TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

\_\_&\_\_

**ĐỒ ÁN MÔN CƠ SỞ DỮ LIỆU NÂNG CAO**

**ĐỀ TÀI: TÌM HIỂU VỀ MONGODB**

**Giảng viên hướng dẫn:**Ths. Lương Trần Hy Hiến

**Sinh viên thực hiện:**Nhóm 80

Đặng Phúc Toàn 43.01.104.183

Lê Thanh Tuấn

Nguyễn Hoàng Duy

*Thành phố Hồ Chí Minh – Năm 2019*

1. **TÌM HIỂU SƠ LƯỢC NOSQL**
2. **GIỚI THIỆU VỀ NOSQL**

Với hầu hết các thời kỳ web, Hệ quản trị cơ sở dữ liệu quan hệ dựa trên SQL đã thống trị hầu hết các hệ Quản trị Cơ sở dữ liệu. Tuy nhiên, thời gian gần đây, một cách tiếp cận mới đã bắt đầu biết đến là NoSQL, tạo ra sự thay thế cho các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.

NoSQL còn có nghĩa là Non-Relational - không ràng buộc. Tuy nhiên, thuật ngữ đó ít phổ dụng hơn và ngày nay người ta thường dịch NoSQL thành Not Only SQL - Không chỉ SQL. NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm

Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các cơ sở dữ liệu quan hệ nguồn mở nhỏ nhưng không sử dụng SQL cho truy vấn.

Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: một thế hệ CSDL không ràng buộc, phân tán, nguồn mở, khả năng mở rộng theo chiều ngang, có thể lưu trữ, xử lý từ một lượng rất nhỏ cho tới hàng petabytes dữ liệu trong hệ thống có độ chịu tải, chịu lỗi cao với những đòi hỏi về tài nguyên phần cứng thấp.

Một số đặc điểm nhận dạng cho thế hệ CSDL mới này bao gồm: schema-free, hỗ trợ mở rộng dễ dàng, API đơn giản, nhất quán cuối (eventual consistency), không giới hạn không gian dữ liệu,...

Sau đây là danh sách các CSDL NoSQL:

**1. Wide Column Store / Column Families:** Hadoop/HBase – Apache, BigTable – Google, Cassandra - Facebook/Apache, Hypertable - Zvents Inc/Baidu, Cloudera, SciDB, Mnesia, Tablets,…

**2. Key-Value Store/Tuple store**

a. Key/value cache in RAM: memcached, Citrusleaf database, Velocity, Redis, Tuple space,...

b. Key/value save on disk: Memcachedb, Berkeley DB, Tokyo Cabinet, Redis,...

c. Eventually Consistent Key Value Store: Amazon Dynamo, Voldemort, Dynomite, KAI, Cassandra, Hibari, Project Voldemort,…

d. Ordered key-value store: NMDB, Memcachedb, Berkeley DB,...

e. Distributed systems: Apache River, MEMBASE, Azure Table Storage, Amazon Dynamo,...

**3. Document Store:** Apache Jackrabbit, CouchDB, IBM Lotus Notes Storage Format (NSF), **MongoDB**, Terrastore, ThruDB, OrientDB, RavenDB,...

**4. Graph Database**: Neo4J, Sones, AllegroGraph, Core Data, DEX, FlockDB, InfoGrid, OpenLink Virtuoso,...

Tuy cùng mang những đặc điểm chung của NoSQL nhưng mỗi CSDL NoSQL cũng có những đặc điểm riêng, và vì thế thường được dùng cho những dự án khác nhau. Ví dụ:

MongoDB và Redis là những lựa chọn tốt cho việc lưu trữ các dữ liệu thống kê ít được đọc mà lại được viết thường xuyên.

Hadoop, một CSDL dạng tự do, phân tán làm tốt công việc lưu trữ các dữ liệu lớn như các con số thống kê thời tiết hoặc công việc phân tích nghiệp vụ.

Memcachedb, một CSDL nhất thời chóng tàn, tuyệt vời trong lưu trữ các phiên làm việc web, các khóa, và các con số thống kê ngắn hạn.

Cassandra và Riak (các lưu trữ dư thừa, tự động tạo bó cluster) làm tốt trong các môi trường với các ứng dụng có tính sẵn sàng cao, khi thời gian sống tối đa là sống còn.

Để tìm hiểu sâu hơn về các CSDL hiện đại NoSQL, chúng ta đi nghiên cứu chi tiết CSDL đặc trưng là **MongoDB.**

1. **ƯU ĐIỂM NOSQL**

- Nó là nguồn mở: Các sản phẩm nguồn mở đưa ra cho những người phát triển với một vài lợi ích lớn lao, trong đó có tình trạng không có chi phí của chúng. Những lợi ích khác: phần mềm nguồn mở có xu hướng sẽ là tin cậy hơn, an ninh hơn và nhanh hơn để triển khai so với các lựa chọn thay thế sở hữu độc quyền.

- Việc mở rộng phạm vi là mềm dẻo: NoSQL thay thế câu thần chú cũ của các nhà quản trị CSDL về 'mở rộng phạm vi' với một thứ mới: 'mở rộng ra ngoài'. Thay vì bổ sung thêm các máy chủ lớn hơn để điều khiển nhiều tải dữ liệu hơn, thì CSDL NoSQL cho phép một công ty phân tán tải qua nhiều máy chủ khi mà tải gia tăng.

- Các CSDL NoSQL khác nhau cho những dự án khác nhau:  MongoDB và Redis là những lựa chọn tốt cho việc lưu trữ các dữ liệu thống kê ít được đọc mà lại được viết thường xuyên, như một số đếm truy cập web chẳng hạn.

 Hadoop, một CSDL dạng tự do, phân tán làm tốt công việc lưu trữ các dữ liệu lớn như các con số thống kê thời tiết hoặc công việc phân tích nghiệp vụ.

 Memcache, một CSDL nhất thời chóng tàn, tuyệt vời trong lưu trữ các phiên làm việc web, các khóa, và các con số thống kê ngắn hạn.  Cassandra và Riak (các lưu trữ dư thừa, tự động tạo bó cluster) làm tốt trong các môi trường với các ứng dụng có tính sẵn sàng cao, khi thời gian sống tối đa là sống còn.

- NoSQL được các hãng lớn sử dụng: Các công ty như Amazon, BBC, Facebook và Google dựa vào các CSDL NoSQL.

- NoSQL phù hợp với công nghệ đám mây: Các máy chủ ngày nay là không đắt và có thể dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu của ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các các kết nối mạnh. Các CSDL NoSQL hầu hết sử dụng bộ nhớ qua đĩa như là vị trí ghi đầu tiên

- Vì thế ngăn ngừa được sự thực thi không ổn định của I/O. Và vì NoSQL lưu trữ dữ liệu thường thúc đẩy được tính mở rộng phạm vi theo chiều ngang thông qua việc ngăn chia, chúng có khả năng tận dụng được việc cung cấp mềm dẻo của đám mây.

**III.DOCUMENT STORE:**

- CSDL Document được thiết kế để quản lý và lưu trữ dữ liệu ở dạng document. Những document này được mã hóa về các dạng chuẩn như là XML, JSON (Javascript Option Notation) hay BSON (Binary JSON). Khác với các kiểu lưu trữ dạng Key-value, giá trị của cột trong các CSDL document chứa dữ liệu bán cấu trúc (Semi-Structured Data), đặc biệt là cặp thuộc tính name (key) – value. Một cột có thể chứa hàng trăm các thuộc tính như vậy, số lượng, loại thuộc tính được lưu lại có thể khác nhau giữa các dòng. Một điểm khác nữa so với các kiểu lưu trữ dữ liệu dạng Key-value đơn giản là cả key và value đều có thể tìm kiếm trong CSDL Document. Vd: Document store

-CSDL Document phù hợp cho việc lưu trữ và quản lý tập dữ liệu có kích thước lớn như là tài liệu văn bản, tin nhắn, cũng như biểu diễn một thực thể CSDL như là Product hay Customer (tài liệu khái niệm trong XML).

- Các CSDL tiêu biểu cho CSDL Document: CouchDB (JSON), MongoDB (BSON),… đều là mà nguồn mở (open source), hướng document (ducument oriented) và có lược đồ tự do (shema free).

**MongoDB**

API: BSON.

Protocol: C.

Query Method: dynamic object-based language & MapReduce,

Replication: Master Slave & Auto-Sharding.

Written in: C++.

Concurrency: Update in Place.

Misc:Indexing, GridFS,

1. **MONGODB**

**1.Giới thiệu và ưu điểm của MongoDB**

1. **Giới thiệu**

Trong những gương mặt góp phần làm suy tàn đế chế SQL thì MongoDB nổi lên là một CSDL đáng tin cậy và dễ dùng nhất. Mongo viết bằng C++. Nó thích hợp cho các ứng dụng tầm trung trở lên. Nếu tỉ lệ lượng dữ liệu ghi vào CSDL của ứng dụng lớn hơn lượng đọc thì đây càng là lựa chọn hợp lý.

[**MongoDB**](https://jobs.hybrid-technologies.vn/blog/mongodb-la-gi/)**là một cơ sở dữ liệu NoSQL hướng đối tượng, đơn giản, linh động và có thể mở rộng.**

Nó dựa trên mô hình lưu trữ NoSQL document. Các đối tượng dữ liệu được lưu trữ dưới dạng các tài liệu riêng biệt bên trong một collection – thay vì lưu trữ dữ liệu vào các cột và hàng của cơ sở dữ liệu quan hệ truyền thống.

Ngôn ngữ MongoDB là triển khai một kho lưu trữ dữ liệu cung cấp hiệu suất cao, tính sẵn sàng cao và tự động mở rộng

MongoDB sử dụng **JSON** hoặc **BSON** document để lưu trữ dữ liệu.

**-JSON** là một kiểu dữ liệu mở trong [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị". Về cấu trúc, nó mô tả một vật thể bằng cách bọc những vật thể con trong vật thể lớn hơn trong dấu ngoặc nhọn ({ }). JSON là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình

**-**[**BSON**](http://bsonspec.org/)**là mã hóa nhị phân của các tài liệu giống như JSON mà MongoDB sử dụng khi lưu trữ tài liệu trong các bộ sưu tập. Nó bổ sung hỗ trợ cho các loại dữ liệu như Ngày và nhị phân không được hỗ trợ trong JSON.**

Các bản phân phối chung cho MongoDB hỗ trợ Windows, Linux, Mac OS

Trước khi đi vào tìm hiểu kỹ hơn về MongoDB, chúng ta làm quen với một số khái niệm cơ bản của MongoDB:

-Database là một Ô chứa dữ liệu ở mức vật lý (physical), mỗi database sẽ có nhiều collection và được thiết lập lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDB thường có thể tạo nhiều cơ sở dữ liệu.

* Văn bản (Document) là đơn vị cơ bản của dữ liệu trong MongoDB, nó tương đương với một dòng trong CSDL quan hệ
* Bộ sưu tập (Collection) có thể được coi như tương đương với một bảng.
* MongoDB có thể lưu trữ nhiều CSDL độc lập, mỗi CSDL này có các bộ sưu tập và điều khoản riêng của mình
* MongoDB đi kèm với một trình tiện ích JavaScript đơn giản nhưng mạnh mẽ, nó hữu ích trong quản trị và thao tác dữ liệu.
* Mỗi văn bản có một khóa đặc biệt, đó là “\_id”, nó là duy nhất trong bộ sưu tập của văn bản.

**Văn bản(Document)**

Văn bản là một khái niệm quan trọng trong MongoDB. Văn bản bao gồm tập hợp các khóa với các giá trị tương ứng.

Ví dụ: *{"greeting" : "Hello, world!"}*

Văn bản trên gồm một khóa là “greeting”, với giá trị là “Hello, world!”. Các văn bản có thể chứa nhiều cặp khóa/giá trị.

Ví dụ: *{"greeting" : "Hello, world!", "foo" : 3}*

Một số lưu ý:

* Các cặp khóa/ giá trị trong văn bản được sắp xếp. Văn bản trên sẽ khác với văn bản sau

*{"foo" : 3, "greeting" : "Hello, world!"}*

* Khóa trong văn bản là một chuỗi
* MongoDB phân biệt chữ hoa chữ thường
* Văn bản trong MongoDB không được chứa những khóa giống nhau. Ví dụ văn bản sau là không hợp lệ

*{"greeting" : "Hello, world!", "greeting" : "Hello, MongoDB!"}*

**Bộ sưu tập(Collection)**

Bộ sưu tập là một nhóm các văn bản. Nếu văn bản tương đương với dòng trong CSDL quan hệ thì bộ sưu tập tương đương với bảng.

Bộ sưu tập là một Schema-Free, nghĩa là các văn bản có hình dạng khác nhau có thể cùng được lưu trữ trong 1 bộ sưu tập.

Ví dụ các văn bản sau có thể cùng được lưu trong một bộ sưu tập:

*{"greeting" : "Hello, world!"}*

*{"foo" : 5}*

Bộ sưu tập được xác định bởi tên của nó là một chuỗi UTF-8

Các đặc trưng của MongoDB:

* Lưu trữ hướng văn bản: Văn bản theo phong cách JSON với những lược đồ động đơn giản
* Hỗ trợ chỉ mục đầy đủ: chỉ mục trên bất kỳ các thuộc tính
* Tính sao lặp và tính sẵn sàng cao: mở rộng
* Auto-sharding: mở rộng theo chiều ngang mà không ảnh hưởng đến chức năng
* Truy vấn: đa dạng, truy vấn dựa trên văn bản
* Cập nhật nhanh:
* Map/Reduce
* GridFS: lưu trữ file với bất kỳ kích cỡ nào mà không làm phức tạp ngăn xếp
* Hỗ trợ thương mại: hỗ trợ doanh nghiệp, đào tào, tư vấn

****\_id****: Là trường bắt buộc có trong mỗi document. Trường \_id đại diện cho một giá trị duy nhất trong document MongoDB. Trường \_id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một \_id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.

****Collection****: Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.

****Cursor****: Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.

****Database****: Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một mấy chủ MongoDB có thể chứa nhiều Database.

****Field****:Làmộtcặpnamevalue trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.

****Index****: Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

1. **Ưu điểm của MongoDB**

Open Source:

* MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn

Hiệu năng cao:

* Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).
* Thử nghiệm cho thấy tốc độ insert, tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL ([So sánh hiệu suất của MongoDB với MySQL](https://quantrimang.com/so-sanh-hieu-suat-hoat-dong-cua-mongodb-va-sql-server-2008-84584))

Tại sao MongoDB có hiệu năng cao như thế? có các lý do sau:

* MongoDB lưu dữ liệu dạng JSON, khi bạn insert nhiều đối tượng thì nó sẽ là insert một mảng JSON gần như với trường hợp insert 1 đối tượng
* Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong RDBMS, khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan như trong RDBMS.
* Dữ liệu trong MongoDB được [đánh chỉ mục](https://stackjava.com/mongodb/danh-chi-muc-la-gi-danh-chi-muc-trong-mongodb.html) ([đánh index](https://stackjava.com/mongodb/danh-chi-muc-la-gi-danh-chi-muc-trong-mongodb.html)) nên khi truy vấn nó sẽ tìm rất nhanh.
* Khi thực hiện insert, find… MongoDB sẽ khóa các thao tác khác lại, ví dụ khi nó thực hiện find(), trong quá trình find mà có thêm thao tác insert, update thì nó sẽ dừng hết lại để chờ find() xong đã.

Dữ liệu linh hoạt:

* MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng field, kiểu dữ liệu… bạn có thể insert thoải mái dữ liệu mà mình muốn.

Là Rich Query Language:

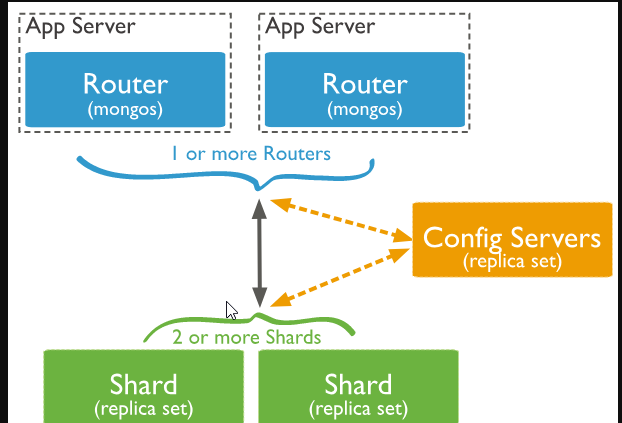
* MongoDB là một rich query language tức là nó có sẵn các method để thực hiện create, read, update, delete dữ liệu (CRUD)

Tính sẵn có:

* MongoDB hỗ trợ [replica set](https://docs.mongodb.com/manual/replication/) nhằm đảm bảo việc sao lưu và khôi phục dữ liệu

Khả năng mở rộng Horizontal Scalability:

* Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:



1. **So sánh MySQL và MongoDB**

|  |  |  |
| --- | --- | --- |
|  | **MySQL** | **MongoDB** |
| Viết bởi ngôn ngữ | C++, C | C++, C và JavaScript |
| Kiểu | RDBMS(Hệ thống quản lý cơ sở dữ liệu quan hệ | Hướng tài liệu |
| Các điểm chính | Table , Row ,Column | Collection , Document, Field |
| License | GPL v2 / Giấy phép thương mại có sẵn OD | OD GNU AGPL v3.0 / Giấy phép thương mại có sẵn OD |
| Lược đồ | Strict | Dynamic |
| Các tính năng chính | Tìm kiếm và đánh chỉ số full text ,Hỗ trợ nhân rộng tích hợp, Trigger, SubSELECT,Truy vấn bộ nhớ đệm,Hỗ trợ SSL,Hỗ trợ Unicode,Công cụ lưu trữ khác nhau với các đặc tính hiệu suất khác nhau | Auto-sharding,Native replication,Hỗ trợ mô hình dữ liệu nhúng,Chỉ số phụ toàn diện,Hỗ trợ ngôn ngữ truy vấn phong phú,Hỗ trợ công cụ lưu trữ khác nhau |
| Sử dụng tốt nhất cho | Cấu trúc dữ liệu phù hợp với bảng và hàng,Sự phụ thuộc mạnh mẽ vào,các giao dịch nhiều hàng,Cập nhật thường xuyên và sửa đổi khối lượng lớn bản ghi,Bộ dữ liệu tương đối nhỏ | Tải ghi cao,Lược đồ không ổn định,DB của bạn được thiết lập để phát triển lớn,Dữ liệu dựa trên vị trí,HA (tính sẵn sàng cao) trong môi trường không ổn định là bắt buộc,Không có quản trị viên cơ sở dữ liệu (DBA) |
| Scaling | Theo chiều dọc | Theo chiều ngang |

# **2.Cài đặt MongoDB**

Trên Windows:

1. Tải MongoDB từ link bên dưới:

<https://www.mongodb.com/download-center?jmp=nav#enterprise>

1. Chạy file vừa tải về

3.Sau khi chạy xong file, vào thư mục C:\ProgramFiles\MongoDB\Server\3.4\bin , chạy file mongod để khởi động mongoDB

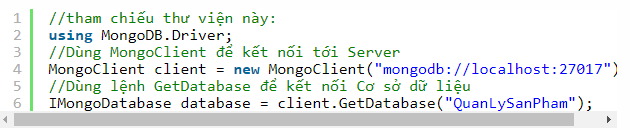
4.Dùng địa chỉ 127.0.0.1:27017 để tạo connection đến mongoDB

Chi tiết tại: <https://techtalk.vn/blog/posts/bai-1-gioi-thieu-mongodb-va-cach-cai-dat>

**3. Kết nối và truy vấn CSDL MongoDB**

Vì bản Demo của nhóm tụi em làm bằng ngôn ngữ C# nên sau đây tụi em sẽ giới thiệu về kết kết nối dữ liệu MongoDB bằng C#

-Để kết nối tới MongoDB ta dùng lệnh sau:



**mongodb://localhost:27017**

Là cú pháp kết nối lên Server. Chúng ta chỉ đổi localhost thành tên Server(địa chỉ IP) và Port. Còn mọi thứ phải giữ nguyên

Lệnh: client.**GetDatabase**(“**QuanLySanPham**“); Dùng để kết nối tới Cơ sở dữ liệu tên là **QuanLySanPham**, do đó bạn muốn kết nối tới Cơ sở dữ liệu khác thì đổi tên chỗ này là xong.

* Truy vấn csdl:
  + Hàm tạo database và record

MongoServer mongo = MongoServer.Create();

mongo.Connect();

var db = mongo.GetDatabase("Organization");

using (mongo.RequestStart(db))

{

var collection = db.GetCollection<BsonDocument>("Employees");

BsonDocument Employee = new BsonDocument()

.Add("\_id", System.Guid.NewGuid().ToString())

.Add("Name", txtName.Text)

.Add("Eamil", txtEmail.Text)

.Add("Phone", txtPhone.Text);

collection.Insert(Employee);

textBox1.Text = textBox1.Text + System.Environment.NewLine + "Inserted....";

mongo.Disconnect();

ReadAll();

}

* Tìm kiếm một record

MongoServer mongo = MongoServer.Create();

mongo.Connect();

textBox1.Text = textBox1.Text + System.Environment.NewLine + "Searching....";

var db = mongo.GetDatabase("Organization");

using (mongo.RequestStart(db))

{

var collection = db.GetCollection<BsonDocument>("Employees");

var query = new QueryDocument("Name", txtNameSearch.Text);

foreach (BsonDocument item in collection.Find(query))

{

BsonElement Name = item.GetElement("Name");

BsonElement Eamil = item.GetElement("Email");

BsonElement Phone = item.GetElement("Phone");

BsonElement id = item.GetElement("\_id");

txtNameSearch.Text = Name.Value.ToString();

txtEamilSearch.Text = Eamil.Value.ToString();

txtPhoneSearch.Text = Phone.Value.ToString();

txtID.Text = id.Value.ToString();

}

}

mongo.Disconnect();

* Đọc tất cả các Record

MongoServer mongo = MongoServer.Create();

mongo.Connect();

textBox1.Text = textBox1.Text + System.Environment.NewLine + "Reading All....";

var db = mongo.GetDatabase("Organization");

using (mongo.RequestStart(db))

{

var collection = db.GetCollection<BsonDocument>("Employees");

foreach (BsonDocument item in collection.FindAll())

{

BsonElement Name = item.GetElement("Name");

BsonElement Eamil = item.GetElement("Eamil");

textBox1.Text= textBox1.Text+System.Environment.NewLine+string.Format("Name: {0}, email: {1}", Name.Value.ToString(), Eamil.Value.ToString());

}

}

mongo.Disconnect();

* Cập nhật record

MongoServer mongo = MongoServer.Create();

mongo.Connect();

var db = mongo.GetDatabase("Organization");

using (mongo.RequestStart(db))

{

var collection = db.GetCollection<BsonDocument>("Employees");

BsonDocument Employee = new BsonDocument()

.Add("\_id", txtID.Text)

.Add("Name", txtNameSearch.Text)

.Add("Eamil", txtEamilSearch.Text)

.Add("Phone", txtPhoneSearch.Text);

collection.Save(Employee);

textBox1.Text = textBox1.Text + System.Environment.NewLine + "Update....";

mongo.Disconnect();

}

* Xóa record trong mongo

MongoServer mongo = MongoServer.Create();

mongo.Connect();

var db = mongo.GetDatabase("Organization");

using (mongo.RequestStart(db))

{

var collection = db.GetCollection<BsonDocument>("Employees");

collection.Remove(new QueryDocument("\_id", txtID.Text));

textBox1.Text = textBox1.Text + System.Environment.NewLine + "deleted....";

mongo.Disconnect();

}

Tài liệu tham khảo

1. <https://docs.mongodb.com/manual/tutorial/>
2. <https://en.wikipedia.org/wiki/MongoDB>
3. <https://techtalk.vn/>