**Assignment 1**

**MONGODB CRUD OPERATIONS**

Name**: Aditya Kangune**

Roll number: **33323**

Batch: **K11**

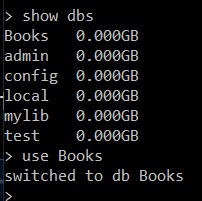
**Aim:** Create a database with suitable example using MongoDB and implement:

* Inserting and saving document (batch insert, insert validation).
* Removing document
* Updating document (document replacement, using modifiers, up inserts, updating multiple documents, returning updated documents)
* Execute at least 10 queries on any suitable MongoDB database that demonstrates following:
* Find and find One (specific values)
* Query criteria (Query conditionals, OR queries, $not, Conditional semantics)
* Type-specific queries (Null, Regular expression, Querying arrays)
* $ where queries
* Cursors(Limit,skip,sort, advanced query options)

**// Opening MongoDB in windows command shell:**

* + 1. Download MongoDB and install it.
    2. Create a new path variable in environment variables to use mongo shell anywhere in the device.
    3. Use commands:
       1. mongod
       2. mongo

**// Create commands:**



> show collections

Book store

> db.createCollection(Books)

**// Shows current database**

{ "ok" : 1 }

> show collections

Book store

Books

> use Books

switched to db Books

> db.Books.find().pretty()

> db.Books.insertMany([{name:"Harry Potter", author:"JK Rowling", id:123456789, price:490, pages:370}, {name:"The mortal instruments", author:"Cassandra Clare", id:4453356543, price:260, pages:290}, {name:"Who will cry when you die", id:7656445678, price:240, pages:120}, {name:"An Autobiography of a Yogi", id:8768905678, author:"Paramhansa Yogananda", price:390, pages:560}, {name:"The Subtle art of not giving a \*uck", author:"Mark Manson", id:2574103698, price:450, pages:230}, {name:"Who Moved My Cheese?", author:"Spencer Johnson", id:9865741520, price:345, pages:180}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("611fa8bffc6e6eecbe413420"),

ObjectId("611fa8bffc6e6eecbe413421"),

ObjectId("611fa8bffc6e6eecbe413422"),

ObjectId("611fa8bffc6e6eecbe413423"),

ObjectId("611fa8bffc6e6eecbe413424"),

ObjectId("611fa8bffc6e6eecbe413425")

]

}

**// Adding user**

> db.createUser({

... user:"Aditya",

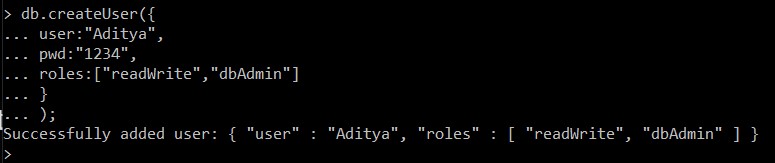
... pwd:"1234",

... roles:["readWrite","dbAdmin"]

... }

... );

Successfully added user: { "user" : "Aditya", "roles" : [ "readWrite", "dbAdmin" ] }



**// Creating new collection(like tables in NoSQL)**

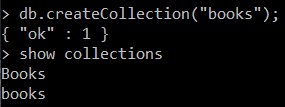
> db.createCollection("books");

{ "ok" : 1 }

> show collections

Books

books



**// Inserting one record and viewing it**

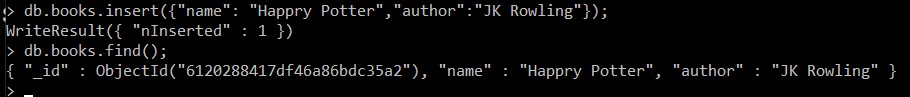
// It automatiacally adds \_id, which is a unique value so we don’t have to worry about auto increament etc like we do in SQL

> db.books.insert({"name": "Happry Potter","author":"JK Rowling"});

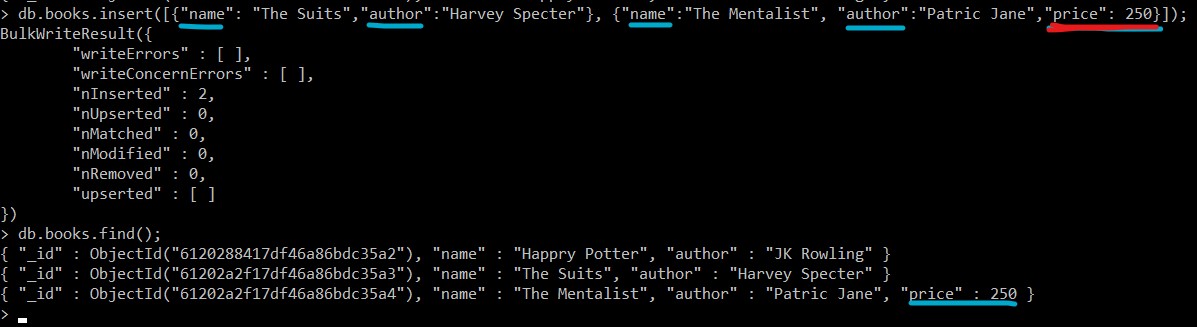
WriteResult({ "nInserted" : 1 })

> db.books.find();

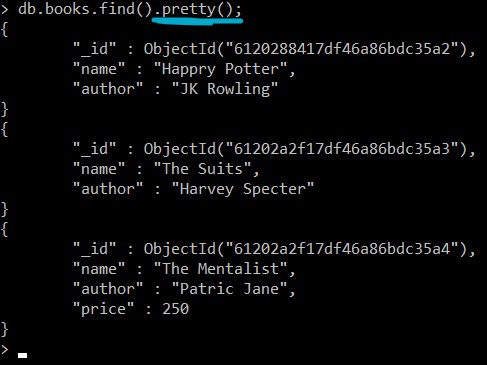
{ "\_id" : ObjectId("6120288417df46a86bdc35a2"), "name" : "Happry Potter", "author" : "JK Rowling" }



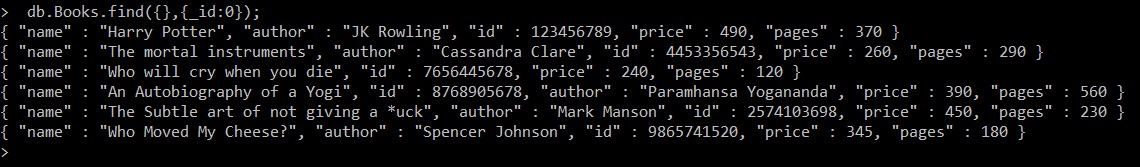
**// Inserting a record with an extra field:**

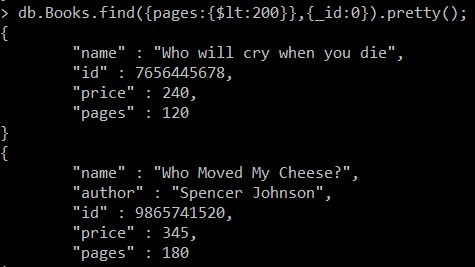


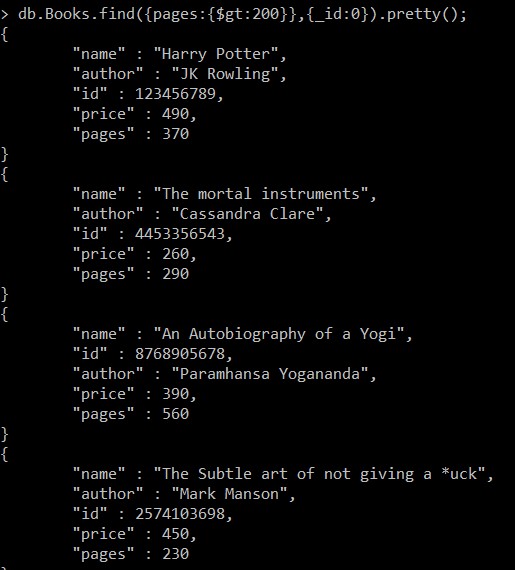
**// Using pretty to get cleaner output**



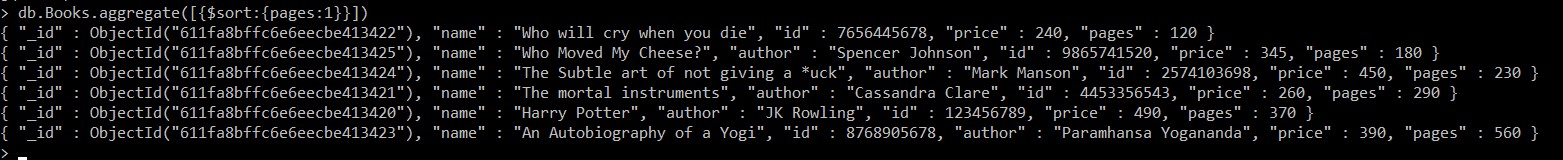
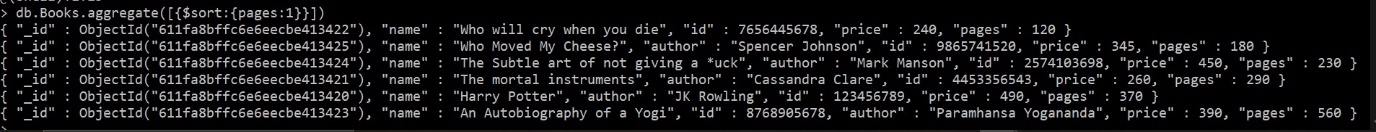
**// Output after create commands**



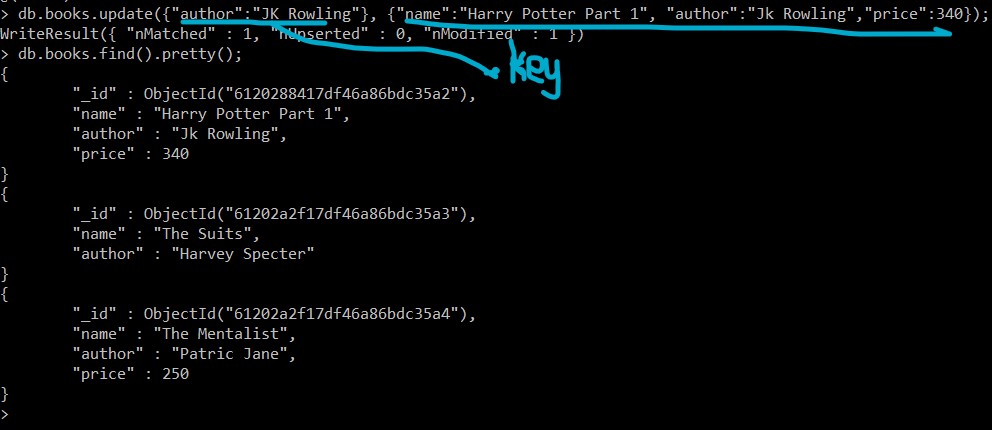




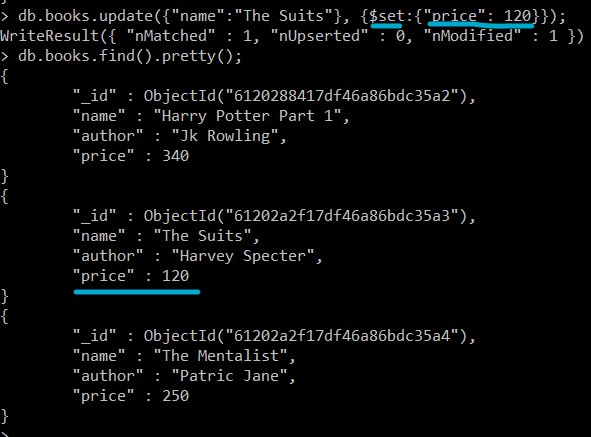
**// Sorting**



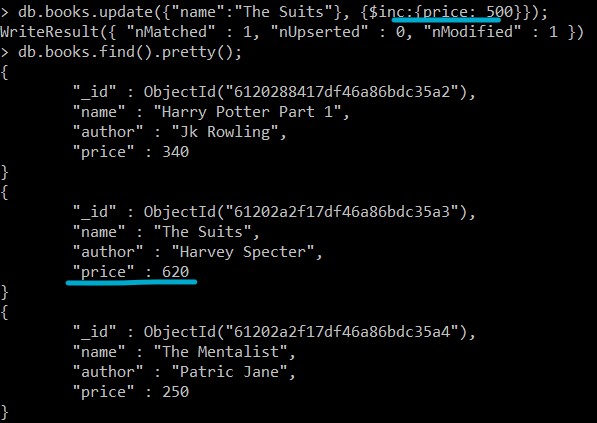
**// Updating by using “author” as key, and changing the name as well as adding a new field of “price” to it.**



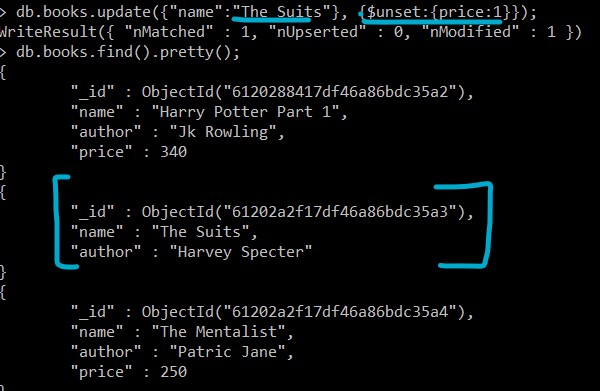
**// Using $set to keep whatever was there previously and adding new field**



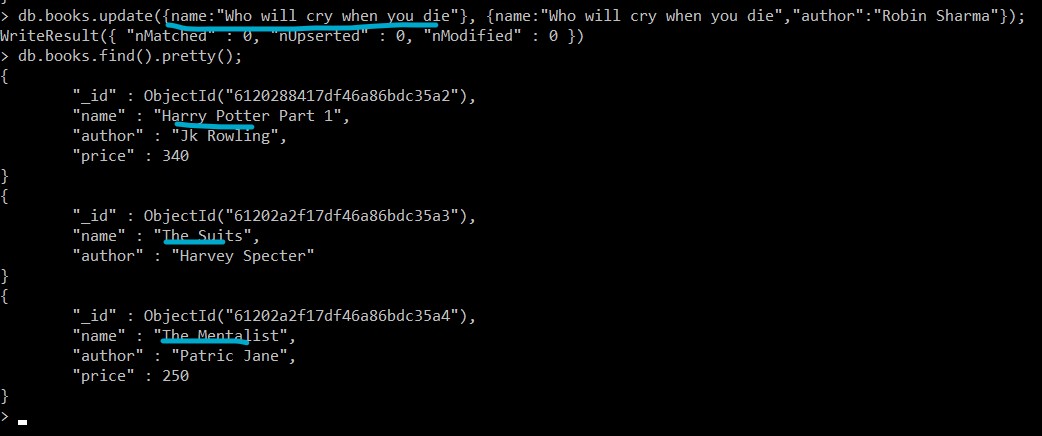
**// Increameting price by 500**



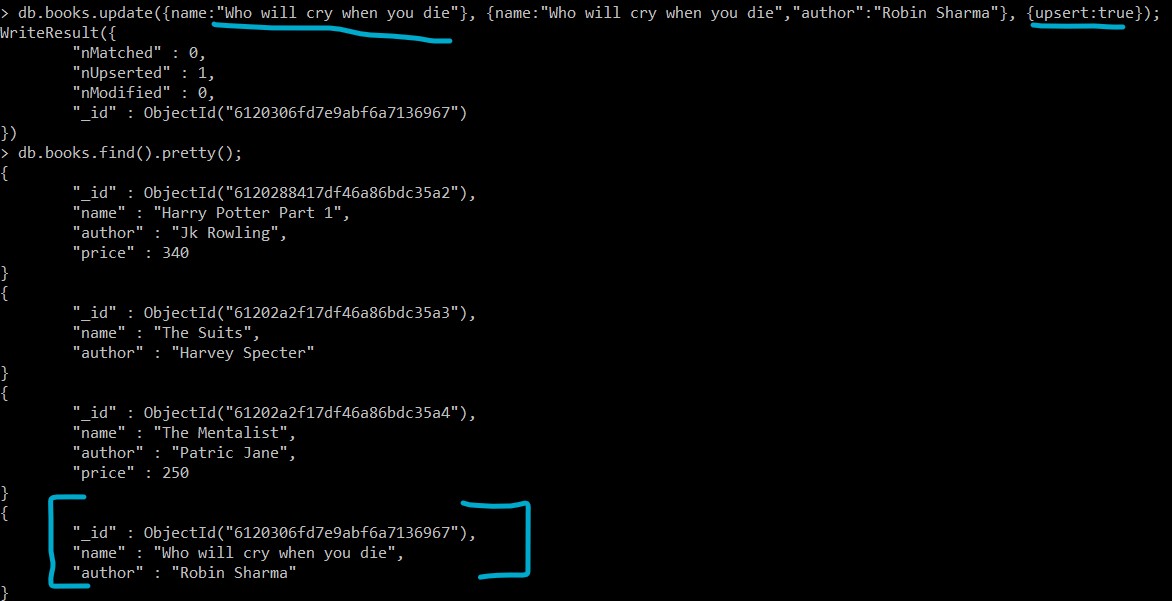
**// Using unset to remove a field**



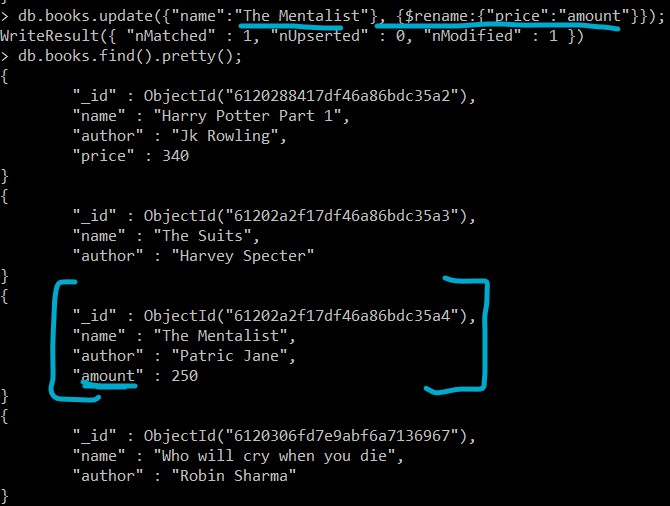
**// Searching for an entry that is not there – We don’t ger any error nor is the record created**



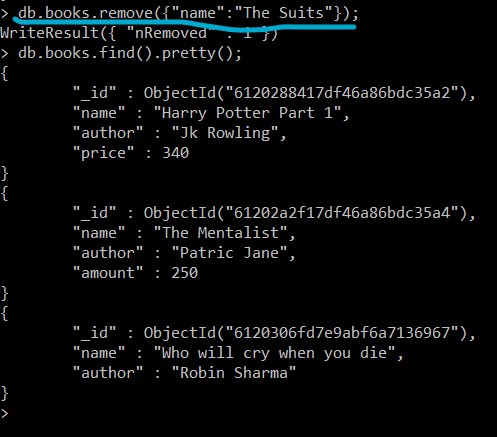
**// Creating a new entry if not found in the collection by setting upsert to true**



**// Changing “price” to “amount” using rename**



**// Removing a particular record**



**//Removing only one record if there are multiple records with same name**



**// Adding nested records using arrays**

**// Adding many records with nested arrays**



> db.books.find().pretty();

{

"\_id" : ObjectId("6120288417df46a86bdc35a2"),

"name" : "Harry Potter Part 1",

"author" : "Jk Rowling",

"price" : 340

}

{

"\_id" : ObjectId("61202a2f17df46a86bdc35a4"),

"name" : "The Mentalist",

"author" : "Patric Jane",

"amount" : 250

}

{

"\_id" : ObjectId("6120306fd7e9abf6a7136967"),

"name" : "Who will cry when you die",

"author" : "Robin Sharma"

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a7"),

"name" : "Harry Potter",

"author" : "JK Rowling",

"id" : 123456789,

"price" : 490,

"pages" : 370,

"address" : [

{

"street" : "Bakers Street",

"city" : "London",

"country" : "UK"

}

],

"members" : [

"mem1",

"mem2"

]

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a8"),

"name" : "The mortal instruments",

"author" : "Cassandra Clare",

"id" : 4453356543,

"price" : 260,

"pages" : 290,

"address" : [

{

"street" : "Times Square",

"city" : "New York",

"country" : "USA"

}

],

"members" : [

"mem1",

"mem2",

"mem3"

]

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a9"),

"name" : "Who will cry when you die",

"id" : 7656445678,

"price" : 240,

"pages" : 120

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35aa"),

"name" : "An Autobiography of a Yogi",

"id" : 8768905678,

"author" : "Paramhansa Yogananda",

"price" : 390,

"pages" : 560

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35ab"),

"name" : "The Subtle art of not giving a \*uck",

"author" : "Mark Manson",

"id" : 2574103698,

"price" : 450,

"pages" : 230

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35ac"),

"name" : "Who Moved My Cheese?",

"author" : "Spencer Johnson",

"id" : 9865741520,

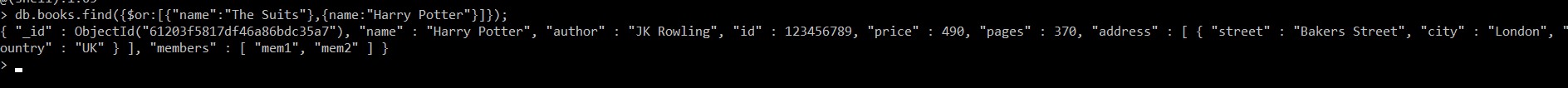
"price" : 345,

"pages" : 180

}

>

**// $or operator**

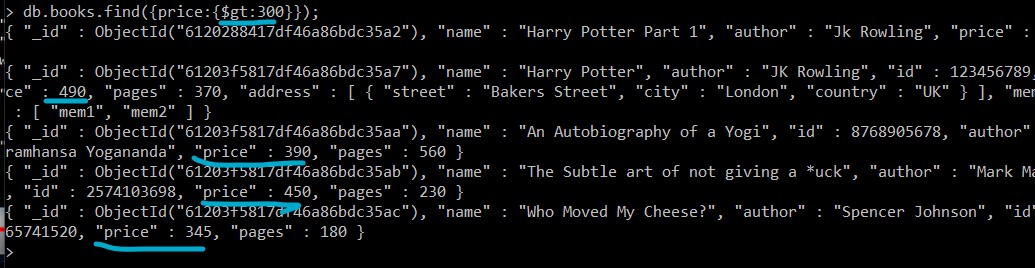


> db.books.find({$or:[{"name":"The Suits"},{name:"Harry Potter"}]});

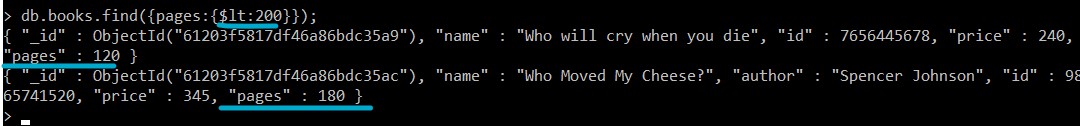
{ "\_id" : ObjectId("61203f5817df46a86bdc35a7"), "name" : "Harry Potter", "author" : "JK Rowling", "id" : 123456789, "price" : 490, "pages" : 370, "address" : [ { "street" : "Bakers Street", "city" : "London", "country" : "UK" } ], "members" : [ "mem1", "mem2" ] }

>

**// Getting book with price greater than 300**



**// Getting book with less than 200 pages**



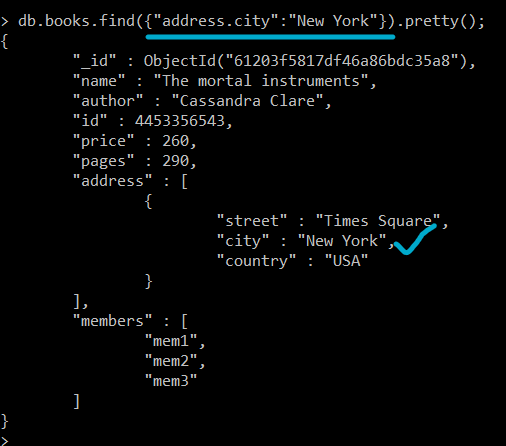
> db.books.find({pages:{$lt:200}});

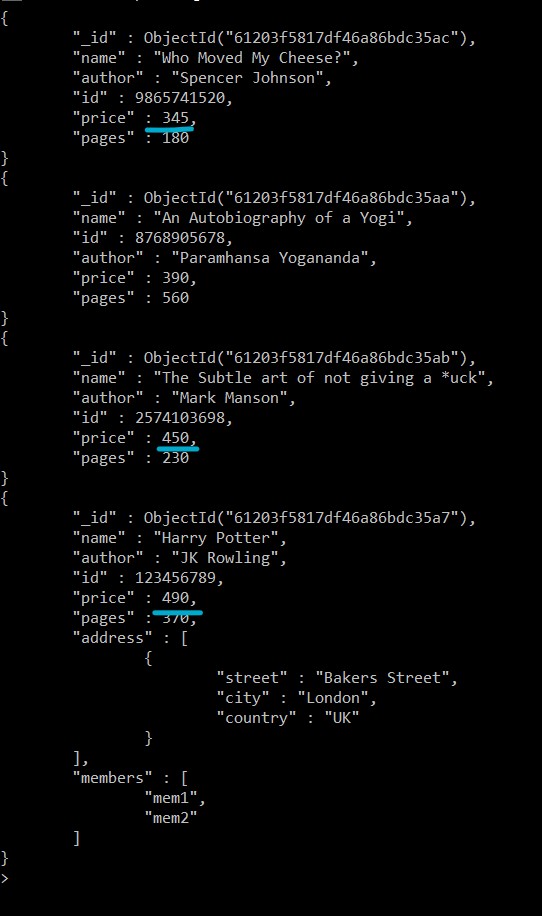
{ "\_id" : ObjectId("61203f5817df46a86bdc35a9"), "name" : "Who will cry when you die", "id" : 7656445678, "price" : 240, "pages" : 120 }

{ "\_id" : ObjectId("61203f5817df46a86bdc35ac"), "name" : "Who Moved My Cheese?", "author" : "Spencer Johnson", "id" : 9865741520, "price" : 345, "pages" : 180 }

>

**// Finding records inside a nested record(row)**



**// Sorting all the books according to price(ascending)**

> db.books.find().sort({price:1}).pretty();

{

"\_id" : ObjectId("61202a2f17df46a86bdc35a4"),

"name" : "The Mentalist",

"author" : "Patric Jane",

"amount" : 250

}

{

"\_id" : ObjectId("6120306fd7e9abf6a7136967"),

"name" : "Who will cry when you die",

"author" : "Robin Sharma"

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a9"),

"name" : "Who will cry when you die",

"id" : 7656445678,

"price" : 240,

"pages" : 120

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a8"),

"name" : "The mortal instruments",

"author" : "Cassandra Clare",

"id" : 4453356543,

"price" : 260,

"pages" : 290,

"address" : [

{

"street" : "Times Square",

"city" : "New York",

"country" : "USA"

}

],

"members" : [

"mem1",

"mem2",

"mem3"

]

}

{

"\_id" : ObjectId("6120288417df46a86bdc35a2"),

"name" : "Harry Potter Part 1",

"author" : "Jk Rowling",

"price" : 340

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35ac"),

"name" : "Who Moved My Cheese?",

"author" : "Spencer Johnson",

"id" : 9865741520,

"price" : 345,

"pages" : 180

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35aa"),

"name" : "An Autobiography of a Yogi",

"id" : 8768905678,

"author" : "Paramhansa Yogananda",

"price" : 390,

"pages" : 560

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35ab"),

"name" : "The Subtle art of not giving a \*uck",

"author" : "Mark Manson",

"id" : 2574103698,

"price" : 450,

"pages" : 230

}

{

"\_id" : ObjectId("61203f5817df46a86bdc35a7"),

"name" : "Harry Potter",

"author" : "JK Rowling",

"id" : 123456789,

"price" : 490,

"pages" : 370,

"address" : [

{

"street" : "Bakers Street",

"city" : "London",

"country" : "UK"

}

],

"members" : [

"mem1",

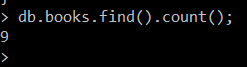
"mem2"

]

}

>

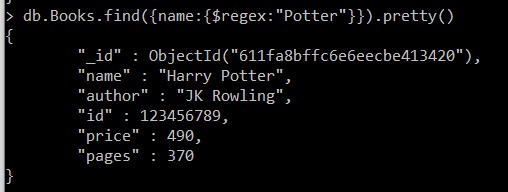
**// Using count() function**



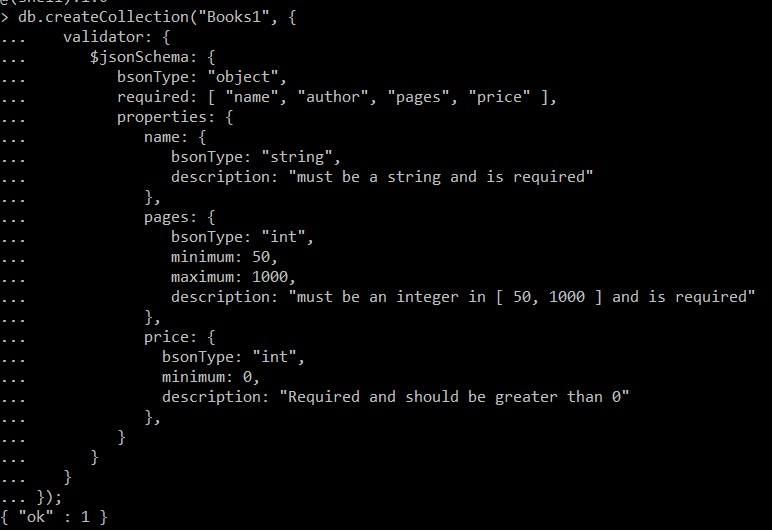
**// Counting occurrence of a specific record**



**// Using regex to find the word “Potter”**

****

**// Using validators**

****

> db.createCollection("Books1", {

... validator: {

... $jsonSchema: {

... bsonType: "object",

... required: [ "name", "author", "pages", "price" ],

... properties: {

... name: {

... bsonType: "string",

... description: "must be a string and is required"

... },

... pages: {

... bsonType: "int",

... minimum: 50,

... maximum: 1000,

... description: "must be an integer in [ 50, 1000 ] and is required"

... },

... price: {

... bsonType: "int",

... minimum: 0,

... description: "Required and should be greater than 0"

... },

... }

... }

... }

... });

{ "ok" : 1 }

**// Using push, pop for arrays:**

**BEFORE push:**

****

**AFTER push:**

****