

MongoDB- NoSQL

Thi Nam Nguyen

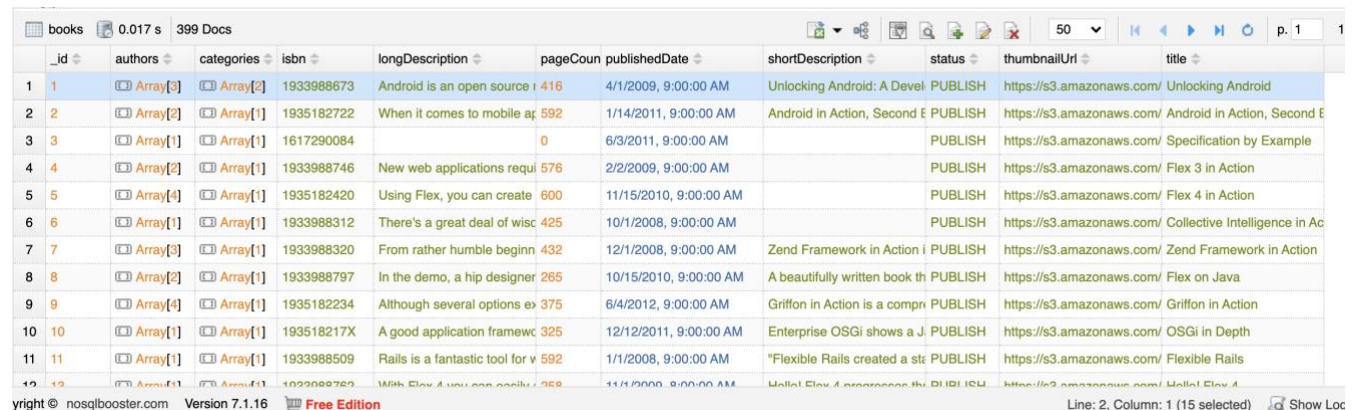
Master in Big Data and Data Science
UCM (2022-2023)

Data source: [GitHub MongoDB NoSQL](#)

Introduction

The objective of this project is to select one dataset and use MongoDB to obtain the desired results. To carry out this analysis, I have chosen a JSON file of all Books published from 1993 to 2014. I focused on obtaining the information regarding the number of books, the books that have longest pages, the authors who have the most publications, etc. In addition, my work will use functions like *filter*, *insert*, *delete*, *update*, *match*, *project*, *group* and *aggregate* to show the results.

The data structure: the data set is classified in 10 different fields

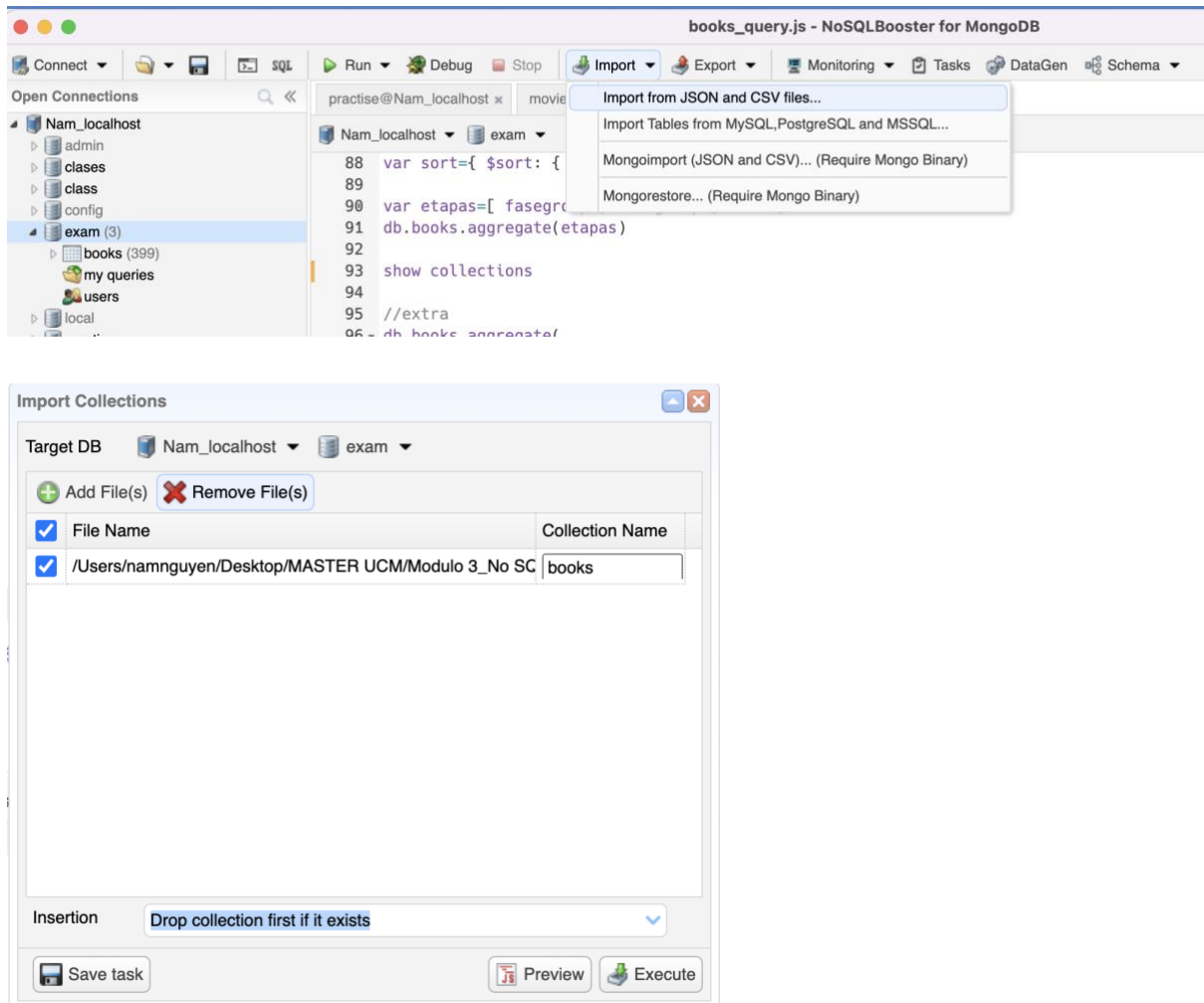


	_id	authors	categories	isbn	longDescription	pageCount	publishedDate	shortDescription	status	thumbnailUrl	title
1	1	Array[3]	Array[2]	1933988673	Android is an open source i	416	4/1/2009, 9:00:00 AM	Unlocking Android: A Devel	PUBLISH	https://s3.amazonaws.com/	Unlocking Android
2	2	Array[2]	Array[1]	1935182722	When it comes to mobile a	592	1/14/2011, 9:00:00 AM	Android in Action, Second E	PUBLISH	https://s3.amazonaws.com/	Android in Action, Second E
3	3	Array[1]	Array[1]	1617290084		0	6/3/2011, 9:00:00 AM		PUBLISH	https://s3.amazonaws.com/	Specification by Example
4	4	Array[2]	Array[1]	1933988746	New web applications requi	576	2/2/2009, 9:00:00 AM		PUBLISH	https://s3.amazonaws.com/	Flex 3 in Action
5	5	Array[4]	Array[1]	1935182420	Using Flex, you can create	600	11/15/2010, 9:00:00 AM		PUBLISH	https://s3.amazonaws.com/	Flex 4 in Action
6	6	Array[1]	Array[1]	1933988312	There's a great deal of wisc	425	10/1/2008, 9:00:00 AM		PUBLISH	https://s3.amazonaws.com/	Collective Intelligence in Ac
7	7	Array[3]	Array[1]	1933988320	From rather humble beginn	432	12/1/2008, 9:00:00 AM	Zend Framework in Action i	PUBLISH	https://s3.amazonaws.com/	Zend Framework in Action
8	8	Array[2]	Array[1]	1933988797	In the demo, a hip designer	265	10/15/2010, 9:00:00 AM	A beautifully written book th	PUBLISH	https://s3.amazonaws.com/	Flex on Java
9	9	Array[4]	Array[1]	1935182234	Although several options ex	375	6/4/2012, 9:00:00 AM	Griffon in Action is a compr	PUBLISH	https://s3.amazonaws.com/	Griffon in Action
10	10	Array[1]	Array[1]	193518217X	A good application framew	325	12/12/2011, 9:00:00 AM	Enterprise OSGi shows a J	PUBLISH	https://s3.amazonaws.com/	OSGi in Depth
11	11	Array[1]	Array[1]	1933988509	Rails is a fantastic tool for v	592	1/1/2008, 9:00:00 AM	"Flexible Rails created a str	PUBLISH	https://s3.amazonaws.com/	Flexible Rails
12	12	Array[1]	Array[1]	1622989722	With Flex 4 you can easily	258	11/1/2009, 9:00:00 AM	Hello! Flex 4 processes the	PUBLISH	https://s3.amazonaws.com/	Hello! Flex 4

- `_id`: as an integer number to identify a book.
- `title`: book's title
- `authors`: book's authors, one book may be written by one or more authors
- `pageCount`: total page numbers.
- `categories`: books are classified by type and subject.
- `isbn`: book's code
- `longDescription`: Describe the content of a book.
- `shortDescription`: Describe the content of a book in a shorter way.
- `status`: shows whether a book is published (**PUBLISH**) or Manning Early Access Program (**MEAP**).
- `thumbnailUrl`: linked to the thumbnail's image.

I. Import dataset

To import the dataset, I first create a new Database name “exam”. Secondly, I chose Import/Import from JSON and CSV files...Next step Insertion chose “Drop collection first if exist”. These processes are shown as the following screenshots:



II. Queries

Once the dataset is input in MongoDB we can see its structure and all information included, which helps us to see the different fields of data where they are located.

The desired information can be obtained by different queries:

1. First, I want to show how many books are in the dataset
`db.books.find().count()`

This results in 399 documents

- Count how many books were written by only one author and how many were written by multiple authors

```
db.books.find({$where: "this.authors.length ==1"}).count()
```

```
db.books.find({$where: "this.authors.length >1"}).count()
```

The previous codes show 211 books were written by only one author and 188 books by multiple authors.

- Now we care about inserting a new book to the dataset

```
var newbook= { "title": "Gone with the wind", "publishedDate": 1936, "authors":  
"Margaret Mitchell", "categories": ["Novel", "fiction"] }
```

```
db.books.insert(newbook)
```

```
db.books.find({"title":"Gone with the wind"})
```

```
1 WriteResult({  
2   "nInserted" : 1  
3 })
```

- And it is possible to delete the book, which has been inserted from the list

```
var newbook={ "title": "Gone with the wind" }
```

```
db.books.deleteMany(newbook)
```

- Count all the books which have no pageCount updated

```
db.books.find({'pageCount': 0}).count()
```

```
1 161
```

```
db.books.find({'pageCount': {$gt:0}}).count()
```

```
238
```

- Show 3 books have the most page numubers

```
var fasegroup={ $group: { "_id":{ "title": "$title", "author":"$authors" }, "page": {  
$max: "$pageCount" } } }
```

```
var fasesort={ $sort: { "page": -1 } }
```

```
var faseLimit={ $limit: 3 }
```

```
var etapas=[ fasegroup, fasesort, faseLimit ]
```

```
db.books.aggregate(etapas)
```

```

/* 1 */
{
  "_id" : {
    "title" : "Essential Guide to Peoplesoft Development and Customization",
    "author" : [ "Tony DeLia", "Galina Landres", "Isidor Rivera", "Prakash Sankaran" ]
  },
  "page" : 1101
},

/* 2 */
{
  "_id" : {
    "title" : "Ten Years of UserFriendly.Org",
    "author" : [ "JD \Illiad\ Frazer" ]
  },
  "page" : 1096
},

/* 3 */
{
  "_id" : {
    "title" : "Java Foundation Classes",
    "author" : [ "Stephen C. Drye", "William C. Wake" ]
  },
  "page" : 1088
}

```

7. The last book was published

```

db.books.find( {}, { 'publishedDate':1, 'title':
'$title', '_id':0 }).sort( { 'publishedDate':-1 })

```

```

1  /* 1 */
2  {
3    "publishedDate" : ISODate( "2014-06-24T09:00:00.000+02:00" ),
4    "title" : "The Well-Grounded Rubyist, Second Edition"
5  },
6

```

8. Show top 3 authors with the most publications

```

var fasegroup = { $group: { "_id": "$authors", "total": { $sum: 1 } } }
var faseordmax = { $sort: { total: -1 } }
var faselimit = { $limit: 3 }
var etapas=[fasegroup, faseordmax, faselimit]
db.books.aggregate(etapas)

```

```

1  /* 1 */
2  {
3    "_id" : [ "Vikram Goyal" ],
4    "total" : 12
5  },
6
7  /* 2 */
8  {
9    "_id" : [
10     [ "Undefined" ]
11   ],
12   "total" : 5
13 },
14
15 /* 3 */
16 {
17   "_id" : [ "Tim Hatton" ],
18   "total" : 3
19 }

```

9. Update all books have empty category as "Undefined"

```
var cat_empty={ 'categories': [0] }  
var actualiza={ $push: { 'catergoies': ["Undefined"] } }  
db.books.updateMany(cat_empty,actualiza)  
db.books.find({ 'catergoies': ["Undefined"] }).count()
```

10. Finally, save a new collection name "**newcategories**" and count how many documents exist in the new collection

```
var faseunwind={ $unwind: "$categories" }  
var faseremoveduplicate={ $project: { "_id": 0 } }  
var faseout={ $out: "newcategories" }  
var etapas=[ faseunwind, faseremoveduplicate, faseout ]  
db.books.aggregate(etapas)  
db.newcategories.find().count()
```

In the new collection (**newcategories**) there are 331 documents.

III. Conclusions

- MongoDB is an attractive option to developers. Its data storage philosophy is simple and immediately understandable to anybody with some programming experience.
- With MongoDB, there are more dynamic options for updating the schema of a collection, such as creating new fields based on an *aggregation pipeline* or updating nested array fields. This benefit is particularly important as databases grow in size.
- In this project I selected a very short and simple **books** collection from 1993 to 2014 which contains 399 books including book title, author, etc. to apply the techniques learned in class. While analyzing the dataset, some missing data have taken account and reorganized in a desired manner.
- The obtained results show that by applying some basic knowledge learned in class it is easy and practical to obtain important and sensible business data and information.