

Tuần 1-2-3 : MongoDB Query

Mục tiêu:

- Cài đặt **MongoDB Server** trên máy tính cá nhân.
- Cài đặt **mongodb-database-tool** để thực hiện import và export database.
- Cài đặt **mongosh** để hỗ trợ định dạng ở chế độ command line.
- Cài đặt **mongodb-compass** để thao tác với database bằng UI (User Interface).
- Tạo được tài khoản **Atlas** trên cloud MongoDB.
- Thực hành bằng mongo hoặc mongosh thực thi các lệnh xem databases, collections.
- Thực hiện các câu truy vấn vào database trên MongoDB và thực thi các lệnh trên Collections như: thêm, xóa, cập nhật documents trong collection (CRUD).

Yêu cầu:

Tạo file MSWord theo định dạng **STT_MSSV_Hoten_Tuan1**, chụp lại các câu lệnh và kết quả chạy (nếu có) vào file MSWord, nộp lại vào cuối buổi thực hành.

Bài 1. Cài đặt

Bài 1.1. Cài đặt MongoDB Server trên máy tính cá nhân (xem chi tiết file hướng dẫn).

Bài 1.2. Cài đặt mongodb-database-tool để thực hiện import và export database (xem chi tiết file hướng dẫn).

Bài 1.3. Cài đặt mongosh để hỗ trợ định dạng ở chế độ command line (xem chi tiết file hướng dẫn).

Bài 1.4. Cài đặt mongodb-compass để làm việc với database bằng UI (User Interface) (xem chi tiết file hướng dẫn).

Bài 1.5. Làm việc với MongoDB Atlas (xem chi tiết file hướng dẫn).

1.5.1. Tạo tài khoản sử dụng MongoDB Atlas.

1.5.2. Load sample data vào Atlas.

Bài 2. Tạo database {import/export}

Lý thuyết:

Dữ liệu được lưu trữ trong MongoDB theo 2 định dạng: BSON hoặc JSON.

1. BSON:

1.1. Backup:

```
mongodump -d <dbName> -o [des-folder/]>
```

```
mongodump -d <dbName> -c <collectionName> -o [des-folder/]
```

Ví dụ:

1. `mongodump -d Test`

→ Backup database Test ra thư mục hiện hành

2. `mongodump -d Test -c dsSinhvien -o sinhvien`

→ Backup collection dsSinhvien trong database Test ra file sinhvien.bson

1.2. [Restore:](#)

```
mongorestore --drop -d <dbName> -c <coName> [des-  
folder/]/[file.bson]
```

Ví dụ:

1. `mongorestore --drop dump/`

→ Xóa và restore tất cả các collections có trong thư mục dump

2. `mongorestore --drop --nsInclude=test.* dump/`

→ Xóa và restore toàn bộ collection trong database test

3. `mongorestore --drop --nsInclude=test.purchaseorders dump/`

hoặc:

```
mongorestore --drop --db=test --collection=purchaseorders  
dump/test/purchaseorders.bson
```

hoặc:

```
mongorestore --drop -d test -c purchaseorders  
dump/test/purchaseorders.bson
```

→ Xóa và restore collections purchaseorders của database test có trong thư mục dump

2. JSON:

2.1. [Export:](#)

```
mongoexport -d <databaseName> -c <collectionName> -o [des-  
folder/]/[file.json]
```

2.2. [Import:](#)

```
mongoimport --drop -d <databaseName> -c <collectionName> [des-  
folder/]/[file.json]
```

Một số lệnh cơ bản trên mongo/mongosh:

`use {database_name}`: chuyển sang database_name (nếu không có tạo mới).

`show dbs`: hiển thị toàn bộ database đã có.

`show collections`: hiển thị toàn bộ collections của database hiện hành.

`db.{collection_name}.find()`: hiển thị toàn bộ document có trong collection_name

`db.{collection_name}.find(projection)`: hiển thị document có trong

collection_name theo các field trong projection.

`db.{collection_name}.find().limit(n)` : hiển thị n document.

`db.{collection_name}.find().skip(n)` : hiển thị các document bỏ qua document thứ n.

Cho database gồm 2 collection là *sinhvien* và *lophoc* tương ứng với 2 file JSON là *sinhvien.json* và *lophoc.json* chứa các document với danh sách sinh viên và lớp học như sau:

<i>sinhvien.json</i>	<i>lophoc.json</i>
<pre>{ _id: ObjectId("620a8bbb2c96dd44ef22230a"), ten: 'Nở', tuoi: 22, diem: 9, monHoc: ['Toan', 'Ngu Van', 'Tin Hoc'], totNghiep: null, lienLac: { email: 'nở@gmail.com', phone: '0999.987.222' } }</pre>	<pre>{ _id: ObjectId("620a8bff7c3b0f4af1d79d98"), Name: 'DHKTPM14', Subject: 'Programming Application', Hours: 5 }</pre>

Bài 2.1. Thực hiện import 2 file JSON này lên MongoDB server.

Bài 2.2. Thực hiện các câu truy vấn các document trong collection như sau:

- 1) `db.sinhvien.find()`
- 2) `db.lophoc.find()`
- 3) `db.sinhvien.find().count()`
- 4) `db.lophoc.find().count()`
- 5) `db.sinhvien.find().limit(2)`
- 6) `db.lophoc.find().skip(2)`
- 7) `db.sinhvien.find().skip(2).limit(2)`
- 8) `db.sinhvien.find({"ten":"Tí"})`
- 9) `db.sinhvien.find({"tuoi":{"$gt":20}})`
- 10) `db.sinhvien.find({"lienLac.email":"ti@gmail.com"})`
- 11) `db.sinhvien.find({"monHoc":"Tin Hoc"})`
- 12) `db.lophoc.find({"Name":"DHKTPM16"})`

Xem kết quả và chụp lại kết quả, nêu ý nghĩa câu query.

Bài 3. Thực hiện lấy các collection trên Atlas vào server local

Bài 3.1. Dùng Mongo Shell/ MongoDB Compass kết nối với Atlas và thao tác với data.

Bài 3.2. Lấy một collection bất kỳ trên Atlas và đưa vào server local.

Bài 4. Truy vấn collection restaurants

Sử dụng collection **restaurants.json** để thực hiện các truy vấn theo yêu cầu.

restaurants.json

```
{
  address: {
    building: '7715',
    coord: [ -73.9973325, 40.61174889999999 ],
    street: '18 Avenue',
    zipcode: '11214' },
  borough: 'Brooklyn',
  cuisine: 'American ',
  grades:
  [
    {date: ISODate("2014-04-16T00:00:00.000Z"), grade: 'A', score: 5},
    {date: ISODate("2013-04-23T00:00:00.000Z"), grade: 'A', score: 2},
    {date: ISODate("2012-04-24T00:00:00.000Z"), grade: 'A', score: 5},
    {date: ISODate("2011-12-16T00:00:00.000Z"), grade: 'A', score: 2}
  ],
  name: 'C & C Catering Service',
  restaurant_id: '40357437'
}
```

1. Hiển thị tất cả các documents có trong collection restaurants.
2. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant_id, name, borough and cuisine.
3. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant_id, name, borough and cuisine và không xuất field _id.
4. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant_id, name, borough and zip code và không xuất field _id.
5. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị là Bronx.
6. Hiển thị 5 documents đầu tiên có trong collection restaurants với field borough có giá trị là Bronx.
7. Hiển thị 5 documents tiếp theo sau khi bỏ qua 5 documents đầu tiên có trong collection restaurants với field borough có giá trị là Bronx.
8. Hiển thị tất cả các documents có trong collection restaurants với điều kiện score trong field grades lớn hơn 90.
9. Hiển thị tất cả các documents có trong collection restaurants với điều kiện score trong field grades lớn hơn 80 và nhỏ hơn 100.
10. Hiển thị tất cả các documents có trong collection restaurants với điều kiện giá trị coord trong field address nhỏ hơn -95.754168.

11. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và score của field grade lớn hơn 70 và giá trị coord trong field address nhỏ hơn -65.754168. *****
12. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và score của field grade lớn hơn 70 và giá trị coord trong field address nhỏ hơn -65.754168.
13. Hiển thị tất cả các documents có trong collection restaurants với điều kiện field cuisine không là 'American' và giá trị grade của field grade là 'A', và fiels borough không là Brooklyn. Sau đó sắp xếp các document theo thứ tự tăng dần của field cuisine.
14. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự bắt đầu là 'Wil'.
15. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự cuối cùng là 'ces'.
16. Hiển thị tất cả các documents có trong collection restaurants, tuy nhiên chỉ xuất các fields restaurant Id, name, borough, cuisine với name có chứa 3 ký tự 'Reg'.
17. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị là Bronx và field cusine có giá trị là American hoặc Chinese.
18. Hiển thị tất cả các documents có trong collection restaurants với field borough có giá trị Staten Island or Queens or Bronxor Brooklyn, chỉ xuất các field restaurant Id, name, borough, cuisine.
19. Hiển thị tất cả các documents có trong collection restaurants với field borough có không là các giá trị Staten Island or Queens or Bronxor Brooklyn, chỉ xuất các field restaurant Id, name, borough, cuisine.
20. Hiển thị tất cả các documents có trong collection restaurants với giá trị score của field grades không lớn hơn 10, chỉ xuất các field restaurant Id, name, borough, cuisine.

Bài 5. Truy vấn collection xe

Tạo database dbQLXe với collection Xe. Thông tin mỗi xe gồm có: mã xe, tên, năm sản xuất, giá, hình ảnh, loại xe. Trong đó loại xe chứa thông tin về mã loại và tên loại.

Xe.json
<pre>{ ma: 'Ya001', ten: 'Yamaha', namsx: 1992, gia: 12000000, hinhanh: 'h2.jpg', loai: { maloi: '001Xemay', tenloai: 'Xe máy' } }</pre>

1. Xuất toàn bộ danh sách xe máy.

2. Xuất ra danh sách xe là ô tô.
3. Xuất danh sách xe máy Yamaha.
4. Xuất danh sách xe máy Honda.
5. Xuất danh sách xe máy Yamaha sản xuất năm 1992.
6. Xuất danh sách xe máy Honda có giá từ 15.000.000 trở lên.
7. Xuất ra các danh sách các xe có Namsx là 1999.
8. Xuất ra danh sách xe ô tô có sản xuất từ năm 2000.
9. Đếm có bao nhiêu xe có năm sản xuất 1992.
10. Xuất ra các xe có giá trên 25.000.000 dưới 655.000.000.
11. Cập nhật mã loại của ô tô thành '001CAR'.
12. Cập nhật giá của tất cả giá xe máy tăng thêm 1.000.000.
13. Cập nhật giá của tất cả giá ô tô lên 0.5 lần.
14. Cập nhật các document có "ten":"Honda" với các Namsx: 2000
15. Thêm 1 thuộc tính màu xe có các giá trị đỏ, đen, trắng cho tất cả xe ô tô.
16. Xuất ra tên và giá của tất cả xe 'Yamaha'.

Bài 6. Truy vấn collection sinhvien, lop

Cho thông tin sinhvien và lop như sau:

sinhvien (file dsSinhvien.bson)

```
{
  _id: ObjectId("6131a2a278b3264add832588"),
  diemTB: 9.5,
  dsDienthoai: [ '0947536844' ],
  email: 'Chuot@gmail.com',
  gioitinh: 'Nam',
  ho: 'Nguyễn Việt',
  malop: 'DHKTPM15B',
  mssv: '19296011',
  ngaysinh: ISODate("2001-09-15T00:00:00.000Z"),
  ten: 'Anh'
}
```

lop (file dsLophoc.bson)

```
{
  _id: ObjectId("6131e5e97cf137f22a235cd0"),
  macn: 'KHMT',
  mslop: 'DHKHMT15A',
  sisoDukien: 80,
  tenlop: 'Đại học Khoa học Máy tính 15A'
}
```

Thực hiện các câu truy vấn sau:

1. Đếm số sinh viên có điểm trung bình từ 5 trở lên.
2. Đếm số sinh viên có điểm trung bình từ 3.0 đến nhỏ hơn 6.5.
3. Liệt kê danh sách sinh viên không có số điện thoại hoặc email.
4. Liệt kê danh sách sinh viên có từ 2 số điện thoại trở lên.

- Liệt kê danh sách sinh viên có lót chữ “Văn”, không phân biệt chữ hoa chữ thường.
- Liệt kê danh sách sinh viên có họ tên chứa chữ “Minh”, không phân biệt chữ hoa chữ thường.

Bài 7. Insert/Update/Delete collection xe

Cú pháp:

```
db.collection.insertMany([ { document } ])
db.collection.update({ <filter> }, { <update> }, { <optional> })
db.collection.updateOne({ <filter> }, { <update> }, { <optional> })
db.collection.updateMany({ <filter> }, { <update> }, { <optional> })
```

Sử dụng collection xe trong database dbQLXe của bài 5 thực hiện các câu sau:

- Thêm vào 3 document với các thuộc tính là xe của sv (tùy ý).
- Tăng giá của tất cả xe Yamaha lên 1200000.
- Cập nhật xe máy Honda có giá thấp nhất 24000000
- Cập nhật giá xe ô tô Ford có giá cao nhất 700000000.
- Cập nhật thêm thuộc tính số lượng là 12 cho xe yamaha
- Cập nhật số lượng của xe Ya004 số lượng tăng gấp 3 lần.
- Cập nhật giá tăng 20% cho xe máy Honda sản xuất từ năm 1990 cho đến nay.
- Tất cả xe cập nhật thêm thuộc tính mauxe(black, white, gold).
- Tất cả xe Ford có năm sản xuất từ trước năm 2000 được cập nhật ngày bán là ngày hiện hành.
- Đổi tất cả tên thuộc tính của tất cả các document trong collection Xe sang English (Ví dụ: ten→name,namsx→year, loai →categories, ...)
- Các xe Ford được cập nhật thêm thuộc tính như sau soluong(20,30,50)

Bài 8. Insert/Update/Delete collection restaurants

Cho thông tin restaurants và zipcode như sau:

restaurants
<pre>{ address: { building: '97-22', coord: [-73.8601152, 40.7311739], street: '63 Road', zipcode: '11374' }, borough: 'Queens', cuisine: 'Jewish/Kosher', grades: [{date: ISODate("2014-11-24T00:00:00.000Z"), grade: 'Z', score: 20}, {date: ISODate("2013-01-17T00:00:00.000Z"), grade: 'A', score: 13}, {date: ISODate("2012-08-02T00:00:00.000Z"), grade: 'A', score: 13}, {date: ISODate("2011-12-15T00:00:00.000Z"), grade: 'B', score: 25}], name: 'Tov Kosher Kitchen', restaurant_id: '40356068' }</pre>

zipcode

```
{
  _id: '01012',
  city: 'CHESTERFIELD',
  loc: [ -72.833309, 42.38167 ],
  pop: 177,
  state: 'MA'
}
```

Lý thuyết:

Các operator trong update:

- Fields: \$currentDate, \$inc, \$min, \$max, \$mul, \$rename, \$set, \$unset
- Array: \$, \$[], \$[<identifier>], \$each

1. Thêm 2 documents restaurants với thông tin như sau:

```
db.restaurants.insertMany([
  {
    address: {street: '63 Ky Dong', zipcode: '11374'},
    cuisine: 'Vietnamese',
    name: 'Lobster Bay',
    restaurant_id: '40356666',
    like: 23000,
    share: 4000
  },
  {
    address: {street: '73rd Floor Landmark 81', zipcode: '11374'},
    cuisine: 'Vietnamese',
    name: 'Truffle',
    restaurant_id: '40357777',
    like: 29000,
    share: 3000
  }
])
```

2. Cập nhật thuộc tính assigndate là ngày hiện hành cho các document có zipcode: '11374'.
3. Cập nhật giá trị 'Euro' cho thuộc tính cuisine của các document có thuộc tính borough là 'Queens'.
4. Tăng lượt like lên 100, lượt share lên 200 cho các nhà hàng Vietnamese.
5. Cập nhật điểm đánh giá cao nhất là 28 cho tất cả các document có zipcode là '10002'.
6. Trong document zipcode, cập nhật dân số (pop) tăng 10% của các city HADLEY thuộc state MA.