

NoSQL Model Comparator

Kertit Salma - Maatouk Marouane

April 10, 2019

1 Context

The main reason of this paper is to show what we both have done so far. First, this paper is about introducing the tool we've been assigned; MongoDB. Then, we will present the work done on the dataset we've been assigned on Github, using MongoDB.

2 MongoDB

MongoDB is a platform-independent document-oriented database program. Its main objective is to handle these data never needing a predefined schema. Actually, MongoDB stores data in a flexible, JSON-like documents way. The document model maps to the objects in our application code, making data easy to work with.

It is about having many collections with many fields, grouped in databases. Which is the structure we find when visualizing the data.

Plus, the fact that MongoDB provides the possibility to store data in a very dynamic way is an advantage over SQL relational databases where you must define and declare the structure of the data prior to inserting it in the database.

3 Dataset Work -Screenshots

When it comes to MongoDB, after downloading it from its website, we needed to create a collection of our database in order to visualize our JSON file. MongoDB makes it possible for us to filter our database, to view it on charts, to add items, to modify them and to remove them.

The dataset, books.json, that we're going to present in this paper is one found in mongo-db-files on the Github link that Mrs.Asaad has shared before.

Here are the screenshots of the different steps and operations we've done.

```
Répertoire de C:\Users\merouane\Desktop\nosql
09/04/2019 15:18 <DIR> .
09/04/2019 15:18 <DIR> ..
04/04/2019 21:48      536 852 books.json
01/01/1980 00:00 <DIR>   Cassandra
01/01/1980 00:00 <DIR>   CosmosDB
01/01/1980 00:00 <DIR>   couchbase
09/04/2019 15:12 <DIR>   DynamoDB
01/01/1980 00:00 <DIR>   Elasticsearch
07/04/2019 21:23      1 763 Hackolade.lnk
01/01/1980 00:00 <DIR>   Hbase
01/01/1980 00:00 <DIR>   json
01/01/1980 00:00 <DIR>   MarkLogic
01/01/1980 00:00 <DIR>   mongodb
01/01/1980 00:00 <DIR>   Neo4j
05/04/2019 14:56 <DIR>   report_img
                2 fichier(s)      538 615 octets
            13 Rép(s) 13 541 572 608 octets libres

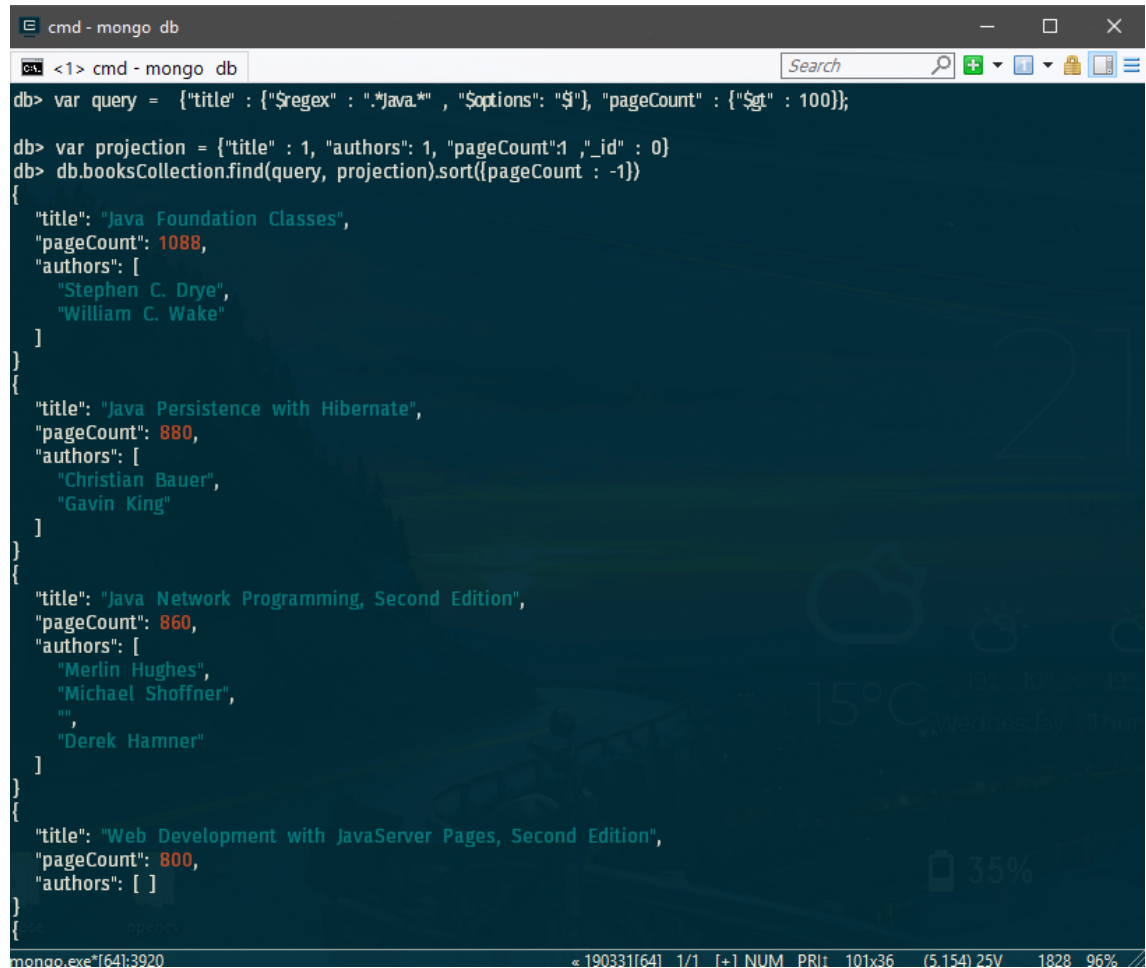
Marouane@DESKTOP-O8D7MVB C:\Users\merouane\Desktop\nosql
$ mongoimport -d db -c booksCollection --file books.json
2019-04-09T21:35:03.503+0100    connected to: localhost
2019-04-09T21:35:04.587+0100    [#####] db.booksCollection
524KB/524KB (100.0%)
2019-04-09T21:35:05.070+0100    [#####] db.booksCollection
524KB/524KB (100.0%)
2019-04-09T21:35:05.071+0100    imported 431 documents

Marouane@DESKTOP-O8D7MVB C:\Users\merouane\Desktop\nosql
$
```

Figure 1: Importing books.json

```
---
db> show collections
booksCollection → 0.494MB / 0.301MB
db> |
```

Figure 2: Show Collections

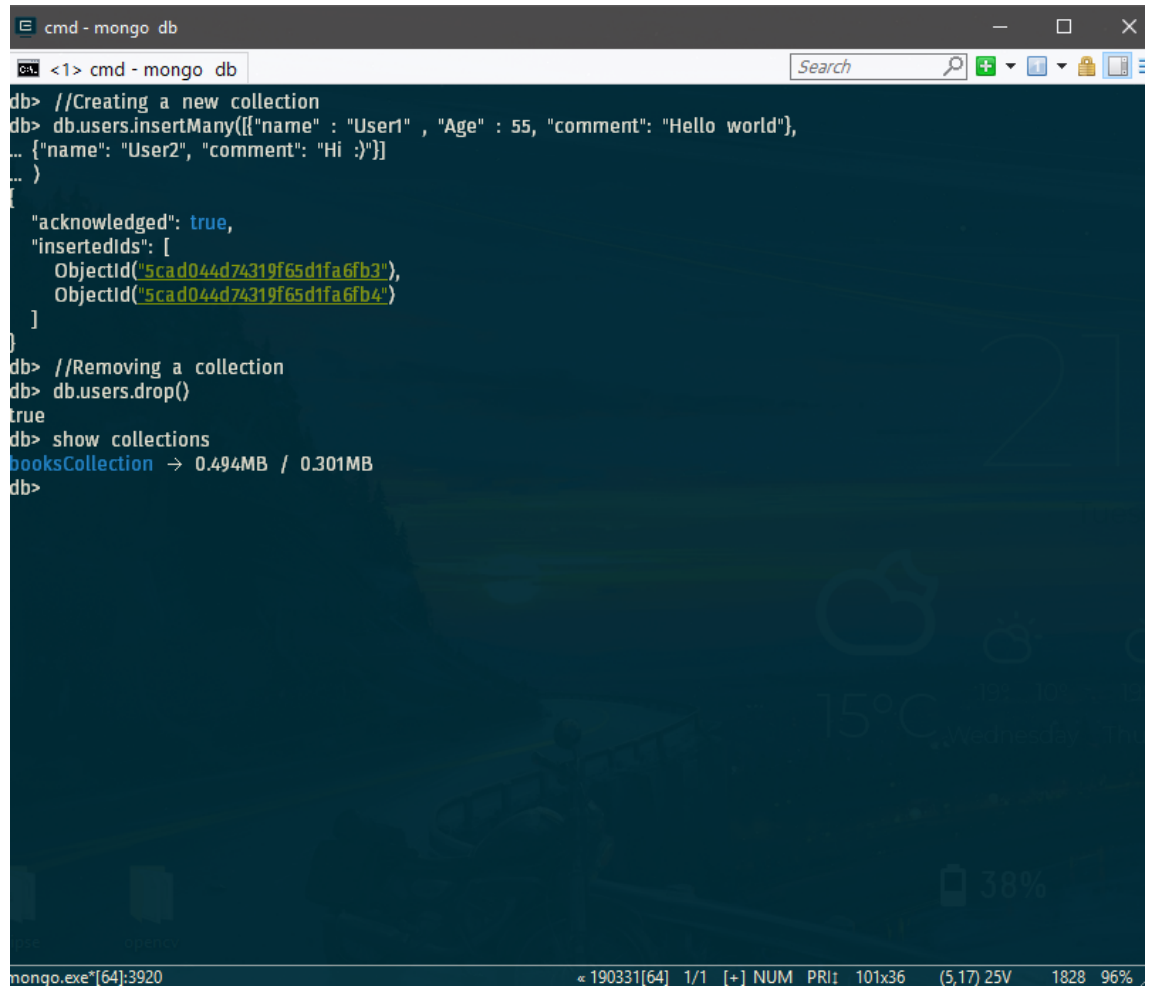


```
cmd - mongo db
<1> cmd - mongo db
db> var query = {"title" : {"$regex" : ".*Java.*" , "$options" : "$i"}, "pageCount" : {"$gt" : 100}};
db> var projection = {"title" : 1, "authors": 1, "pageCount":1 , "_id" : 0}
db> db.booksCollection.find(query, projection).sort({"pageCount" : -1})
{
  "title": "Java Foundation Classes",
  "pageCount": 1088,
  "authors": [
    "Stephen C. Drye",
    "William C. Wake"
  ]
}
{
  "title": "Java Persistence with Hibernate",
  "pageCount": 880,
  "authors": [
    "Christian Bauer",
    "Gavin King"
  ]
}
{
  "title": "Java Network Programming, Second Edition",
  "pageCount": 860,
  "authors": [
    "Merlin Hughes",
    "Michael Shoffner",
    "",
    "Derek Hamner"
  ]
}
{
  "title": "Web Development with JavaServer Pages, Second Edition",
  "pageCount": 800,
  "authors": [ ]
}
{
  "title": "..."
}
```

Figure 3: DataSelectQuery

```
cmd - mongo db
<1> cmd - mongo db
db> db.booksCollection.find().limit(1)
{
  "_id": 1,
  "title": "Specification by Example",
  "isbn": "167290084",
  "pageCount": 0,
  "publishedDate": ISODate("2011-06-03T07:00:00Z"),
  "thumbnailURL": "https://ss.amazonaws.com/AKIA/CSRLADLUMVRPFDQbook-thumb-images/adzic.jpg",
  "status": "PUBLISH",
  "authors": [
    "Gojko Adzic"
  ],
  "categories": [
    "Software Engineering"
  ]
}
db>
db> //Add a new author
db> db.booksCollection.updateOne({"_id": 1}, {$push : {"authors": "Rowling"}})
{
  "acknowledged": true,
  "matchedCount": 1,
  "modifiedCount": 1
}
db> db.booksCollection.find().limit(1)
{
  "_id": 1,
  "title": "Specification by Example",
  "isbn": "167290084",
  "pageCount": 0,
  "publishedDate": ISODate("2011-06-03T07:00:00Z"),
  "thumbnailURL": "https://ss.amazonaws.com/AKIA/CSRLADLUMVRPFDQbook-thumb-images/adzic.jpg",
  "status": "PUBLISH",
  "authors": [
    "Gojko Adzic",
    "Rowling"
  ],
  "categories": [
    "Software Engineering"
  ]
}
db>
db>
```

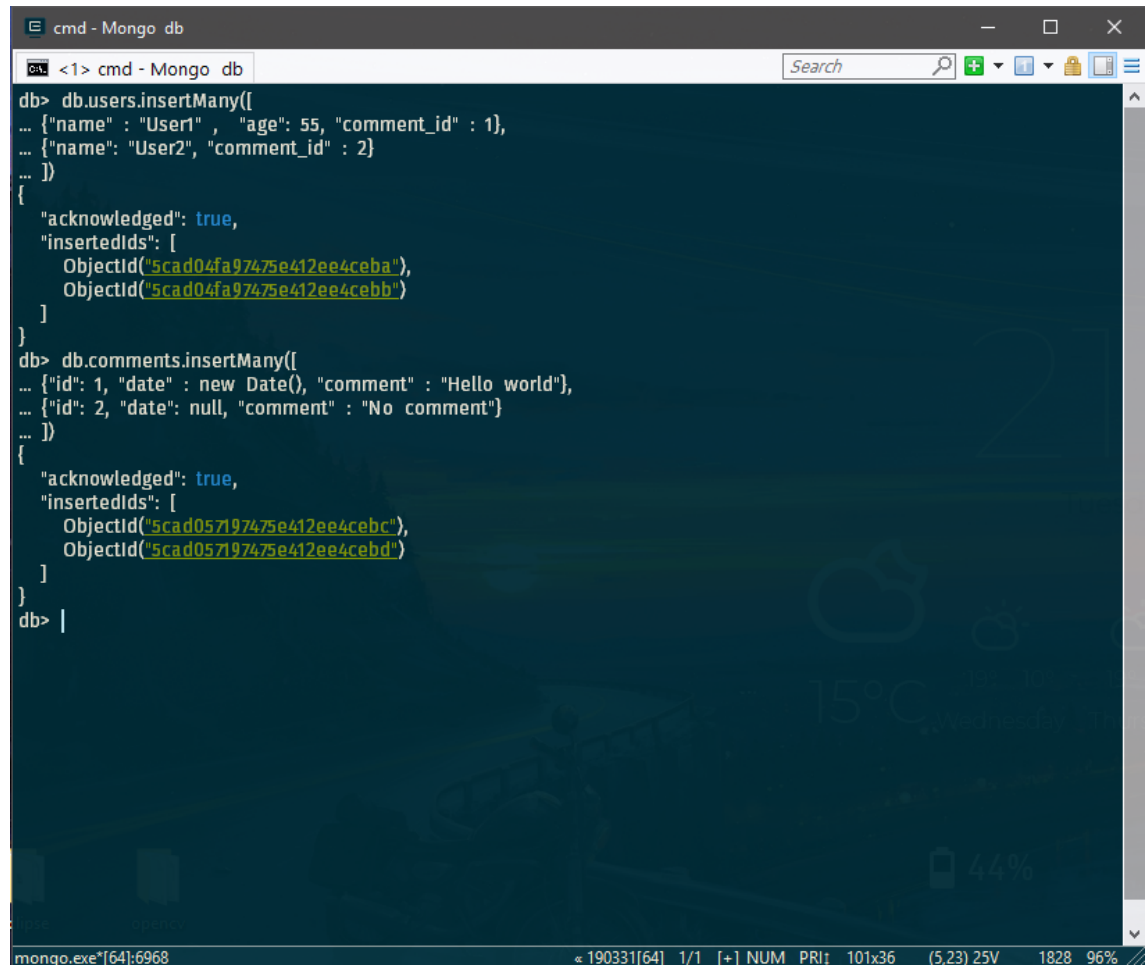
Figure 4: dataSelectNupdate



```
cmd - mongo db
<1> cmd - mongo db
Search
db> //Creating a new collection
db> db.users.insertMany([{"name" : "User1" , "Age" : 55, "comment": "Hello world"},
... {"name": "User2", "comment": "Hi :)"}]
... }
{
  "acknowledged": true,
  "insertedids": [
    ObjectId("5cad044d74319f65d1fa6fb3"),
    ObjectId("5cad044d74319f65d1fa6fb4")
  ]
}
db> //Removing a collection
db> db.users.drop()
true
db> show collections
booksCollection → 0.494MB / 0.301MB
db>
```

The screenshot shows a terminal window titled 'cmd - mongo db'. The user enters several MongoDB commands. First, they use `db.users.insertMany()` to insert two documents into the 'users' collection. The response shows the operation was successful with two object IDs. Then, they use `db.users.drop()` to remove the collection, which returns `true`. Finally, they use `show collections` to list the collections, showing 'booksCollection' with a size of 0.494MB. The terminal window has a dark blue background with a faint image of a person on a boat. The status bar at the bottom shows 'mongo.exe [64]:3920' and system information.

Figure 5: newCollections



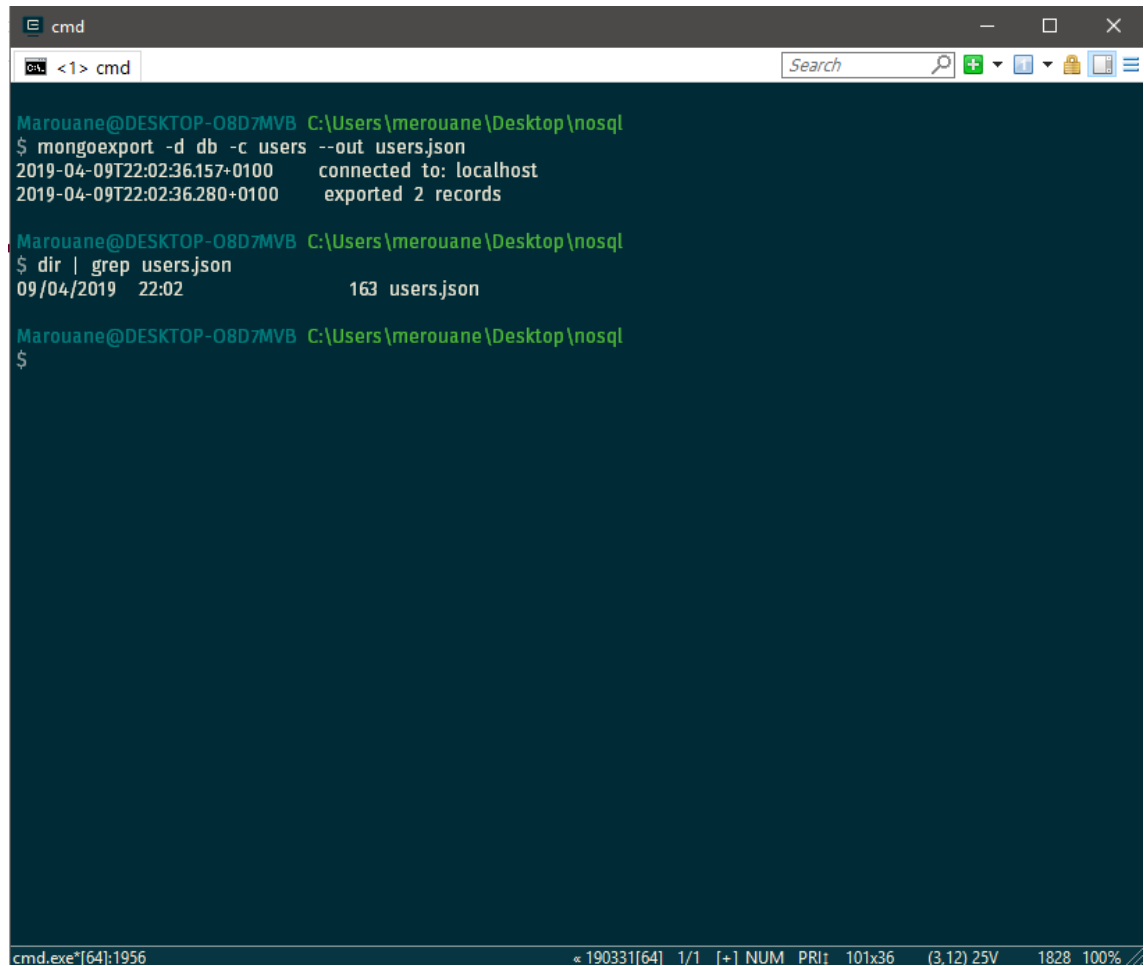
```
cmd - Mongo db
db> db.users.insertMany([
... {"name" : "User1" , "age": 55, "comment_id" : 1},
... {"name": "User2", "comment_id" : 2}
... ])
{
  "acknowledged": true,
  "insertedIds": [
    ObjectId("5cad04fa97475e412ee4ceba"),
    ObjectId("5cad04fa97475e412ee4cebb")
  ]
}
db> db.comments.insertMany([
... {"id": 1, "date" : new Date(), "comment" : "Hello world"},
... {"id": 2, "date": null, "comment" : "No comment"}
... ])
{
  "acknowledged": true,
  "insertedIds": [
    ObjectId("5cad057197475e412ee4cebc"),
    ObjectId("5cad057197475e412ee4cebd")
  ]
}
db> |
```

The screenshot shows a terminal window titled "cmd - Mongo db". The user has executed two MongoDB commands. The first command, `db.users.insertMany()`, inserts two documents into the `users` collection. The second command, `db.comments.insertMany()`, inserts two documents into the `comments` collection. Both commands return a JSON object indicating successful insertion with `"acknowledged": true` and a list of `insertedIds`. The terminal window has a dark background with a faint, stylized image of a person's face. The status bar at the bottom shows "mongo.exe* [64]:6968" and various system metrics.

Figure 6: Insert

```
cmd - mongo db
db> db.users.aggregate({
... $lookup: {
... from: "comments",
... localField: "comment_id",
... foreignField: "id",
... as: "users_comments"
... }
... })
{
  "cursor": {
    "firstBatch": [
      {
        "_id": ObjectId("5cad04fa97475e412ee4ceba"),
        "name": "User1",
        "age": 55,
        "comment_id": 1,
        "users_comments": [
          {
            "_id": ObjectId("5cad057197475e412ee4cebc"),
            "id": 1,
            "date": ISODate("2019-04-09T20:49:53.805Z"),
            "comment": "Hello world"
          }
        ]
      },
      {
        "_id": ObjectId("5cad04fa97475e412ee4cebb"),
        "name": "User2",
        "comment_id": 2,
        "users_comments": [
          {
            "_id": ObjectId("5cad057197475e412ee4cebd"),
            "id": 2,
            "date": null,
            "comment": "No comment"
          }
        ]
      }
    ]
  }
}
```

Figure 7: Join



```
cmd
<1> cmd

Marouane@DESKTOP-O8D7MVB C:\Users\merouane\Desktop\nosql
$ mongoexport -d db -c users --out users.json
2019-04-09T22:02:36.157+0100    connected to: localhost
2019-04-09T22:02:36.280+0100    exported 2 records

Marouane@DESKTOP-O8D7MVB C:\Users\merouane\Desktop\nosql
$ dir | grep users.json
09/04/2019  22:02                163 users.json

Marouane@DESKTOP-O8D7MVB C:\Users\merouane\Desktop\nosql
$
```

cmd.exe*[64]:1956 190331[64] 1/1 [+1] NUM PRI: 101x36 (3,12) 25V 1828 100%

Figure 8: Export