

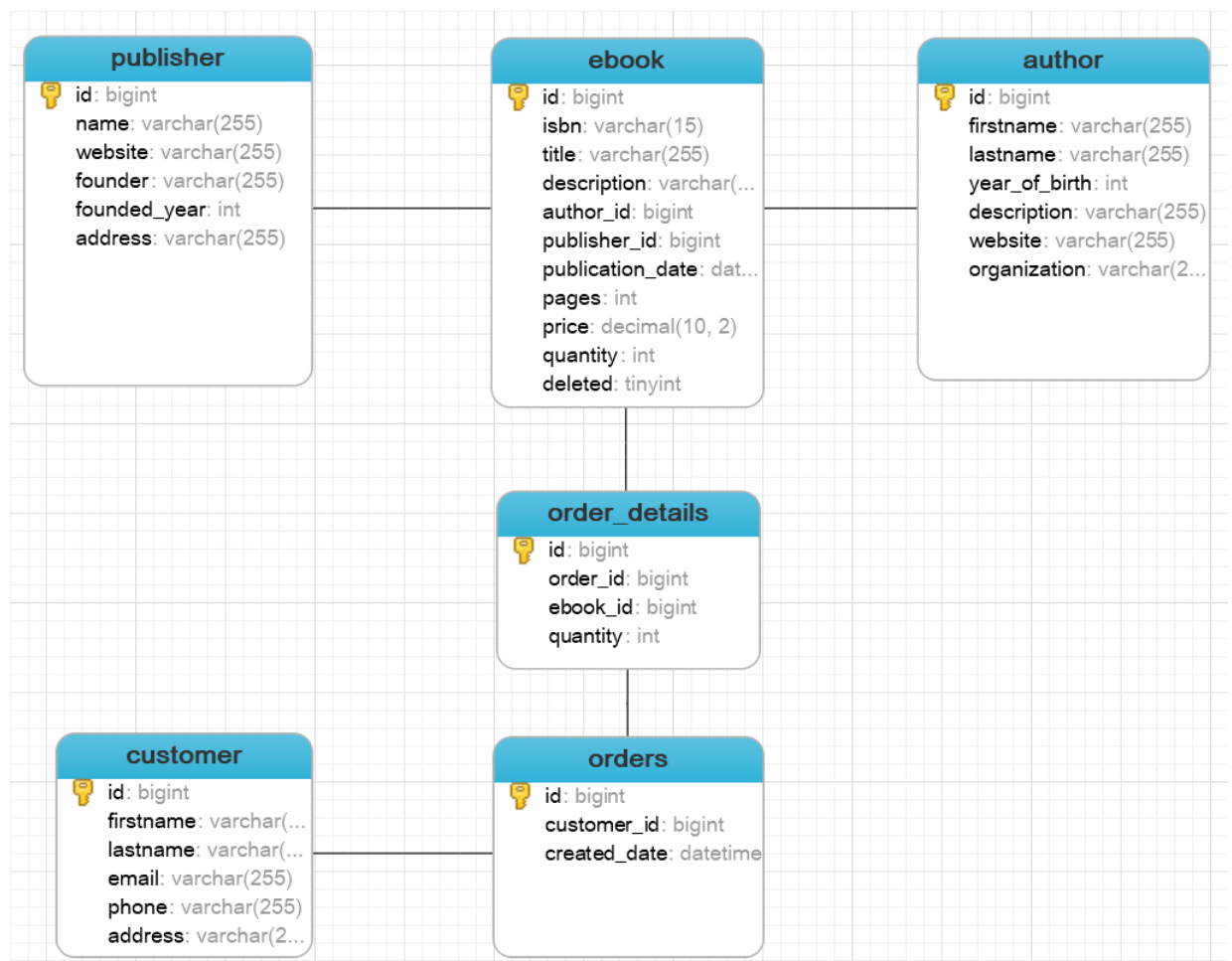
Bài tập Spring MVC – Tuần 04

I. Mô tả ngắn gọn

Bạn cần xây dựng một hệ thống Backend Restful Web Service sử dụng Spring MVC. Chức năng chủ yếu của hệ thống này là quản lý thông tin các ebooks, cho phép người dùng tìm kiếm các đầu sách theo nhiều tiêu chí khác nhau; order sách và quản lý các orders. Chúng ta sẽ gọi tên hệ thống này là **eb-shop**.

II. Lược đồ ER

Dưới đây là lược đồ ER thể hiện thông tin của các đối tượng trong hệ thống và mối quan hệ giữa các đối tượng đó:



Theo lược đồ ER trên, các thực thể trong hệ thống gồm có:

1. Sách (ebook)

Ebook là các đầu sách được quản lý bởi hệ thống. Thông tin của một ebook gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã đầu sách	Long	Khóa chính. Tự động tăng
Title	Tựa đề của sách	String	Required.
ISBN	International Standard Book Number – Mã số tiêu chuẩn của sách.	String	Required. Uniqued.
Description	Mô tả chung về sách	String	Required.
Author ID	Mã tác giả của sách	Long	Khóa ngoại đến bảng author
Publisher ID	Mã nhà xuất bản	Long	Khóa ngoại đến bảng publisher
Publication Date	Ngày xuất bản	Date	Required.
Pages	Số trang	Integer	Required.
Price	Giá bán	Long	Required.
Quantity	Số lượng sách còn lại trong kho sách.	Integer	Required. Phải ≥ 0
Deleted	Trạng thái của sách đã bị xóa hay chưa.	Boolean	Required. Default là false

2. Tác giả (author)

Thông tin về các tác giả gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã tác giả	Long	Khóa chính. Tự động tăng
Firstname	Tên	String	Required.
Lastname	Họ	String	Required.
Year of Birth	Năm sinh	Integer	Required.
Description	Mô tả chung về tác giả	String	Optional.
Website	Địa chỉ trang web/trang cá nhân của tác giả	String	Optional.
Organization	Nơi làm việc của tác giả (công ty, tổ chức,...)	String	Optional.

3. Nhà xuất bản (publisher)

Thông tin về các nhà xuất bản sách gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã NXB	Long	Khóa chính. Tự động tăng
Name	Tên NXB	String	Required.
Website	Trang web chính thức của NXB	String	Optional.
Founder	Tên người thành lập NXB	String	Optional.
Founded Year	Năm thành lập	Integer	Required.
Address	Địa chỉ NXB	String	Required.

4. Hóa đơn (orders)

Hệ thống cho phép khách hàng đặt mua sách, hệ thống cần lưu trữ lại thông tin hóa đơn của mỗi lần mua sách của khách hàng. Thông tin của một hóa đơn gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã hóa đơn	Long	Khóa chính. Tự động tăng
Customer ID	Mã khách hàng mua sách	Long	Khóa ngoại đến bảng customer
Created Date	Ngày tạo hóa đơn	Date	Required.

Lưu ý: Khách hàng có thể mua một hoặc nhiều cuốn sách trong mỗi lần mua, nên thông tin chi tiết của hóa đơn sẽ được lưu trữ trong bảng Chi tiết hóa đơn bên dưới.

5. Chi tiết hóa đơn (order_details)

Thông tin chi tiết của các hóa đơn, gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã chi tiết hóa đơn	Long	Khóa chính. Tự động tăng
Order ID	Mã hóa đơn	Long	Khóa ngoại đến bảng orders
Ebook ID	Mã sách được đặt mua	Long	Khóa ngoại đến bảng ebook
Quantity	Số lượng đặt mua	Integer	Required. Phải > 0

6. Khách hàng (customer)

Thông tin khách hàng đặt mua sách, gồm có:

Tên	Mô tả	Kiểu dữ liệu (Java)	Ghi chú
ID	Mã khách hàng	Long	Khóa chính. Tự động tăng
Firstname	Tên	String	Required.
Lastname	Họ	String	Required.
Email	Địa chỉ email	String	Required. Uniqued. Phải được validate trước khi save lại.
Phone	Số ĐT	String	Required.
Address	Địa chỉ	String	Required.

III. Yêu cầu chức năng

1. Thêm sách vào kho

Chức năng này cho phép người quản lý sách thêm sách vào kho sách. Sách được xác định dựa trên mã ISBN của sách. Nếu thêm 1 cuốn sách có mã ISBN đã tồn tại trong hệ thống thì hệ thống chỉ update lại các thông tin của sách cũ chứ không tạo sách mới, với số lượng sách - **quantity** được cộng thêm vào số lượng đang có. Nếu sách chưa tồn tại trong hệ thống thì thêm sách mới.

2. Chỉnh sửa thông tin sách

Chức năng này cho phép người quản lý sách chỉnh sửa thông tin của một đầu sách theo mã của đầu sách đó.

Nếu người dùng nhập vào mã của 1 đầu sách không tồn tại thì hệ thống phải trả về thông báo lỗi.

3. Xóa sách

Chức năng này cho phép người quản lý sách xóa một đầu sách khỏi hệ thống theo mã của đầu sách đó.

Hệ thống chỉ thực hiện **soft delete**, tức là khi thực hiện xóa thì set giá trị trường **deleted** của bảng **ebook** thành **true** thay vì xóa hẳn record đó khỏi bảng.

Nếu người dùng nhập vào mã của 1 đầu sách không tồn tại thì hệ thống phải trả về thông báo lỗi.

4. Tìm kiếm sách

Hệ thống cho phép người sử dụng tìm kiếm sách theo một trong (or) hoặc kết hợp (and) của các tiêu chí sau:

- **Tên sách**

Tìm kiếm tất cả các cuốn sách có tên sách chứa 1 chuỗi ký tự được người dùng nhập vào.

- **Mã tác giả**

Tìm kiếm tất cả các cuốn sách của 1 tác giả nào đó theo mã tác giả.

- **Mã nhà xuất bản**

Tìm kiếm tất cả các cuốn sách của 1 NXB nào đó theo mã NXB

- **Khoảng giá**

Tìm kiếm tất cả các cuốn sách có giá trong 1 khoảng nào đó.

Ví dụ: Tìm kiếm các cuốn sách có giá từ 20.000đ đến 80.000đ

- **Mã ISBN**

Tìm kiếm 1 cuốn sách theo mã ISBN của cuốn sách đó.

Kết quả trả về phải được sắp xếp theo một trong các thứ tự sau:

- **Tên sách**

Sắp xếp theo thứ tự bảng chữ cái của tên sách, có thể tăng dần (ASC) hoặc giảm dần (DESC)

- **Giá**

Sắp xếp theo thứ tự giá tăng dần hoặc giảm dần

- **Được mua nhiều**

Sách được đặt mua nhiều nhất được sắp xếp trước.

Kết quả trả về phải theo page với Page index và Page size được nhập vào từ request param. Page index bắt đầu từ 1.

Không trả về các đầu sách đã bị xóa khỏi hệ thống. (Tức là sách có trạng thái **deleted** là true)

5. Thêm tác giả

Thêm tác giả mới vào hệ thống.

6. Sửa thông tin tác giả

Chỉnh sửa thông tin của tác giả theo mã tác giả.

Nếu mã tác giả không tồn tại, hệ thống cần thông báo lỗi.

7. Xóa tác giả

Xóa tác giả khỏi hệ thống theo mã tác giả

Nếu mã tác giả không tồn tại, hệ thống cần thông báo lỗi.

Nếu vẫn còn sách của tác giả đó trong hệ thống thì không cho phép xóa tác giả.

8. Xem thông tin của tác giả

Trả về thông tin của tác giả theo mã tác giả.

Ngoài các thông tin cơ bản của tác giả, cần trả về thêm:

- Số lượng đầu sách của tác giả đó trong hệ thống.
- Danh sách 3 đầu sách bán chạy nhất của tác giả đó.

Nếu mã tác giả không tồn tại, hệ thống cần thông báo lỗi.

9. Thêm NXB

Thêm nhà xuất bản mới vào hệ thống.

10. Sửa thông tin NXB

Chỉnh sửa thông tin của NXB theo mã NXB.

Nếu mã NXB không tồn tại, hệ thống cần thông báo lỗi.

11. Xóa NXB

Xóa NXB khỏi hệ thống theo mã NXB.

Nếu mã NXB không tồn tại, hệ thống cần thông báo lỗi.

Nếu vẫn còn sách của NXB đó trong hệ thống thì không cho phép xóa NXB đó.

12. Xem thông tin của NXB

Trả về thông tin của NXB theo mã NXB.

Ngoài các thông tin cơ bản của NXB, cần trả về thêm:

- Số lượng đầu sách của NXB đó trong hệ thống.
- Danh sách 5 đầu sách bán chạy nhất của NXB đó.

Nếu mã NXB không tồn tại, hệ thống cần thông báo lỗi.

13. Đặt mua sách

Chức năng này cho phép người dùng đặt mua sách trong hệ thống. Khi đặt mua người dùng nhập vào các thông tin sau:

- Mã ISBN của sách đặt mua
- Số lượng sách
- Tên khách hàng
- Email khách hàng
- Số ĐT khách hàng

Khách hàng có thể đặt mua nhiều đầu sách trong mỗi lần mua. (Ví dụ trong 1 lần mua người dùng có thể mua 2 quyển Head First Java, 3 quyển Spring in Action, 1 quyển Thinking in Java).

Khi người dùng request đặt mua, hệ thống cần record lại thông tin order của khách hàng vào 2 bảng orders và order_details, đồng thời lưu lại thông tin của khách hàng vào bảng customer.

Khách hàng được xác định theo email, nếu 1 khách hàng đã tồn tại trong hệ thống rồi thì không cần tạo khách hàng mới nữa.

Trước khi lưu lại thông tin của khách hàng, cần validate email.

Sau khi khách hàng đặt mua thành công một đầu sách nào đó, số lượng đầu sách đó trong kho sẽ bị giảm đi theo số lượng đặt mua.

Nếu khách hàng đặt mua sách quá số lượng sách đang có trong hệ thống, hệ thống phải thông báo lỗi cho người dùng đó. (Ví dụ nếu trong kho chỉ còn 1 quyển Head First Java mà người dùng lại order đến 3 quyển thì hệ thống phải thông báo lỗi).

Khách hàng không thể đặt mua các đầu sách đã bị xóa khỏi hệ thống. (Tức là sách có trạng thái **deleted** là true)

14. Truy xuất thông tin hóa đơn

Hệ thống cần cho phép người dùng truy xuất thông tin hóa đơn theo mã hóa đơn. Các thông tin của hóa đơn được trả về bao gồm:

- Tên khách hàng
- Email
- Số ĐT
- Danh sách sách đặt mua, thông tin mỗi item bao gồm: Tên sách, tên tác giả, tên NXB, giá sách, số lượng sách được đặt mua.
- Tổng giá trị của hóa đơn đó.

15. Các chức năng thống kê

Hệ thống cần có các chức năng thống kê, bao gồm:

- Top 10 đầu sách bán chạy nhất (sắp xếp theo số lượng sách bán ra).
Thông tin của mỗi đầu sách bao gồm: Tên đầu sách, tên tác giả, tên NXB, tổng số lượng sách bán được.
Không trả về các đầu sách đã bị xóa khỏi hệ thống. (Tức là sách có trạng thái **deleted** là true)
- Top 5 tác giả có sách bán chạy nhất (sắp xếp theo số lượng sách bán ra).
Thông tin của mỗi tác giả gồm có: Tên tác giả, website, organization, tổng số lượng sách bán ra.
- Top 5 NXB có sách bán chạy nhất (sắp xếp theo số lượng sách bán ra).
Thông tin của mỗi NXB bao gồm: Tên NXB, website, địa chỉ, tổng số lượng sách bán ra.
- Top 5 khách hàng mua nhiều sách nhất (sắp xếp theo số lượng sách đặt mua).
Thông tin của mỗi khách hàng bao gồm: Tên, email, phone, address, tổng số lượng sách đã đặt mua, tổng giá trị của tất cả các hóa đơn.
- Top 5 khách hàng có tổng giá trị hóa đơn cao nhất (sắp xếp theo tổng giá trị hóa đơn).
Thông tin của mỗi khách hàng bao gồm: Tên, email, phone, address, tổng số lượng sách đã đặt mua, tổng giá trị của tất cả các hóa đơn.

IV. Các yêu cầu về kỹ thuật

1. Sử dụng Hệ quản trị CSDL MySQL để quản lý dữ liệu của hệ thống. Tạo 1 database có tên **ebshop** để lưu trữ và truy xuất dữ liệu.
2. Tạo sẵn dữ liệu ban đầu cho các bảng **publisher** (nhà xuất bản), **ebook** (sách), **author** (tác giả).
Yêu cầu ít nhất 10 nhà xuất bản, 50 đầu sách, 10 tác giả.
Về dữ liệu tham khảo có thể lấy ở các trang sau:
 - <https://it-eb.com/>
 - <http://it-ebooks.info/>
 - <http://www.allitebooks.in/>
 - https://www.amazon.com/s/ref=nb_sb_noss?url=search-alias%3Dstripbooks&field-keywords=java
3. Export database ra định dạng .sql (export cả schema và dữ liệu) và đặt trong thư mục resources để người review có thể import lại để chạy.
4. Sử dụng Spring MVC, tạo các REST API cho tất cả các chức năng của hệ thống. Tất cả các request và response body đều được format dưới dạng JSON.
5. Sử dụng Java-based configuration.
6. Sử dụng Postman để test các API, sau đó save lại dưới dạng Postman collection, đính kèm theo sourcecode đẩy lên Github.

V. Các yêu cầu về bàn giao

1. Cách thức làm việc
 - Theo nhóm.
 - Tú sẽ đóng vai trò là trưởng nhóm có trách nhiệm chính trong việc điều phối hoạt động của nhóm.
 - Tất cả các vấn đề về mặt kỹ thuật hoặc requirements có thể hỏi Hùng NH.
2. Deadline 01
 - Yêu cầu:
 - Đầu tiên, cả nhóm ngồi đọc và phân tích requirements. Sau đó thiết kế database, tạo dữ liệu ban đầu cho các bảng theo yêu cầu.
 - Export và push file **.sql** lên Github.
 - Cả nhóm ngồi lại với nhau tạo bảng phân công công việc cho từng người, nên break ra các task khác nhau, có thể break theo từng chức năng, rồi cho từng người nhận task. Cái này cả nhóm ngồi thảo luận với nhau để break task và phân công sao cho hợp lý nhất. Sử dụng file Google sheet online [này](#).
 - Deadline:
 - 11h sáng thứ 2 tuần sau (23/07)

3. Deadline 02

- Yêu cầu:
 - Tạo bảng mô tả tất cả các REST API cho tất cả các chức năng trong hệ thống theo yêu cầu chức năng ở trên. Điền thông tin vào file [API Docs này](#).
- Deadline:
 - 11h sáng thứ 3 tuần sau (24/07)

4. Deadline 03

- Yêu cầu:
 - Hoàn thành tất cả các chức năng của hệ thống.
- Deadline
 - 12h sáng thứ 2 tuần sau nữa (30/07)