Mongo queries-

1)to stop mongo DB service

net stop MongoDB

2)db.stats()

Outputs some stats about database

2)how to drop a db-

use databaseName

db.dropDatabase()

3)to Drop a collection-

db.myCollection.drop()

how to start mongo db server manually it -

1)mongod

starts the database server(if you have data/db in c drive i.e root directory)

otherwise run this-

mongod --dbpath "path to db folder"

2)

a)mongod --port 27018(by default port is 27017)

b)then for db client use this-

mongo --port 27018

Mongo commands-

1)show dbs

2)use shop(connects with shop db , will create this db,if it does not exsts)

inactual database is only created when you insert some value there. it also returns

a unique id of document created.type of id is ObjectId

this is special data provided by mongo db. it can be used to sort data as it has some

timestamp information.

3) db.products.insertOne({name: "A Book", price: 12.99})

inserts the data and returns the unique id. also creates the collection if it is not there.

it also assigns a unique id, if not given. you can give your id by uisng \_id.it can be number

or any other string. it needs to be unique

4)db.products.find() , note withupdate and delete, it is required to pass {}, returns the cursor(first 20 records)

or

db.products.find().pretty(), pretty is not supported on findOne

or

db.products.find({})

or

db.products.find({}).toArray()

or

db.passengers.find().forEach(passengerData => printjson(passengerData));

db.flightData.find({distance: {$gt: 10000}}).pretty()

5)to see type-

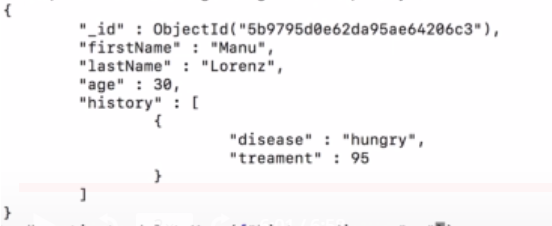
typeof db.numbers.findOne().a

5)db.passengers.find({hobbies: "sports"}).pretty()

here hobies is array ["sports", "coding"]. but mongo db figures it out.it know we mean that one of elements

of array is sports

6)db.patients.deleteMany({"history.disease": "cold"})



here history is array of objets, where disases is one of proeprties.

history: [{disease: hungry, treatment: 95}, {}]. impr- wrap it in quotation marks

5)projection-

db.passengers.find({},{name:1, \_id:0}).pretty()

this will retrieve only name

6)lets say we want to get hobbies of one record, then use-

db.passengers.findOne({name: "Albert Twostone"}).hobbies

it gives us array.note you have to use findOne, so that mongodb knows that we are accessing one record.

we can access properties on document, not on cursor. findMany returns document

5)db.flightData.updateOne({distance: 12000},{$set: {marker: "delete"}})

with update command you have to use atomic operators

6)db.flightData.insertMany([{},{}])

7) > db.flightData.replaceOne({ "\_id" : ObjectId("5cd2644bd6045fd4b709236a")},{

... "departureAirport": "MUC",

... "arrivalAirport": "SFO",

... "aircraft": "Airbus A380",

... "distance": 12000,

... "intercontinental": true

... })