**DANH SÁCH THÀNH VIÊN**

Võ Tấn Đạt\_ 20116031

Lê Hữu Hiệp\_ 20115701

Huỳnh Hồ Nhựt Hào\_20067811

Nguyễn Minh Đức\_20077961

Điểu Phan Quang Dũng\_20093921

# Bài 1: Sách và nhà xuất bản (Books and publishers)

Chúng tôi muốn thiết kế một CSDL MongoDB để lưu trữ dữ liệu về sách (Books) và các nhà xuất bản (Publishers).

* Mỗi nhà xuất bản được đặc trưng bởi tên (name), năm thành lập (year) và quốc gia (country). Một dánh sách các phương tiện để liên hệ. Các phương tiện liên hệ có thể là email, điện thoại, website. Ứng với mỗi một phương tiện liên hệ chỉ lưu một thông tin liên hệ duy nhất. Các phương tiện liên hệ có thể được thêm vào trong tương lai.
* Mỗi cuốn sách được xác định bằng mã ISBN và được đặc trưng bởi tiêu đề sách (title), tiêu đề phụ (subtitle), số trang (number of page), ngôn ngữ (languge) và ngày xuất bản (publication date). Mỗi cuốn sách được xuất bản bởi duy nhất một nhà xuất bản duy nhất. Một nhà xuất bản xuất bản nhiều sách khác nhau.
* Thiết kế lược đồ lưu trữ phải được tối ưu hóa để hiển thị cho mỗi cuốn sách với tất cả thông tin của nó và tên nhà xuất bản.

Yêu cầu: Thiết kế lược đồ lưu trữ dữ liệu để đáp ứng các yêu câu trên (chỉ ra các collection trong CSDL, cấu trúc Document, mối quan hệ, các chiến lược (design pattern) được sử dụng để thiết kế và nêu lý do tại sao dùng chiến lược đó).

Các collection: Publishers, Books

Mối quan hệ

Diagram

Description automatically generated

Thiết kế sơ đồ theo reference patern

Diagram

Description automatically generated with medium confidence

Ta sử dụng extend reference patter vì mục tiêu chính của thiết kế chính là để truy vấn tra cứu thông tin của sách với tất cả thông tin của sách và tên nhà xuất bản. Ta nhúng trường name của nhà xuất bản vào collection Book để tiện cho việc truy vấn đó theo mục đính của đề bài

# Bài 2: Blog Website

Chúng tôi muốn thiết kế CSDL MongoDB để quản lý blog.

* Mỗi người dùng (user) đã đăng ký blog được xác định bằng một tên người dùng (username). Chúng tôi cũng muốn ghi nhận lại phương thức liên hệ của người dùng, chẳng hạn như email, điện thoại, tài khoản Twitter, v.v. Các phương thức liên hệ được ghi nhận lại có thể thay đổi theo thời gian. Một phương thức liên hệ có thể có nhiều hơn một thông tin liên hệ. Mỗi phương thức liên hệ có thể có một trạng thái, ví dụ: "hoạt động", "không hoạt động", "đã xác nhận", v.v.
* Các bài đăng trên blog được đặc trưng bởi tiêu đề (title), mô tả (description) và tác giả (tức là username). Các bài viết nhận được bình luận (comment) từ các người dùng. Mỗi bình luận được mô tả bởi nội dung (text), ai đã bình luận (user) và bình luận lúc nào (timestamp)?
* Ứng dụng sẽ chỉ hiển thị cho mỗi bài đăng 10 bình luận gần đây nhất. Hồ sơ người dùng cũng phải bao gồm tổng số bài đăng và tổng số bài mà người dùng đã bình luận.

Yêu cầu: Thiết kế lược đồ lưu trữ dữ liệu để đáp ứng các yêu câu trên (chỉ ra các collection trong CSDL, cấu trúc Document, mối quan hệ, các chiến lược (design pattern) được sử dụng để thiết kế và nêu lý do tại sao dùng chiến lược đó).

|  |  |
| --- | --- |
| **User**  {\_id:String,  user\_ver:int,  Contacts:[{“K”: “email”,”V”:string, “S”:string},  {“K”: “điện thoại”,”V”:string, “S”:string }, {“K”: “Twitter”,”V”:string, “S”:string },  …  ],  blog\_n: int //số bài đăng  commnet\_n: int //số bài commnet  } | Sử dụng **Attribute Pattern** ở trường “Contacts” vì có nhiều phương thức liên hệ |
| **Blog**  {\_id:String,  title:String,  description: String,  user\_id:String (tác giả)  comment:[0:10]  [{commnet\_id:String,  user\_id: String,  text: String,  timestamp: String,  }]  } | Sử dụng **Computed Pattern** để lưu trữ số lượng bài đăng  Sử dụng **Subset Pattern** để lưu trữ 10 comment gần nhất |
| **Comment**  {commnet\_id:String,  user\_id: String,  text: String,  timestamp: String,  } | Sử dụng **Computed Pattern** để lưu trữ số lượng comment của người dùng |

# Bài 3: Breadcrumbs

Chúng tôi muốn thiết kế CSDL để quản lý sơ đồ trang web.

* Mỗi trang (page) của webisite được đặc trưng bởi đường dẫn tương đối (path), tiêu đề (title), nội dung (body), thời gian tạo (creation timestamp), thời gian cập nhật cuối cùng (last update timestamp) và danh sách các thẻ tag (list of tags).
* Các trang được sắp xếp theo hình cây. Trang chủ (home page) là gốc. Tất cả các trang khác nằm trong các đường dẫn phụ từ gốc.
* Ứng dụng phải xây dựng mã breadcrumb cho mỗi trang. Breadcrumb là một tập hợp các liên kết giúp cho người dùng xác định được vị trí hiện tại của mình trên cấu trúc website, là một danh sách có thứ tự các liên kết chứa tất cả các tổ tiên (ancestors) của trang hiện tại quay trở lại trang chủ, tức là thư mục gốc.

Yêu cầu: Thiết kế lược đồ lưu trữ dữ liệu để đáp ứng các yêu câu trên (chỉ ra các collection trong CSDL, cấu trúc Document, mối quan hệ, các chiến lược (design pattern) được sử dụng để thiết kế và nêu lý do tại sao dùng chiến lược đó).

- Các collections: Page,tags, Breadcrumb.

- Document:

|  |
| --- |
| Tags |
| -\_idTags:<objectID>;  -name:string;  -\_idPage:<objectID>; |
|  |

|  |
| --- |
| Page |
| -\_idPage:<objectID>;  -Path:string;  -title:string;  -body:string;  - creation\_timestamp:ISODATE;  - lastupdate\_timestamp:ISODATE;  -list\_tags[1,2,..]:{  -\_idTags:<objectID>;  }  -\_idBreadcrumb:<objectID>; |
|  |

|  |
| --- |
| Breadcrumb |
| -\_idBreadcrumb:<objectID>;  -location:string;  - ancestors[1,10,00]:{  -path:string;  }  -\_idPage:<objectID>; |
|  |

* Mối quan hệ:

+ Page – tags: Mối quan hệ n-n.

+ Page- Breadcrumb: Mối quan hệ 1-n.

* Design pattern: Extended pattern.
* Lý do chọn: tránh bị trùng lập dữ liệu, khi được cập nhật dữ liệu xác định những gì cần thay đổi, giúp người dùng truy vấn dễ dàng và nhanh chóng hơn bởi theo kiểu reference, dễ dàng trong việc quản lí các thông tin liên quan.

|  |
| --- |
| HomePage |
| -\_idPage:<objectID>;  -Path:string;  -title:string;  -body:string;  - creation\_timestamp:ISODATE;  - lastupdate\_timestamp:ISODATE;  -list\_tags[1,2,..]:{  -\_idTags:<objectID>;  }  -\_idBreadcrumb:<objectID>; |
|  |

|  |
| --- |
| Tags |
| -\_idTags:<objectID>;  -name:string;  -\_idPage:<objectID>; |
|  |



|  |
| --- |
| Breadcrumb |
| -\_idBreadcrumb:<objectID>;  -location:string;  - ancestors[1,10,00]:{  -path:string;  }  -\_idPage:<objectID>; |
|  |

Bài 4 – chuyển phát bưu kiện (Parcel delivery)



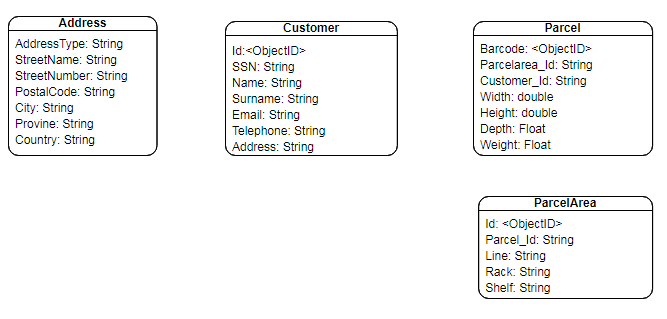
Thiết kế cơ sở dữ liệu MongoDB để quản lý việc chuyển phát bưu kiện theo các yêu cầu sau.

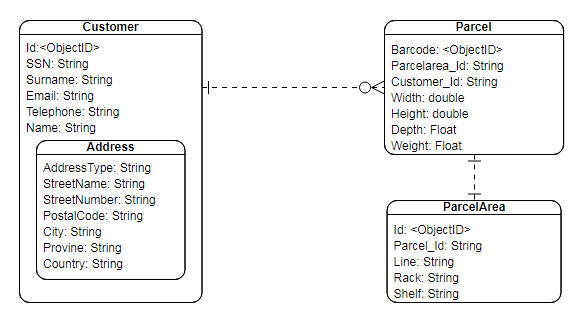
* Khách hàng (customers) của dịch vụ chuyển phát bưu kiện là công dân được xác định bằng số an sinh xã hội (social security number) của họ. Họ có thể là người gửi hoặc người nhận bưu kiện đã chuyển. Họ có đặc trưng bởi tên (name), họ (surname), địa chỉ thư điện tử (email), số điện thoại (telephone number) và các địa chỉ (address) khác nhau, mỗi địa thuộc một loại địa chỉ khác nhau như một địa chỉ thanh toán (billing address), địa chỉ nhà (home address), địa chỉ công sở (work address), …. Mỗi địa chỉ bao gồm tên đường (street name, số nhà (street number), mã bưu chính bưu (postal code), thành phố (city), tỉnh (province), và quốc gia (country).
* Bưu kiện được đặc trưng bởi một mã vạch (barcode) duy nhất và kích thước vật lý (physical dimensions) của chúng như chiều rộng (width), chiều cao (height), độ sâu (depth) và trọng lượng (weight). Chiều rộng, chiều cao và chiều sâu luôn được tính bằng mét. Trọng lượng là luôn luôn được tính bằng kilôgam.
* Thông tin người nhận và người gửi được yêu cầu để chuyển từng bưu kiện phải luôn có sẵn khi truy cập dữ liệu của một bưu kiện. Thông tin người nhận và người gửi cần thiết để chuyển một bưu kiện bao gồm họ, tên đường, số nhà, mã bưu chính, thành phố, tỉnh, quốc gia.
* Kho bưu kiện được chia thành nhiều khu vực (area) khác nhau. Mỗi khu vực được xác định bởi một mã, ví dụ: 'area\_51' và bao gồm các dòng (line) khác nhau. Mỗi dòng được xác định bởi duy nhất mã, ví dụ: 'line\_12' và tạo từ một giá đỡ (rack). Mỗi giá đỡ được xác định bằng mã duy nhất, ví dụ: 'rack\_33' và được tạo thành từ các kệ (shelf). Mỗi bưu kiện được đặt trên một kệ cụ thể của kho hàng, được xác định bằng một mã duy nhất, ví dụ: 'shelf\_99'. Cơ sở dữ liệu được yêu cầu để theo dõi vị trí của từng bưu kiện trong kho.
* Với một bưu kiện cho trước, cơ sở dữ liệu phải được thiết kế để cung cấp một cách hiệu quả đầy đủ vị trí của nó, từ kệ, đến khu vực, thông qua giá để và dòng.
* Với một khách hàng, cơ sở dữ liệu phải được thiết kế để cung cấp hiệu quả tất cả bưu kiện với tư cách là người gửi và tất cả các bưu kiện của người đó với tư cách là người nhận.

Yêu cầu: Thiết kế lược đồ lưu trữ dữ liệu để đáp ứng các yêu câu trên (chỉ ra các collection trong CSDL, cấu trúc Document, mối quan hệ, các chiến lược (design pattern) được sử dụng để thiết kế và nêu lý do tại sao dùng chiến lược đó).

(1) Các collection trong CSDL: Address, Customer, Parcel, PacelArea

(2) Cấu trúc Document:



(3) Các mối liên hệ:

(4) Áp dụng pattern:

Diagram

Description automatically generated

- Sử dụng Extended Reference Pattern có thể tạo ra nhiều trường lưu dữ liệu nhưng khi truy vấn sẽ dễ dàng hơn, nhanh hơn, giảm số nối các collection và tra cứu.

- Khi chúng ta mở collection Parcel, ta có thể biết được các thuộc tính của bưu kiện và thông tin của người gửi, nhận mà không cần phải kết nối tới collection Customer, tránh việc mất thời gian và dễ dàng quản lí các thông tin chuyển phát bưu kiện.

# Bài 5: Museum exhibitions

Chúng tôi muốn thiết kế cơ sở dữ liệu để quản lý triển lãm bảo tàng theo các yêu cầu.

* Các bảo tàng (museums) được đặc trưng bởi tên (name), địa chỉ (address), số điện thoại (phone) và trang web (website) nếu có. Địa chỉ bao gồm tọa độ địa lý (coordinates), tên đường (street), số nhà (nmuber), mã bưu điện (postal code) và thành phố (city).
* Các vật phẩm (items) được trưng bày trong bảo tàng được xác định bằng một số lũy tiến và được đặc trưng bởi một tiêu đề (title), mô tả (description) và danh sách tên tác giả (authors). Các vật phẩm được phân loại là phát hiện khảo cổ học, hoặc tranh, hoặc tác phẩm điêu khắc.
* Cơ sở dữ liệu phải ghi lại tất cả các đặc tính chính của vật phẩm, chẳng hạn như kích thước của nó (tùy vào loại vật phẩm, ví dụ chiều rộng, chiều cao, trọng lượng, v.v.). Mỗi đối tượng địa lý có ít nhất một tên và một giá trị, và có thể là một đơn vị đo lường. Ví dụ, vật liệu chính là đặc điểm của một phát hiện khảo cổ học, kích thước hình học là đặc điểm của một bức tranh. Đối với mỗi vật phẩm, bảo tàng mà nó thuộc về phải được ghi lại, với tên bảo tàng thường xuyên được truy cập cùng với đặt điểm chính đó.
* Một số cuộc triển lãm được tổ chức trong mỗi bảo tàng. Triển lãm được đặc trưng bởi một tiêu đề (title), một mô tả (description), danh sách tên người phụ trách (curators). Bạn phải ghi lại tất cả các vật phẩm liên quan đến mỗi cuộc triển lãm; chúng có thể lên tới hàng trăm vật phẩm. Một vật phẩm có thể là một phần của các cuộc triển lãm khác nhau. Hơn nữa, mỗi cuộc triển lãm có thể được tổ chức bởi một số bảo tàng trong các thời kỳ khác nhau. Bạn phải ghi lại ngày bắt đầu và ngày kết thúc của mỗi triển lãm ở mỗi bảo tàng.
* Với một vật phẩm, cơ sở dữ liệu phải được thiết kế để cung cấp một cách hiệu quả tên của bảo tàng sở hữu nó.
* Với một cuộc triển lãm, cơ sở dữ liệu phải được thiết kế để cung cấp một cách hiệu quả tên của bảo tàng và tọa độ địa lý nơi nó đã được tổ chức.
* Hơn nữa, với một cuộc triển lãm, danh sách các vật phẩm có trong cuộc triển lãm và số lượng của chúng phải được đưa ra một cách hiệu quả.

Yêu cầu: Thiết kế lược đồ lưu trữ dữ liệu để đáp ứng các yêu câu trên (chỉ ra các collection trong CSDL, cấu trúc Document, mối quan hệ, các chiến lược (design pattern) được sử dụng để thiết kế và nêu lý do tại sao dùng chiến lược đó).

1. Đưa ra đủ và đúng các collection cần thiết (2đ)

* Bảo tàng
* Địa chỉ
* Vật phẩm
* Triển lãm

1. Đưa ra cấu trúc document: đủ các trường, đúng kiểu dữ liệu (2đ)

Text, letter

Description automatically generated Text, letter

Description automatically generated

Text, table

Description automatically generated Text, table, letter

Description automatically generated

1. Mối quan hệ: dùng embedding hoặc reference (trường tham chiếu); bản số (cũng thể hiện được mối quan hệ) (2đ)

Diagram

Description automatically generated

1. Áp dụng pattern: đưa ra được cấu trúc lưu trữ có sử dụng pattern (2đ)Diagram

   Description automatically generated
2. Nêu lý do áp dụng pattern: Giải thích lợi ích khi áp dụng pattern (2đ)

* Reference pattern: để truy vấn nhanh tên Bảo tàng mà không cần đến những trường khác, theo yêu cầu.
* Attribute pattern: để lưu các đặc tính của vật phẩm, bởi mỗi loại vật phẩm có những thuộc tính khác nhau.
* Schema Versioning Pattern: vừa truy vấn nhanh địa chỉ không cần thông qua collections DiaChi theo yêu cầu, vừa không cần tạo thêm collections DiaChi nhưng ít dùng tới.