

MONGODB - tuần 3

Lý thuyết

- Skip, limit và sort

```
db.products.find({}).skip(2).limit(2);
```

```
db.products.find({}).sort({price: -1});
```

// 1:tăng dần; -1: giảm dần

- \$regex

```
db.products.find({name: {$regex: "Charger"}}); // name có chứa "Charger"
```

```
db.products.find({name: {$regex: "^AC3 Case"}}); // name bắt đầu bằng "AC3 Case"
```

```
db.products.find({name: {$regex: "Warranty$"}}); // name kết thúc bằng "Warranty"
```

```
db.products.find({name: {$regex: "phone service", $options: 'i'}});
```

// i: case-insensitive

```
db.products.find({name: /^AC3 Case/}).sort({price: -1, rating: 1});
```

```
db.products.find({name: /AC3|ac7/i}); // name chứa AC3 hoặc ac7, i: case-insensitive
```

- Subdocument

```
db.products.find({'limits.voice.n': {$gt: 20}});
```

```
db.products.find({'limits.sms.units': {$exists: true}});
```

```
db.products.find({'limits.voice.n': {'$not': {$eq: 400}}});
```

```
db.products.find({'limits': {$exists: true}, 'limits.voice.n': {'$not': {$eq: 400}}});
```

- Array

\$elemMatch:

```
{<field>: {$elemMatch: [<query1>, <query2>, ...]}}
```

<https://www.mongodb.com/docs/manual/reference/operator/query/elemMatch/>

// \$elemMatch: is an array query operator that matches documents that contain an array field and the array field has at least one element that satisfies all the specified queries.

```
db.products.find({'type': {$elemMatch: {$eq: "accessory", $ne: "charger"}}});
```

```
db.products.find({'type': {$eq: "accessory", $ne: "charger"}});
```

```

db.products.find({'for.0': 'ac3'}); // tìm các document có phần tử đầu tiên của for là ac3
db.products.find({}, {'for': {'$slice': 2}}); // hiển thị 2 phần tử đầu tiên của mảng
db.products.find({}, {'for': {'$slice': -2}}); // hiển thị 2 phần tử cuối cùng của mảng
db.products.find({}, {'for': {'$slice': [2, 3]}}); // skip 2 phần tử đầu và giới hạn số lượng
hiển thị là 3
db.products.find({'additional_tariffs': {'$size': 2}}); // tìm các document có chiều dài
mảng 'additional_tariffs' là 2

```

- Array of embedded documents

```

db.products.find({'additional_tariffs.kind': /^fe/, 'additional_tariffs.amount': 2.25})
// các thành phần chỉ cần thoả trong 'additional_tariffs'
db.products.find({'additional_tariffs':
    {$elemMatch: {kind: /^fe/, amount: 2.25}}});
// cả hai thành phần phải thoả trong cùng 1 subdocument con của mảng
'additional_tariffs'

```

Bài tập

Import restaurants.json vào database tên lab09 và collection tên restaurants
 mongoimport --db=lab09 --collection=restaurants --file=restaurants.json --jsonArray

1. Hiển thị tất cả document trong restaurants
2. Đếm số lượng documents của restaurants
3. Hiển thị id, name, phone và categories của tất cả các document trong restaurants
4. Hiển thị 5 nhà hàng đầu tiên có stars > 3, sort theo thứ tự từ cao đến thấp
5. Hiển thị 5 nhà hàng tiếp theo sau khi skip 5 nhà hàng đầu tiên có stars > 3, sort theo thứ tự từ cao đến thấp
6. Tìm các nhà hàng có vị trí latitude < -95.754168
7. Tìm các nhà hàng không có món ăn Californian, điểm stars >= 4, và latitude < -65.754168, sắp xếp theo thứ tự stars giảm dần
8. Hiển thị _id, name, stars, categories, phone, và email của các nhà hàng có tên bắt đầu bằng 'Wil'

9. Hiển thị `_id`, `name`, `stars`, `categories`, `phone`, và `email` của các nhà hàng có tên kết thúc bằng 'ces'
10. Hiển thị `_id`, `name`, `stars`, `categories`, `phone`, và `email` của các nhà hàng có tên chứa các kí tự 'Reg'.
11. Tìm các nhà hàng có cả ba món Russian, Chinese và Vietnamese
12. Tìm các nhà hàng có số lượng phần tử trong `grade` bằng 5
13. Tìm các nhà hàng có ít nhất một phần tử trong `grades` lớn hơn hoặc bằng 20 và nhỏ hơn 30
14. Tìm các nhà hàng có số điện thoại bắt đầu bằng 770 và email kết thúc bằng '.com'