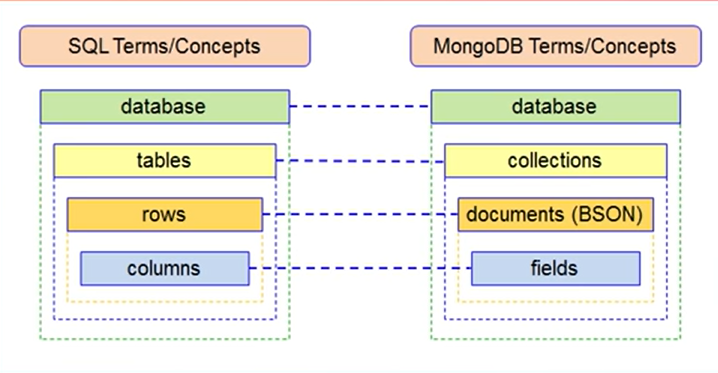
Mongo DB

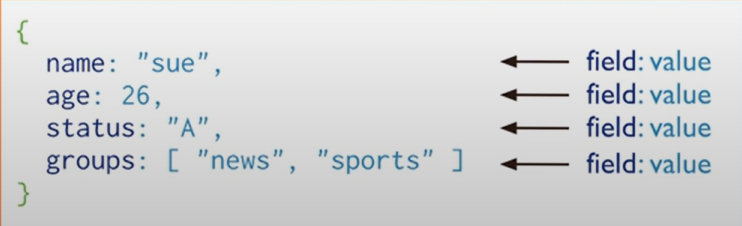
1. MongoDB is a document database designed for ease of development and scalling
2. It is intuitive and easy to use enterprise edition
3. Available as community and enterprise edition
4. “mongo” is the command-line shell that connects to a specific instance of mongod
5. “mongod” is the “Mongo Daemon” its basically the host process for the database
6. The difference between the MySQL(relational database) and the mongoDB is expressed as below :



1. In MongoDB the data is stored in the form of the collections



1. And in the collections the data is represent as the :



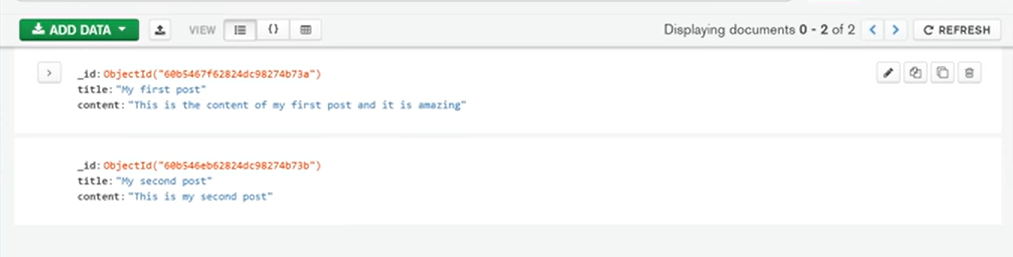
1. To install the MongoDB go to the mongodb.com and than install the mongoDB community server and download it
2. For making the use of the mongoDB via the cmd or power shell, we have to go to the C:\Program Files\MongoDB\Server\5.0\bin and copy this location and add to the environment variable’s path, So now we can use the mongoDB via the cmd or power shell
3. We can also use the mongoDB by mongo compass also
4. We can also connect the mongoDB which is on the external server like that of the aws or etc and than we can also connect to it by the help of the mongoDB Compass
5. We can also run the mongoDB on the local system by connecting it with the mongod which is already installed and can be accessed by the **mongodb://localhost:27017** statatment or port, And we can connect to it by clicking the connect button below in the compass
6. And than go to the databases anf there you can use the existing database and also make the new database by the clicking of the create database and while creating we should give the one collection name
7. In mongoDb we can also make an capped collection and due to it when the data more than the limit is being stored than the old data would be automatically deleted

🡪Example, if we put the cap of the 200 than when the 201th data is being entered than the 1st data in that collection would be deleted and this process goes on

1. We can insert the data in the database by clicking on the **Add Data** in compass and than insert document and enter the title and the content in it in the double quotation marks, and the every content is separated by the coma example:



1. And than we can see the inserted data in the compass as :



1. And we can also edit the data by clicking on it in the compass, we can also change the type of the value and even also make the new field
2. In an collection we can make the data’s field more or less than the other, It is allowed in the mongoDB but it is not allowed in the schemas in the MYSQL
3. In the mongosh which is in the end of the mongo compass, we can also write the command of the mongo
4. Now, We will not use the compass which is an user interface, we will use the cmd or the power shell for now for learning
5. So, before writing any command in the terminal, we first have to write the mongo in it to active the mongod in the system
6. And now we can write the different commands in it, suppose the **show dbs** is used to see the databases in the system
7. As there was the primary key in the MQSQL, there is the ObjectId in the mongoDB by which we can uniquely identify an object
8. We can also use the mongoDB atlas to make the database on the server, the one of it is free to use
9. And by going to the mongoDB atlas, First build your first cluster and add the details asked by them
10. Than make an username and password
11. Than add the current ip address to it, so that we can remotely access it using that ip for using it in the compass and we can also make change in the database
12. And than the cluster will be created, And to connect it with the local system’s compass go to the connect button🡪connect using MongoDB Compass🡪copy the connection link shown there and paste it in the compass🡪And than you will be connected to the cluster and do any thing

**All MongoDb commands you will ever need (MongoDb Cheatsheet)**

In this post, we will see a comprehensive list of all the MongoDB commands you will ever need as a MongoDB beginner. This list covers almost all the most used commands in MongoDB.

I will assume that you are working inside a collection named 'comments' on a MongoDB database of your choice

**Database Commands**

1. **View all databases**

show dbs

1. **Create a new or switch databases**

use dbName

1. **View current Database**

db

1. **Delete Database**

db.dropDatabase()

**Collection Commands**

1. **Show Collections**

show collections

1. **Create a collection named 'comments'**

db.createCollection('comments')

1. **Drop a collection named 'comments'**

db.comments.drop()

**Row(Document) Commands**

1. **Show all Rows in a Collection**

db.comments.find()

1. **Show all Rows in a Collection (Prettified)**

db.comments.find().pretty()

1. **Find the first row matching the object**

db.comments.findOne({name: 'Harry'})

1. **Insert One Row**

db.comments.insert({

'name': 'Harry',

'lang': 'JavaScript',

'member\_since': 5

})

1. **Insert many Rows**

db.comments.insertMany([{

'name': 'Harry',

'lang': 'JavaScript',

'member\_since': 5

},

{'name': 'Rohan',

'lang': 'Python',

'member\_since': 3

},

{'name': 'Lovish',

'lang': 'Java',

'member\_since': 4

}])

1. **Search in a MongoDb Database**

db.comments.find({lang:'Python'})

1. **Limit the number of rows in output**

db.comments.find().limit(2)

1. **Count the number of rows in the output**

db.comments.find().count()

1. **Update a row**

db.comments.update({name: 'Shubham'},

{'name': 'Harry',

'lang': 'JavaScript',

'member\_since': 51

}, {upsert: true})

1. **Mongodb Increment Operator**

db.comments.update({name: 'Rohan'},

{$inc:{

member\_since: 2

}})

1. **Mongodb Rename Operator**

db.comments.update({name: 'Rohan'},

{$rename:{

member\_since: 'member'

}})

1. **Delete Row**

db.comments.remove({name: 'Harry'})

1. **Less than/Greater than/ Less than or Eq/Greater than or Eq**

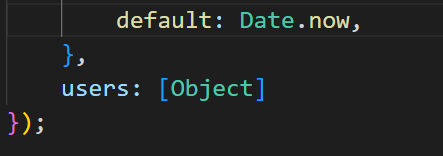
db.comments.find({member\_since: {$lt: 90}})

db.comments.find({member\_since: {$lte: 90}})

db.comments.find({member\_since: {$gt: 90}})

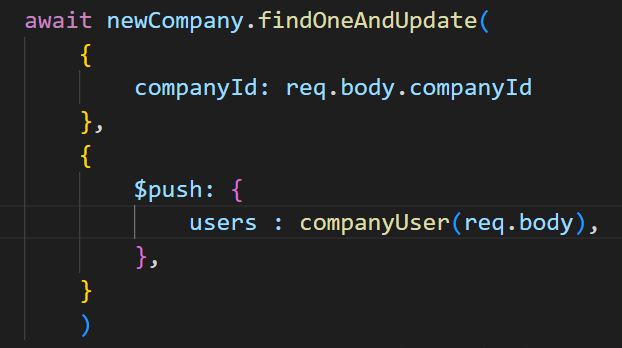
db.comments.find({member\_since: {$gte: 90}})

🡪To store an schema in the other schema in the form of the array we have to use the following method:



🡪We have to make the array symbol as show in the figure in it we have to metion the object, if we want to insert only string or number only than we have to mention that only

🡪And now by the help of the following code we can add the Object of the any schema in the array



🡪In this method we have used the inbuilt function named the findOneAndUpdate and in which we will search an existing card which meets our company id in the json and where it meets we will add the our object schema which is companyUser which is imported from an .js file and by the help of the $push method we will push it in the array which is defined by the name users

🡪We can also find the wheather the document with the specific item and specific value is present in the database or not by the help of the following method

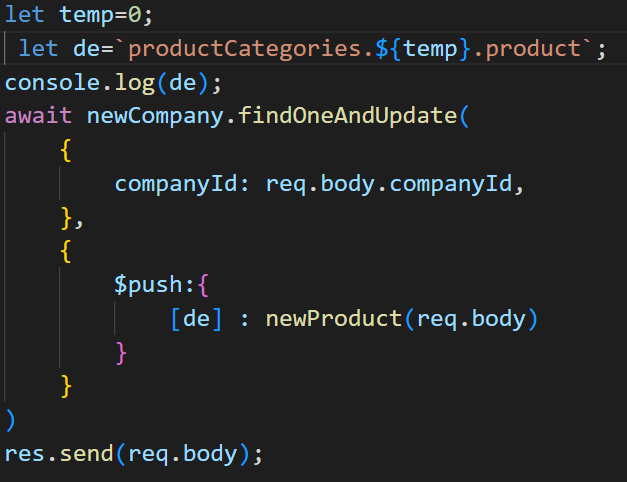


🡪We can find inside an array of the array by the help of the statement like the following one:

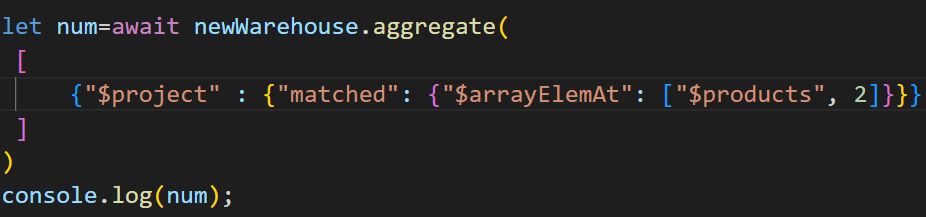


🡪If the document containing all the requirments exists than the notes would come with the documents containing all the documents in the form of the newCompany model and we can see them by making them first save in an variable and than res.send(<Variable-name>), And if there are no document fulfilling such conditions are present than it will give nothing in the output it will return []

🡪We can also use the java-script variable while working with the mongoDB by the help of only the square brackets than only it will take the value from js otherwise it will not take the value from the js, Example is as follow :

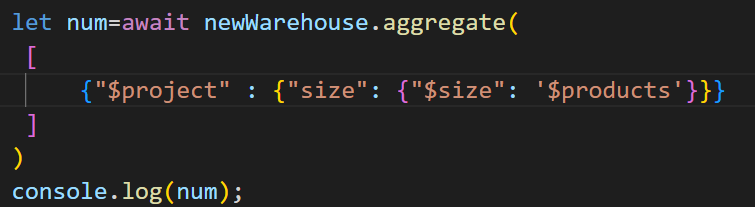


🡺We can retrive data from the mongoDB’s collection’s array by the help of the following statement



🡪Where **newWarehouse** is the collection name, **products** is the name of array present in the collection, So change only two things keep other things as it is

🡺We can get the size of the array which is stored in the collection of the mongoDB by the help of the following method:

  
🡪Here, only two things are variable other are not to be varied, The things that are tobe varied are newWarehouse which is an collection and the other one is products which is the name of the array present in the collection of the newWarehouse