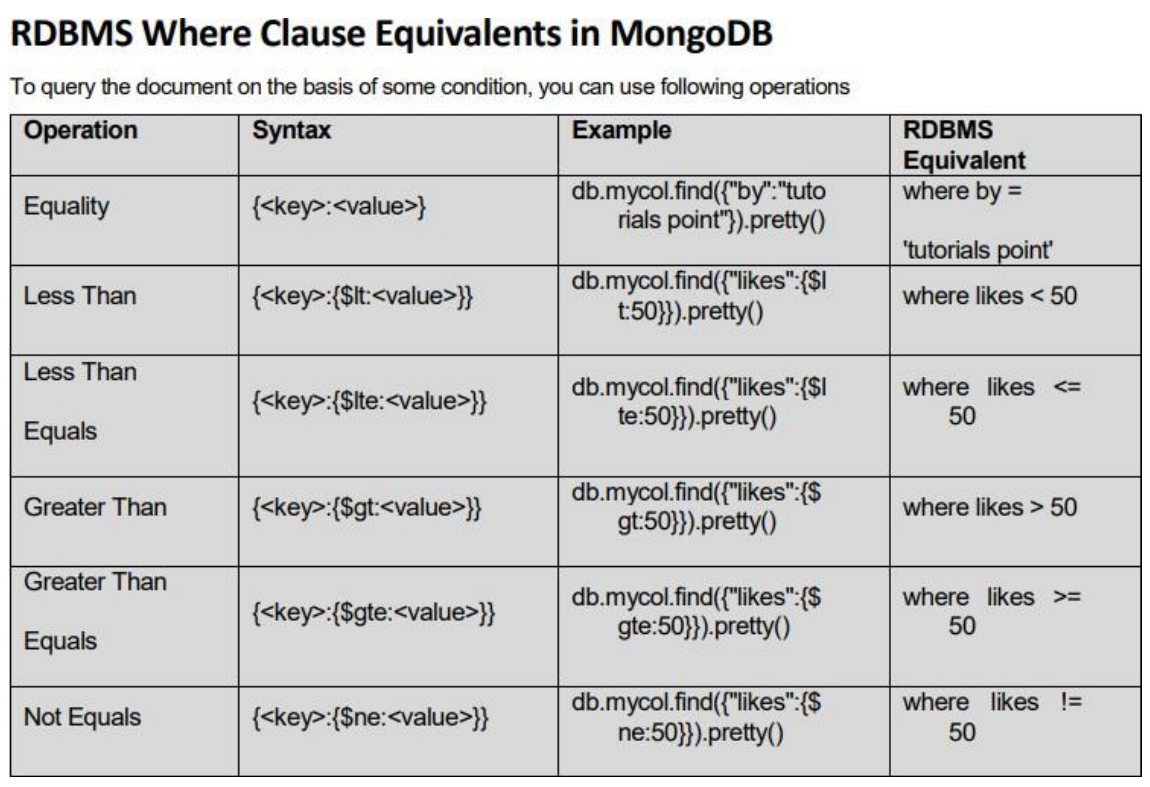
find() method will display all the documents in a non-structured way.

Alternatively, you can use Pretty method to display in structured form.

```
>db.mycol.find().pretty()
```

Example:

```
>db.mycol.find().pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
  "by": "tutorials point",
  "url": "http://www.tutorialspoint.com",
  "tags": ["mongodb", "database", "NoSQL"],
  "likes": "100"
}
>
```



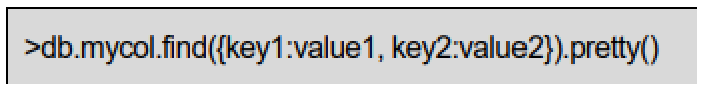
RDBMS Where Clause Equivalents in MongoDB

To query the document on the basis of some condition, you can use following operations

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value>}	db.mycol.find({"by":"tutorials point"}).pretty()	where by = 'tutorials point'
Less Than	{<key>:{$lt:<value>}}	db.mycol.find({"likes":{$lt:50}}).pretty()	where likes < 50
Less Than Equals	{<key>:{$lte:<value>}}	db.mycol.find({"likes":{$lte:50}}).pretty()	where likes <= 50
Greater Than	{<key>:{$gt:<value>}}	db.mycol.find({"likes":{$gt:50}}).pretty()	where likes > 50
Greater Than Equals	{<key>:{$gte:<value>}}	db.mycol.find({"likes":{$gte:50}}).pretty()	where likes >= 50
Not Equals	{<key>:{$ne:<value>}}	db.mycol.find({"likes":{$ne:50}}).pretty()	where likes != 50

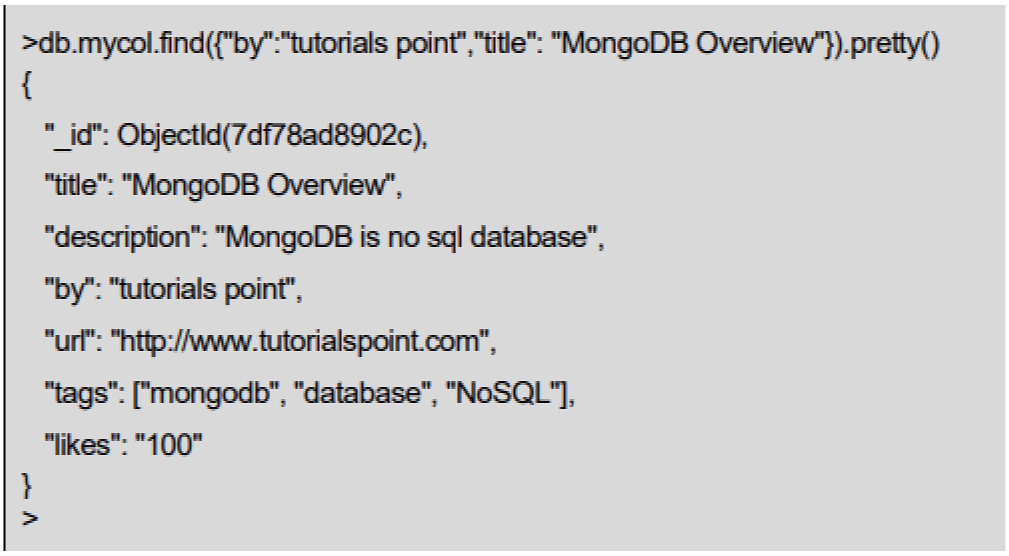
Mongo DB – AND statement:

In the find() method if you pass multiple keys by separating them by ',' then MongoDB treats it AND condition. Basic syntax of AND is shown below:



```
>db.mycol.find({key1:value1, key2:value2}).pretty()
```

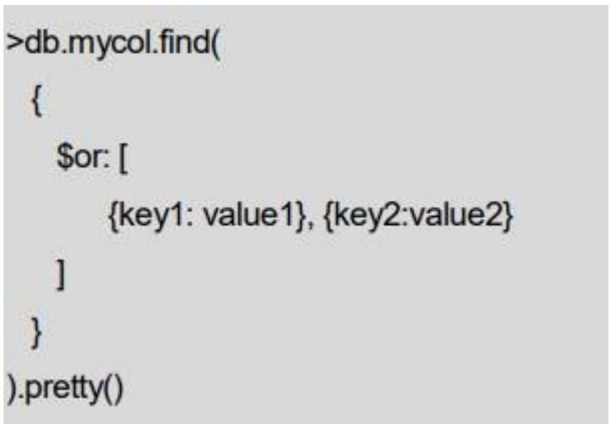
Example:



```
>db.mycol.find({"by":"tutorials point","title": "MongoDB Overview"}).pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
  "by": "tutorials point",
  "url": "http://www.tutorialspoint.com",
  "tags": ["mongodb", "database", "NoSQL"],
  "likes": "100"
}
>
```

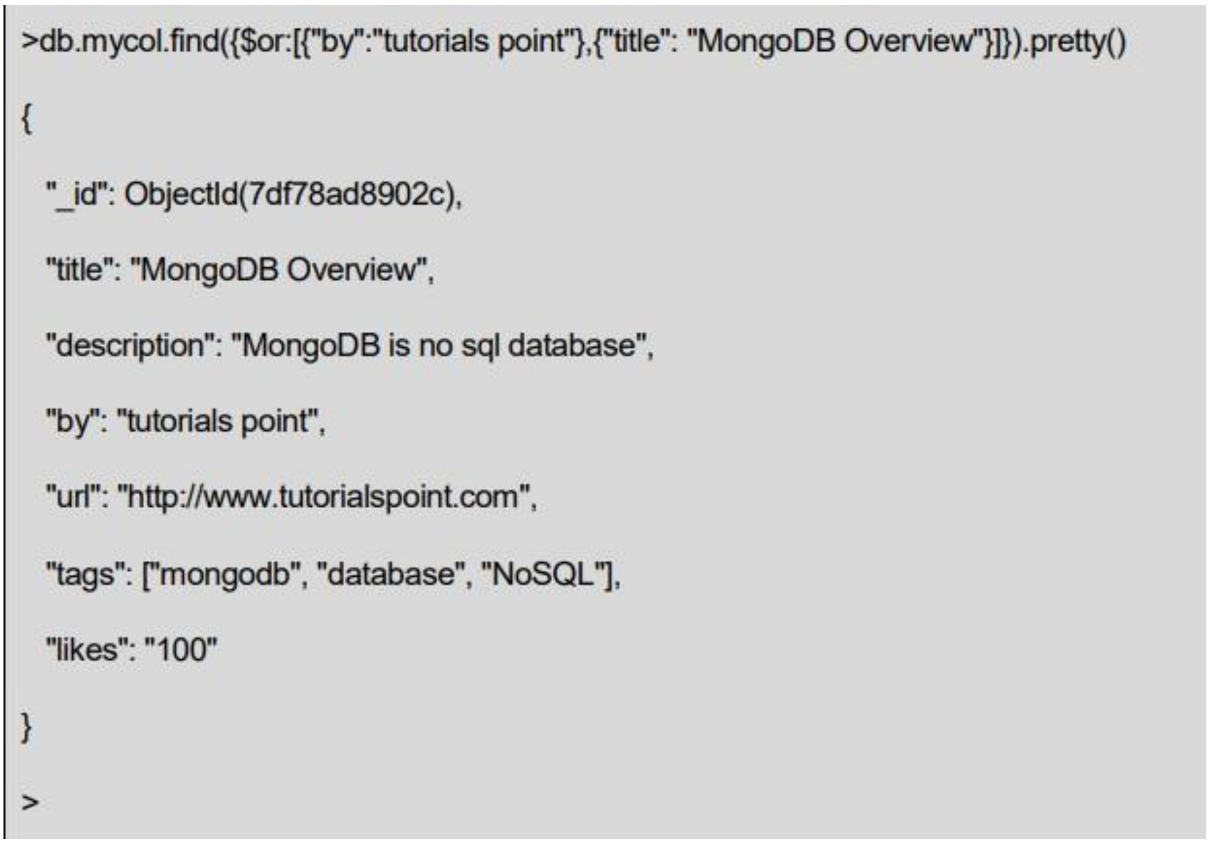
Mongo DB – OR statement:

To query documents based on the OR condition, you need to use $or keyword. Basic syntax of OR is shown below:



```
>db.mycol.find(
  {
    $or: [
        {key1: value1}, {key2:value2}
    ]
  }
).pretty()
```

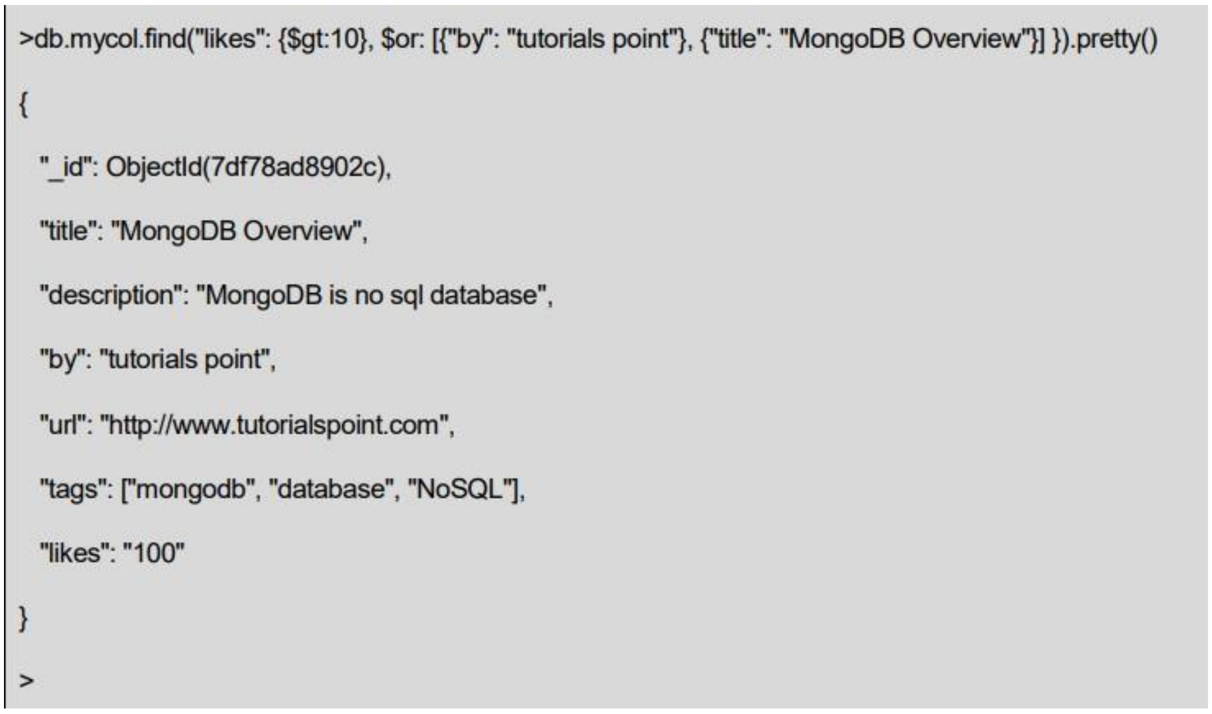
Example:



```
>db.mycol.find({$or:[{"by":"tutorials point"},{"title": "MongoDB Overview"}]}).pretty()

{

  "_id": ObjectId(7df78ad8902c),

  "title": "MongoDB Overview",

  "description": "MongoDB is no sql database",

  "by": "tutorials point",

  "url": "http://www.tutorialspoint.com",

  "tags": ["mongodb", "database", "NoSQL"],

  "likes": "100"

}

>
```

Mongo DB – AND - OR Together:

Below given example will show the documents that have likes greater than 100 and whose title is either 'MongoDB Overview' or by is 'tutorials point'. Equivalent sql where clause is 'where likes>10 AND (by = 'tutorials point' OR title = 'MongoDB Overview')'



```
>db.mycol.find("likes": {$gt:10}, $or: [{"by": "tutorials point"}, {"title": "MongoDB Overview"}] }).pretty()

{

   "_id": ObjectId(7df78ad8902c),

   "title": "MongoDB Overview",

   "description": "MongoDB is no sql database",

   "by": "tutorials point",

   "url": "http://www.tutorialspoint.com",

   "tags": ["mongodb", "database", "NoSQL"],

   "likes": "100"

}

>
```

EXAMPLE SCHEMA IN MONGO DB

Example Suppose a client needs a database design for his blog website and see the differences between RDBMS and MongoDB schema design. Website has the following requirements.

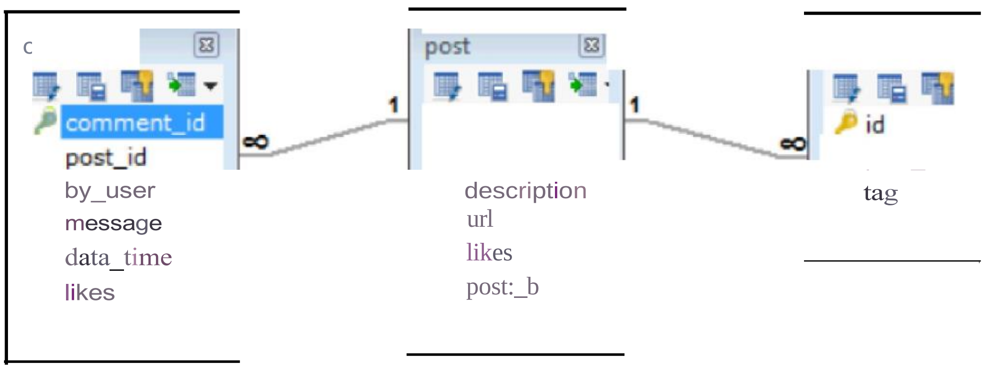
• Every post has the unique title, description and url.

• Every post can have one or more tags.

• Every post has the name of its publisher and total number of likes.

• Every Post have comments given by users along with their name, message, data-time and likes.

• On each post there can be zero or more comments. In RDBMS schema design for above requirements will have minimum three tables.

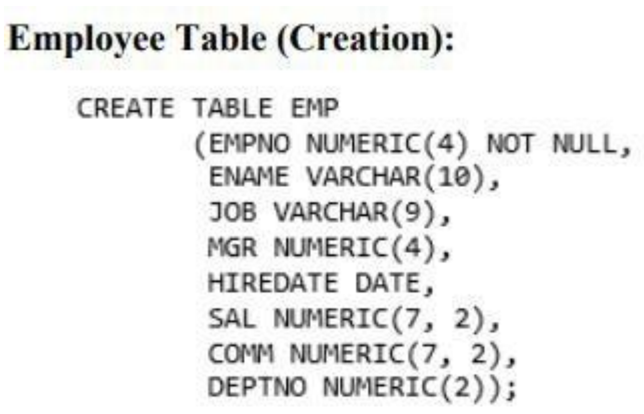


While in Mongo☐B sdhema design will have one collection post and has lle following s1mcture.:

```
  id: POST ID
title: TITLE_OF_POST,
description: POST_DESCRIPTION,
by: POST_BY,
url: URL_OF_POST.
tags: [TAG1, TAG2, TAG3],
likes: TOTAL_LIKES,
comments [

     user:'OOMMENT  BY',
     message: TEXT,
     dateCr-eatedl: DATE_TIME,
     like: LIKES
  },
  {
     user:'OOMMENT BY',
     message: TEXT,
     dateCr-eatedl: DATE_TIME,
     like: LIKES

]
```

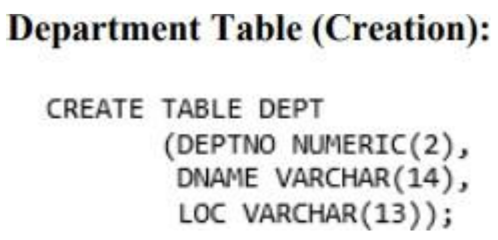
TASK

- You have to recreate database that was created in LAB-5 which had following tables



Employee Table (Creation):

```
CREATE TABLE EMP
        (EMPNO NUMERIC(4) NOT NULL,
         ENAME VARCHAR(10),
         JOB VARCHAR(9),
         MGR NUMERIC(4),
         HIREDATE DATE,
         SAL NUMERIC(7, 2),
         COMM NUMERIC(7, 2),
         DEPTNO NUMERIC(2));
```



Department Table (Creation):

```
CREATE TABLE DEPT
        (DEPTNO NUMERIC(2),
         DNAME VARCHAR(14),
         LOC VARCHAR(13));
```

- Now convert above RDBMS into MongoDB schema and create collection.
- Insert at least 6 documents.

Display following queries

- List all the employee names, their salaries.
- List all the employee names, jobs, and salaries who have not been given any commission.
- List all the employee identity numbers, names, jobs, and salaries are greater than 1500
- List all the employee names and jobs who are not CLERK, ANALYST, and SALESMAN.
- Name of employees in specific department.