



Bài 3

Các thao tác với document

- › Thao tác với document
- › Insert Document
- › Query Document
- › Update Document
- › Delete Document
- › Limiting Records
- › Sorting Records

- Phương thức insert() để chèn dữ liệu vào collection, bạn cần sử dụng phương thức insert() hoặc save() của MongoDB.

- **Cú pháp**

- Cú pháp cơ bản của lệnh **insert()** như sau:

db.COLLECTION_NAME.insert(document)

Ví dụ thêm tài liệu vào collection users

```
db.users.insert({name: "Admin Manager",email:"admin@gmail.com",password:"123456"})
```

- Thêm nhiều tài liệu bằng cách truyền mảng các đối tượng tài liệu

```
db.users.insert([
  {name: "User 1",email:"user1@gmail.com",password:"123456"},
  {name: "User 2",email:"user2@gmail.com",password:"123456"},
  {name: "User 3",email:"user3@gmail.com",password:"123456"}
])
```

- Để truy vấn dữ liệu từ bộ sưu tập MongoDB, bạn cần sử dụng phương thức find() của MongoDB.
- Cú pháp cơ bản của phương thức find () như sau:
db.COLLECTION_NAME.find()
- Phương thức find() sẽ hiển thị tất cả các tài liệu theo cách không có cấu trúc.

```
> db.users.find()
{ "_id" : ObjectId("62c8ed3f929267bea65df699"), "name" : "Admin Manager", "email" : "admin@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69a"), "name" : "User 1", "email" : "user1@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69b"), "name" : "User 2", "email" : "user2@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69c"), "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }
>
```

Phương thức find().pretty()

- Phương thức pretty() khi được kết hợp sẽ định dạng dữ liệu
- Cú pháp: `db.COLLECTION_NAME.find().pretty()`

- Ví dụ:

```
> db.users.find().pretty()
{
  "_id" : ObjectId("62c8ed3f929267bea65df699"),
  "name" : "Admin Manager",
  "email" : "admin@gmail.com",
  "password" : "123456"
}
{
  "_id" : ObjectId("62c8eec0929267bea65df69a"),
  "name" : "User 1",
  "email" : "user1@gmail.com",
  "password" : "123456"
}
{
  "_id" : ObjectId("62c8eec0929267bea65df69b"),
  "name" : "User 2",
  "email" : "user2@gmail.com",
  "password" : "123456"
}
{
  "_id" : ObjectId("62c8eec0929267bea65df69c"),
  "name" : "User 3",
  "email" : "user3@gmail.com",
  "password" : "123456"
}
```

Phương thức find() – Chỉ lấy các field chỉ định

- Phương thức pretty() khi được kết hợp sẽ định dạng dữ liệu

- Cú pháp:

```
db.COLLECTION_NAME.find({}, {KEY1:1,...}).pretty()
```

- Trong cú pháp trên ta thấy có KEY1:1 => nghĩa là được chọn, KEY2:0 => nghĩa là không chọn
- Ví dụ sau đây loại bỏ _id trong kết quả trả về

```
db.users.find({}, {_id:0})
{ "name" : "Admin Manager", "email" : "admin@gmail.com", "password" : "123456" }
{ "name" : "User 2", "email" : "myemail@gmail.com", "password" : "123456" }
{ "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }
```

- find(), có thể truy vấn với điều kiện nào đó
- Ví dụ sau đây, truy vấn tìm user có name = Admin Manager

```
db.users.find({name:"Admin Manager"})
{ "_id" : ObjectId("62c8ed3f929267bea65df699"), "name" :
"Admin Manager", "email" : "admin@gmail.com", "password" :
"123456" }
```

- Để truy vấn nhiều điều kiện kết hợp sử dụng **\$and**, đảm bảo các điều kiện thỏa mãn thì mới có dữ liệu trả về. Sau đây là cú pháp cơ bản của AND

```
db.mycol.find({ $and: [ {<key1>:<value1>}, { <key2>:<value2>} ] })
```

Ví dụ sau đây truy vấn theo name và email

```
db.users.find({ $and: [{name:"Admin Manager"},  
  { email:"admin@gmail.com"}] })  
{ "_id" : ObjectId("62c8ed3f929267bea65df699"), "name" :  
  "Admin Manager", "email" : "admin@gmail.com", "password" :  
  "123456" }
```


- Để truy vấn với một trong các điều kiện thỏa mãn sử dụng **\$or**, đảm bảo một trong các điều kiện thỏa mãn thì mới có dữ liệu trả về. Sau đây là cú pháp cơ bản của AND

```
db.mycol.find({ $or: [ {<key1>:<value1>}, { <key2>:<value2>} ] })
```

- Ví dụ sau đây truy vấn theo name hoặc email

```
db.users.find({ $or: [{name:"Admin Manager"}, { email:"user1@gmail.com"}]})
{ "_id" : ObjectId("62c8ed3f929267bea65df699"), "name" : "Admin Manager", "email" : "admin@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69a"), "name" : "User 1", "email" : "user1@gmail.com", "password" : "123456" }
```

- Để truy vấn gần đúng chúng ta sử dụng mẫu ký tự như trong biểu thức chính quy tương tự như javascript Expression

```
db.users.find({name: /mẫu ký tự/i})
```

- Ví dụ sau đây sẽ truy vấn tất cả các users với thuộc tính name có ký tự user

```
db.users.find({name: /user/i})
{ "_id" : ObjectId("62c8eec0929267bea65df69a"), "name" : "User 1", "email" : "user1@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69b"), "name" : "User 2", "email" : "user2@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69c"), "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }
```

- Với \$in, dữ liệu sẽ thỏa mãn các điều kiện trong mảng đã cho

```
db.users.find({field: {$in: ["value 1", "value 3"]}})
```

- Ví dụ sau đây sẽ truy vấn tất cả các users với thuộc tính name User 1 và User 3

```
db.users.find({name: {$in: ["User 1", "User 3"]}})
{ "_id" : ObjectId("62c8eec0929267bea65df69a"), "name" : "User 1", "email" :
"user1@gmail.com", "password" : "123456" }
{ "_id" : ObjectId("62c8eec0929267bea65df69c"), "name" : "User 3", "email" :
"user3@gmail.com", "password" : "123456" }
```

- Với \$not \$in, sẽ loại bỏ các dữ liệu thỏa mãn các điều kiện trong mảng đã cho

```
db.users.find({field: {$not: {$in: ["value 1", "value2"]}}})
```

- Ví dụ sau đây sẽ truy vấn loại bỏ tất cả các users với thuộc tính name User 1 và User 3

```
db.users.find({name: {$not: {$in: ["User 1", "User 3"]}}})  
{ "_id" : ObjectId("62c8ed3f929267bea65df699"), "name" : "Admin Manager", "email" :  
  "admin@gmail.com", "password" : "123456" }  
{ "_id" : ObjectId("62c8eec0929267bea65df69b"), "name" : "User 2", "email" :  
  "user2@gmail.com", "password" : "123456" }
```

findOne() – Trả về một tài liệu

- findOne(), chỉ trả về một tài liệu theo điều kiện nào đó
- Ví dụ sau đây, truy vấn tìm user có name = Admin Manager

```
db.users.findOne({name:"Admin Manager"})
{
  "_id" : ObjectId("62c8ed3f929267bea65df699"),
  "name" : "Admin Manager",
  "email" : "admin@gmail.com",
  "password" : "123456"
}
```

- Phương thức update () cập nhật các giá trị trong tài liệu hiện có.

```
db.COLLECTION_NAME.update(ĐIỀU_KIỆN, DỮ_LIỆU_UPDATE)
```

- Ví dụ sau đây sẽ cập nhật email của users có name là User 2

```
db.users.update({name:"User 2"}, {$set: {email: "myemail@gmail.com"}})  
=> Kết quả  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Phương thức remove () xóa các tài liệu thỏa mãn điều kiện nào đó.

```
db.COLLECTION_NAME.update(ĐIỀU_KIỆN, DỮ_LIỆU_UPDATE)
```

- Ví dụ sau đây sẽ cập nhật email của users có name là User 2

```
db.users.update({name:"User 2"}, {$set: {email: "myemail@gmail.com"}})  
=> Kết quả  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Phương thức update () cập nhật các giá trị trong tài liệu hiện có.

```
db.users.remove({field:"value"})  
=> Nếu chỉ muốn xóa một tài liệu  
db.users.remove({field:"value"}, 1)
```

- Ví dụ sau đây sẽ xóa users có name là User 1

```
db.users.remove({name:"User 1"}, 1)  
=> Kết quả  
WriteResult({ "nRemoved" : 1 })
```


- Phương thức limit() sẽ giới hạn số tài liệu trả về
- Phương thức skip() bỏ qua số tài liệu

```
db.COLLECTION_NAME.find().limit(NUMBER).skip(NUMBER)
```

- Ví dụ sau đây chỉ lấy về 2 tài liệu, bỏ qua 1 tài liệu

```
db.users.find({}, {_id:0}).limit(2).skip(1)
{ "name" : "User 2", "email" : "myemail@gmail.com", "password" : "123456" }
{ "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }
```

Phương thức sort() - sắp xếp dữ liệu

- Phương thức sort() sử dụng khi cần sắp xếp dữ liệu theo trường nào đó
- Sắp xếp có thể là tăng dần, giảm dần
- Cú pháp

```
- Sắp xếp tăng dần  
db.users.find({}, {_id:0}).sort({name:1})  
- Sắp xếp giảm dần  
db.users.find({}, {_id:0}).sort({name:-1})
```

- Ví dụ sau đây sắp xếp theo name tăng dần

```
db.users.find({}, {_id:0}).sort({name:1})  
{ "name" : "Admin Manager", "email" : "admin@gmail.com", "password" : "123456" }  
{ "name" : "User 2", "email" : "myemail@gmail.com", "password" : "123456" }  
{ "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }
```

- Ví dụ sau đây sắp xếp theo name giảm dần

```
db.users.find({}, {_id:0}).sort({name:-1})  
{ "name" : "User 3", "email" : "user3@gmail.com", "password" : "123456" }  
{ "name" : "User 2", "email" : "myemail@gmail.com", "password" : "123456" }  
{ "name" : "Admin Manager", "email" : "admin@gmail.com", "password" : "123456" }
```

THANK FOR WATCH !



TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn