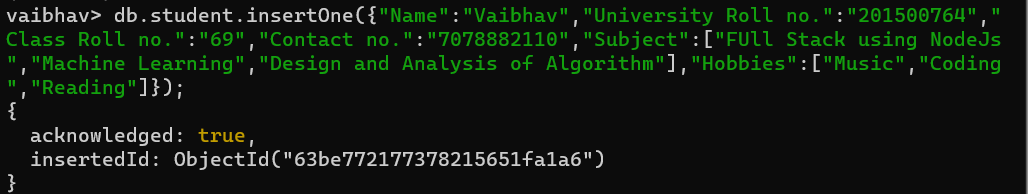
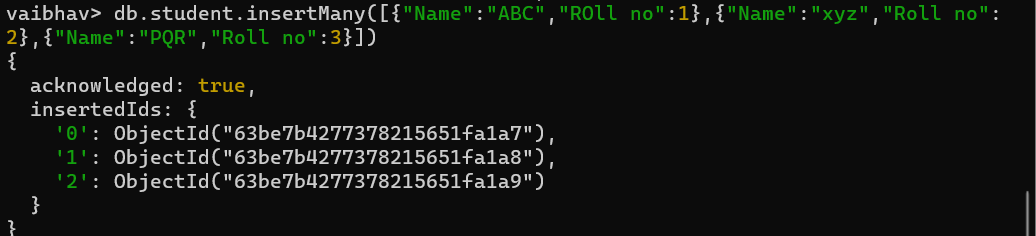
Different Commands and their usage

* Use <database name> - To create a database
* 
* db.createCollection(“<collection name>”) – To create a collection



* db.<collection name>.insertOne({}) – to insert data inside a collection



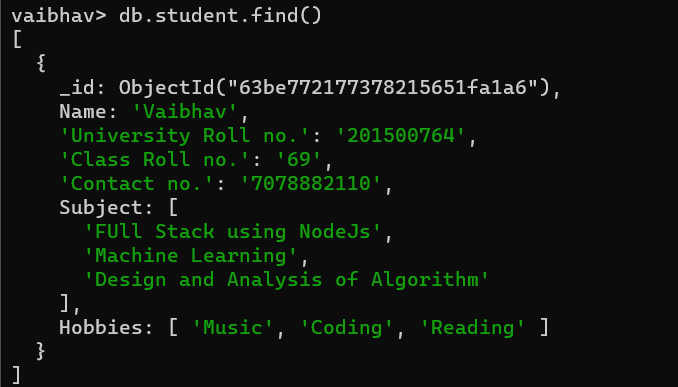
* db.<collection name>.insertMany([{},{},{}]) – to insert many records at a time
* 
* db – it shows in which database we are working



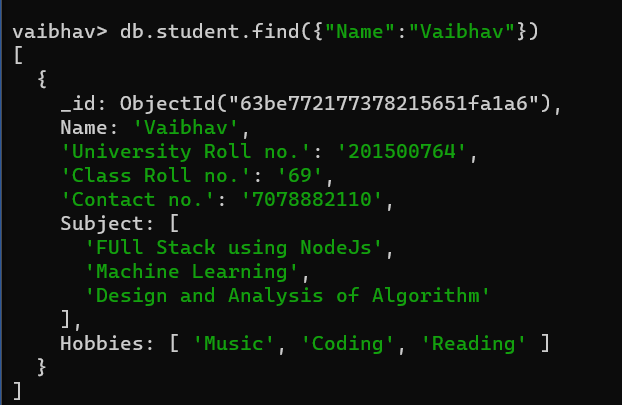
* show collection/db.getCollectionName – to get all the collection inside a databse



* db.<collection name>.find() – to find all the data stored in the collection

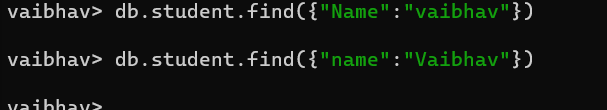


* db.<collection name>.find({“key”:”value”})



This command will check the exact value of key and pair (case sensitive)

When it is not able to find any such data then it will print a blank line



In case there are many keys with the same value than it will show all the keys having the same values

When we use find , it gives all the data related to the key like on finding tha name it gives all the data associated with the name.

So as to tackle this and to check only that key value exist other than all the value we use find with projection.

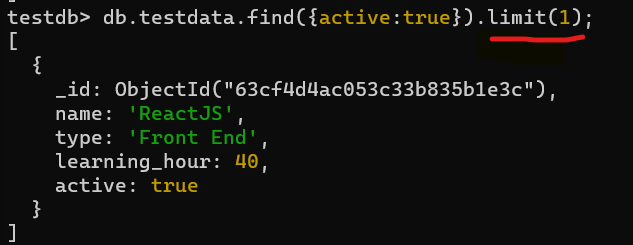
* db.testdata.find({key:value},{key:1}}



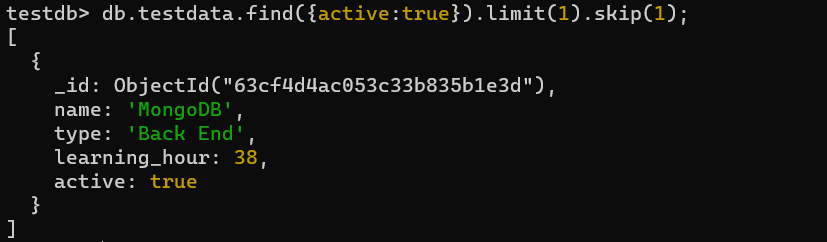
If we don’t want even the id of it we will use



* To find only one document without using findone we will use



* To find one document but not the first one we will use skip



We can change the value of skip to skip the first x found document

For printing only one we will use

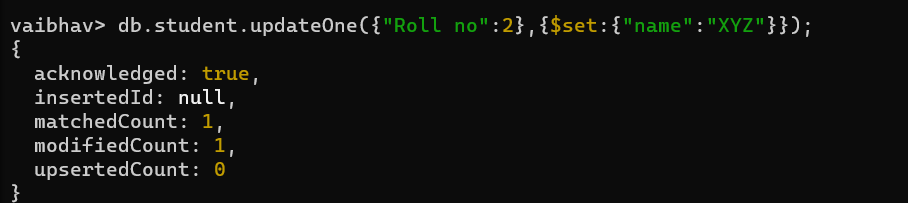
* db.<collectionname>.findOne({“key” : ”value”})

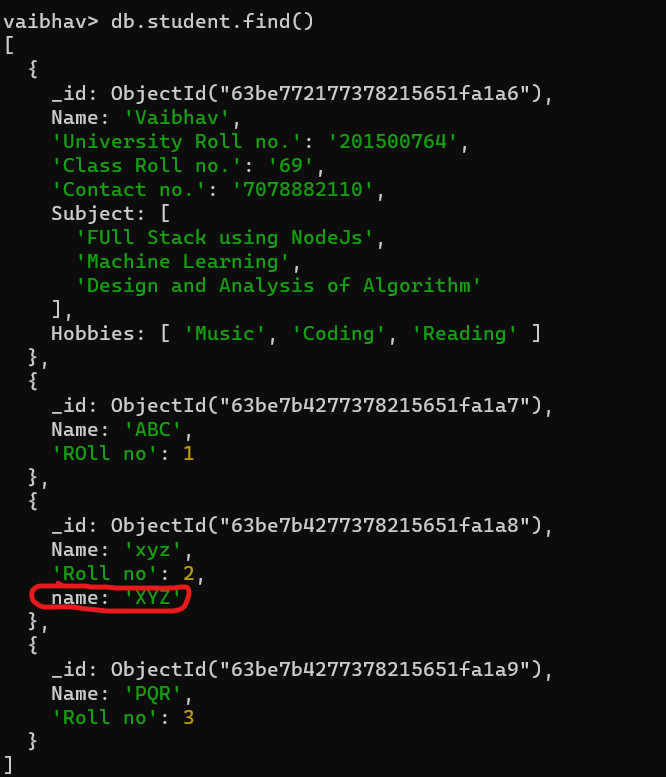
pretty doesn’t work with findone

* db.<collection name>.updateOne({“key1” : “value1” , {$set:{“key2” : “value2”}});

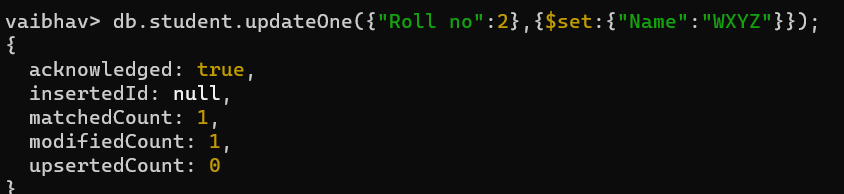
this command will check which data has key and value exact similar to the key1 and value1. If it finds it will update the value of key2 by value2. In case there exist no key2 than it will add new entry in the collection

On inserting unmatched key2 it will create a new entry

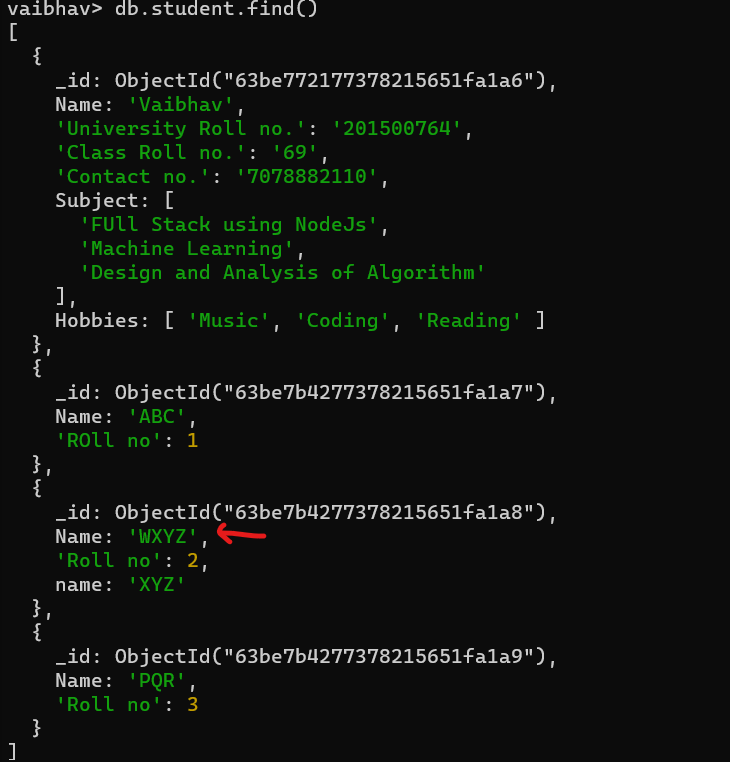




On inserting correct value of key2



* db.countDocument/db.estimatedDoucmentCount



Operators in MongoDB

To retirieve the document with exact id\_value ‘L0009100’

Db.inventory.find({“\_id”:{$eq :”L0009100”}})

We make user to perform different task to various users

There are different types of user

1. read
2. write
3. dbAdmin
4. dbowner
5. userAdmin

Now to create any user we will use this command

db.createUser({name,password,role})

db.createUser({user:”xyz”,pwd:”pass@123”,roles:[role:”read”,db:”supermarket”]})

where supermarket is the name of the database

Now using above approach password is visible therefore we will use this command

db.createUser({user:”xyz”,pwd:passwordPrompt(),roles:[role:”read”,db:”supermarket”]})

using this on hitting enter it will ask for password in astrik

# Query Operators

## Comparison Operators

$eq - equal to

$gt - greater than

$gte - greater than or equal to

$lt - less than

$lte - less than or equal to

$ne - not equal to

## Logical Operators

$and - logical and

$not - logical not

$or - logical or

## Element Operators

$exists - exists

$type - type

## Existence Operators

$in - in

$nin - not in

# Mongo DB Queries

## Basic Queries

show dbs // show all databases

use <db\_name> // use database

db.dropDatabase() // to delete database (must present in database)

## Create, show & delete collections

db.createCollection("<collection\_name>") // create collection

show collections // show all collections

db.getCollectionNames() // show all collections

db.<collection\_name>.drop() // delete collection

## Inserting Documents

db.<collection\_name>.insertOne({<document>}) // insert one document

db.<collection\_name>.insertMany([{<document>}, {<document>}, ...]) // insert many documents

## Deleting Documents

db.<collection\_name>.deleteOne({<document}>) // delete one document

db.<collection\_name>.deleteMany({<document>}) // delete many documents

## Updating Documents

db.<collection\_name>.updateOne({<document>} //to find , {//operator : {<document>}}) // update one document

db.<collection\_name>.updateMany([{<document>}, {<document>}]) // update many documents

## Count

db.<collection\_name>.documentCount() // count all documents

db.<collection\_name>.documentCount({<document>}) // count all documents having matched key value pair

db.<collection\_name>.estimatedDocumentCount()

## To import json database

mongoimport --jsonArray --db supermarket --collection product --file "C:\Users\Khushal\_agarwal\Downloads\pro.json"

## Retrieve Data

db.<collection\_name>.find() // show all documents in collection

db.<collection\_name>.find({key:value}) // all records having matched key value pair

db.<collection\_name>.findOne({key:value}) // first record having matched key value pair

db.testdata.find({name:'Html'},{name:1}) // Show only name field

db.testdata.find({name:'Html'},{\_id:0,name:1}) // \_id also not shown

db.testdata.find({active:true}).limit(1) //find only one document

db.testdata.find({active:true}).limit(1).skip(2) // to skip any number of documents exist

## User commands

db.createUser({user:"khush",pwd:passwordPrompt(),roles:[{role:"read",db:supermarket}]})

mongo installation

mongosh setup

mongo cli tools - export import

basic CRUD operation

18 general operators (regex,logical,array,arithmetic,indexes)

index - text index,ascending,descending,multi key

data modelling

authentication - adding a user with roles

administration - backup restore