MongoDB

It is a no-sql database, it stores unstructured data in the database

In RDBMS you have to store structured data in a table

i.e., Assume a user table has user\_id, user\_name & dob: then you can have the data only for 3 columns maximum, it is structured, you can’t have extra data if you need at the later stage, its possible only if you update the whole table with new columns

In MongoDB you can have different set of data for each record if you wish to have, in mongodb you don’t have tables to store records, you have something called documents which represents records, these documents are stored in the collection which is like a table.

Collection: Group of documents

Document: Record/Row in the collection that will have data which could be unstructured data

In MongoDB documents are not represented as columns instead documents are represented in key value pairs which are enclosed in {}, like javascript object

{key:value, key:value,…}

A value can be another document also, means it can have embedded document with key & value

ex:   
{name: “Alex”, “dob”:”2000-10-15”, phone : [99999,893939]}

{name : {first:”Alex”, last:”Jack”}, “dob”:”2000-12-16”}

You can perform operations in mongodb using the inbuilt functions provided by mongodb, like SQL has queries MongoDB has functions

You can download the mongodb.zip which is an installation folder, once you extract the folder you can start mongodb using a command called mongod, this command starts mongodb service and looks for a folder in c:/ drive, that folder is C:/data/db, it will store all the database informations in the c drive, if you are able to start the mongod then you can interact with the mongodb using mongo shell, mongo is another command which interacts with mongodb to perform operations on the MongoDB.

It means at the beginning you should be using 2 commands:

>> mongod - starts the mongodb database

>> mongo - it opens a mongodb terminal which is shell where you can perform operation on the mongodb database

Minimum 2 command prompts we must use

1. mongod
2. mongo

Your command prompt should recognize the path of the mongod & mongo hence set path in both the command prompts.

How to make mongod to locate the data/db in different drive

>> mongod --dbpath D://data//db

* Above command helps mongod to search data/db in D drive

>> mongod

* Above command searches data/db in C drive

By default mongodb runs in 27017, but we can change the port as well

>> mongod --port 27018 --dbpath D://data/db

* Above command starts mongodb in 27018 port & searches the data/db in D drive
* Now mongo command needs to connect to 27018, as by default it connects to 27017, for that you need to use

>> mongo --port 27018

Some of the important functions you can use in MongoDB are:

show databases: shows the databases list

use database\_name: switches to the database

db.createCollection(“collection-name”): creates the collection like a table

show collections: lists all the collections in the database

db.collection-name.insertOne(…): to insert one document

db.collection-name:insertMany(..): to insert bulk document

db.collection-name.find(..): to retrieve the document

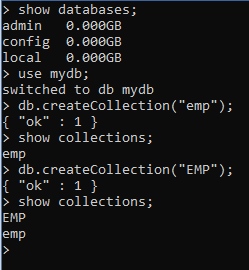
db.collection-name.remove(..): to remove the document

db.collection-name.updateOne(..): to update the document

db.collection-name.updateMany(..): to update many documents

db.dropDatabase(): to drop the database

Note: MongoDB is case sensitive.

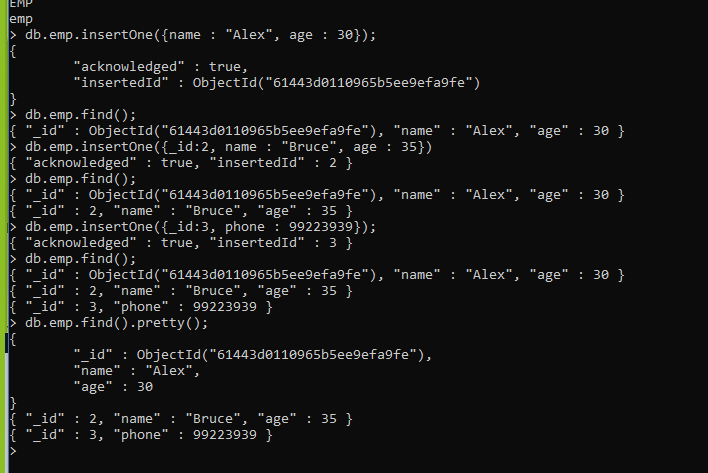


You can notice two collections with the same name emp is created as MongoDB is case sensitive

How to store the document in the collection

db.collection\_name.insertOne({key : value,…});

Ex: db.emp.insertOne({name:”…”, age:30});



The above commands like

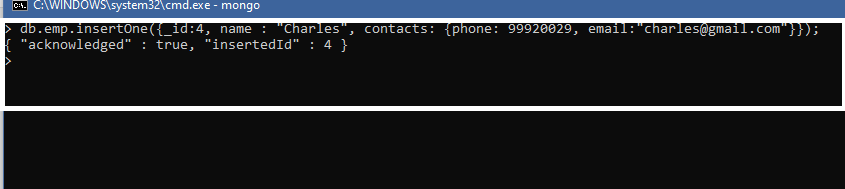
insertOne(): is storing the document, and status returns the auto-generated id if not passed, else returns the id that is passed, here \_id is the primary key column

find(): it returns all the documents present in the collection

find().pretty(): it is for formatting the output:

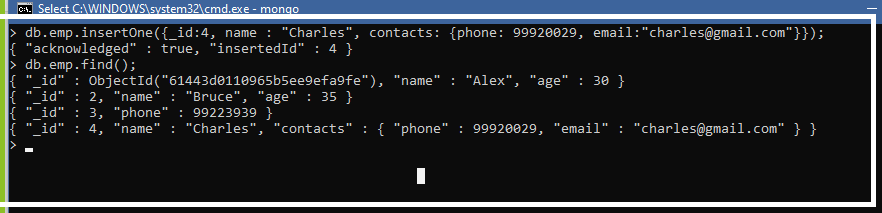
How to store the embedded document in a document

Embedded means a value that keeps another value in document format

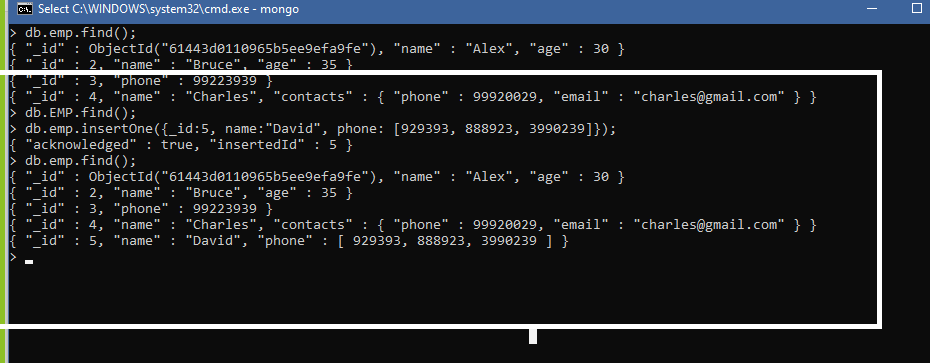


Here contacts is storing phone & email inside it

You can use find() to see the output



We can also store arrays in the document



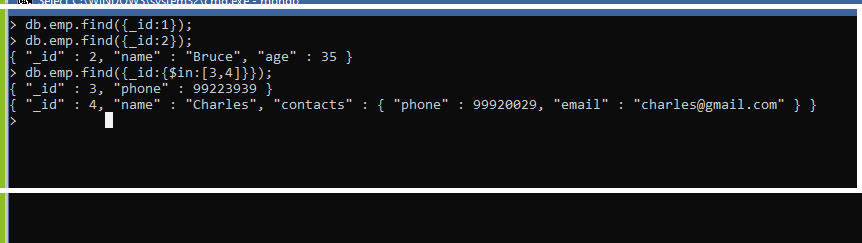
Exercise: Create a collection with the name users store

* name which will have firstname & lastname as an embedded document
* contacts which will have email-id & phone numbers as an array
* dob which will have date hint: new Date(“yyyy-MM-dd”)

How to find data based on some conditions

db.emp.find({\_id:1}): returns the document having \_id:1

db.emp.find({\_id:{$in:[1,2]}}): select documents applying conditions in the list

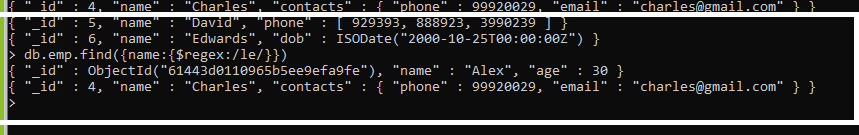


You can also apply conditions within the range

You can use $gt, $lt, $regex

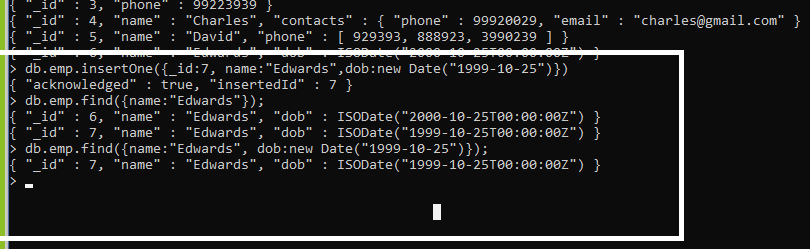
db.emp.find({\_id:{$gt:2, $lt:4}}): returns the document between 2 and 4

db.emp.find({name:{$regex:/le/}}): returns the document having the characters ‘le’



You can also apply multiple conditions:

db.emp.find({name:”Alex”, dob:new Date(“2000-10-20”)})



Bulk Insert: You can store multiple document at the same time using insertMany([{},{}]);

