

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM  
KHOA HỆ THỐNG THÔNG TIN VÀ VIỆN THÁM



BÁO CÁO ĐỒ ÁN  
MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU  
ĐỀ TÀI: SO SÁNH MONGODB VÀ MYSQL

Sinh viên thực hiện:    Phạm Minh Tuấn                    0950080115  
   Nguyễn Mạnh Hùng Vĩ                    0950080117  
Lớp:    09\_ĐHCNTT3

Thành phố Hồ Chí Minh, tháng 11, năm 2022

## Mục lục

Lời mở đầu.....	1
I. GIỚI THIỆU .....	2
II. PHÁT TRIỂN ỨNG DỤNG SỬ DỤNG MONGODB VS. MYSQL.....	4
III. NGHIÊN CỨU SO SÁNH.....	8
IV. SO SÁNH HIỆU NĂNG.....	11
a.  Thao tác chèn (insert):.....	11
b.  Thao tác truy vấn .....	13
c.  Thao tác cập nhật (update) .....	15
d.  Thao tác xóa (Delete) .....	18
V. GIỚI THIỆU DEMO .....	21
5.1 Tổng quan về demo .....	21
5.2 Công cụ sử dụng .....	21
5.3 Xây dựng các bảng:.....	21
5.4 Mô hình quan hệ cơ sở dữ liệu:.....	21
5.5 Biểu đồ UseCase .....	22
5.4 Mô tả chức năng .....	23
5.4.1 Staff (người dùng bình thường) .....	23
5.4.2 Admin (quản trị viên) .....	23
V. KẾT LUẬN .....	25
Tài liệu tham khảo .....	26

## **Lời mở đầu**

Trong những năm gần đây mọi thứ đều được cơ sở dữ liệu hoá đã có những bước phát triển vượt bậc trong nước cũng như trên thế giới và được ứng dụng rộng rãi trong các lĩnh vực của cuộc sống. Đặc biệt là trong ngành kinh tế nó góp phần đáng kể cho sự phát triển kinh tế của nước ta trong quá trình hội nhập.

Việc ứng dụng cơ sở dữ liệu đã mang lại bước đột phá mới cho công tác quản lý bán hàng, giúp doanh nghiệp nắm bắt thông tin về hàng hóa, vật tư, thông tin khách hàng, và trạng thái các đơn đặt hàng... một cách chính xác và nhanh chóng. Từ đó người quản lý có thể đưa ra các kế hoạch và quyết định đúng đắn nhằm nâng cao kết quả hoạt động.

Trong bài báo này, chúng tôi sẽ cố gắng trình bày một cách so sánh nghiên cứu cơ sở dữ liệu phi quan hệ và cơ sở dữ liệu quan hệ. Chúng tôi chủ yếu tập trung trình bày vào một triển khai của công nghệ cơ sở dữ liệu SQL, cụ thể là MySQL, và tạo ra một so sánh với việc triển khai cơ sở dữ liệu quan hệ khác, cụ thể là MongoDB, và do đó giải thích tại sao MongoDB hiệu quả hơn MySQL. Chúng tôi cũng sẽ trình bày những ưu điểm của sử dụng cơ sở dữ liệu phi quan hệ so với cơ sở dữ liệu quan hệ. Cơ sở dữ liệu SQL được sử dụng để phát triển điển đàn là MySQL

## I. GIỚI THIỆU

Nếu một vài năm trước, một ứng dụng thường chỉ được sử dụng để có hàng nghìn người dùng đến hàng chục nghìn người dùng trong trường hợp đặc biệt, hiện tại có những ứng dụng có hàng triệu người dùng và những người được kết nối 24/7, 365 ngày mỗi năm. Điều quan trọng là sử dụng một cơ sở dữ liệu thích hợp, mà hỗ trợ kết nối đồng thời hàng trăm ngàn người dùng.

Cơ sở dữ liệu quan hệ được sử dụng rộng rãi trong hầu hết các ứng dụng và chúng có hiệu suất tốt khi chúng xử lý một lượng dữ liệu hạn chế. Để xử lý một khối lượng lớn dữ liệu như internet, đa phương tiện và phương tiện truyền thông xã hội việc sử dụng cơ sở dữ liệu quan hệ truyền thống là không hiệu quả. Để khắc phục điều này vấn đề thuật ngữ “NO SQL” đã được giới thiệu. Thuật ngữ NoSQL được đặt ra bởi Carlo Strozzi vào năm 1998 và đề cập đến việc cơ sở dữ liệu phi quan hệ, thuật ngữ này sau đó được giới thiệu lại trong 2009 bởi Eric Evans. Gần đây, thuật ngữ này đã nhận được nghĩa khác, cụ thể là "Not Only SQL", nhẹ hơn biến thể của việc xác định thuật ngữ, so với ý nghĩa trước đây của nó, phản quan hệ .

NoSQL, không phải là một công cụ, mà là một phương pháp bao gồm một số công cụ bổ sung và cạnh tranh . Ưu điểm chính của cơ sở dữ liệu NoSQL là, không giống như cơ sở dữ liệu quan hệ, nó có thể xử lý dữ liệu phi cấu trúc như tài liệu, e-mail, đa phương tiện và mạng xã hội một cách hiệu quả. Cơ sở dữ liệu phi quan hệ không sử dụng các nguyên tắc RDBMS (Hệ thống quản lý cơ sở dữ liệu quan hệ) và không lưu trữ dữ liệu trong các bảng, lược đồ không cố định và có mô hình dữ liệu rất đơn giản. Thay vào đó, chúng sử dụng các khóa nhận dạng và dữ liệu có thể được tìm thấy dựa trên các khóa được chỉ định.

Có bốn chiến lược để lưu trữ dữ liệu trong cơ sở dữ liệu phi quan hệ, như được trình bày trong và chúng như sau:

1. Key-Value (Khóa-Giá trị) - Key-Value databases (Khóa-Giá trị cơ sở dữ liệu) chẳng hạn như Riak, là các từ điển phân tán theo khái niệm và không có lược đồ xác định trước; chúng là lược đồ ít hơn. Khóa có thể là khóa tổng hợp hoặc tự tạo và giá trị có thể là bất kỳ thứ gì: chuỗi, JSON, BLOB, v.v.

2. Document (Tài liệu) - Couchbase và MongoDB [2] là những cơ sở dữ liệu dựa trên tài liệu phổ biến nhất. Chúng linh hoạt trong loại nội dung vì chúng không có lược

đều được xác định trước. Về mặt khái niệm, chúng làm việc với các loại tài liệu khác nhau: JSON, BSON, XML và BLOB [2, 8]. Về cơ bản, chúng chỉ đại diện cho một chuyên môn hóa của key-value databases (khóa-giá trị cơ sở dữ liệu). Một tài liệu được viết/đọc bằng một phím. Bên cạnh chức năng Key-Value (Khóa-Giá trị), cơ sở dữ liệu dựa trên tài liệu bổ sung các chức năng khác nhau để tìm tài liệu dựa trên nội dung của chúng.

3. Column – Databases (Cột – Cơ sở dữ liệu) từ danh mục BigTable, chẳng hạn như HBase và Hypertable là loại cột và phải có lược đồ được xác định trước [2, 4]. Dữ liệu được lưu trữ trong các ô được nhóm trong các cột và các cột được nhóm hợp lý thành các họ cột. Về mặt lý thuyết, chúng có thể chứa một số lượng không giới hạn (giới hạn tùy thuộc vào việc triển khai) các cột có thể được tạo khi chạy hoặc khi định nghĩa lược đồ.

4. Graph-Oriented (Định hướng theo đồ thị) – Chiến lược này có thể hỗ trợ các truy vấn dữ liệu phức tạp cũng được thực hiện trong một khoảng thời gian tương đối nhỏ hơn so với các cơ sở dữ liệu khác sử dụng các chiến lược được đề cập ở trên.

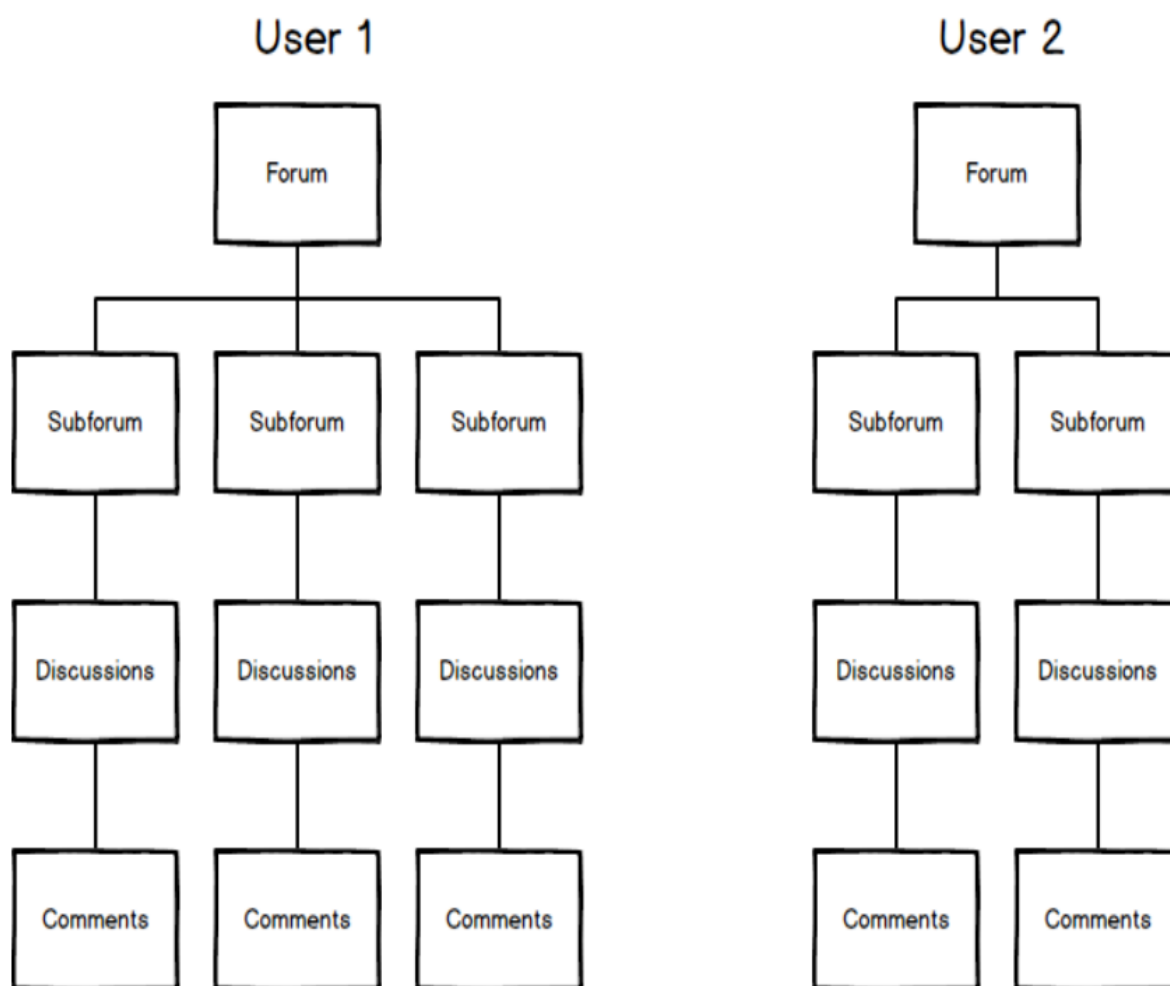
Ngoài ra, cơ sở dữ liệu phi quan hệ cung cấp tính linh hoạt cao khi thêm hoặc xóa thuộc tính khỏi cơ sở dữ liệu vì chúng không có lược đồ cơ sở dữ liệu cố định. Tùy thuộc vào yêu cầu của ứng dụng, chúng ta có thể sử dụng các loại cơ sở dữ liệu NoSQL khác nhau và mỗi cơ sở dữ liệu NoSQL có các tính năng, mô hình dữ liệu và lựa chọn kiến trúc của cơ sở dữ liệu tùy thuộc vào ứng dụng.

Trong bài báo này, chúng tôi tập trung vào một trong những công nghệ NoSQL, cụ thể là MongoDB, và so sánh với MySQL để làm nổi bật lý do tại sao MongoDB hiệu quả hơn MySQL. Ngoài ra, chúng tôi sẽ trình bày những ưu điểm của việc sử dụng cơ sở dữ liệu phi quan hệ trong ứng dụng diễn đàn, sử dụng MongoDB làm cơ sở dữ liệu NoSQL.

## II. PHÁT TRIỂN ỨNG DỤNG SỬ DỤNG MONGODB VS. MYSQL

Chúng tôi đã thực hiện một nghiên cứu so sánh giữa MongoDB và MySQL dựa trên các khái niệm và lệnh của chúng được sử dụng cho các hoạt động khác nhau.

Nghiên cứu so sánh này dựa trên sự phát triển của một diễn đàn có cấu trúc động, tùy thuộc vào sở thích của người dùng. Sử dụng một cơ sở dữ liệu quan hệ như MySQL [9], cấu trúc sẽ là tĩnh và mỗi người dùng sẽ phải tuân theo một cấu trúc được thiết lập hoàn toàn trong cơ sở dữ liệu (ví dụ: diễn đàn - diễn đàn con - thảo luận - nhận xét). Do đó, mỗi người dùng buộc phải tuân theo cấu trúc, được minh họa trong Hình 2-1.



Hình 2-1: Cấu trúc cộng đồng sử dụng MySQL.

Trong Hình 2-1, mỗi người dùng có số lượng diễn đàn con khác nhau trong một diễn đàn, nhưng phương pháp thực hiện mà mỗi người dùng phải đáp ứng là giống nhau: đầu tiên, họ cần tạo một diễn đàn, sau đó là các diễn đàn con, tiếp theo là tạo của các cuộc thảo luận và cuối cùng là bằng cách bổ sung các ý kiến trong cuộc thảo luận.

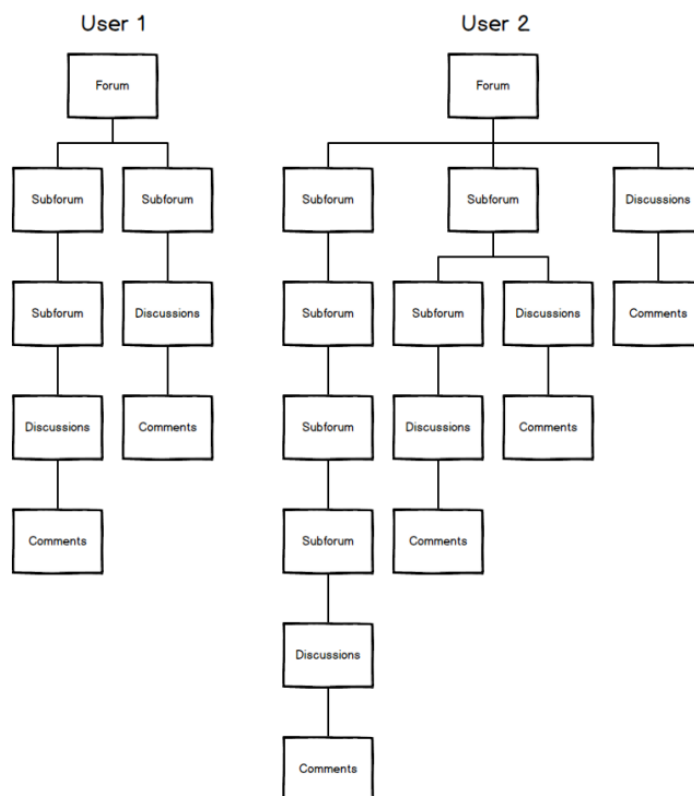
Trong ứng dụng này, cơ sở dữ liệu phi quan hệ, MongoDB, đã được chọn – cơ sở dữ

liệu dựa trên các tài liệu JSON có lược đồ động, được viết bằng C++. MongoDB ban đầu được ra mắt vào năm 2009, hiện đang được phát triển và đang mở rộng. MongoDB là một cơ sở dữ liệu có thể được sử dụng cả trong các dự án nhỏ có vài nghìn người dùng, nhưng chủ yếu dành cho các sản phẩm và ứng dụng chứa hàng trăm nghìn người dùng, bao gồm Craigslist, eBay, Foursquare hoặc New York Times.

Cơ sở dữ liệu MongoDB chứa một tập hợp các bộ sưu tập. Một bộ sưu tập không có lược đồ được xác định trước như bảng và lưu trữ dữ liệu dưới dạng tài liệu BSON (đối tượng giống như JSON được mã hóa nhị phân) .

Một tài liệu là một tập hợp các trường và có thể được coi là một hàng trong một bộ sưu tập. Nó có thể chứa các cấu trúc phức tạp như danh sách hoặc thậm chí toàn bộ tài liệu. Mỗi tài liệu có một trường ID, được sử dụng làm khóa chính và mỗi bộ sưu tập có thể chứa bất kỳ loại tài liệu nào, nhưng các truy vấn và chỉ mục chỉ có thể được áp dụng trên các bộ sưu tập.

Sử dụng cơ sở dữ liệu phi quan hệ, chẳng hạn như MongoDB, để phát triển một diễn đàn, cho phép cấu trúc dành riêng cho từng người dùng, do đó, mỗi người dùng có khả năng tổ chức diễn đàn của mình theo một cách riêng, như trong Hình 2-2.



Hình 2-2: Cấu trúc cộng đồng sử dụng MongoDB.

Hình 2-2 cho thấy rằng việc sử dụng cơ sở dữ liệu phi quan hệ, cấu trúc tĩnh trong đó một diễn đàn chứa các diễn đàn con, các diễn đàn con chứa các cuộc thảo luận và các cuộc thảo luận chứa các bình luận, không bắt buộc phải đáp ứng. Như vậy, một diễn đàn con có thể chứa diễn đàn con khác, diễn đàn này có thể chứa diễn đàn con khác, diễn đàn này chứa diễn đàn con khác, v.v. Ngoài ra còn có khả năng đính kèm các cuộc thảo luận trực tiếp vào một diễn đàn mà không cần phải tạo một diễn đàn con để mở một cuộc thảo luận mới.

Ứng dụng được phát triển bằng Symfony2, một khung công tác PHP mã nguồn mở cho phép dễ dàng tích hợp cơ sở dữ liệu MongoDB. Việc tích hợp cơ sở dữ liệu MongoDB trong khung được thực hiện thông qua một gói, một gói phụ trợ phải được đăng ký trong danh sách các gói được sử dụng trong ứng dụng.

Sau khi gói đã được tích hợp vào Symfony2, cấu hình cơ sở dữ liệu đã được thực hiện (lựa chọn máy chủ - trong trường hợp này là máy chủ cục bộ và lựa chọn tên cơ



sở dữ liệu - chọn cơ sở dữ liệu không tồn tại không dẫn đến lỗi, nhưng việc tạo của cơ sở dữ liệu), như sau:

```
# app/config/config.yml
```

```
doctrine_mongodb:
```

```
connections:
```

```
default:
```

```
server: mongodb://localhost:27017
```

```
options: { }
```

```
default_database: my_database
```

```
document_managers:
```

```
default:
```

```
auto_mapping: true
```

### III. NGHIÊN CỨU SO SÁNH

Để hiểu thêm về cách sử dụng MongoDB, Bảng 3.1 cho thấy sự so sánh giữa các thuật ngữ được sử dụng trong MySQL và MongoDB tương ứng. Như được hiển thị trong Bảng 3.1, trong MongoDB, một số thuật ngữ MySQL, chẳng hạn như bảng hoặc hàng, có tên khác, cụ thể là bộ sưu tập, tài liệu BSON tương ứng. Nói cách khác, chúng ta có thể nói rằng MongoDB chứa các bộ sưu tập, các bộ sưu tập chứa các tài liệu và một tài liệu chứa nhiều trường.

Không giống như MySQL, nơi cơ sở dữ liệu được trình bày dưới dạng đồ họa dưới dạng bảng, trong MongoDB, cơ sở dữ liệu có cấu trúc đồ họa sau:

```
{
  "_id": "d4acaf3a76e4378b853eb15fde216722",
  "username": "andra",
  "email": "andra@gmail.com",
} {
  "_id": "d4rvgf3a76e4378b853tr15fde216722",
  "username": "ioana",
  "email": "ioana@gmail.com",
}
```

Ví dụ trên hiển thị cơ sở dữ liệu cho người dùng, mỗi người dùng có ID duy nhất và được tạo tự động, Tên người dùng và một địa chỉ email.

Ứng dụng sẽ có 3 lớp người dùng, cụ thể là **administrators** (quản trị viên), **MODERATORS**(người điều hành) và **regular users**(người sử dụng thường xuyên). Mỗi người dùng có quyền tạo một diễn đàn/diễn đàn con riêng tư và có toàn quyền quản trị viên trên diễn đàn/diễn đàn con đó. Ở trong một diễn đàn con, người điều hành có quyền sửa/xóa diễn đàn con và họ cũng có thể kiểm duyệt các cuộc thảo luận khác của người dùng, trong khi người dùng thường xuyên chỉ được phép đăng các cuộc thảo luận và để lại bình luận. Nếu một cơ sở dữ liệu quan hệ đã được sử dụng, Cột cho các diễn đàn và diễn đàn con nên xuất hiện người dùng diễn đàn, mặc dù người dùng bình thường sẽ không bao giờ có quyền tạo, sửa hoặc xóa chúng, trừ khi chúng là quản trị viên của diễn đàn cụ thể đó. Sử dụng MongoDB, Những lĩnh vực này

liên quan đến diễn đàn và diễn đàn con sẽ chỉ xuất hiện cho người dùng có quyền đó (người điều hành và quản trị viên), do đó làm giảm đáng kể không gian lưu trữ, sẽ cao hơn nhiều bằng cách sử dụng MySQL.

Trên thực tế, những vai trò này không được gán cho người dùng, mà cho chính họ. Điều này là do người dùng có thể không bao giờ là người quản trị trên diễn đàn, mà chỉ người sử dụng thường xuyên và không có điểm nào để gán quản trị viên cho người dùng đó (sẽ luôn không có hiệu lực), trong khi diễn đàn sẽ luôn có trường này, bởi vì quản trị viên của nó thực tế là tác giả của nó.

Như trong cơ sở dữ liệu quan hệ, MongoDB cũng có mối quan hệ one-to-many (một-nhiều), nhưng trong trường hợp này, khái niệm khóa ngoại không được sử dụng; thay vào đó, khái niệm chú thích được sử dụng. Do đó, trong trường hợp này, liên quan đến một diễn đàn, kết nối giữa diễn đàn và các diễn đàn con của nó như sau: trong tài liệu diễn đàn, diễn đàn con được tham chiếu bằng cách sử dụng chú thích `@MongoDB\EmbedMany (targetDocument = "Subforum")`, và trong tài liệu Subforum, diễn đàn được tham chiếu bởi chú thích sau đây đó là kết nối hai tài liệu: `@MongoDB\ReferenceOne (targetDocument = "Forum", inversedBy="Subforums")`.

```
class Forum
```

```
{
```

```
/**
```

```
* @MongoDB\EmbedMany(targetDocument="Subforum")
```

```
*/
```

```
public $subforums = array();
```

```
}
```

```
Class Subforum
```

```
{
```

```
/**
```

```
* @MongoDB\ReferenceOne(targetDocument="Forum",
```

```
inversedBy="subforums")
```

```
*/
```

```
protected $forum;
```

```
}
```

Để tạo ra một cấu trúc linh hoạt cho diễn đàn, kết nối thực tế giữa diễn đàn và các tiêu mục của nó, và cũng giữa diễn đàn - thảo luận, diễn đàn con – diễn đàn con, diễn đàn con- Các cuộc thảo luận được thực hiện bằng ID của họ là duy nhất và cho phép một liên kết động giữa các yếu tố của diễn đàn. Vì vậy, một cuộc thảo luận có thể dễ dàng gắn vào cả hai vào một diễn đàn và với một diễn đàn con, nhờ kết nối động giữa các bộ sưu tập. Ngoài ra, nếu trong tương lai, nó mong muốn chèn một trường mới trong cơ sở dữ liệu, điều này có thể dễ dàng đạt được bằng cách chèn tên và loại trong tài liệu tương ứng mà không ảnh hưởng đến dữ liệu thực tế được lưu trữ trong cơ sở dữ liệu và không cần xác định lại toàn bộ cấu trúc cơ sở dữ liệu. Trong chương tiếp theo, chúng tôi sẽ trình bày sự khác biệt giữa hai cơ sở dữ liệu, MongoDB và MySQL, về mặt hiệu suất, sau khi thực hiện các hoạt động khác nhau trên cả hai cơ sở dữ liệu.

## IV. SO SÁNH HIỆU NĂNG

Để làm nổi bật những ưu điểm của việc sử dụng cơ sở dữ liệu phi quan hệ MongoDB so với cơ sở dữ liệu quan hệ MySQL, các thao tác khác nhau đã được thực hiện trên hai cơ sở dữ liệu. Các thao tác này là bốn thao tác cơ bản có thể được thực hiện trên bất kỳ cơ sở dữ liệu nào, cụ thể là:

1. Insert (Chèn)
2. Select (query) (Chọn (truy vấn))
3. Update (Cập nhật)
4. Delete (Xóa).

Do kết quả kiểm tra phụ thuộc vào máy tính thực hiện các kiểm tra này, điều quan trọng cần lưu ý là tất cả các kết quả được trình bày bên dưới đều thu được từ các nghiên cứu được thực hiện trên máy tính có các đặc điểm sau: Windows 7 Ultimate 64-bit, bộ xử lý Intel Core i3 (2,4 GHz), bộ nhớ RAM 4 GB.

### a. Thao tác chèn (insert):

Chúng tôi bắt đầu thử nghiệm với việc tạo cơ sở dữ liệu mà không có bất kỳ nội dung, cả trong MongoDB và MySQL. Cấu trúc của hai cơ sở dữ liệu tương tự nhau và chúng có cùng một số của các cột/trường và bảng/tài liệu. Các phần tử chung của hai cơ sở dữ liệu là:

- Bảng/tài liệu người dùng tương ứng với các cột: id, username, password, email.
- Bảng/tài liệu diễn đàn tương ứng với các cột: id, tiêu đề, tác giả, thông tin (mô tả ngắn).
- Bảng/tài liệu diễn đàn con tương ứng với các cột: id, tiêu đề, tác giả, thông tin, tạo, cập nhật.
- Bảng/tài liệu bài viết tương ứng với các cột: id, tiêu đề, tác giả, tạo, cập nhật, nội dung.
- Bảng/tài liệu bình luận tương ứng với các cột: id, tác giả, tạo, nội dung, approved

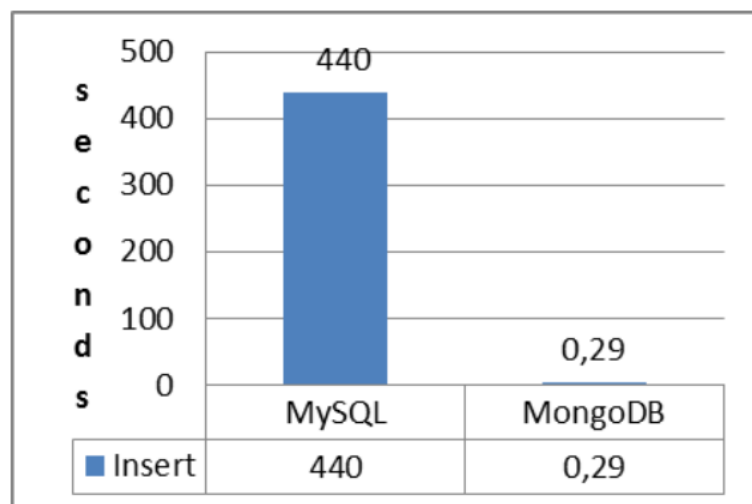
Ngoài các lĩnh vực phổ biến này, còn tồn tại các lĩnh vực các trường liên kết các bảng khác trong MySQL và các tài liệu trong MongoDB, và chúng thường là những khóa ngoại để xác định

Ví dụ: một diễn đàn con thuộc về diễn đàn cha mẹ của nó hoặc tác giả của một bình luận nhất định.

Chèn dữ liệu bắt đầu trong cả hai cơ sở dữ liệu với người dùng chèn. Đối với mỗi cơ sở dữ liệu, 10.000 người dùng đã được chèn. ID của họ là được tạo tự động bởi cả cơ sở dữ liệu và cho tên người dùng, mật khẩu và địa chỉ email. Chúng tôi đã sử dụng PHP các chức năng như MD5, RAND, Subr và Str\_Shuffle.

Để ghi thời gian cần thiết để chèn các yếu tố trong cơ sở dữ liệu, chúng tôi đã sử dụng microtime chức năng PHP, trong đó ghi lại thời gian từ khi bắt đầu thời gian chạy tập lệnh và cho đến khi nó hoàn thành.

Do đó, sau khi chạy hai tập lệnh, một cho MySQL và một cho Mongo, Hình 4.1 cho thấy 10.000 người dùng đã được chèn vào MySQL trong 440 giây, trong khi ở MongoDB, thời gian chỉ 0,29 giây.



Hình 4.1: Chèn người dùng của MySQL và MongoDB.

Khi người dùng đã được đưa vào cơ sở dữ liệu, việc chèn dữ liệu được chuyển cho các diễn đàn, diễn đàn con, thảo luận và bình luận. Tất cả ID của tất cả các bảng đã được tạo tự động trong cả MySQL và MongoDB, và tạo các trường khác mà chúng tôi đã sử dụng các chức năng được mô tả ở trên. Vì một diễn đàn con không thể tồn tại nếu diễn đàn hoặc một diễn đàn khác diễn đàn con không được tạo đầu tiên và vì không có bình luận nào có thể tồn tại riêng của nó vì nó phải được sở hữu bởi một cuộc thảo luận, chúng tôi bắt đầu chèn các diễn đàn, sau đó là chèn các cuộc thảo luận, và cuối cùng các ý kiến.

Để thử nghiệm, nó đã được quyết định chèn 5000 hàng cho mỗi bảng một phần, như sau: 5000 Diễn đàn, 5000 diễn đàn con, tương ứng 5000 thảo luận và 5000 bình luận.

Do đó, sau khi chèn dữ liệu vào hai cơ sở dữ liệu, Thời gian chèn kết quả:

MySQL: 1010 giây, MongoDB 3.3331 giây.

Chúng tôi nhận thấy rằng MongoDB dành ít thời gian hơn MySQL, để chèn một lượng lớn thông tin trong cơ sở dữ liệu.

### **b. Thao tác truy vấn**

Để kiểm tra hiệu suất của các hoạt động truy vấn, chúng tôi đã thực hiện hai Chọn (truy vấn), cụ thể là:

- Lựa chọn tất cả các cuộc thảo luận mà người dùng tham dự và với một ngày khác với một cái nhất định;
- Lựa chọn tất cả người dùng từ cơ sở dữ liệu và thảo luận bắt đầu bởi mỗi người dùng.

Select đầu tiên MySQL có cú pháp sau (Select 1-MySQL):

```
mysql_connect('localhost','root','');
mysql_select_db('routinie');
$query = "SELECT d.username, d.dtitle, d.dcontent,
d.created, s.id, f.id
FROM discussion d
inner join subforum s on s.id = d.subforumid
inner join forum f on f.id = s.forumid
where d.username = `andra`
and d.created <> '2014-11-27'";
$query_run = mysql_query($query);
```

và cú pháp MongoDB sau (Select 1- MongoDB):

```
$m = new MongoClient();
$db = $m->selectDB('routinie'); $collection = new MongoClient($db,
'categories');
$date = '1417091683';
```

```

$q = array(
'subcategories.topics.discussions.dauthor' =>
'andra',
'subcategories.topics.discussions.created' =>
array( '$ne' => new MongoInt32($date))
);
$cursor = $collection->find($q);

```

Select thứ hai MySQL có cú pháp sau đây (Select 2- MySQL):

```

mysql_connect('localhost','root','');
mysql_select_db('routinie');
$query = "select u.id, u.username, count(d.dtitle)
from users u
left outer join discussion d on d.userid
= u.id
group by u.id, u.username";
$query_run = mysql_query($query);

```

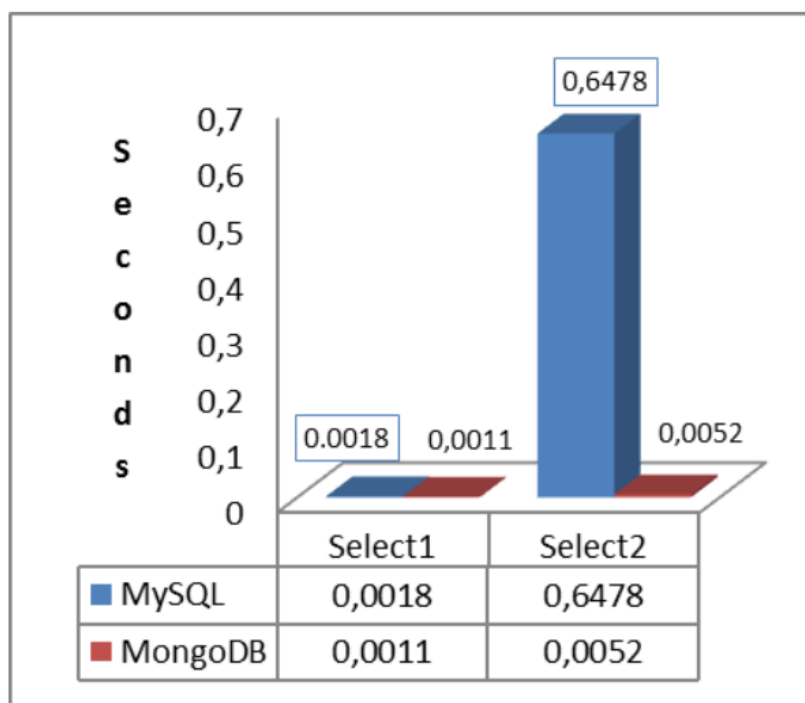
và cú pháp MongoDB có dạng như sau (Select 2- MongoDB):

```

$m = new MongoClient();
$db = $m->selectDB('routinie');
$collection = new MongoCollection($db,
'categories');
$user =
'subcategories.topics.discussions.dauthor';
$find_users = $collection->distinct($user);
foreach($find_users as $find_user) {
$discussions = $collection-
>count(array('subcategories.topics.discussions.dau
thor' => $find_user ));

```





Hình 4-1: thời gian thao tác select MySQL vs. MongoDB

Select đầu tiên được thực hiện trong MySQL trong 0,0018 giây, trong khi ở MongoDB, nó đã được thực hiện trong 0,0011 giây và select thứ hai được thực hiện trong 0,6478 giây trong MySQL và 0,0052 giây trong MongoDB. Từ biểu đồ, chúng tôi nhận thấy rằng MongoDB dành ít thời gian hơn MySQL, để thực hiện các hoạt động truy vấn như trong Hình 4-2.

### c. Thao tác cập nhật (update)

Để kiểm tra hoạt động cập nhật trong cơ sở dữ liệu, Sau hai truy vấn đã được thực thi:

- Cập nhật bình luận được viết bởi một người dùng cụ thể (Update 1)
- Cập nhật địa chỉ email của người dùng, (Update 2).

Updating đầu tiên có cú pháp MySQL có dạng như sau (Update 1- mysql):

```
mysql_connect('localhost','root','');
mysql_select_db('routinie');
$query = "update `comments`
set content = 'This is the new content.'
WHERE id = 10594
and username = `andra`";
$query_run = mysql_query($query);
```

và cú pháp MongoDB sau (Update 1- MongoDB):

```
$m = new MongoClient();  
$db = $m->selectDB('routinie');  
$collection = new MongoCollection($db,  
'categories');  
$criteria = array(  
'subcategories.topics.discussions.comments.cauthor  
' => 'andra',  
'subcategories.topics.discussions.comments.$id'  
=> new  
MongoId('5c936263f3428a40227908d5a3847c0b');  
);  
$collection->update(  
$criteria,  
array('subcategories.topics.discussions.comments.c  
content' => "new content")  
);
```

Bản cập nhật thứ hai có cú pháp MySQL có dạng như sau đây (Update 2- MySQL):

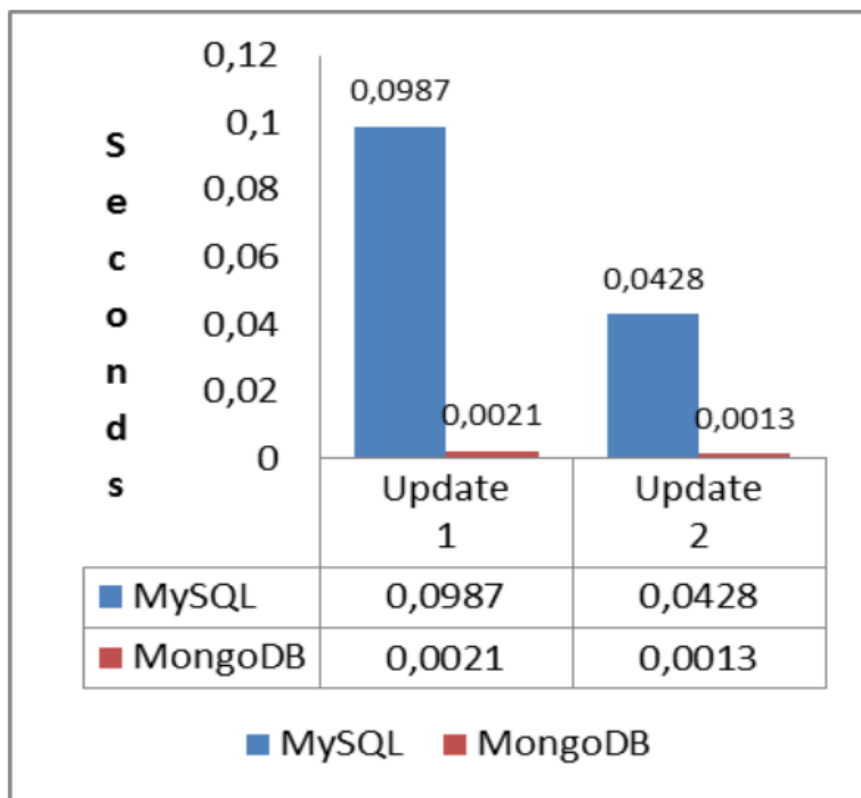
```
mysql_connect('localhost','root','');  
mysql_select_db('routinie');  
$query = "update `users`  
set email = 'andra.olah@gmail.com'  
WHERE id = 1012";  
$query_run = mysql_query($query);
```

và cú pháp MongoDB có dạng như sau:

```

$m = new MongoClient();
$db = $m->selectDB('routinie');
$collection = new MongoClient($db,
'categories');
$criteria = array(
'users.$id' => new
MongoId('92cc227532d17e56e07902b254dfad10');
);
$collection->update($criteria, array('users.email' =>
"andra.olah@gmail.com")
);

```



Hình 4-3: thời gian update của MySQL và MongoDB

Bản update đầu tiên được thực hiện trong MySQL trong 0,0987 giây, trong khi ở MongoDB, nó đã được thực hiện trong 0,0021 giây và cập nhật thứ hai được thực hiện trong 0,0428 giây trong MySQL và 0,0013 giây trong MongoDB. Từ Hình 4-3, chúng tôi nhận thấy MongoDB đã dành ít thời gian hơn MySQL, để thực hiện cập nhật hoạt động như trong Hình 4-3.

#### **d. Thao tác xóa (Delete)**

Như các thao khác được mô tả ở trên, chúng tôi đã thực hiện hai truy vấn xóa cho mỗi cơ sở dữ liệu.

Truy vấn đầu tiên đã xóa tất cả các nhận xét được đăng bởi người dùng (Delete 1). Truy vấn thứ hai đã xóa tất cả các diễn đàn được tạo bởi một người dùng cụ thể (Delete 2). Điều quan trọng là phải đề cập rằng với việc xóa một diễn đàn, tất cả các tiêu thư, nhận xét và các cuộc thảo luận có trong diễn đàn được chỉ định cũng là tự động xóa, trong cả MySQL và MongoDB.

Delete đầu tiên có cú pháp MySQL như sau đây (Delete 1-MySQL):

```
mysql_connect('localhost','root','');  
mysql_select_db('routinie');  
$query = "delete from comments where username =  
`andra`";  
$query_run = mysql_query($query);
```

và cú pháp MongoDB có dạng như sau (Delete 1- MongoDB):

```
$m = new MongoClient();  
$db = $m->selectDB('routinie');  
$collection = new MongoCollection($db,  
'categories');  
$criteria =  
array('subcategories.topics.discussions.comments.c  
author' => 'andra');  
$collection->update(  
$criteria,  
array('$unset' =>  
array('subcategories.topics.discussions.comments'  
=> true)),  
array('multiple' => true)
```

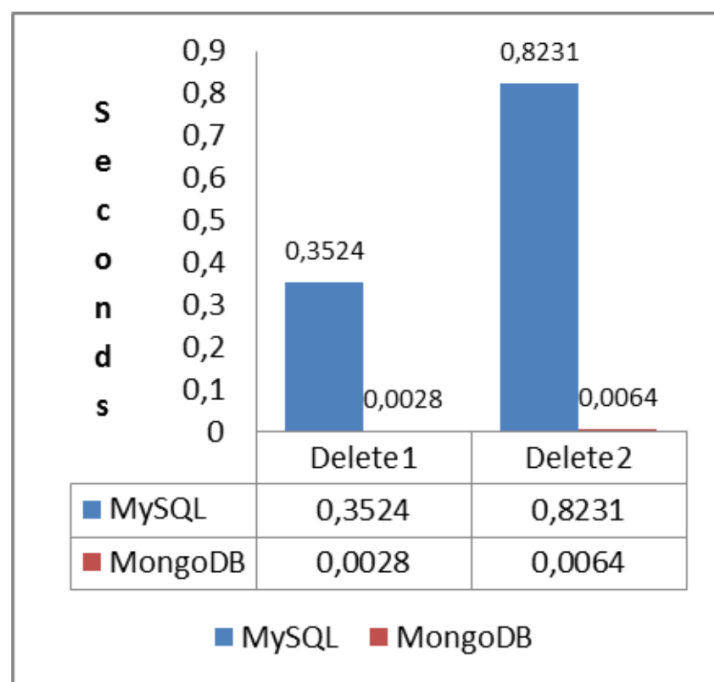
);

và cú pháp MongoDB có dạng như sau (Delete 2- MySQL):

```
mysql_connect('localhost','root','');  
mysql_select_db('routinie');  
$query = "delete FROM `forum` WHERE username  
=`andra`";  
$query_run = mysql_query($query);
```

và cú pháp MongoDB sau (Delete 2- MongoDB):

```
$m = new MongoClient();  
$db = $m->selectDB('routinie');  
$collection = new MongoCollection($db,  
'categories');  
$q = array('admins.username' => 'andra');  
$cursor = $collection->remove($q);
```



Hình4-4: Thời gian delete của MySQL và MongoDB

Từ hình 4-4, chúng tôi nhận thấy rằng delete đầu tiên là được thực hiện trong mysql trong 0,3524 giây, trong khi ở MongoDB nó được thực hiện trong 0,0028 giây và xóa thứ hai là được thực hiện trong 0,8231 giây trong MySQL và 0,0064 giây trong MongoDB.

MongoDB cung cấp thời gian thực hiện thấp hơn mysql trong tất cả bốn hoạt động cơ bản, điều này rất cần thiết khi một ứng dụng sẽ cung cấp hỗ trợ cho hàng ngàn người dùng đồng thời. Do đó, so sánh ở trên, chứng minh rằng cho một lượng lớn dữ liệu MongoDB có hiệu suất tốt và nó được ưa thích hơn MySQL.

## V. GIỚI THIỆU DEMO

### 5.1 Tổng quan về demo

Xây dựng một web forum đáp ứng các nhu cầu như đăng bài viết, bình luận và trả lời bình luận đối với người dùng. Và quản lý người dùng, subforum đối với quản trị viên. Xây dựng cơ sở dữ liệu bằng MySQL trên phpMyAdmin để dễ dàng quản lý và cập nhật dữ liệu một cách nhanh chóng.

### 5.2 Công cụ sử dụng

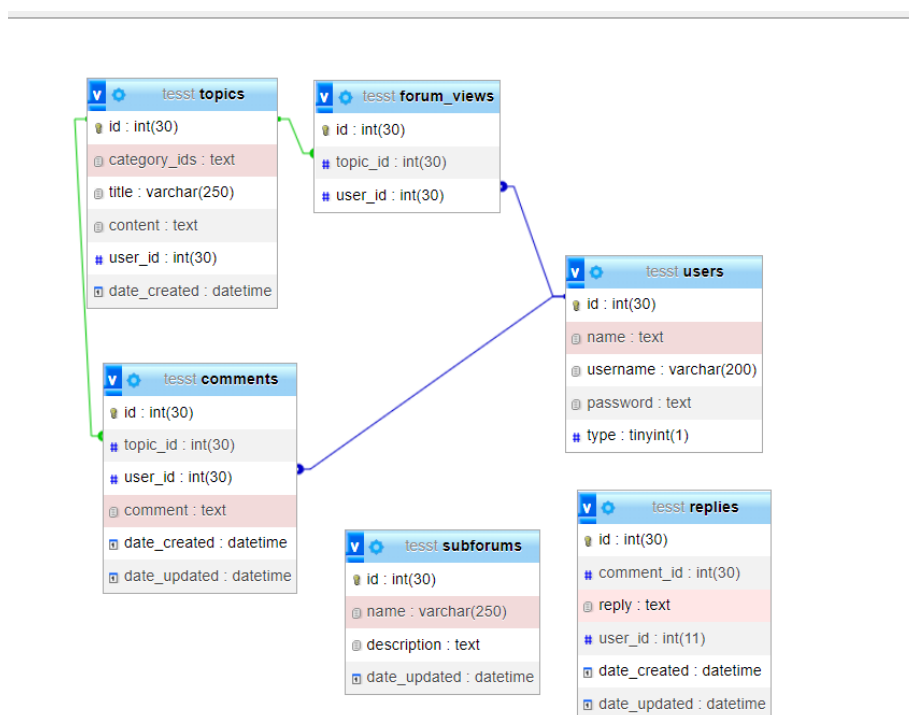
+) Visual Studio Code

+) Xampp

### 5.3 Xây dựng các bảng:

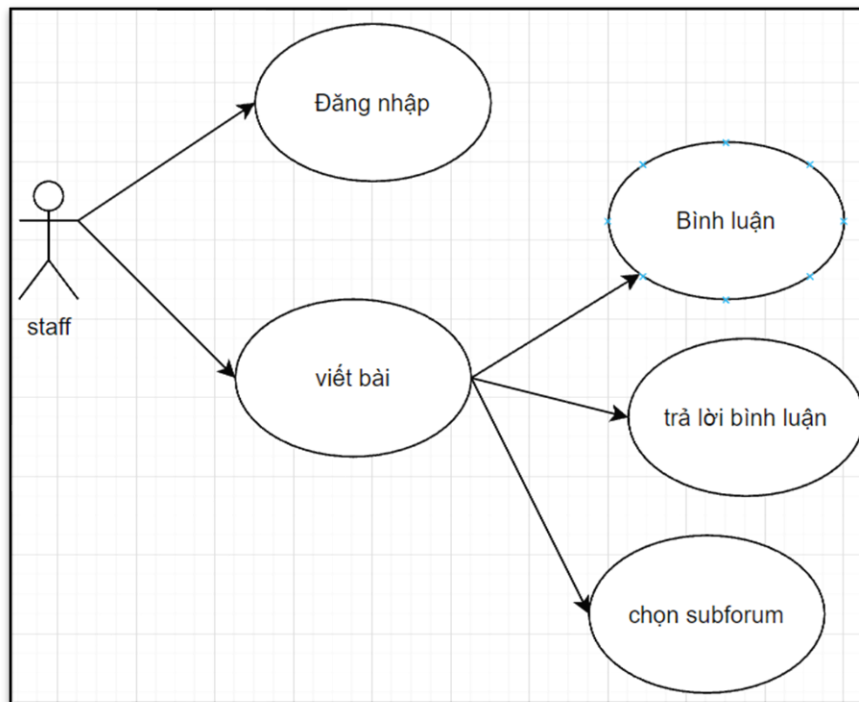
1. Comments (id, topic\_id, user\_id, comment, date\_created, date\_updated)
2. forum\_view (id, topic\_id, user\_id)
3. replies (id, comment\_id, reply, user\_id, date\_created, date\_updated)
4. subforums (id, name, description, date\_updated)
5. topics (id, category\_ids, title, content, user\_id, date\_created)
6. users (id, name, username, password, type)

### 5.4 Mô hình quan hệ cơ sở dữ liệu:

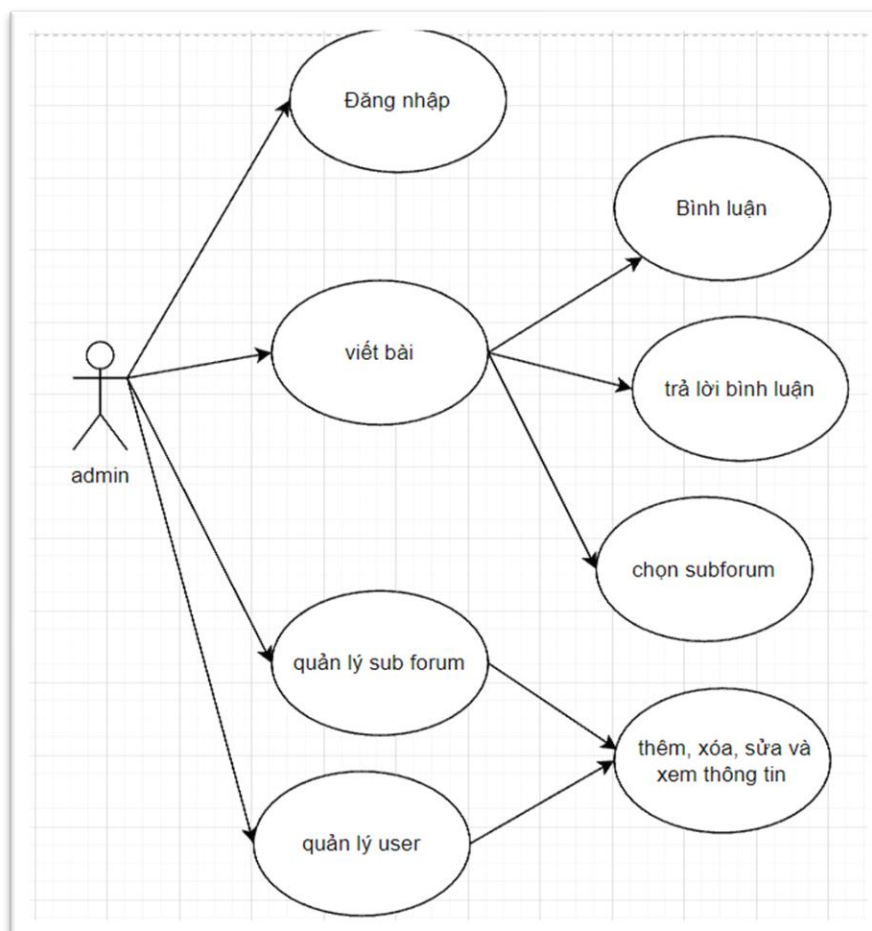


## 5.5 Biểu đồ UseCase

Các chức năng mà staff (người dùng bình thường) được phép thực hiện trên web



Các chức năng mà admin (quản trị viên) được phép thực hiện trên web





## 5.4 Mô tả chức năng

### 5.4.1 Staff (người dùng bình thường)

Chức năng đăng nhập và đăng xuất

Mô tả:

- Khi người dùng muốn sử dụng web forum nếu đã có tài khoản thì người dùng sẽ nhập tài khoản mật khẩu để đăng nhập, còn không người dùng sẽ được cấp tài khoản thông qua quản trị viên và khi tài khoản người dùng được thêm vào thành công người dùng sẽ đăng nhập và sử dụng web forum.
- Sau khi người dùng không còn muốn sử dụng web forum người dùng có thể ấn nút đăng xuất và trở về màn hình đăng nhập.

Chức năng home (trang chủ)

Mô tả:

- Khi vào trang chủ người dùng có thể thấy số lượng người dùng, bài viết và subforum và hiển thị danh sách các subforum.

Chức năng topic (bài viết)

Mô tả

- Khi người dùng đăng nhập có thể chọn vào phần topic (bài viết) để đăng bài viết và lựa chọn subforum mà bản thân muốn đăng.
- Xem danh sách bài viết.
- Bình luận và trả lời các bình luận khác của bài viết.

### 5.4.2 Admin (quản trị viên)

Chức năng đăng nhập, đăng xuất

Mô tả:

- Quản trị viên là người duy nhất có thể đăng nhập mà không cần đăng ký
- Sau khi quản trị viên không còn muốn sử dụng web forum quản trị viên có thể ấn nút đăng xuất và trở về màn hình đăng nhập.

Chức năng home (trang chủ)

Mô tả:

- Khi vào trang chủ quản trị viên có thể thấy số lượng người dùng, bài viết và subforum và hiển thị danh sách các subforum.

Chức năng topic (bài viết)

Mô tả:

- Có thể thêm, sửa, xoá bài viết
- Xem danh sách bài viết.
- Khi quản trị viên đăng nhập có thể chọn vào phần topic (bài viết) để đăng bài viết và lựa chọn subforum mà quản trị viên muốn đăng.
- Bình luận và trả lời các bình luận khác của bài viết.

Chức năng user (người dùng)

Mô tả:

- Có thể thêm, sửa, xoá người dùng
- Xem danh sách người dùng
- Tìm người dùng

Chức năng subforum

Mô tả:

- Có thể thêm, sửa, xoá subforum
- Xem danh sách subforum
- Tìm subforum

## V. KẾT LUẬN

Chuyển đổi từ cơ sở dữ liệu quan hệ sang cơ sở dữ liệu phi quan hệ có thể là một thách thức về nhiều mặt, trong đó cần nghiên cứu kỹ tất cả các loại cơ sở dữ liệu phi quan hệ có thể có và tìm ra giải pháp tối ưu cho ứng dụng cụ thể và tạo ra một cơ sở dữ liệu phi quan hệ có thể cung cấp chính xác các tính năng và hoạt động truy vấn giống như cơ sở dữ liệu sẽ thay thế. Nhưng việc sử dụng cơ sở dữ liệu quan hệ sẽ không bao giờ kết thúc, bởi vì nó cung cấp cho chúng ta một bộ tính năng vô song, bằng cách duy trì tính toàn vẹn và khả năng mở rộng của dữ liệu. Về cơ bản, công việc của nhà phát triển là quyết định cơ sở dữ liệu nào sẽ được sử dụng để đáp ứng các yêu cầu của ứng dụng.

Đối với demo forum thì cơ sở dữ liệu phi quan hệ phù hợp nhất là MongoDB, bởi vì diễn đàn sẽ có hàng nghìn người dùng, thậm chí hàng chục nghìn và MongoDB đã tùy chỉnh để phù hợp với từng người dùng bằng cách tạo các diễn đàn riêng, mỗi diễn đàn có sở hữu cấu trúc linh hoạt và năng động. Nhưng chúng tôi lại chọn MySQL để ứng dụng vào demo vì nó có tính chặt chẽ, trực quan hơn và có tính bảo mật tốt

Ưu điểm của việc sử dụng MongoDB được làm nổi bật hơn nữa bằng cách tiến hành các thử nghiệm và diễn giải kết quả của chúng, điều này đã được trình bày trong chương trước. MongoDB cung cấp thời gian thực thi thấp hơn MySQL trong cả bốn thao tác cơ bản, điều này rất cần thiết khi một ứng dụng cần cung cấp hỗ trợ cho hàng nghìn người dùng cùng một lúc.

Chúng ta có thể chọn MongoDB thay vì MySQL nếu ứng dụng cần nhiều dữ liệu và lưu trữ nhiều dữ liệu và truy vấn nhiều dữ liệu. Cũng cần lưu ý rằng trong cùng một ứng dụng, đôi khi mỗi người dùng có thể cần các cài đặt tùy chỉnh của riêng mình và cơ sở dữ liệu quan hệ không cho phép tùy chỉnh hoàn toàn chỉ dựa trên nhu cầu của người dùng. Do đó, ngày càng có nhiều ứng dụng bắt đầu sử dụng cơ sở dữ liệu phi quan hệ vì chúng cung cấp một cấu trúc linh hoạt hơn có thể định hình theo nhu cầu của mỗi người dùng; chúng được thiết kế để lưu trữ lượng lớn dữ liệu và chúng là cơ sở dữ liệu không chuẩn hóa, giúp tăng hiệu suất.

## **Tài liệu tham khảo**

Sách:

- [1] K. Sanobar, M. Vanita, “SQL Support over MongoDB using Metadata”, International Journal of Scientific and Research Publications, Volume 3, Issue 10, October 2013
- [2] T. Frătean, Bazele de date NoSQL – o analiză comparativă, To Day Software Magazine, number 10 [Online]. Available:  
<http://www.todaysoftmag.ro/article/304/bazele-de-date-nosql-o-analizacomparativa>, acceseed oct. 2014.
- [3] S. Hoberman, “Data Modeling for MongoDB”, Publisher by Technics Publications, LLC 2 Lindsley Road Basking Ridge, NJ 07920, USA, ISBN 978-1-935504-70-2, 2014.
- [4] S. Hall (2014, Jul) MySQL vs MongoDB, jul, 2014, Available:  
<http://www.scriptrock.com/articles/mysql-vs-mongodb>, acceseed nov. 2014
- [5] F. Potencier, “The symfony 1.3 & 1.4 Reference Guide”, Publisher: Sensio SA (November 4, 2009), ISBN: 978-2918390145, pp 278.
- [6] MongoDB, Inc. (2015, Aprilie), MongoDB Ops Manager Manual Release 1.6, [Online]. Available:  
<https://docs.opsmanager.mongodb.com/current/opsmanager-manual.pdf>
- [7] R. P Padhy, M. R. Patra, S. C. Satapathy, “RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database’s”, International Journal of Advance Engineering Sciences and Technologies, Vol. 11, Issue No. 1, 015-030, 2011.
- [8] J. Clarence, M. Tauro, S. Aravindh, A. B. Shreeharsha, “Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Database”, International Journal of Computer Applications, ISSN 0975 – 888, Volume 48– No.20, June 2012.
- [9] M. Irfan (March 14, 2014) Tools and tips for analysis of MySQL’s Slow Query Log Oracle Corporation. [Online]. Available:

<http://www.percona.com/blog/2014/03/14/tools-and-tips-for-analysis-of-mysqls-slow-query-log/>

- [10] Z. Wei-Ping, LI Ming-Xin, H. Chen, “Using MongoDB to Implement Textbook Management System instead of MySQL”, IEEE 3rd International Conference on Communication Software and Networks

source:

<https://code-projects.org/student-enrollment-in-php-with-source-code/>

<https://codeshack.io/crud-application-php-pdo-mysql/>

[https://www.w3resource.com/php/function-reference/mysqli\\_affected\\_rows.php](https://www.w3resource.com/php/function-reference/mysqli_affected_rows.php)

Link:

<https://www.mongodb.com/docs/manual/reference/method/js-collection/>

<https://www.youtube.com/watch?v=3xHrMwy-Y5A&t=163s>

<https://www.youtube.com/watch?v=SMZaOPVRAIc>