MongoDB Shell or Mongo

* Mongo means mongo shell
* Collection of documents (objects) are called collection to show run show collections
* Data base consist set of collections to show run show dbs
* For check data in collection run db.collectionName.find().pretty()
* All in one db.collectionName.find({ name: “adil” }).pretty().limit(2)
* For active databse run db

1. Install it with complete option
2. Create a environmental variable
3. Run mongo –version to check that it installs or not

Create Database, collection, and documents

* To create or use Database we use (use dataBaseName)
* To create collection and insert document in selecting database we use (db.collectionName.insertOne({name: “adil younas” }) )

Important note: You Can Delete Database Also But Make One Thing Asure That You Using The Database You Want To Delete

db.dropDatabase()

Important note: You Can rename collection

db.collectionName.renameCollection(“newName”,true)

Important note: You Can delete collection

db.collectionName.drop()

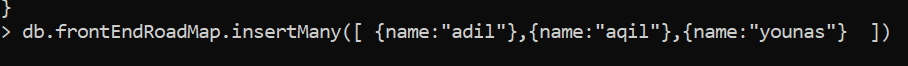
MongoDB Shell CRUD Operations on document

<https://www.mongodb.com/docs/manual/crud/>

Create

insertOne( { } ) => insertOne

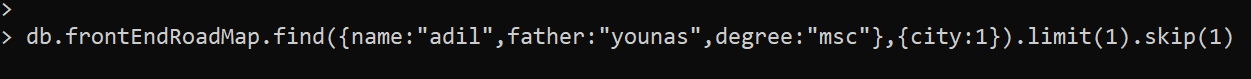
insertMany( [ { },{ },{ } ] ) => insertMany



Read



* Query means I have so many adil but find adil whose father name is younas and did msc degree.
* Projection means I have so many fields there so, if 1 then show only one field which is city but if it is 0 then hide city and show all (hide = 0 or show = 1) so, zoom it,
* Limit(1) means show the first match
* Limit(1).Skip(1) means show only one but skip the first one



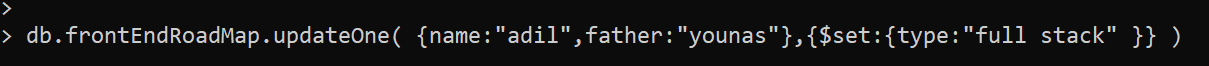
Update

Query = > searching key value pairs

$set = > is a operator

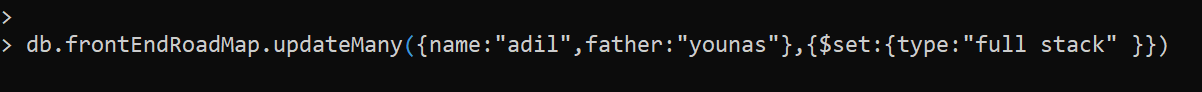
Update only one match field

Db.collectionName.updateOne({query},{$set:{ key: value}} )



Update all match fields

Db.collectionName.updateMany({query},{$set:{ key: value}} )

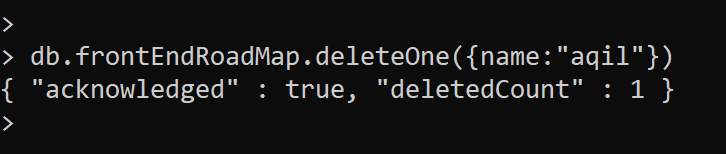


Delete

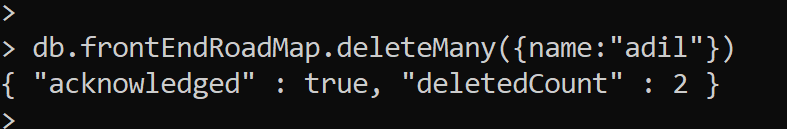
delete one match fields

db.collectionName.deleteOne( { } ) Enter

&

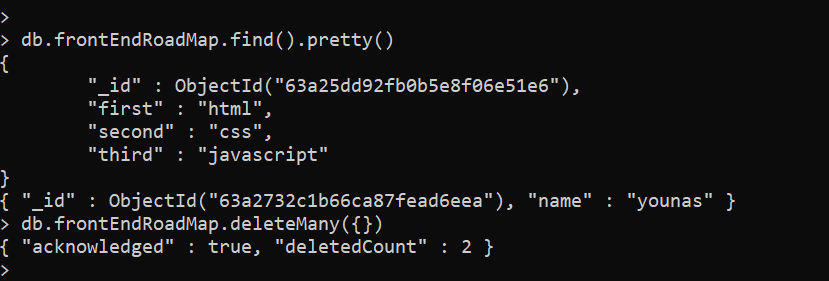


delete match fields



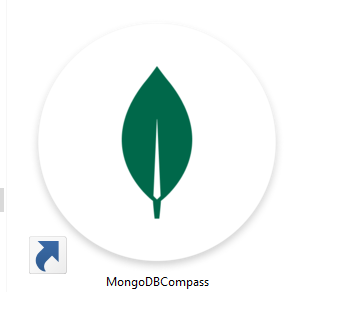
Delete documents present in collection

db.collectionName.deleteMany( { } ) Enter



MongoDBShell die here….Now we discuss MongoDB compass

MongoDB compass is an application through which you can perform all the action above in very simple manners



Mongoose

<https://www.npmjs.com/package/mongoose>

“Mongoose = We Get Data From (HTML Forms, Postman) And Land It On URL through node using express And Mongoose Send That Data To Database”.

Note: An error kicked my balls when I use this code. This code officially belong to mongoose document which is not working in my case.

const mongoose = require('mongoose');

mongoose.connect('mongodb://127.0.0.1:27017/test')

.then(() => console.log('Connected!'));

Correct Code

const mongoose = require('mongoose');

mongoose.set('strictQuery', true); //remove error

mongoose.connect('mongodb://127.0.0.1:27017/students')

  .then(() => console.log('Connected!'))

.catch((e)=>console.log(e))

Create Schema, Create Model/collections and run a async function to put value in model

1. First you must define the schema
2. Create collection or model and in it define you are using “Student” as a collection name and students as a schema like (“Student”,students)
3. Because we must ensure we put data after receiving the data so we use async await function
4. We can push one or multi objects in create( ) like create( [ { },{ } ] ) create( { } )

//how to create schema

const students = new mongoose.Schema({

  name: String,

  workout: Boolean,

  height: Number,

});

//how to create model/collections

//collection name, schema you define

const Students = new mongoose.model("Student", students);

// 3. put data/document in schema

const adder = async () => {

  const data = await Students.create({

    name: "adil",

    workout: true,

    height: 6,

  });

};

adder();

We can push multiple objects in create() function like

const adder = async () => {

  const data = await Students.create([

    {

      name: "adil",

      workout: true,

      height: 6,

    },

    {

      name: "aqil",

      workout: true,

      height: 5,

    },

  ]);

};

adder();

collectionName.find()

const adder = async () => {

  const data = await Students.find()

  console.log(data);

};

adder();

Operators

<https://www.mongodb.com/docs/manual/reference/operator/query/>

const adder = async () => {

  const data = await Students.find({height:{$eq:6}})

  console.log(data);

};

adder();

Note: Remember this syntax and forgot everything

{ age:{ $eq:50} }

* Operators are use with values

Following are the difficult one

$regex = attached with the query value

$options = for string case insensitivity