

- Chuẩn bị các câu lệnh xóa bằng cách sử dụng lớp BulkWrite() của Trình quản lý trình điều khiển MongoDB.
- Lớp BulkWrite() có phương thức **delete()** nhận vào 2 tham số, mỗi tham số là một mảng
- Tham số thứ nhất là mảng điều kiện lọc
- Tham số thứ hai là mảng chỉ định số bản ghi xóa là 1 hay nhiều
- Lưu ý: Trong tham số thứ 2 nếu **limit = 1** thì xóa 1 bản ghi đầu tiên tìm được, ngược lại là 0 thì xóa tất cả các bản ghi phù hợp

```
$bulkWrite->delete(  
  ['name' => 'Liton'],  
  ['limit' => 1] // Nếu 0 thì xóa các bản ghi phù hợp  
);  
$result = $conn->executeBulkWrite("mydb.customers", $bulkWrite);
```



# Bài 7

## MongoDB với NodeJS, PHP, Java, C#

- › Các thao tác với CSDL MongoDB sử dụng NodeJS, PHP, Java, C#

- Trong dự án NodeJS hãy sử dụng lệnh cài đặt gói thư viện mongodb  
`npm install mongodb`



- Để tạo cơ sở dữ liệu trong MongoDB, hãy bắt đầu bằng cách tạo đối tượng MongoClient, sau đó chỉ định URL kết nối với địa chỉ ip chính xác và tên của cơ sở dữ liệu bạn muốn tạo.
- MongoDB sẽ tạo cơ sở dữ liệu nếu nó không tồn tại và tạo kết nối với nó.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  console.log("Database created!");
  db.close();
});
```

- *Lưu ý:* Để biết chuỗi kết nối, bạn có thể khởi động máy chủ mongodb như các bài trước

```
C:\Users\luong>mongo
MongoDB shell version v5.0.9
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4955403d-c782-43d3-b94d-ba00c38c43b4") }
MongoDB server version: 5.0.9
```

- Để tạo collection trong MongoDB, hãy sử dụng phương thức **createCollection()**

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/"; // chuỗi kết nối
MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb"); // sử dụng CSD đã tạo
  // tạo collection tên là customers
  dbo.createCollection("customers", function (err, res) {
    if (err) throw err;
    console.log("Collection created!");
    db.close();
  });
});
```

- Để chèn một bản ghi (tài liệu) vào một bộ sưu tập, sử dụng phương thức **insertOne()**
- Tham số đầu tiên của phương thức insertOne() là một đối tượng json giá trị của mỗi trường trong tài liệu bạn muốn chèn. Nó cũng có một hàm callback, hàm callback này có tham số lỗi và tham số kết quả.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = { name: "Company Inc", address: "Highway 37" };
  dbo.collection("customers").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

- Để tìm tài liệu trong một bộ sưu tập, bạn có thể lọc kết quả bằng cách sử dụng phương thức find()
- Đối số đầu tiên của phương thức find() là một đối tượng truy vấn để lọc kết tìm kiếm
- Nếu đối tượng truy vấn rỗng thì sẽ hiển thị tất cả bản ghi trong collection

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var query = { address: "Highway 37" }; // đối tượng truy vấn
  dbo.collection("customers").find(query).toArray(function (err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```



- Sử dụng phương thức sort() để sắp xếp kết quả theo thứ tự tăng dần hoặc giảm dần.
- Phương thức sort() nhận một tham số là một đối tượng xác định kiểu sắp xếp, đối tượng này có key là trường mà bạn muốn sắp xếp, và value (1 hoặc -1) là kiểu sắp xếp

- VD sắp xếp theo name  
`{ name: 1 } // ascending`  
`{ name: -1 } // descending`

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var mysort = { name: 1 }; // đối tượng sắp xếp
  dbo.collection("customers").find().sort(mysort).toArray(function (err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

- Để giới hạn kết quả trong MongoDB, sử dụng phương thức limit ().
- Phương thức limit() nhận một tham số, xác định có bao nhiêu tài liệu để trả về.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var query = {};
  dbo.collection("customers").find().limit(5).toArray(function (err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

- MongoDB không phải là cơ sở dữ liệu quan hệ, nhưng có thể thực hiện phép left outer join bằng cách sử dụng \$lookup.
- \$lookup cho phép chỉ định collection muốn kết hợp với collection hiện tại và những trường nào phù hợp.
- Hãy xem xét collection “categories” và collection “products”

```
categories collection
[
  { _id: 1, name: 'Áo Nam', status: 1 },
  { _id: 2, name: 'Áo Nữ', status: 1 },
  { _id: 3, name: 'Đồng Hồ', status: 1 }
]
```

```
product collection
[
  { _id: 1, name: 'Chocolate Heaven', category_id: 1 },
  { _id: 2, name: 'Tasty Lemons', category_id: 1 },
  { _id: 3, name: 'Vanilla Dreams', category_id: 2 }
];
```

- Tham khảo cú pháp join với **\$lookup**

```
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");

  dbo.collection('categories').aggregate([
    {
      $lookup:
      {
        from: 'products',
        localField: '_id',
        foreignField: 'category_id',
        as: 'products'
      }
    }
  ]).toArray(function (err, res) {
    if (err) throw err;
    console.log(JSON.stringify(res));
    db.close();
  });
});
```

Kết quả truy vấn tương tự như sau

```
[
  {
    "_id": 1, "name": "Áo Nam", "status": 1,
    "products": [
      { "_id": 1, "name": "Chocolate Heaven",
"category_id": 1 },
      { "_id": 2, "name": "Tasty Lemons", "category_id":
1 }
    ]
  },
  {
    "_id": 2, "name": "Áo Nữ", "status": 1,
    "products": [
      { "_id": 3, "name": "Vanilla Dreams", "category_id":
2 }
    ]
  },
  {
    "_id": 3, "name": "Đồng Hồ", "status": 1,
    "products": []
  }
]
```

- Để xóa một bản ghi hãy sử dụng phương thức delete().
- Tham số đầu tiên của phương thức delete() là một đối tượng truy vấn xác định tài liệu nào cần xóa. Xóa tất cả các bản ghi tìm được
- Ngoài ra có phương thức deleteOne() xóa một bản ghi tìm được đầu tiên

```
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myquery = { address: 'Mountain 21' }; // đối tượng tìm kiếm
  dbo.collection("customers").delete(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

- Bạn có thể cập nhật bản ghi bằng cách sử dụng phương thức updateOne().
- Tham số đầu tiên của phương thức updateOne () là một đối tượng truy vấn xác định tài liệu nào cần cập nhật.
- **Lưu ý:** Nếu truy vấn tìm thấy nhiều bản ghi, chỉ lần xuất hiện đầu tiên được cập nhật.

```
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function (err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");

  var myquery = { address: "Valley 345" }; //đối tượng truy vấn
  // dữ liệu cập nhật mới
  var newvalues = { $set: { name: "Mickey", address: "Canyon 123" } };
  dbo.collection("customers").updateOne(myquery, newvalues, function (err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

- Để có thể làm việc với CSDL mongodb trong php bạn cần cài đặt thêm driver
- Tải mongodb driver tại đây: <https://pecl.php.net/package/mongodb>
- Sau khi tải về giải nén và copy file php\_mongodb.dll và thư mục module mở rộng của xampp, thường là **C:\xampp\php\ext**
- Hiện tại trên máy này là **D:\xampp\php\ext**

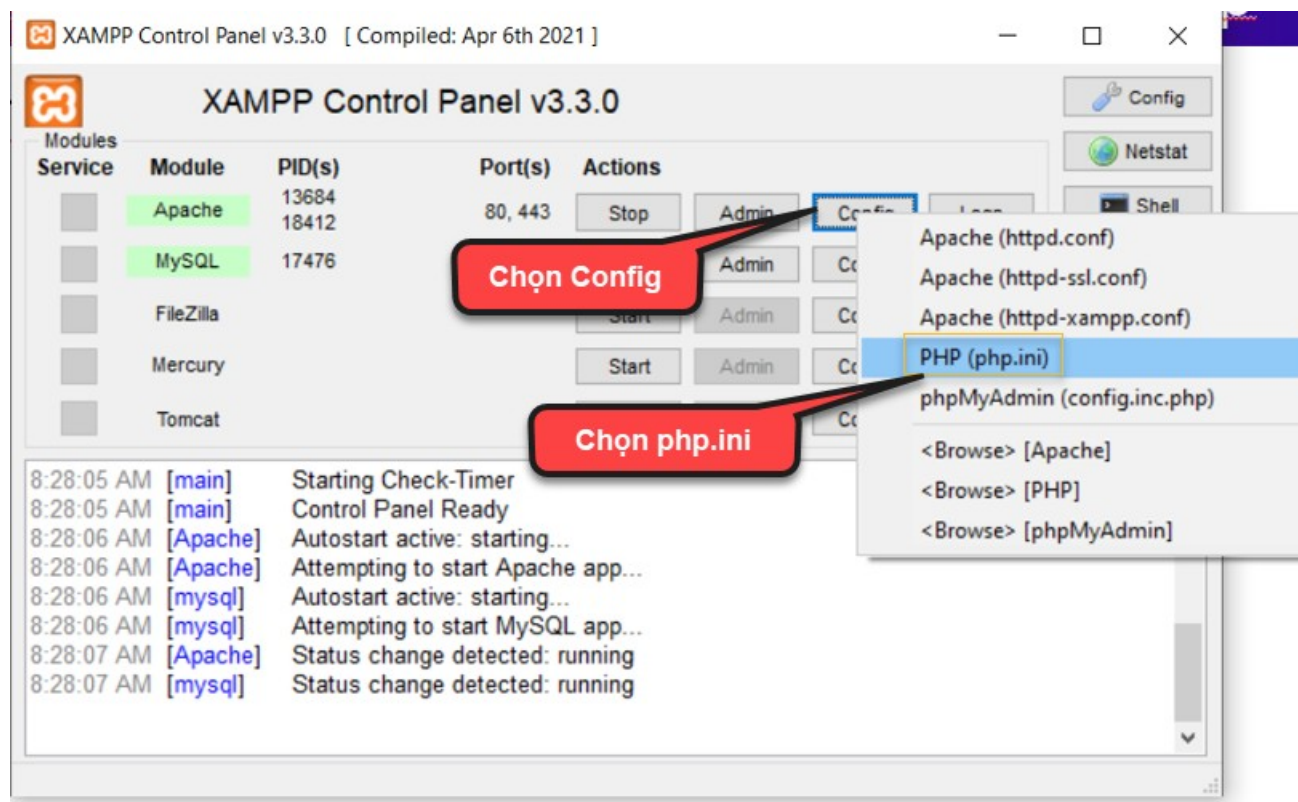
Local Disk (D:) > xampp > php > ext

Name	Date modified	Type	Size
php_intl.dll	2/24/2022 9:39 AM	Application extens...	381 KB
php_ldap.dll	2/24/2022 9:39 AM	Application extens...	252 KB
php_mbstring.dll	2/24/2022 9:39 AM	Application extens...	1,414 KB
php_mongodb.dll	12/8/2018 2:57 PM	Application extens...	837 KB
php_mysqli.dll	2/24/2022 9:39 AM	Application extens...	110 KB
php_oci8_12c.dll	2/24/2022 9:39 AM	Application extens...	150 KB

**Lưu ý:** Chọn mongodb phù hợp với phiên bản php đang cài trên máy  
VD trên máy này php 7.4 thì tìm đến phiên bản mongodb 1.8.2  
Trong danh sách tải về, tìm để tải bản Thread Safe (TS) x64

- Tiếp theo tìm mở file php.ini và thêm dòng sau vào cuối file, sau đó stop -> start lại apache

`extension = php_mongodb.dll`





- Mở trình duyệt lên truy cập vào phpinfo của localhost

<http://localhost/dashboard/phpinfo.php>

- Di chuyển xuống dưới sẽ có thông tin của mongodb

mongodb		
MongoDB support		enabled
MongoDB extension version	1.8.2	
MongoDB extension stability	stable	
libbson bundled version	1.17.2	
libmongoc bundled version	1.17.2	
libmongoc SSL	enabled	
libmongoc SSL library	OpenSSL	
libmongoc crypto	enabled	
libmongoc crypto library	libcrypto	
libmongoc crypto system profile	disabled	
libmongoc SASL	enabled	
libmongoc ICU	disabled	
libmongoc compression	disabled	
libmongocrypt bundled version	1.0.4	
libmongocrypt crypto	enabled	
libmongocrypt crypto library	libcrypto	
Directive	Local Value	Master Value
<b>mongodb.debug</b>	no value	no value

- Tạo một thư mục dự án và file index.php soạn code sau

```
try {  
    $manager = new MongoDB\Driver\Manager("mongodb://localhost:27017/");  
} catch (MongoDBDriverException $e) {  
    echo $e->getMessage();  
}
```

- Nếu kết nối thành công thì không có thông báo gì.
- Ngược lại sẽ có thông báo lỗi

- Chuẩn bị các câu lệnh insert bằng cách sử dụng lớp BulkWrite() của Trình quản lý trình điều khiển MongoDB.
- Lớp BulkWrite() có phương thức **insert()** nhận vào mảng dữ liệu là thông tin bản ghi của collection đang cần thêm mới
- Cuối cùng cần thực thi lệnh thêm mới bằng phương thức executeBulkWrite() có sẵn trong biến kết nối

```
$data1 = [  
    "name" => "William Ka",  
    "address" => "Central st 885"  
];  
$data2 = [  
    "name" => "William Ko",  
    "address" => "Central st 125"  
];  
  
// Chuẩn bị dữ liệu trước khi thực thi  
$bulkWrite = new MongoDB\Driver\BulkWrite();  
$bulkWrite->insert($data1);  
$bulkWrite->insert($data2);  
  
// Thực hiện thêm mới  
$manager->executeBulkWrite("mydb.customers", $bulkWrite);
```

```
Kết quả sau khi thêm thành công  
MongoDB\Driver\WriteResult Object  
(  
    [nInserted] => 2  
    [nMatched] => 0  
    [nModified] => 0  
    [nRemoved] => 0  
    [nUpserted] => 0  
    [upsertedIds] => Array...  
)
```

- Để chuẩn bị thực thi truy vấn danh sách bản ghi, hãy sử dụng lớp `MongoDB\Driver\Query($filter, $option)`
- Lớp này có 2 tham số đầu vào là **filter** (mảng các điều kiện lọc) **option** là các tùy chọn khác tương ứng với câu truy vấn cần thực hiện

```
$filter = []; // nếu muốn lọc kết quả, truyền mảng ['name' => 'Amy']  
$option = []; // các tùy chọn  
$read = new MongoDB\Driver\Query($filter, $option);  
$customers = $manager->executeQuery("mydb.customers", $read);
```

- Sau đó có thể duyệt dữ liệu như mã bên  
**Customer List**

STT	Name	Address
1	William Ka	Central st 885
2	William Ko	Central st 125
3	Company Inc	Highway 37

```
<h2>Customer List</h2>  
<table class="table table-bordered">  
  <thead>  
    <tr>  
      <th>STT</th>  
      <th>Name</th>  
      <th>Address</th>  
    </tr>  
  </thead>  
  <tbody>  
    <?php foreach($customers as $key => $cus) : ?>  
      <tr>  
        <td><?php echo $key + 1;?></td>  
        <td><?php echo $cus->name;?></td>  
        <td><?php echo $cus->address;?></td>  
      </tr>  
    <?php endforeach ;?>  
  </tbody>  
</table>
```

- Chuẩn bị các câu lệnh cập nhật bằng cách sử dụng lớp BulkWrite() của Trình quản lý trình điều khiển MongoDB.
- Lớp BulkWrite() có phương thức **update()** nhận vào 3 tham số, mỗi tham số là một mảng
- Tham số thứ nhất là mảng điều kiện lọc
- Tham số thứ hai là mảng dữ liệu mới cần update
- Tham số thứ ba là tùy chọn update 1 hay nhiều bản ghi
- Lưu ý: Trong tham số thứ 3 nếu **multi = false** thì cập nhật 1 bản ghi đầu tiên tìm được

```
$bulkWrite = new MongoDB\Driver\BulkWrite();  
$bulkWrite->update(  
    ['name' => 'Amy'],  
    ['$set' => ['name' => 'Liton', 'address' => 'Apple st 888']],  
    ['multi' => false, 'upsert' => false]  
);  
  
$result = $conn->executeBulkWrite("mydb.customers", $bulkWrite);
```

- Chuẩn bị các câu lệnh xóa bằng cách sử dụng lớp BulkWrite() của Trình quản lý trình điều khiển MongoDB.
- Lớp BulkWrite() có phương thức **delete()** nhận vào 2 tham số, mỗi tham số là một mảng
- Tham số thứ nhất là mảng điều kiện lọc
- Tham số thứ hai là mảng chỉ định số bản ghi xóa là 1 hay nhiều
- Lưu ý: Trong tham số thứ 2 nếu **limit = 1** thì xóa 1 bản ghi đầu tiên tìm được, ngược lại là 0 thì xóa tất cả các bản ghi phù hợp

```
$bulkWrite->delete(  
    ['name' => 'Liton'],  
    ['limit' => 1] // Nếu 0 thì xóa các bản ghi phù hợp  
);  
$result = $conn->executeBulkWrite("mydb.customers", $bulkWrite);
```

- Để kết nối CSDL MongoDB trong C#, trước tiên cần tạo một dự án C# mới
- Sau đó Add NuGet Package tên là MongoDB.Driver và dự án này và soạn code sau

```
using MongoDB.Driver;
using System;
using System.Linq;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");

            var dbList = dbClient.ListDatabases().ToList();

            Console.WriteLine("Danh sach CSDL dang co: ");
            foreach (var db in dbList)
            {
                Console.WriteLine(db);
            }
        }
    }
}
```

Kết quả

```
C:\Windows\system32\cmd.exe
Danh sach CSDL dang co:
{ "name" : "admin", "sizeOnDisk" : NumberLong(40960), "empty" : false }
{ "name" : "config", "sizeOnDisk" : NumberLong(73728), "empty" : false }
{ "name" : "demo_db", "sizeOnDisk" : NumberLong(49152), "empty" : false }
{ "name" : "local", "sizeOnDisk" : NumberLong(73728), "empty" : false }
{ "name" : "my_db", "sizeOnDisk" : NumberLong(81920), "empty" : false }
{ "name" : "mydb", "sizeOnDisk" : NumberLong(155648), "empty" : false }
Press any key to continue . . .
```

- Trong khi kết nối thành công, thì cần chọn CSDL và collection để thao tác dữ liệu

```
using MongoDB.Bson;
using MongoDB.Driver;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
        }
    }
}
```



- Trước tiên cần tạo một BsonDocument mới để chèn vào cơ sở dữ liệu.

```
using MongoDB.Bson;
using MongoDB.Driver;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Khai báo đối tượng BsonDocument
            var document = new BsonDocument
            {
                { "name", "Peter Paula" },
                { "address", "Lowstreet 6" }
            };

            // thực hiện thêm mới
            collection.InsertOne(document);
        }
    }
}
```

- Để lấy bản ghi đầu tiên trong bộ sưu tập, chúng ta có thể sử dụng phương thức `FirstOrDefault` hoặc `FirstOrDefaultAsync` và in kết quả ra bảng điều khiển.

```
using MongoDB.Bson;
using MongoDB.Driver;
using System;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Đọc bản ghi đầu tiên
            var customer = collection.Find(new BsonDocument()).FirstOrDefault();
            Console.WriteLine(customer.ToString());
        }
    }
}
```

- Để lấy danh sách dữ liệu trong bộ sưu tập, chúng ta có thể sử dụng phương thức `find()`

```
using MongoDB.Bson;
using MongoDB.Driver;
using System;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Đọc bản ghi đầu tiên
            var customers = collection.Find(new BsonDocument()).ToList();
            foreach (var customer in customers)
            {
                Console.WriteLine(customer.ToString());
            }
        }
    }
}
```

```
using MongoDB.Bson;
using MongoDB.Driver;
using System;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Đọc bản ghi đầu tiên
            var customers = collection.Find(new BsonDocument()).ToList();
            foreach (var customer in customers)
            {
                Console.WriteLine(customer.ToString());
            }
        }
    }
}
```

- Sử dụng UpdateMany() để cập nhật dữ liệu
- Phương thức updateMany nhận vào 2 tham số là:
  - Filter: lọc kết quả
  - Update data

```
using MongoDB.Bson;
using MongoDB.Driver;
using System;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Đọc bản ghi đầu tiên có Filter
            var update = Builders<BsonDocument>.Update.Set("name", "Peter KaKa")
                                                         .Set("address", "Peter KaKa");
            var filter = Builders<BsonDocument>.Filter.Eq("name", "Peter KaKa");

            collection.UpdateMany(filter, update);
        }
    }
}
```

- Sử dụng UpdateMany() để cập nhật dữ liệu
- Phương thức updateMany nhận vào 2 tham số là:
  - Filter: lọc kết quả
  - Update data

```
using MongoDB.Bson;
using MongoDB.Driver;
using System;

namespace MongoDBConnectionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            MongoClient dbClient = new MongoClient("mongodb://localhost:27017/");
            // chọn CSDL
            var database = dbClient.GetDatabase("mydb");
            // chọn collection
            var collection = database.GetCollection<BsonDocument>("customers");
            // Đọc bản ghi đầu tiên có Filter
            var update = Builders<BsonDocument>.Update.Set("name", "Peter KaKa")
                                                         .Set("address", "Peter KaKa");
            var filter = Builders<BsonDocument>.Filter.Eq("name", "Peter KaKa");

            collection.UpdateMany(filter, update);
        }
    }
}
```

# 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](mailto:tuyensinh@bachkhoa-aptech.edu.vn)**



**[www.bachkhoa-aptech.edu.vn](http://www.bachkhoa-aptech.edu.vn)**