

MONGODB - CRUD

Create - Read - Update - Delete

Lý thuyết

- Tạo, xóa database

```
use university;
db.dropDatabase();
show dbs;
Show databases;
```

- Tạo, xóa collection

```
show collections;
db.createCollection('student');
db.student.renameCollection('students');
db.students.drop();
```

- Insert dữ liệu

- db.collection.insertOne()

```
db.students.insertOne({full_name: 'Peter Pan'});
```

- db.collection.insertMany()

```
db.students.insertMany([
  {full_name: 'Arya Stark'},
  {full_name: 'Khal Grogo'}
]);
```

- Cập nhật dữ liệu

- db.collection.replaceOne({}, {})

```
{ "_id": ObjectId("6644664cb270dd605702633a"),
  Full_name: 'Paul',
  Age: 20 }
db.students.replaceOne(
  { "_id": ObjectId("6644664cb270dd605702633a") },
  { full_name: 'Paul Pan' }
);
```

```

db.students.replaceOne(
  {full_name: 'Paula Pan'},
  {first_name: 'Paula', last_name: 'Lee',
    age: 20, email: 'paulalee@gmail.com'},
  {upsert: true} // trường hợp không tìm được
  document thỏa đk thì thêm mới
);

```

Lưu ý: _id không thể thay đổi

- db.collection.findOneAndReplace()

```

db.students.insertMany([
  {"_id": 1011, first_name: "Lenny"},
  {"_id": 1013, first_name: "Lenny"},
  {"_id": 1017, first_name: "Lenny"},
  {"_id": 1021, first_name: "Lenny"},
]);

```

```

db.students.findOneAndReplace(
  {first_name: "Lenny"},
  {full_name: "Lenny Liu", email:
"lenny@abc.com"}
); // thay thế document đầu tiên được tìm thấy

```

- db.collection.updateOne({}, {})

- \$set: // thay thế giá trị nếu trường cần cập nhật đã có và thêm mới nếu chưa có

```

db.students.updateOne(
  {full_name: 'Lenny Liu'},
  {$set: {age: 21,
    subject: ['science', 'math']
  }});

db.students.updateMany(
  {first_name: "Lenny"},
  {$set: {age: 23,

```

```

        subject: ['math']]

    });

    db.students.updateMany({}, {$set: {age: 21}});
- $unset: // loại bỏ trường subject ra khỏi document cần
  cập nhật
    db.students.updateOne(
        {first_name: "Lenny"},
        {$unset: {subject: 1}}
    );

- $push: $addToSet // thêm 1 giá trị vào cuối 1 mảng nếu
  mảng đó tồn tại và thêm mảng mới nếu chưa tồn tại
    db.students.updateOne(
        {full_name: "Paul Pan"},
        {$push: {subject: "science"}});

    db.students.updateOne(
        {full_name: "Paul Pan"},
        {$addToSet: {subject: "chemistry"}});

    db.students.updateOne(
        {full_name: "Arya Stark"},
        {$push:
            {subject: {$each: ["science",
                "history"]}}});
        - > subject: ["math", "science",
            "history"]

    db.students.updateOne(
        {full_name: "Arya Stark"},
        {$push:
            {subject: ["science", "history"]}});
        - > subject: ["math", ["science",
            "history"]]

- $pull: // remove 1 phần tử ra khỏi mảng

```

```

db.students.updateOne(
  {full_name: "Arya Stark"},
  {$pull:
    {subject: "science"}});
$pop: //remove phần tử đầu tiên hoặc cuối cùng của
mảng
db.students.findOneAndUpdate(
  {full_name: "Arya Stark"},
  {$pop: {subject: -1}},
  {returnNewDocument : true}
);
//1: phần tử cuối cùng; -1: phần tử đầu tiên
$pullAll
db.students.findOneAndUpdate(
  {},
  {$pullAll:
    {subject: ["science", "math"]}},
  {returnNewDocument : true});

```

- \$inc: //tăng hoặc giảm giá trị

```

db.students.updateOne(
  {first_name: "Paula"},
  {$inc: {age: 3}} // tăng tuổi thêm 3 đơn vị
);

```

- findOneAndUpdate() → trả về document trước khi được chỉnh sửa

- Xóa dữ liệu

- db.collection.deleteOne()

```

db.students.deleteOne({full_name: "Khal Grogo"});

```

- db.collection.deleteMany()

```

db.students.deleteMany({age: 23});

```

- db.collection.deleteMany({}) - xoá toàn bộ documents trong collection hiện tại -

db.collection.drop()

- db.collection.findOneAndDelete() → trả về document bị xóa

```

db.students.findOneAndDelete({full_name: "Khal

```

```
Grogo"}});
```

- Truy vấn cơ bản

- `db.collection.find() = db.collection.find({})`

```
db.students.find() = db.students.find({})
```

= `SELECT * FROM students;`

- `db.collection.find({}, {})` → ({nội dung cần tìm}, {hiển thị ra bên ngoài})

```
db.students.find({}, {full_name: 1, email: 1}) // _id
```

= `SELECT full_name, email FROM students;`

```
db.students.find({full_name: "Paul Pan"}, {full_name: 1, age: 1, _id: 0})
```

= `SELECT full_name, age FROM students WHERE full_name='Paul Pan';`

- `db.collection.find().pretty()`

- `db.collection.findOne()`

```
db.students.findOne({first_name: "Paula"})
```

- `db.collection.distinct()` → kết quả trả về 1 array

```
db.students.distinct("age");
```

= `SELECT DISTINCT age FROM students;`

```
db.students.distinct("first_name", {age: 22})
```

= `SELECT DISTINCT first_name FROM students WHERE age=22;`

- `db.collection.countDocuments({})` - - đếm số lượng document trong 1 collection -

```
db.students.countDocuments();
```

= `SELECT COUNT(*) FROM students;`

```
db.students.countDocuments({age: 20});
```

= `SELECT COUNT(*) FROM students WHERE age=20;`

- AND, OR, NOT, IN, NIN, ALL

- `$and: [{},{},{}]`

```
db.students.countDocuments({
  $and: [{age: 22}, {first_name: "Lala"}]
});
```

```
= db.students.countDocuments({age: 22, first_name:
```

```
"Lala"});
```

= `SELECT COUNT(*) FROM students`

WHERE age=22 AND first_name='Lala';

- \$or

```
db.students.find(
  {$or: [{age: 22}, {first_name: "Lala"}]},
  {first_name: 1, _id: 0}
);
= SELECT first_name FROM students
  WHERE age=22 OR first_name="Lala";
```

- \$in

```
db.students.updateMany(
  {first_name: {$in: ["Paula", "Lala"]}},
  {$set: {age: 22}}
);
```

```
db.students.find({age: {$in: [21, 22]}});
= SELECT * FROM students WHERE age IN (21, 22);
```

- \$nin - not in

```
db.students.find({age: {$nin: [21, 23]}});
= SELECT * FROM students WHERE age NOT IN (21, 23);
```

- \$all - > array

```
db.students.find({subject: {$all: ["math",
  "science"]}}
  ));
```

- Các toán tử so sánh

- \$eq: = equal

```
db.students.find({age: 22}) =
db.students.find({age: {$eq: 22}});
```

- \$ne: != not equal

```
db.students.find({age: {$ne: 22}})
```

- \$lt: < less than

```
db.students.find({age: {$lt: 22}})
```

- \$lte: <= less than equal

```
db.students.find({age: {$lte: 22}}).count();
db.students.countDocuments({age: {$lte: 22}});
```

- \$gt: > greater than

```
db.students.find({age: {$gt: 21}}).count()
```

- \$gte: >= greater than equal

```
db.students.find({age: {$gte: 21}}).count()
```

```
db.students.find({age: {$gte: 18, $lte: 30}})
```

Bài tập

1. Tạo database tên 'mydatabase'
2. Tạo collection tên 'students' trong db 'mydatabase'
3. Nhập dữ liệu sau vào collection 'students'
 - a. name: John; age: 22, subject: math, science, history
 - b. name: Alice; age: 25; subject: history, science
 - c. name: Izzy; age: 23; subject: math, science
 - d. name: Penny; age: 22; subject: math, history
 - e. name: Peter; age: 18; subject: history
 - f. name: Paul; age: 18; subject: math
 - g. name: John; age: 24, subject: math, history
 - h. name: Joe; age: 23, subject: history, email: joe@abc.edu.vn
4. Truy vấn tất cả các document trong collection 'students'
5. Truy vấn tên của các sinh viên trong collection 'students'
6. Tìm các sinh viên có độ tuổi bằng 22
7. Cập nhật tuổi của John bằng 21
8. Thêm một trường grade với giá trị là "A" cho tất cả các documents trong collection 'students'
9. Cập nhật grade là "B" cho sinh viên tên 'Alice'
10. Tìm các sinh viên có độ tuổi từ 21 đến 23
11. Tìm các sinh viên có tên là Paul hoặc Peter hoặc Penny
12. Tìm các sinh viên học cả 2 môn math và science
13. Tìm các sinh viên có tên John và có độ tuổi là 21 hoặc 24
14. Hiển thị name và age của sinh viên có tham gia subject là 'math' 15.
- Đổi tên John thành Johnathan với sinh viên có tên John và tuổi là 21 16.
- Hiển thị các độ tuổi khác nhau trong collection students
17. Xoá sinh viên có tên Alice ra khỏi collection 'students'
18. Xoá trường 'grade' trong collection 'students'
19. Xoá collection 'students'