

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



HỆ CƠ SỞ DỮ LIỆU

Ứng dụng mua bán sách trực tuyến

Học kỳ 231 - Nhóm NLTT - Lớp TN01

GVHD:	Nguyễn Thị Ái Thảo
SV:	Nguyễn Hoài Trung 2110631
	Nguyễn Châu Long 2111666
	Hồ Trọng Nhân 2111899
	Giản Định Thái 2112278



Mục lục

1 Đặc tả nghiệp vụ	3
1.1 Mô tả nghiệp vụ	3
1.2 Các ràng buộc dữ liệu	5
1.3 Các ràng buộc ngữ nghĩa	6
1.4 Các thay đổi so với Bài tập lớn 1	6
2 Mô hình quan hệ thực thể	6
2.1 Mô hình quan hệ thực thể mở rộng	6
2.2 Link Mô hình quan hệ thực thể mở rộng	7
3 Lược đồ cơ sở dữ liệu	7
3.1 Lược đồ cơ sở dữ liệu	7
3.2 Link Lược đồ cơ sở dữ liệu	7
4 Đặc tả cơ sở dữ liệu	9
4.1 DBMS sử dụng	9
4.2 Cấu trúc các file SQL	9
4.3 Tạo các bảng	9
4.4 Các ràng buộc dữ liệu	14
4.4.1 Ràng buộc về số điện thoại của nhà cung cấp	14
4.4.2 Ràng buộc về số điện thoại khi thêm vào đơn hàng	15
4.4.3 Ràng buộc về số điện thoại của khách hàng (người lớn)	15
4.4.4 Ràng buộc về số lượng quyển sách trong đơn hàng phải là số dương	15
4.4.5 Ràng buộc về số lượng quyển sách hiện có phải lớn hơn 0	16
4.4.6 Ràng buộc về số lượng quyển sách có trong 1 series phải lớn hơn 0	16
4.4.7 Ràng buộc điểm đánh giá của một quyển sách phải trong khoảng từ 1 đến 5	16
4.4.8 Ràng buộc về mã ISBN phải là 13 ký tự số	17
4.4.9 Ràng buộc về giá của quyển sách phải lớn hơn 0	17
4.4.10 Ràng buộc về phần trăm giảm giá của đợt giảm giá không được âm	18
4.4.11 Ràng buộc về phần trăm giảm giá của mã giảm giá không được âm	18
4.4.12 Ràng buộc về giá trị giảm tối đa của mã giảm giá phải lớn hơn 0	18
4.5 Các ràng buộc ngữ nghĩa	19
4.5.1 Kiểm tra chỉ cho phép người lớn đánh giá những quyển sách đã mua	19
4.5.2 Kiểm tra chỉ cho phép sách có khuyến mãi cùng thể loại khuyến mãi	20
4.5.3 Kiểm tra tuổi của người lớn phải lớn hơn bằng 16	21
4.5.4 Kiểm tra tuổi của trẻ em phải bé hơn 16	21
4.5.5 Kiểm tra chỉ cho phép khách hàng mua những quyển sách có độ tuổi phù hợp	22
4.5.6 Kiểm tra chỉ cho phép đơn hàng chứa những quyển sách cùng nhà cung cấp	23
4.5.7 Kiểm tra chỉ cho phép khách hàng mua những quyển sách điện tử 1 lần, với số lượng cuốn sách bằng 1	24
4.5.8 Kiểm tra khách hàng chỉ có thể mua số lượng sách bé hơn bằng số lượng sách đang được cung cấp	25
4.5.9 Kiểm tra khách hàng chỉ có thể sử dụng mã giảm giá còn hạn và còn số lượt sử dụng	25
4.5.10 Kiểm tra hạng thành viên của khách hàng khi sử dụng mã giảm giá	27
4.5.11 Kiểm tra khách hàng chỉ có thể sử dụng mã giảm giá thỏa giá trị tối thiểu	28
4.6 Các trigger, hàm và thủ tục	30
4.6.1 Câu 1.2.1: Viết các thủ tục để thêm (insert), sửa (update) và xóa (delete) dữ liệu vào một bảng dữ liệu	30
4.6.2 Câu 1.2.2: Viết 2 trigger để kiểm soát các hành động INSERT, UPDATE, DELETE trên một số bảng đã tạo	33
4.6.3 Câu 1.2.3: Viết 2 thủ tục trong đó chỉ chứa các câu truy vấn để hiển thị dữ liệu và tham số đầu vào là giá trị trong mệnh đề WHERE và/hoặc HAVING	37
4.6.4 Câu 1.2.4 Viết 2 hàm có sử dụng câu lệnh IF, có truy vấn dữ liệu và kiểm tra tham số đầu vào	39



5 Đặc tả ứng dụng	41
5.1 Đặc tả kỹ thuật	41
5.2 Cấu trúc code	41
5.3 Tạo kết nối với Database	42
5.3.1 Kết nối đến Database	42
5.3.2 Thực hiện truy vấn để lấy/thay đổi dữ liệu từ Database	42
5.4 Các chức năng đã hiện thực	43
5.4.1 Câu 5a Màn hình thêm/sửa sách	43
5.4.2 Câu 5b Màn hình hiển thị các cuốn sách của nhà cung cấp, xóa sách	56
5.4.3 Câu 5c Màn hình tác giả yêu thích của bạn	62
5.4.4 Màn hình thêm - Màn hình đặt sách của khách hàng:	64



1 Đặc tả nghiệp vụ

1.1 Mô tả nghiệp vụ

Một doanh nghiệp kinh doanh sách cần xây dựng hệ cơ sở dữ liệu cho ứng dụng mua bán sách trực tuyến tại Việt Nam.

Doanh nghiệp cho phép các nhà cung cấp có thể thêm, sửa, xóa thông tin các cuốn sách được bán trên hệ thống bao gồm tên cuốn sách, mã số định danh sách, thể loại của cuốn sách (một cuốn sách có thể có nhiều thể loại), thể loại gồm các thể loại chính như sau ('Kinh doanh', 'Truyện tranh', 'Giáo dục', 'Hư cấu', 'Sức khỏe', 'Lịch sử', 'Luật', 'Thần thoại', 'Y học', 'Chính trị', 'Lãng mạn', 'Tôn giáo', 'Khoa học', 'Self-help', 'Thể thao', 'Công nghệ', 'Du lịch', 'Thơ ca'), độ tuổi đọc tối thiểu của cuốn sách, giá của cuốn sách (giá tiền của một quyển sách phải lớn hơn hoặc bằng 0), ngôn ngữ, phiên bản cuốn sách, ngày xuất bản của cuốn sách, nhà xuất bản, mã ISBN gồm 13 ký tự số. Ngoài ra sách còn có **số điểm đánh giá của cuốn sách** dựa trên đánh giá khách hàng mua sách. Sách phải được xếp vào 3 kiểu phân biệt là KindleBook, AudioBook, Physical book.

- KindleBook cần lưu thêm thông tin về kích thước file, số trang.
- AudioBook cần lưu thêm thông tin về thời lượng, kích thước file.
- Physical book lưu thêm thông tin về định dạng (bìa cứng, bìa mềm...), khổ sách, số trang, cân nặng, tình trạng (mới, cũ).

1.1.1 Công thức tính điểm đánh giá cuốn sách

$$\text{Số điểm đánh giá} = \frac{\sum \text{điểm đánh giá từ người mua sách}_i}{\text{số đánh giá nhận được}}$$

Sách có thể không thuộc series nào hoặc thuộc về tối đa một series. Một series có thể chứa nhiều cuốn sách. Series cần có thông tin về tên, mã định danh series, số cuốn sách thuộc series đó (lớn hơn 0).

Sách phải được cung cấp bởi một nhà cung cấp, số lượng sách có thể cung cấp được lưu lại cho mỗi cuốn. Nhà cung cấp có các thông tin như địa chỉ, mã định danh, tên, số điện thoại (gồm 10 số có thể bắt đầu bằng 1 hoặc bắt đầu bằng 0), tỉ lệ chiết khấu mà doanh nghiệp thỏa thuận với nhà cung cấp (lớn hơn 0% và bé hơn hoặc bằng 100%). Mỗi nhà cung cấp lưu lại đều phải cung cấp từ một cuốn sách trở lên.

Doanh nghiệp có thể tạo các chương trình khuyến mãi áp dụng trên thể loại sách. Khuyến mãi cần có thêm mã định danh chương trình khuyến mãi, ngày bắt đầu/kết thúc, tên khuyến mãi, giá trị khuyến mãi (theo đơn vị %), thể loại sách áp dụng. Một cuốn sách có thể có nhiều chương trình khuyến mãi có thể áp dụng (**có thể loại ứng với thể loại trong chương trình khuyến mãi**) nhưng chỉ tính khuyến mãi tối đa từ một chương trình. Một chương trình khuyến mãi cũng có thể áp dụng cho nhiều quyển sách.

Các cuốn sách cần có thông tin về tác giả. Một cuốn sách có thể không có tác giả (sách kinh thánh...) hoặc sáng tác bởi nhiều tác giả, một tác giả cũng có thể sáng tác nhiều cuốn sách. Mỗi tác giả cần có mã số định danh tác giả và bút danh của tác giả đó. Tác giả có thể được theo dõi bởi nhiều khách hàng, một khách hàng cũng có thể theo dõi nhiều tác giả.

Khách hàng cần có các thông tin bao gồm mã số định danh cho tài khoản, tên (hiển thị trong ứng dụng với các người dùng khác), tên đăng nhập (duy nhất), mật khẩu tài khoản, giới tính (nhận một trong 3 giá trị là "M" (Nam), "F" (Nữ) và "O" (khác)), ngày sinh, tuổi tính được từ ngày sinh. Khách hàng phải được xếp vào 2 loại riêng biệt dựa trên độ tuổi là Child và Adult.

- **Child có tuổi nhỏ hơn 16**, phải được giám hộ bởi một Adult.
- **Adult có độ tuổi từ 16 trở lên**, có thể giám hộ nhiều Child. Adult còn có thêm các thuộc tính như số điện thoại (số điện thoại của đơn hàng/khách hàng là người trưởng thành gồm 10 số bắt đầu bằng số 0), tổng số tiền của các đơn hàng đã đặt, hạng thành viên (sử dụng cho các ưu đãi của ứng dụng), một hoặc nhiều địa chỉ. **Hạng thành viên được tính trên tổng số tiền đơn hàng đã trả trên thực tế (không tính số tiền được khuyến mãi). Hạng thành viên được chia thành Đồng/Bạc/Vàng/Kim Cương, ứng với tổng số tiền đã trả tối thiểu 0/1/5/20 (triệu vnd).**



Khách hàng có thể tìm kiếm các cuốn sách theo các thuộc tính của sách và cho vào các đơn hàng. Đơn hàng chỉ có thể chứa một hoặc nhiều cuốn sách thuộc cùng nhà cung cấp, một cuốn sách cũng có thể ở trong nhiều đơn hàng khác nhau. Với cuốn sách trong đơn hàng cần có thông tin về số lượng của mỗi cuốn. Đơn hàng chỉ có thể tạo bởi một người. Người dùng có thể theo dõi tình trạng vận chuyển và thanh toán của đơn hàng. Đơn hàng cần có thông tin như mã định danh cho đơn hàng, thời gian tạo, thời gian bắt đầu vận chuyển, loại vận chuyển, người vận chuyển, chi phí vận chuyển, tổng giá trị các cuốn sách đã chọn chưa bao gồm khuyến mãi, tổng khuyến mãi cho đơn hàng, số tiền phải thanh toán, phương thức thanh toán, thời gian thanh toán, trạng thái đơn hàng (gồm "Hoàn tất", "Đang giao", "Đã hủy"), địa chỉ nhận hàng, tên người nhận, số điện thoại người nhận. Sách trong cùng đơn hàng chỉ thuộc về một nhà cung cấp, cần lưu thêm thông tin về thời gian lấy đơn hàng từ nhà cung cấp, trạng thái lấy của hệ thống đối với đơn hàng được lấy từ nhà cung cấp ("Đã lấy", "Chưa lấy", "Đã hủy"). Người dùng chỉ có thể thêm cuốn sách vào đơn hàng nếu có số tuổi lớn hơn số tuổi yêu cầu của cuốn sách và số lượng cho mỗi cuốn sách không được lớn hơn số sách cung cấp của nhà cung cấp. Sách thuộc loại KindleBook hoặc AudioBook chỉ được mua với số lượng tối đa 1 và không mua lại. Nếu một quyển sách có nhiều khuyến mãi khác nhau, chỉ áp dụng khuyến mãi có giá trị lớn nhất. Các chi phí được tính toán theo bảng Công thức tính các chi phí đơn hàng.

1.2: Công thức tính các chi phí đơn hàng

$$\text{Item total} = \sum (\text{Price}_i * \text{Amount}_i)$$

$$\text{Item discount} = \sum (\text{Discount value}_i / 100 * \text{Price}_i * \text{Amount}_i)$$

Promotion discount = ($\text{Item total} - \text{Item discount}$) * Promotion value/100, nếu Promotion discount > Maximum promo, thì được tính bằng Maximum promo.

Total discount = Promotion discount + Item discount.

Grand total = Item total + Ship fee - Total discount

Trong đó:

- Item total: tổng giá trị các cuốn sách đã chọn chưa bao gồm khuyến mãi (vnđ)
- Price_i, amount_i: giá tiền (vnđ) và số lượng tương ứng cuốn sách thứ i trong đơn hàng
- Item discount: tổng tiền được giảm từ chương trình khuyến mãi
- Promotion discount: tổng số tiền được giảm từ mã khuyến mãi (vnđ)
- Promotion value: giá trị khuyến mãi từ mã khuyến mãi (%), thỏa điều kiện Item total >= giá trị tối thiểu, ngày xác nhận nằm trong thời gian giới hạn, với giá trị tối thiểu và thời gian lưu trong mã khuyến mãi.
- Maximum promo: giá trị giảm giá tối đa của mã khuyến mãi (giá trị này phải lớn hơn 0)
- Total discount: tổng số tiền được giảm (vnđ)
- Discount value_i: giá trị được giảm tối đa theo chương trình khuyến mãi (%) cho quyển sách tương ứng, thỏa điều kiện thẻ loại, ngày bắt đầu, ngày kết thúc của chương trình khuyến mãi.
- Grand total: số tiền phải thanh toán thực tế (vnđ)
- Ship fee: phí vận chuyển (vnđ)

Mỗi đơn hàng phải được lấy trực tiếp từ duy nhất một nhà cung cấp và giao đến cho khách hàng bởi bên thứ ba, một nhà cung cấp có thể cung cấp cho nhiều đơn hàng. Cần lưu lại thời điểm nhận đơn hàng từ nhà cung cấp, tình trạng để kiểm soát việc đã nhận hay chưa. Khi nhận được tiền đơn hàng, cần thực hiện thanh toán cho nhà cung cấp. Cần lưu lại các thông tin về trạng thái (gồm "Đã trả", "Chưa trả", "Đã hủy"), thời điểm thanh toán, phương thức thanh toán, số tiền thanh toán tính theo **Công thức phí thanh toán nhà cung cấp**



1.3: Công thức tính phí thanh toán cho nhà cung cấp

Tiền thanh toán cho nhà cung cấp (vnd) = Item total (vnd) * tỉ lệ chiết khấu ứng với từng nhà cung cấp(%) / 100.

Adult có thể xác nhận cho đơn hàng của Child chịu giám sát hoặc của bản thân. Một Adult có thể xác nhận cho nhiều đơn hàng nhưng đơn hàng chỉ được xác nhận bởi một Adult. Doanh nghiệp có thể tặng cho các khách hàng mã khuyến mãi với số lượng nhất định. Adult sở hữu các mã khuyến mãi và thông tin về số lượt sử dụng còn lại của mã khuyến mãi đó. Mã khuyến mãi có thể sở hữu bởi nhiều Adult và một Adult cũng có thể có nhiều loại mã khuyến mãi khác nhau. Mã khuyến mãi có các thông tin về tên, mã code, ngày bắt đầu, kết thúc, giá trị tối thiểu đơn hàng áp dụng, số tiền giảm tối đa (lớn hơn 0), giá trị khuyến mãi (tính theo % áp dụng trên giá trị đơn hàng, bé hơn bằng 100% và lớn hơn 0%), số lần sử dụng được cấp phát, loại thành viên được sử dụng. Đơn hàng chỉ có thể dùng các mã khuyến mãi hoặc nhận ưu đãi từ chương trình khuyến mãi nếu thỏa các điều kiện về thời gian khuyến mãi, giá trị tối thiểu (nếu có) được ghi trong mã khuyến mãi/chương trình khuyến mãi. **Người trưởng thành (Adult) chỉ sử dụng được các mã giảm giá (promotion code) mà có hạng thành viên bé hơn bằng hạng thành viên tài khoản của họ. Số lượt sử dụng mã khuyến mãi còn lại được tính bằng Công thức tính số lượt sử dụng mã khuyến mãi**

1.4: Công thức tính số lượt sử dụng mã khuyến mãi

Số lượt sử dụng mã khuyến mãi còn lại = số lượt sử dụng được cấp phát - số đơn hàng có dùng mã khuyến mãi sau khi đã được xác nhận.

Adult chỉ có thể cho đánh giá bằng điểm (thang điểm là số nguyên từ 1 đến 5) các cuốn sách đã mua. Một cuốn sách có thể được đánh giá bởi nhiều người. Một người cũng có thể đánh giá nhiều cuốn sách, mỗi người có thể đánh giá nhiều lần nhưng chỉ tính lần đánh giá cuối đối với một cuốn sách.

Adult có thể nhận xét bằng lời đối với các cuốn sách đã mua. Hệ thống cần lưu thông tin về nội dung, thời gian, mã định danh cho lời nhận xét. Một Adult có thể có nhiều lời nhận xét, có thể nhận xét một cuốn sách nhiều lần, một nhận xét phải thuộc về một Adult. Một cuốn sách có thể nhận được nhiều nhận xét, một nhận xét phải thuộc về một cuốn sách. Người dùng có thể xem nhận xét từ các người dùng khác.

1.2 Các ràng buộc dữ liệu

- Mã ISBN của cuốn sách gồm 13 ký tự số.
- Số điểm đánh giá là số nguyên từ 1 đến 5 điểm.
- Giá tiền của một quyển sách phải lớn hơn hoặc bằng 0.
- Số điện thoại (của đơn hàng/của khách hàng là người trưởng thành) gồm 10 số bắt đầu bằng số 0.
- Số điện thoại của nhà cung cấp gồm 10 số (có thể bắt đầu bằng 1 hoặc bắt đầu bằng 0).
- Giới tính của khách hàng là kí tự chữ chỉ nhận một trong 3 giá trị là "M" (Nam), "F" (Nữ) và "O" (khác).
- Số lượng quyển sách (có trong một series, có trong đơn hàng, được cung cấp bởi nhà cung cấp) phải là số nguyên lớn hơn 0.
- Các user_name của người dùng là duy nhất.
- Giá trị khuyến mãi của chương trình khuyến mãi(discount value), mã khuyến mãi(promotion value) phải là số lớn hơn 0% và bé hơn bằng 100%.
- Giá trị giảm giá tối đa của mã khuyến mãi (maximum promo) phải lớn hơn 0.
- Tỉ lệ chiết khấu ứng với từng nhà cung cấp phải là số lớn hơn 0% và bé hơn bằng 100%.



1.3 Các ràng buộc ngữ nghĩa

- Người trưởng thành (Adult) chỉ có thể cho đánh giá điểm các cuốn sách đã mua.
- Các chương trình khuyến mãi áp dụng cho sách được lưu phải có thẻ loại áp dụng phải có trong các thẻ loại của cuốn sách đó.
- Trẻ em (Child) có tuổi nhỏ hơn 16, người trưởng thành (Adult) có tuổi từ 16 trở lên.
- Hạng thành viên của Adult được tính trên tổng số tiền đơn hàng đã trả trên thực tế (không tính số tiền được khuyến mãi). Hạng thành viên được chia thành Đồng/Bạc/Vàng/Kim Cương, ứng với tổng số tiền đã trả tối thiểu 0/1/5/20 (triệu vnđ).
- Người trưởng thành (Adult) chỉ sử dụng được các mã giảm giá (promotion code) mà có hạng thành viên bé hơn bằng hạng thành viên tài khoản của họ.
- Người dùng chỉ có thể thêm cuốn sách vào đơn hàng nếu có số tuổi lớn hơn số tuổi yêu cầu của cuốn sách và số lượng cho mỗi cuốn sách không được lớn hơn số sách cung cấp của nhà cung cấp. Sách thuộc loại KindleBook hoặc AudioBook chỉ được mua với số lượng tối đa 1 và không mua lại.
- Đơn hàng chỉ có thể dùng các mã khuyến mãi hoặc nhận ưu đãi từ chương trình khuyến mãi nếu thỏa các điều kiện về thời gian khuyến mãi, giá trị tối thiểu (nếu có) được ghi trong mã khuyến mãi/chương trình khuyến mãi.
- Đơn hàng chỉ có thể chứa một hoặc nhiều cuốn sách thuộc cùng nhà cung cấp.

1.4 Các thay đổi so với Bài tập lớn 1

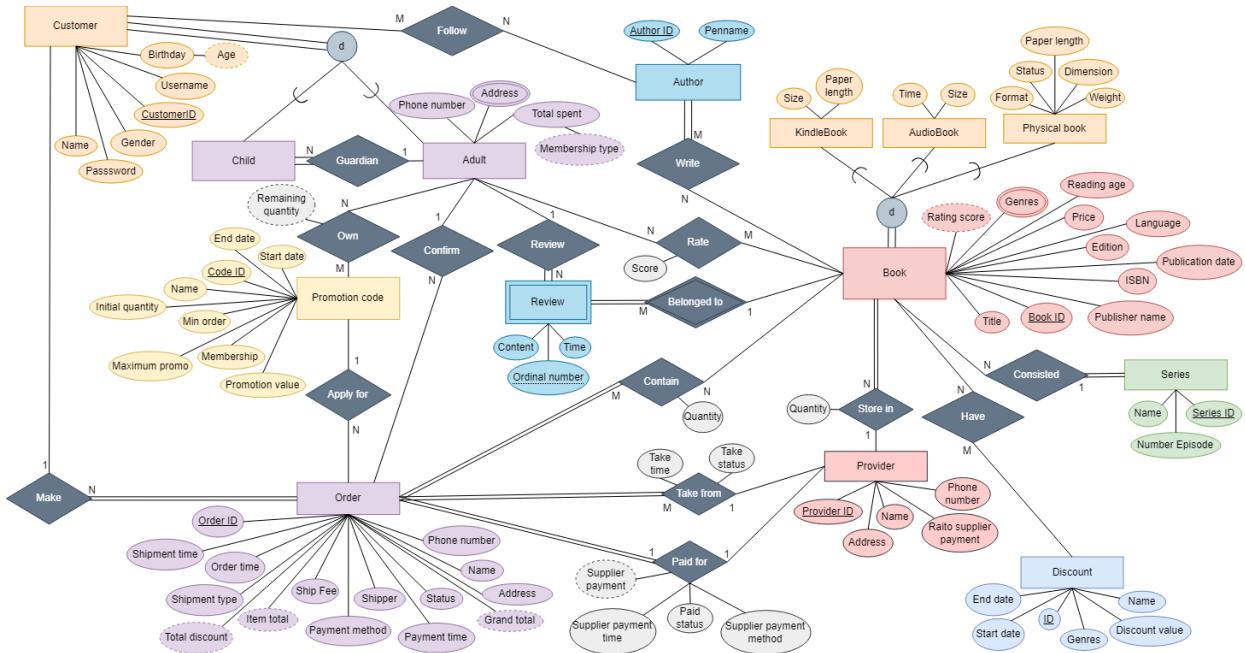
Dựa trên những nhận xét của cô và những vấn đề trong thực tế, nhóm đã có một số thay đổi sau :

- Thể loại các cuốn sách là: ('Kinh doanh','Truyện tranh','Giáo dục','Hư cấu','Sức khỏe','Lịch sử','Luật','Thần thoại','Y học','Chính trị','Lãng mạn','Tôn giáo','Khoa học','Self-help','Thể thao','Công nghệ','Du lịch','Thơ ca')
- Người trưởng thành (Adult) có lưu thêm thông tin về tổng số tiền của các đơn hàng đã hoàn tất.
- Trạng thái đơn hàng gồm 3 loại "Hoàn tất", "Đang giao", "Đã hủy".
- Trạng thái lấy sách gồm 3 loại "Đã lấy", "Chưa lấy", "Đã hủy".
- Trạng thái đơn trả tiền cho nhà cung cấp gồm 3 loại "Đã trả", "Chưa trả", "Đã hủy".

2 Mô hình quan hệ thực thể

2.1 Mô hình quan hệ thực thể mở rộng

Từ mô tả nghiệp vụ, nhóm đã vẽ mô hình quan hệ thực thể mở rộng (EERD) như sau:



Hình 1: Mô hình quan hệ thực thể mở rộng

2.2 Link Mô hình quan hệ thực thể mở rộng

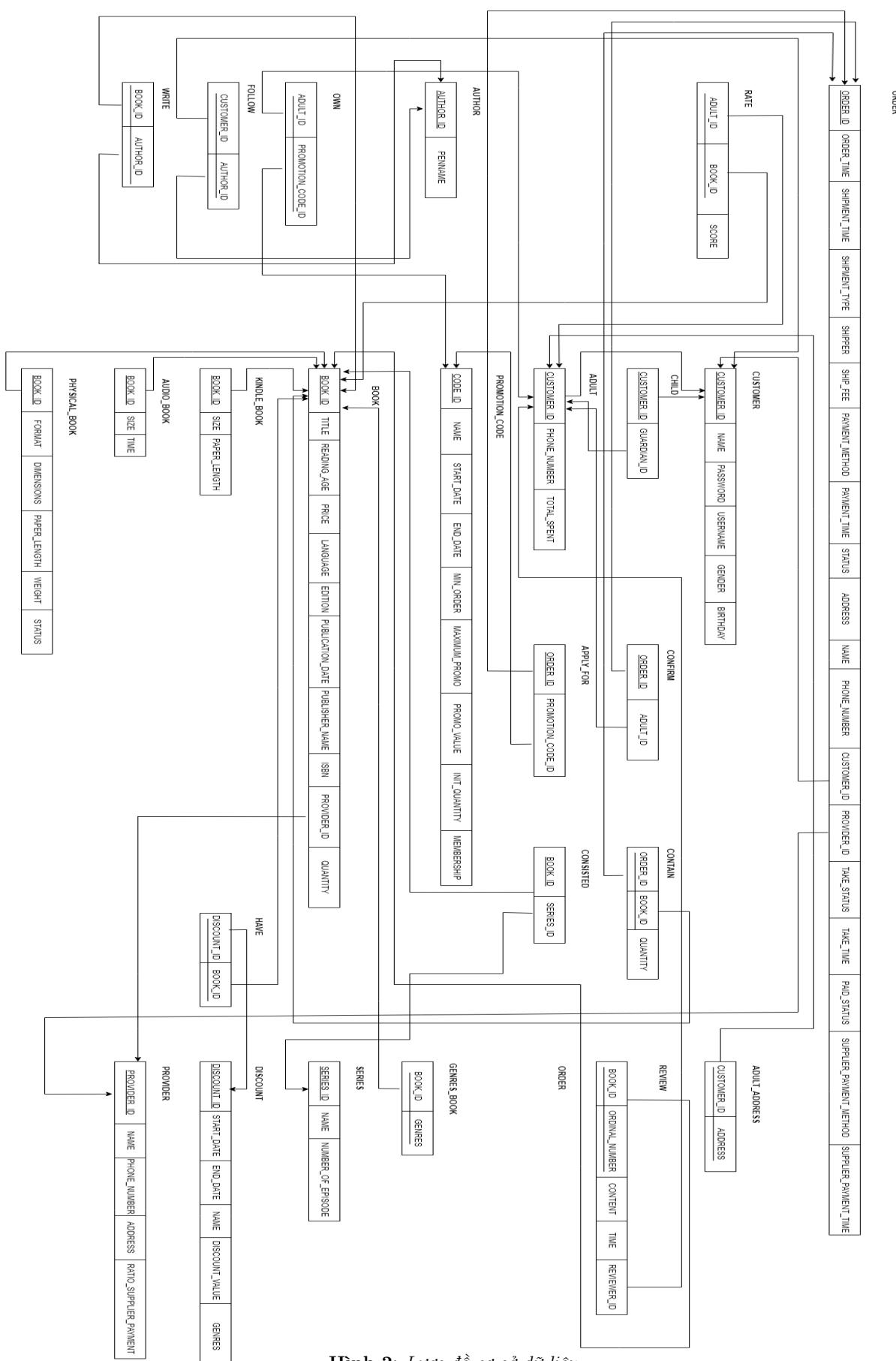
Để xem chi tiết hơn EERD, vui lòng truy cập: [\[HK231-TN01_NLTT\] EERD](#)

3 Lược đồ cơ sở dữ liệu

3.1 Lược đồ cơ sở dữ liệu

3.2 Link Lược đồ cơ sở dữ liệu

Để xem chi tiết hơn lược đồ cơ sở dữ liệu, vui lòng truy cập: [\[HK231-TN01_NLTT\] Schema](#)



Hình 2: Lược đồ cơ sở dữ liệu



4 Đặc tả cơ sở dữ liệu

4.1 DBMS sử dụng

Nhóm sử dụng DBMS là MYSQL 8.0. Nhóm chọn MYSQL vì các lý do chính sau:

- MYSQL là một hệ quản trị cơ sở dữ liệu (DBMS) mã nguồn mở, miễn phí.
- MYSQL sử dụng ngôn ngữ SQL.
- MYSQL còn có một cộng đồng sử dụng đông đảo, với các tài liệu, diễn đàn được cập nhật liên tục, giúp nhóm dễ dàng tìm kiếm tài liệu và giải đáp các thắc mắc
- MYSQL có tính tương thích với nhiều hệ điều hành và hỗ trợ nhiều loại ngôn ngữ khác nhau

4.2 Cấu trúc các file SQL

Cơ sở dữ liệu được khởi tạo bằng việc chạy các file theo thứ tự sau đây:

Tên file	Mô tả
Create.sql	Tạo schema và tất cả bảng trong đó với các quy định về kiểu dữ liệu.
Data_constraint.sql	Tạo các trigger để ràng buộc các dữ liệu nhập vào.
Meaning_constraint.sql	Tạo các trigger thể hiện các ràng buộc ngữ nghĩa.
Function_for_cal_price_order.sql	Tạo các hàm để tính tiền phải trả cho một đơn hàng.
Meaning_constraint...price.sql	Tạo các thủ tục để lọc ra các mã giảm giá và phần trăm giảm.
Input.sql	Nhập dữ liệu vào các bảng trong schema.
Procedure_Book.sql	Tạo các thủ tục sử dụng cho giao diện như thêm sách, lọc sách,...
Procedure_1.2.1.sql	Tạo các thủ tục cho các yêu cầu câu 1.2.1 của đề bài.
Procedure_for_cal_price_order.sql	Tạo các thủ tục để tính tiền phải trả cho một đơn hàng.
1.2.2 and 1.2.4.sql	Tạo các thủ tục cho các yêu cầu câu 1.2.2 và 1.2.4 của đề bài.

Bảng 1: Mô tả các file SQL

4.3 Tạo các bảng

Hệ cơ sở dữ liệu của nhóm gồm 25 bảng, trong đó được tạo bởi các lệnh sau:

```
1      CREATE SCHEMA 'assign_db' ;
```

Listing 1: Tạo Schema

```
1          CREATE TABLE assign_db.author (
2              author_id int NOT NULL primary key auto_increment ,
3              penname varchar(255) NOT NULL unique
4      );
```

Listing 2: Tạo bảng author

```
1                  CREATE table assign_db.customer(
2          customer_id int NOT NULL AUTO_INCREMENT primary key,
3          name_ varchar(255) NOT NULL ,
4          password_ varchar(255) NOT NULL ,
5          username varchar(255) NOT NULL unique ,
6          gender char(1) NOT NULL ,
7          birthday DATE NOT NULL
8      );
```

Listing 3: Tạo bảng customer



```
1 Create table assign_db.provider(
2     provider_id int NOT NULL AUTO_INCREMENT primary key,
3     name_ varchar(255) NOT NULL,
4     phone_number char(10) NOT NULL,
5     address varchar(255) NOT NULL,
6     ratio_supplier_payment double NOT NULL
7 );
```

Listing 4: Tạo bảng provider

```
1 Create table assign_db.discount(
2     discount_id int NOT NULL AUTO_INCREMENT primary key,
3     start_date datetime NOT NULL,
4     end_date datetime NOT NULL,
5     name_ varchar(255) NOT NULL,
6     discount_value int NOT NULL,
7     genres ENUM('Kinh doanh', 'Truyện tranh', 'Giáo dục', 'Hư cấu', 'Sức khỏe', 'Lịch sử',
8                 'Luật', 'Thần thoại', 'Y học', 'Chính trị', 'Lãng mạn', 'Tôn giáo', 'Khoa học',
9                 'Self-help', 'Thể thao', 'Công nghệ', 'Du lịch', 'Thơ ca') not null
8 );
```

Listing 5: Tạo bảng discount

```
1 Create table assign_db.book(
2     book_id int NOT NULL AUTO_INCREMENT primary key,
3     title varchar(255) NOT NULL,
4     reading_age int NOT NULL,
5     price double NOT NULL,
6     language_ varchar(255) NOT NULL,
7     edition varchar(255) NOT NULL,
8     publication_date DATE NOT NULL,
9     publisher_name varchar(255) NOT NULL,
10    isbn varchar(13) not null,
11    provider_id int NOT NULL,
12    quantity int NOT NULL,
13    FOREIGN KEY (provider_id) references provider(provider_id) On update restrict
14        on delete restrict
14 );
```

Listing 6: Tạo bảng book

```
1 create table assign_db.kindle_book(
2     book_id int NOT NULL primary key,
3     size int NOT NULL,
4     paper_length int NOT NULL,
5     FOREIGN KEY (book_id) references book(book_id) On update cascade on delete
6         cascade
6 );
```

Listing 7: Tạo bảng kindle_book

```
1 create table assign_db.audio_book (
2     book_id int NOT NULL primary key,
3     size int NOT NULL,
4     time_ TIME Not NULL,
5     FOREIGN KEY (book_id) references book(book_id) On update cascade on delete
6         cascade
6 );
```

Listing 8: Tạo bảng audio_book



```
1  create table assign_db.physical_book(
2      book_id int NOT NULL primary key,
3      format_ varchar(255) NOT NULL,
4      dimensions varchar(255) NOT NULL,
5      paper_length int NOT NULL,
6      weight double NOT NULL,
7      status_ varchar(255) NOT NULL,
8      FOREIGN KEY (book_id) references book(book_id) On update cascade on delete
9      cascade
10 );
```

Listing 9: Tạo bảng physical_book

```
1  create table assign_db.series(
2      series_id int NOT NULL primary key auto_increment,
3      name_ varchar(255) NOT NULL,
4      number_of_episode int NOT NULL
5 );
```

Listing 10: Tạo bảng series

```
1  create table assign_db.order_(
2      order_id int NOT NULL primary key auto_increment,
3      order_time datetime NOT NULL,
4      shipment_time datetime ,
5      shipment_type varchar(255) NOT NULL,
6      shipper varchar(255),
7      ship_fee double not null,
8      payment_method varchar(255) not null,
9      payment_time datetime,
10     status_ ENUM('Hoàn tất', 'Đang giao', 'Đã hủy') not null,
11     address varchar(255) not null,
12     name_ varchar(255) not null,
13     phone_number char(10) not null,
14     customer_id int not null,
15     provider_id int not null,
16     take_status ENUM('Đã lấy', 'Chưa lấy', 'Đã hủy') not null,
17     take_time datetime,
18     paid_status ENUM('Đã trả', 'Chưa trả', 'Đã hủy') not null,
19     supplier_payment_method varchar(255),
20     supplier_payment_time datetime,
21     FOREIGN KEY (customer_id) references customer(customer_id) On update restrict
22     on delete restrict,
23     FOREIGN KEY (provider_id) references provider(provider_id) On update restrict
24     on delete restrict
25 );
```

Listing 11: Tạo bảng order

```
1  create table assign_db.promotion_code(
2      code_id int NOT NULL primary key,
3      name_ varchar(255) NOT NULL,
4      start_date datetime NOT NULL,
5      end_date datetime NOT NULL,
6      min_order int NOT NULL,
7      maximum_promo int NOT NULL,
8      promo_value double NOT NULL,
9      init_quantity int NOT NULL,
10     membership ENUM ("Đồng", "Bạc", "Vàng", "Kim cương")
11 );
```

Listing 12: Tạo bảng promotion_code



```
1 create table assign_db.adult(
2     customer_id int NOT NULL,
3     phone_number char(10) not null,
4     total_spent int not null default 0,
5     FOREIGN KEY (customer_id) references customer(customer_id) On update
restrict on delete restrict
6 );
```

Listing 13: Tạo bảng adult

```
1 create table assign_db.child(
2     customer_id int NOT NULL primary key,
3     guardian_id int NOT NULL,
4     FOREIGN KEY (customer_id) references customer(customer_id) On update restrict
on delete restrict,
5     FOREIGN KEY (guardian_id) references adult (customer_id) On update restrict
on delete restrict
6 );
```

Listing 14: Tạo bảng child

```
1 create table assign_db.confirm(
2     order_id int not NULL primary key,
3     adult_id int not NULL,
4     foreign key (adult_id) references adult(customer_id) On update restrict on
delete restrict,
5     foreign key (order_id) references order_(order_id) On update cascade on
delete cascade
6 );
```

Listing 15: Tạo bảng confirm

```
1 create table assign_db.apply_for(
2     order_id int not NULL primary key,
3     promotion_code_id int not NULL,
4     foreign key (order_id) references order_(order_id) On update cascade on
delete cascade,
5     foreign key (promotion_code_id) references promotion_code(code_id) On update
restrict on delete restrict
6 );
```

Listing 16: Tạo bảng apply_for

```
1 create table assign_db.contain(
2     order_id int not NULL,
3     book_id int not NULL,
4     quantity int not NULL,
5     foreign key (order_id) references order_(order_id) On update cascade on
delete cascade,
6     foreign key (book_id) references book(book_id) On update restrict on delete
restrict,
7     CONSTRAINT pk_contain PRIMARY KEY (order_id, book_id)
8 );
```

Listing 17: Tạo bảng contain

```
1 create table assign_db.consisted(
2     book_id int not NULL primary key,
3     series_id int not NULL,
```



```
4      foreign key (book_id) references book(book_id) On update cascade on delete
5      cascade,
6      foreign key (series_id) references series(series_id) On update restrict on
7      delete restrict
8 );
```

Listing 18: Tạo bảng consisted

```
1 CREATE table assign_db.rate(
2     adult_id int NOT NULL,
3     book_id int NOT NULL,
4     score int NOT NULL,
5     CONSTRAINT pk_rate PRIMARY KEY (adult_id, book_id),
6     foreign key (adult_id) references adult (customer_id) On update restrict on
7     delete restrict,
8     foreign key (book_id) references book (book_id) On update restrict on delete
9     restrict
10 );
```

Listing 19: Tạo bảng rate

```
1 CREATE table assign_db.own (
2     adult_id int NOT NULL,
3     promotion_code_id int NOT NULL,
4     CONSTRAINT pk_own PRIMARY KEY (adult_id, promotion_code_id),
5     foreign key (adult_id) references adult (customer_id) On update restrict on
6     delete restrict,
7     foreign key (promotion_code_id) references promotion_code (code_id) On update
8     restrict on delete restrict
9 );
```

Listing 20: Tạo bảng own

```
1
2 CREATE table assign_db.follow (
3     customer_id int NOT NULL,
4     author_id int NOT NULL,
5     CONSTRAINT pk_follow PRIMARY KEY (customer_id,author_id ),
6     foreign key (customer_id) references customer (customer_id) On update
7     restrict on delete restrict,
8     foreign key (author_id) references author (author_id) On update restrict on
9     delete restrict
10 );
```

Listing 21: Tạo bảng follow

```
1 CREATE table assign_db.write_ (
2     book_id int NOT NULL,
3     author_id int NOT NULL,
4     CONSTRAINT pk_follow PRIMARY KEY (book_id,author_id ),
5     foreign key (book_id) references book (book_id) On update cascade on delete
6     cascade,
7     foreign key (author_id) references author (author_id) On update restrict on
8     delete restrict
9 );
```

Listing 22: Tạo bảng write_

```
1 Create table assign_db.review(
2     book_id int NOT NULL,
```



```
3     ordinal_number int NOT NULL,
4     content varchar(255) not null,
5     time_ datetime not null,
6     reviewer_id int not null,
7     constraint pk_review Primary key (book_id, ordinal_number),
8     foreign key (book_id) references book (book_id) On update cascade on delete
9     cascade,
10    foreign key (reviewer_id) references adult(customer_id) On update restrict on
11    delete restrict
12  );
```

Listing 23: Tạo bảng review

```
1  create table assign_db.genres_book(
2    book_id int not null,
3    genres ENUM('Kinh doanh', 'Truyện tranh', 'Giáo dục', 'Hư cấu', 'Sức khỏe', 'Lịch
4    sử', 'Luật', 'Thần thoại', 'Y học', 'Chính trị', 'Lãng mạn', 'Tôn giáo', 'Khoa học',
5    'Self-help', 'Thể thao', 'Công nghệ', 'Du lịch', 'Thơ ca') not null,
6    constraint pk_genres_book primary key (book_id, genres),
7    foreign key (book_id) references book (book_id) On update cascade on delete
8    cascade
9  );
```

Listing 24: Tạo bảng genres

```
1  create table assign_db.adult_address (
2    customer_id int not null,
3    address varchar(255) not null,
4    constraint pk_adult_address primary key (customer_id,address),
5    foreign key (customer_id) references adult(customer_id) On update restrict on
6    delete restrict
7  );
```

Listing 25: Tạo bảng adult_address

4.4 Các ràng buộc dữ liệu

Nhóm cũng đã thực hiện các ràng buộc dữ liệu như sau:

4.4.1 Ràng buộc về số điện thoại của nhà cung cấp

```
1  create Trigger assign_db.provider_phone_number_insert
2    before insert on assign_db.provider
3    for each row
4    Begin
5      if not (new.phone_number  regexp '^[0]+[0-9]{9}' or new.phone_number  regexp ,
6             '^[1]+[0-9]{9}') then
7        signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số, b
8       ắt đầu bằng 0 hoặc 1';
9      end if;
10 End;
11 |
12 create Trigger assign_db.provider_phone_number_update
13   before update on assign_db.provider
14   for each row
15   Begin
16     if not (new.phone_number  regexp '^[0]+[0-9]{9}' or new.phone_number  regexp ,
17             '^[1]+[0-9]{9}') then
18       signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số, b
19      ắt đầu bằng 0 hoặc 1';
20     end if;
21 End;
```



Listing 26: Kiểm tra số điện thoại của nhà cung cấp phải có 10 chữ số, bắt đầu bằng 0 hoặc 1

4.4.2 Ràng buộc về số điện thoại khi thêm vào đơn hàng

```
1 create Trigger assign_db.order__phone_number_insert
2 before insert on assign_db.order_
3 for each row
4 Begin
5   if not (new.phone_number  regexp '^[0]+[0-9]{9}') then
6     signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số và
7       bắt đầu bằng 0';
8   end if;
9 End;
10
10 create Trigger assign_db.order__phone_number_update
11 before update on assign_db.order_
12 for each row
13 Begin
14   if not (new.phone_number  regexp '^[0]+[0-9]{9}') then
15     signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số và
16       bắt đầu bằng 0';
17   end if;
17 End;
```

Listing 27: Kiểm tra số điện thoại của order phải có 10 chữ số và bắt đầu bằng 0

4.4.3 Ràng buộc về số điện thoại của khách hàng (người lớn)

```
1 create Trigger assign_db.adult__phone_number_insert
2 before insert on assign_db.adult_
3 for each row
4 Begin
5   if not (new.phone_number  regexp '^[0]+[0-9]{9}') then
6     signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số và
7       bắt đầu bằng 0';
8   end if;
9 End;
10
10 create Trigger assign_db.adult__phone_number_update
11 before update on assign_db.adult_
12 for each row
13 Begin
14   if not (new.phone_number  regexp '^[0]+[0-9]{9}') then
15     signal sqlstate '45000' set message_text ='Số điện thoại phải có 10 chữ số và
16       bắt đầu bằng 0';
17   end if;
17 End;
```

Listing 28: Kiểm tra số điện thoại của adult phải có 10 chữ số và bắt đầu bằng 0

4.4.4 Ràng buộc về số lượng quyển sách trong đơn hàng phải là số dương

```
1 create Trigger assign_db.contain_quantity_insert
2 before insert on assign_db.contain
3 for each row
4 Begin
5   if (new.quantity <=0) then
6     signal sqlstate '45000' set message_text ='Số cuốn sách phải lớn hơn 0';
```



```
7     end if;
8 End;
9 |
10 create Trigger assign_db.contain_quantity_update
11 before update on assign_db.contain
12 for each row
13 Begin
14   if (new.quantity <=0) then
15     signal sqlstate '45000' set message_text ='Số cuốn sách phải lớn hơn 0';
16   end if;
17 End;
```

Listing 29: Kiểm tra số lượng quyển sách có trong đơn hàng phải lớn hơn 0

4.4.5 Ràng buộc về số lượng quyển sách hiện có phải lớn hơn 0

```
1 create Trigger assign_db.book_insert
2 before insert on assign_db.book
3 for each row
4 Begin
5   if (new.quantity <0) then
6     signal sqlstate '45000' set message_text ='Số cuốn sách phải lớn hơn bằng 0';
7   end if;
8 End;
9 |
10 create Trigger assign_db.book_update
11 before update on assign_db.book
12 for each row
13 Begin
14   if (new.quantity <0) then
15     signal sqlstate '45000' set message_text ='Số cuốn sách phải lớn hơn bằng 0';
16   end if;
17 End;
```

Listing 30: Kiểm tra số lượng quyển sách được cung cấp bởi nhà cung cấp phải lớn hơn 0

4.4.6 Ràng buộc về số lượng quyển sách có trong 1 series phải lớn hơn 0

```
1 create Trigger assign_db.series_insert
2 before insert on assign_db.series
3 for each row
4 Begin
5   if (new.number_of_episode <= 0) then
6     signal sqlstate '45000' set message_text ='Số quyển sách phải lớn hơn 0';
7   end if;
8 End;
9 |
10 create Trigger assign_db.series_update
11 before update on assign_db.series
12 for each row
13 Begin
14   if (new.number_of_episode <= 0) then
15     signal sqlstate '45000' set message_text ='Số quyển sách phải lớn hơn 0';
16   end if;
17 End;
```

Listing 31: Kiểm tra số cuốn sách có trong series phải lớn hơn 0

4.4.7 Ràng buộc điểm đánh giá của một quyển sách phải trong khoảng từ 1 đến 5



```
1  create Trigger assign_db.rate_score_insert
2  before insert on assign_db.rate
3  for each row
4  Begin
5    if (new.score < 1 or new.score > 5) then
6      signal sqlstate '45000' set message_text ='Số điểm phải từ 1 đến 5';
7    end if;
8  End;
9  |
10 create Trigger assign_db.rate_score_update
11 before update on assign_db.rate
12 for each row
13 Begin
14   if (new.score < 1 or new.score > 5) then
15     signal sqlstate '45000' set message_text ='Số điểm phải từ 1 đến 5';
16   end if;
17 End;
18 |
```

Listing 32: Kiểm tra điểm đánh giá cuốn sách phải lớn hơn bằng 1 và bé hơn bằng 5

4.4.8 Ràng buộc về mã ISBN phải là 13 ký tự số

```
1  create Trigger assign_db.book_ISBN_insert
2  before insert on assign_db.book
3  for each row
4  Begin
5    if not (new.ISBN regexp '^[0-9]{13}') then
6      signal sqlstate '45000' set message_text ='Mã ISBN của cuốn sách phải là 13 k
    ý tự chữ số';
7    end if;
8  End;
9  |
10 create Trigger assign_db.book_ISBN_update
11 before update on assign_db.book
12 for each row
13 Begin
14   if not (new.ISBN regexp '^[0-9]{13}') then
15     signal sqlstate '45000' set message_text ='Mã ISBN của quyền sách phải là 13
     ký tự chữ số';
16   end if;
17 End;
```

Listing 33: Kiểm tra mã ISBN phải là 13 ký tự chữ số

4.4.9 Ràng buộc về giá của quyền sách phải lớn hơn 0

```
1  create Trigger assign_db.book_price_insert
2  before insert on assign_db.book
3  for each row
4  Begin
5    if (new.price < 0) then
6      signal sqlstate '45000' set message_text ='Giá tiền của quyền sách phải lớn h
     ơn hoặc bằng 0';
7    end if;
8  End;
9  |
10 create Trigger assign_db.book_price_update
11 before update on assign_db.book
12 for each row
13 Begin
14   if (new.price < 0) then
```



```
15      signal sqlstate '45000' set message_text ='Giá tiền của quyền sách phải lớn h
16          ơn hoặc bằng 0';
17  end if;
18 End;
```

Listing 34: Kiểm tra giá tiền của sách phải lớn hơn 0

4.4.10 Ràng buộc về phần trăm giảm giá của đợt giảm giá không được âm

```
1 create Trigger assign_db.discount_discount_value_insert
2 before insert on assign_db.discount
3 for each row
4 Begin
5   if (new.discount_value < 0 or new.discount_value > 100) then
6     signal sqlstate '45000' set message_text ='Phần trăm giảm giá phải lớn hơn 0
và bé hơn bằng 100';
7   end if;
8 End;
9 |
10 create Trigger assign_db.discount_discount_value_update
11 before update on assign_db.discount
12 for each row
13 Begin
14   if (new.discount_value < 0 or new.discount_value > 100) then
15     signal sqlstate '45000' set message_text ='Phần trăm giảm giá phải lớn hơn 0
và bé hơn bằng 100';
16   end if;
17 End;
```

Listing 35: Kiểm tra giá trị giảm giá của discount phải lớn hơn 0

4.4.11 Ràng buộc về phần trăm giảm giá của mã giảm giá không được âm

```
1 create Trigger assign_db.promotion_code_promo_value_insert
2 before insert on assign_db.promotion_code
3 for each row
4 Begin
5   if (new.promo_value < 0 or new.promo_value > 100) then
6     signal sqlstate '45000' set message_text ='Phần trăm giảm giá của mã khuyến m
ãi phải lớn hơn 0 và bé hơn bằng 100';
7   end if;
8 End;
9 |
10 create Trigger assign_db.promotion_code_promo_value_update
11 before update on assign_db.promotion_code
12 for each row
13 Begin
14   if (new.promo_value < 0 or new.promo_value > 100) then
15     signal sqlstate '45000' set message_text ='Phần trăm giảm giá của mã khuyến m
ãi phải lớn hơn 0 và bé hơn bằng 100';
16   end if;
17 End;
18 |
```

Listing 36: Kiểm tra giá trị giảm giá của promotion_code phải lớn hơn 0

4.4.12 Ràng buộc về giá trị giảm tối đa của mã giảm giá phải lớn hơn 0

```
1 create Trigger assign_db.promotion_code_maximum_promo_insert
2 before insert on assign_db.promotion_code
```



```
3 for each row
4 Begin
5   if (new.maximum_promo < 0) then
6     signal sqlstate '45000' set message_text ='Giá trị giảm tối đa của mã giảm gi
á phải lớn hơn 0';
7   end if;
8 End;
|
10 create Trigger assign_db.promotion_code_maximum_promo_update
11 before update on assign_db.promotion_code
12 for each row
13 Begin
14   if (new.maximum_promo < 0) then
15     signal sqlstate '45000' set message_text ='Giá trị giảm tối đa của mã giảm gi
á phải lớn hơn 0';
16   end if;
17 End;
```

Listing 37: Kiểm tra mức giảm giá tối đa của promotion_code phải lớn hơn 0

4.5 Các ràng buộc ngữ nghĩa

Nhóm cũng đã thực hiện ràng buộc ngữ nghĩa như sau:

4.5.1 Kiểm tra chỉ cho phép người lớn đánh giá những quyền sách đã mua

```
1 Create Trigger assign_db.rate_only_bought_insert
2 before insert on assign_db.rate
3 for each row
4 BEGIN
5   Declare is_bought_book INT default 0;
6   if (new.book_id <= 0) then
7     signal sqlstate '45000' set message_text ='Mã sách không hợp lệ';
8   end if;
9
10  if (new.adult_id <= 0) then
11    signal sqlstate '45000' set message_text ='Mã khách hàng không hợp lệ';
12  end if;
13
14  if not (select exists (select * from book where book.book_id = book_id)) then
15    signal sqlstate '45000' set message_text ='Sách không tồn tại';
16  end if;
17
18  if not (select exists (select * from adult where adult.customer_id =
customer_id)) then
19    signal sqlstate '45000' set message_text ='Người dùng không tồn tại';
20  end if;
21
22  With book_bought AS
23    (SELECT book_id, adult_id
24     FROM confirm
25     Inner join contain on confirm.order_id=contain.order_id
26       Join order_ on order_.order_id=contain.order_id
27       Where book_id=new.book_id and adult_id=new.adult_id and order_.status_='Hoàn
tất')
28
29    SELECT Count(*)
30    INTO is_bought_book
31    FROM book_bought;
32
33    if (is_bought_book<=0) then
34      signal sqlstate '45000' set message_text ='Người dùng phải mua mới được quyền
đánh giá điểm cuốn sách';
35    end if;
36 END;
37 |
```



```
38 Create Trigger assign_db.rate_only_bought_update
39 before update on assign_db.rate
40 for each row
41 BEGIN
42     Declare is_bought_book INT default 0;
43     if (new.book_id <= 0) then
44         signal sqlstate '45000' set message_text ='Mã sách không hợp lệ';
45     end if;
46
47     if (new.adult_id <= 0) then
48         signal sqlstate '45000' set message_text ='Mã khách hàng không hợp lệ';
49     end if;
50
51     if not (select exists (select * from book where book.book_id = book_id)) then
52         signal sqlstate '45000' set message_text ='Sách không tồn tại';
53     end if;
54
55     if not (select exists (select * from adult where adult.customer_id =
56         customer_id)) then
57         signal sqlstate '45000' set message_text ='Người dùng không tồn tại';
58     end if;
59
60     With book_bought AS
61     (SELECT book_id, adult_id
62     FROM confirm
63     Inner join contain on confirm.order_id=contain.order_id
64     Join order_ on order_.order_id=contain.order_id
65     Where book_id=new.book_id and adult_id=new.adult_id and order_.status_='Hoàn
66     tất')
67
68     SELECT Count(*)
69     INTO is_bought_book
70     FROM book_bought;
71
72     if (is_bought_book<=0) then
73         signal sqlstate '45000' set message_text ='Người dùng phải mua mới được quyền
74     đánh giá điểm cuốn sách';
75     end if;
76 END;
```

Listing 38: Kiểm tra chỉ cho phép người lớn rate những quyển sách đã mua

4.5.2 Kiểm tra chỉ cho phép sách có khuyến mãi cùng thẻ loại khuyến mãi

```
1 Create Trigger assign_db.have_match_genres_insert
2 before insert on assign_db.have_
3 for each row
4 BEGIN
5     Declare is_having_genres INT default 0;
6
7     With book_genres As
8     (Select book_id
9     From genres_book p1
10    inner join discount p2 on p1.genres=p2.genres
11    where book_id=new.book_id and discount_id=new.discount_id)
12
13     Select Count(*)
14     Into is_having_genres
15     from book_genres;
16
17     If (is_having_genres<=0) then
18         signal sqlstate '45000' set message_text ='Sách không thuộc thẻ loại của chương
19         trình khuyến mãi này';
20     end if;
21 END;
```



```
22 Create Trigger assign_db.have_match_genres_update
23 before update on assign_db.have_
24 for each row
25 BEGIN
26     Declare is_having_genres INT default 0;
27
28     With book_genres As
29         (Select book_id
30          From genres_book p1
31         inner join discount p2 on p1.genres=p2.genres
32         where book_id=new.book_id and discount_id=new.discount_id)
33
34     Select Count(*)
35     Into is_having_genres
36     from book_genres;
37
38     If (is_having_genres<=0) then
39         signal sqlstate '45000' set message_text ='Sách không thuộc thể loại của chương trình khuyến mãi này';
40     end if;
41 END
```

Listing 39: Kiểm tra chỉ cho phép quyền sách có khuyến mãi cùng thể loại với thể loại khuyến mãi

4.5.3 Kiểm tra tuổi của người lớn phải lớn hơn bằng 16

```
1 Create Trigger assign_db.adult_age_16_insert
2 before insert on assign_db.adult
3 for each row
4 BEGIN
5     Declare customer_birthday date;
6     Declare age INT default 0;
7     SELECT birthday into customer_birthday
8     FROM customer
9     WHERE customer_id=new.customer_id;
10    SELECT TIMESTAMPDIFF(YEAR, customer_birthday, CURDATE()) into age;
11    If (age<16) then
12        signal sqlstate '45000' set message_text ='Tuổi của tài khoản trưởng thành phải lớn hơn bằng 16';
13    end if;
14 END
15 |
16
17 Create Trigger assign_db.adult_age_16_update
18 before update on assign_db.adult
19 for each row
20 BEGIN
21     Declare customer_birthday date;
22     Declare age INT default 0;
23     SELECT birthday into customer_birthday
24     FROM customer
25     WHERE customer_id=new.customer_id;
26     SELECT TIMESTAMPDIFF(YEAR, customer_birthday, CURDATE()) into age;
27     If (age<16) then
28        signal sqlstate '45000' set message_text ='Tuổi của tài khoản trưởng thành phải lớn hơn bằng 16';
29     end if;
30 END
```

Listing 40: Kiểm tra tuổi của adult phải lớn hơn bằng 16

4.5.4 Kiểm tra tuổi của trẻ em phải bé hơn 16



```
1 Create Trigger assign_db.adult_child_16_insert
2 before insert on assign_db.child
3 for each row
4 BEGIN
5     Declare customer_birthday date;
6     Declare age INT default 0;
7     SELECT birthday into customer_birthday
8     FROM customer
9     WHERE customer_id=new.customer_id;
10    SELECT TIMESTAMPDIFF(YEAR, customer_birthday, CURDATE()) into age;
11    If (age>16) then
12        signal sqlstate '45000' set message_text ='Tuổi của tài khoản trẻ em phải bé
13        hơn 16';
14    end if;
15 END
16 |
17 Create Trigger assign_db.child_age_16_update
18 before update on assign_db.child
19 for each row
20 BEGIN
21     Declare customer_birthday date ;
22     Declare age INT default 0;
23     SELECT birthday into customer_birthday
24     FROM customer
25     WHERE customer_id=new.customer_id;
26     SELECT TIMESTAMPDIFF(YEAR, customer_birthday, CURDATE()) into age;
27     If (age>=16) then
28         signal sqlstate '45000' set message_text ='Tuổi của tài khoản trẻ em phải bé
29        hơn 16';
30    end if;
31 END
```

Listing 41: Kiểm tra tuổi của child phải bé hơn 16

4.5.5 Kiểm tra chỉ cho phép khách hàng mua những quyển sách có độ tuổi phù hợp

```
1
2 Create Trigger assign_db.contain_age_insert
3 before insert on assign_db.contain
4 for each row
5 BEGIN
6     Declare customer_id_of_order int;
7     Declare customer_birthday date ;
8     Declare age INT default 0;
9     Declare book_reading_age int;
10
11    Select customer_id into customer_id_of_order
12    From order_
13    where order_.order_id=new.order_id;
14
15    SELECT birthday into customer_birthday
16    FROM customer
17    WHERE customer_id=customer_id_of_order;
18
19    SELECT TIMESTAMPDIFF(YEAR, customer_birthday, CURDATE()) into age;
20
21    Select reading_age
22    into book_reading_age
23    From book
24    Where book.book_id=new.book_id;
25
26    If (age<book_reading_age) then
27        signal sqlstate '45000' set message_text ='Tuổi của tài khoản không đủ để đọc
28        cuốn sách này';
29    end if;
30 END
```



```
31 Create Trigger assign_db.contain_age_update
32 before update on assign_db.contain
33 for each row
34 BEGIN
35     Declare customer_id_of_order int;
36     Declare customer_birthday date ;
37     Declare age INT default 0;
38     Declare book_reading_age int;
39
40     Select customer_id into customer_id_of_order
41     From order_
42     where order_.order_id=new.order_id;
43     SELECT birthday into customer_birthday
44     FROM customer
45     WHERE customer_id=customer_id_of_order;
46     SELECT TIMESTAMPDIFF(YEAR, customer_birthday , CURDATE()) into age;
47
48     Select reading_age
49     into book_reading_age
50     From book
51     Where book.book_id=new.book_id;
52
53     If (age<book_reading_age) then
54         signal sqlstate '45000' set message_text ='Tuổi của tài khoản không đủ để đọc
55         cuốn sách này';
56     end if;
57
58 END
```

Listing 42: Kiểm tra chỉ cho phép khách hàng mua những quyển sách có độ tuổi phù hợp

4.5.6 Kiểm tra chỉ cho phép đơn hàng chứa những quyển sách cùng nhà cung cấp

```
1 Create Trigger assign_db.contain_same_provider_insert
2 before insert on assign_db.contain
3 for each row
4 BEGIN
5     Declare num_provider int;
6     Select count(Distinct provider_id)
7         into num_provider
8         from contain c1
9         inner join book b1 on c1.book_id=b1.book_id
10        where c1.order_id=new.order_id;
11
12     If (num_provider>=2) then
13         signal sqlstate '45000' set message_text ='Các cuốn sách cần thuộc cùng một
14         nhà cung cấp.';
15     end if;
16
17 | Create Trigger assign_db.contain_same_provider_update
18 before update on assign_db.contain
19 for each row
20 BEGIN
21     Declare num_provider int;
22     Select count(Distinct provider_id)
23         into num_provider
24         from contain c1
25         inner join book b1 on c1.book_id=b1.book_id
26        where c1.order_id=new.order_id;
27
28     If (num_provider>=2) then
29         signal sqlstate '45000' set message_text ='Các cuốn sách cần thuộc cùng một
30         nhà cung cấp.';
31     end if;
32
33 |
```



Listing 43: Kiểm tra chỉ cho phép đơn hàng chứa những quyền sách cùng nhà cung cấp

4.5.7 Kiểm tra chỉ cho phép khách hàng mua những quyền sách điện tử 1 lần, với số lượng cuốn sách bằng 1

```
1 Create trigger assign_db.contain_ebook_buy_1_insert
2 before insert on assign_db.contain
3 for each row
4 BEGIN
5     Declare is_kindle_book INT;
6     Declare is_audio_book INT;
7     Declare customer_id_of_order INT;
8     Declare is_bought INT;
9
10    Select customer_id
11        into customer_id_of_order
12        from order_
13        where order_id=new.order_id;
14
15    Select count(*)
16        into is_bought
17        from order_ o1
18        inner join contain c1 on o1.order_id=c1.order_id
19        where o1.customer_id=customer_id_of_order and c1.book_id=new.book_id;
20
21    Select count(*)
22        into is_kindle_book
23        from kindle_book
24        where book_id=new.book_id;
25
26    Select count(*)
27        into is_audio_book
28        from audio_book
29        where book_id=new.book_id;
30
31    if((is_kindle_book!=0 or is_audio_book!=0) and is_bought >1) then
32        signal sqlstate '45000' set message_text ='Các cuốn sách bản điện tử chỉ được
33            mua một lần.';
34    end if;
35
36    if((is_kindle_book!=0 or is_audio_book!=0) and new.quantity >1) then
37        signal sqlstate '45000' set message_text ='Các cuốn sách bản điện tử chỉ được
38            mua 1 quyền. ';
39    end if;
40 END
41 |
42 Create trigger assign_db.contain_ebook_buy_1_update
43 before update on assign_db.contain
44 for each row
45 BEGIN
46     Declare is_kindle_book INT;
47     Declare is_audio_book INT;
48     Declare customer_id_of_order INT;
49     Declare is_bought INT;
50
51     Select customer_id
52         into customer_id_of_order
53         from order_
54         where order_id=new.order_id;
55
56     Select count(*)
57         into is_bought
58         from order_ o1
59         inner join contain c1 on o1.order_id=c1.order_id
```



```
59      where o1.customer_id=customer_id_of_order and c1.book_id=new.book_id;
60
61      Select count(*)
62      into is_kindle_book
63      from kindle_book
64      where book_id=new.book_id;
65
66      Select count(*)
67      into is_audio_book
68      from audio_book
69      where book_id=new.book_id;
70
71      if((is_kindle_book!=0 or is_audio_book!=0) and is_bought >1) then
72          signal sqlstate '45000' set message_text ='Các cuốn sách bản điện tử chỉ được
73          mua một lần.';
74      end if;
75
76      if((is_kindle_book!=0 or is_audio_book!=0) and new.quantity >1) then
77          signal sqlstate '45000' set message_text ='Các cuốn sách bản điện tử chỉ được
78          mua 1 quyển.';
79      end if;
79 END
```

Listing 44: Kiểm tra chỉ cho phép khách hàng mua một cuốn sách bản điện tử 1 lần, với số lượng cuốn của sách điện tử bằng 1

4.5.8 Kiểm tra khách hàng chỉ có thể mua số lượng sách bé hơn bằng số lượng sách đang được cung cấp

```
1 Create trigger assign_db.contain_book_quantity_insert
2 before insert on assign_db.contain
3 for each row
4 BEGIN
5     Declare book_quantity INT;
6     Select quantity into book_quantity
7     from book
8     where book_id=new.book_id;
9
10    if(book_quantity<new.quantity) then
11        signal sqlstate '45000' set message_text ='Không đủ số lượng sách。';
12    end if;
13 END
14 |
15 Create trigger assign_db.contain_book_quantity_update
16 before update on assign_db.contain
17 for each row
18 BEGIN
19     Declare book_quantity INT;
20     Select quantity into book_quantity
21     from book
22     where book_id=new.book_id;
23
24    if(book_quantity<new.quantity) then
25        signal sqlstate '45000' set message_text ='Không đủ số lượng sách。';
26    end if;
27 END
28 |
```

Listing 45: Kiểm tra khách hàng chỉ có thể mua số lượng sách bé hơn bằng số lượng sách đang được cung cấp

4.5.9 Kiểm tra khách hàng chỉ có thể sử dụng mã giảm giá còn hạn và còn số lượt sử dụng



```
1 Create trigger assign_db.apply_for_insert_check_date_and_quantity
2 before insert on assign_db.apply_for
3 for each row
4 Begin
5   Declare customer_apply_id int;
6   Declare used_quantity int;
7   Declare time_of_order datetime;
8   Declare start_date_of_promotion datetime;
9   Declare end_date_of_promotion datetime;
10  Declare initial_quantity int;
11
12  Select customer_id into customer_apply_id
13    from order_
14   where order_id=new.order_id;
15
16  Select order_time into time_of_order
17    from order_
18   where order_id=new.order_id;
19
20  Select start_date into start_date_of_promotion
21    from promotion_code
22   where code_id=new.promotion_code_id;
23
24  Select init_quantity into initial_quantity
25    from promotion_code
26   where code_id=new.promotion_code_id;
27
28  Select end_date into end_date_of_promotion
29    from promotion_code
30   where code_id=new.promotion_code_id;
31
32  Select count(*)
33    into used_quantity
34    from apply_for a1
35   inner join order_o1 on a1.order_id=o1.order_id
36   where a1.promotion_code_id=new.promotion_code_id and o1.customer_id =
customer_apply_id;
37
38  If time_of_order <start_date_of_promotion then
39    signal sqlstate '45000' set message_text ='Mã giảm giá này chưa đến thời
gian sử dụng.';
40  end if;
41
42  If time_of_order >end_date_of_promotion then
43    signal sqlstate '45000' set message_text='Mã giảm giá này đã quá thời gian
sử dụng.';
44  end if;
45
46  if initial_quantity-used_quantity <=0 then
47    signal sqlstate '45000' set message_text="Hết hạn sử dụng mã giảm giá";
48  end if;
49 END
50 |
51 Create trigger assign_db.apply_for_update_check_date_and_quantity
52 before update on assign_db.apply_for
53 for each row
54 Begin
55   Declare customer_apply_id int;
56   Declare used_quantity int;
57   Declare time_of_order datetime;
58   Declare start_date_of_promotion datetime;
59   Declare end_date_of_promotion datetime;
60   Declare initial_quantity int;
61
62   Select customer_id into customer_apply_id
63     from order_
64   where order_id=new.order_id;
65
66   Select order_time into time_of_order
67     from order_
68   where order_id=new.order_id;
```



```
69      Select start_date into start_date_of_promotion
70          from promotion_code
71          where code_id=new.promotion_code_id;
72
73      Select init_quantity into initial_quantity
74          from promotion_code
75          where code_id=new.promotion_code_id;
76
77      Select end_date into end_date_of_promotion
78          from promotion_code
79          where code_id=new.promotion_code_id;
80
81      Select count(*)
82          into used_quantity
83          from apply_for a1
84          inner join order_ o1 on a1.order_id=o1.order_id
85          where a1.promotion_code_id=new.promotion_code_id and o1.customer_id =
86              customer_apply_id;
87
88      If time_of_order <start_date_of_promotion then
89          signal sqlstate '45000' set message_text ='Mã giảm giá này chưa đến thời
90          gian sử dụng.';
91      end if;
92      If time_of_order >end_date_of_promotion then
93          signal sqlstate '45000' set message_text='Mã giảm giá này đã quá thời gian
94          sử dụng.';
95      end if;
96
97      if initial_quantity-used_quantity <=0 then
98          signal sqlstate '45000' set message_text="Hết hạn sử dụng mã giảm giá";
99      end if;
100 END
```

Listing 46: Kiểm tra khách hàng chỉ có thể sử dụng mã còn hạn và còn số lượt sử dụng

4.5.10 Kiểm tra hạng thành viên của khách hàng khi sử dụng mã giảm giá

```
1 Create Trigger assign_db.apply_for_check_membership_insert
2 before insert on assign_db.apply_for
3 for each row
4 BEGIN
5     Declare membership_of_promo ENUM ("Đồng", "Bạc", "Vàng", "Kim cương");
6     declare total_spent_cus Int default 0;
7
8     select total_spent into total_spent_cus
9         from adult
10        join confirm on confirm.adult_id=adult.customer_id
11        where confirm.order_id=new.order_id;
12
13    Select membership into membership_of_promo
14        from promotion_code
15        where code_id=new.promotion_code_id;
16
17
18    if(membership_of_promo="Bạc" and total_spent_cus<1000000) then
19        signal sqlstate '45000' set message_text='Bạn phải là thành viên Bạc trở lên
20        mới sử dụng được mã giảm giá này。';
21    end if;
22
23    if(membership_of_promo="Vàng" and total_spent_cus<5000000) then
24        signal sqlstate '45000' set message_text='Bạn phải là thành viên Vàng trở lên
25        mới sử dụng được mã giảm giá này。';
26
27    if(membership_of_promo="Kim cương" and total_spent_cus<10000000) then
```



```
28     signal sqlstate '45000' set message_text='Bạn phải là thành viên Kim cương trả
29     ở lên mới sử dụng được mã giảm giá này.';
30   end if;
31 |
32 Create Trigger assign_db.apply_for_check_membership_update
33 before update on assign_db.apply_for
34 for each row
35 BEGIN
36   Declare membership_of_promo ENUM ("Đồng", "Bạc", "Vàng", "Kim cương");
37   declare total_spent_cus Int default 0;
38 |
39   select total_spent into total_spent_cus
40   from adult
41   join confirm on confirm.adult_id=adult.customer_id
42   where confirm.order_id=new.order_id;
43 |
44   Select membership into membership_of_promo
45   from promotion_code
46   where code_id=new.promotion_code_id;
47 |
48 |
49 |
50   if(membership_of_promo="Bạc" and total_spent_cus<1000000) then
51     signal sqlstate '45000' set message_text='Bạn phải là thành viên Bạc trả lên
52     mới sử dụng được mã giảm giá này.';
53   end if;
54 |
55   if(membership_of_promo="Vàng" and total_spent_cus<5000000) then
56     signal sqlstate '45000' set message_text='Bạn phải là thành viên Vàng trả lên
57     mới sử dụng được mã giảm giá này.';
58   end if;
59 |
60   if(membership_of_promo="Kim cương" and total_spent_cus<10000000) then
61     signal sqlstate '45000' set message_text='Bạn phải là thành viên Kim cương trả
62     ở lên mới sử dụng được mã giảm giá này.';
```

Listing 47: Kiểm tra hạng thành viên của khách hàng khi sử dụng mã giảm giá

4.5.11 Kiểm tra khách hàng chỉ có thể sử dụng mã giảm giá thỏa giá trị tối thiểu

```
1 Create Trigger assign_db.apply_for_check_minimum_order_insert
2 before insert on assign_db.apply_for
3 for each row
4 BEGIN
5   Declare price_without_promo int;
6   Declare minimum_require int;
7   Select min_order into minimum_require
8   from promotion_code
9   where code_id=new.promotion_code_id;
10 |
11   set price_without_promo=cal_price_order_to_check_min_promo__(new.order_id);
12 |
13   if(price_without_promo<minimum_require) then
14     signal sqlstate '45000' set message_text='Đơn hàng chưa đủ giá trị tối thiểu
15     để sử dụng mã giảm giá này.';
16   end if;
17 |
18 Create Trigger assign_db.apply_for_check_minimum_order_update
19 before update on assign_db.apply_for
20 for each row
21 BEGIN
```



```
23 Declare price_without_promo int;
24     Declare minimum_require int;
25     Select min_order into minimum_require
26     from promotion_code
27     where code_id=new.promotion_code_id;
28
29     set price_without_promo=cal_price_order_to_check_min_promo__(new.order_id);
30
31     if(price_without_promo<minimum_require) then
32         signal sqlstate '45000' set message_text='Đơn hàng chưa đủ giá trị tối thiểu
để sử dụng mã giảm giá này.';
33     end if;
34 END;
35 |
36 DELIMITER ;
```

Listing 48: Kiểm tra khách hàng chỉ có thể sử dụng mã giảm giá thỏa điều kiện giá trị đơn tối thiểu

Trong đó, ràng buộc vừa rồi có gọi một hàm tính tổng tiền của đơn hàng theo id chưa bao gồm mã khuyến mãi là:

```
1  create function cal_price_order_to_check_min_promo__(a_order_id int) returns int
2      READS SQL DATA
3  begin
4      Declare date_of_order datetime;
5      Declare result int;
6
7      Select order_time into date_of_order
8      from order_
9      where order_.order_id=a_order_id;
10
11     with discount_in_time AS (
12         select discount_id, discount_value
13         from discount
14         where start_date<=date_of_order and date_of_order<=end_date),
15
16     book_and_quantity AS(
17         Select contain.book_id, contain.quantity, book.price
18         from contain
19         join book on contain.book_id=book.book_id
20         where order_id=a_order_id),
21
22     book_and_discount AS(
23         select book_and_quantity.book_id, book_and_quantity.quantity,
24             book_and_quantity.price, have_.discount_id
25         from book_and_quantity
26         left join have_ on book_and_quantity.book_id=have_.book_id),
27
28     book_quantity_discount_value AS(
29         Select book_id,quantity, price, discount_in_time.discount_value
30         from book_and_discount
31         left join discount_in_time on book_and_discount.discount_id=discount_in_time.
32         discount_id),
33
34     price_check_of_order As(
35
36     SELECT
37     SUM(subquery.avg_price * subquery.avg_quantity) AS item_total,
38     SUM(subquery.avg_price *subquery.avg_quantity* subquery.max_discount_value/100)
39     AS discount_total,
40     SUM(subquery.avg_price *subquery.avg_quantity* (100-subquery.max_discount_value)
41     /100) as grand_total
42
43     FROM
44     (
45         SELECT
46             AVG(book_id) AS avg_book_id,
47             AVG(quantity) AS avg_quantity,
48             AVG(price) AS avg_price,
49             COALESCE(MAX(discount_value),0) AS max_discount_value
50
51             FROM
52             book_quantity_discount_value
```



```
45     GROUP BY
46         book_id
47     )As subquery
48 )
49
50     Select grand_total into result
51     from price_check_of_order;
52
53     return result;
54 end
55 |
```

Listing 49: Hàm tính tổng tiền của đơn hàng chưa bao gồm mã khuyến mãi

4.6 Các trigger, hàm và thủ tục

4.6.1 Câu 1.2.1: Viết các thủ tục để thêm (insert), sửa (update) và xóa (delete) dữ liệu vào một bảng dữ liệu

Nhóm chọn viết các thủ tục thêm, sửa, xóa dữ liệu vào bảng book.

Đầu tiên là procedure **add_book** dùng để thêm sách. Trong đó procedure sẽ thực hiện các việc sau đây:

- Kiểm tra đảm bảo rằng các thuộc tính của cuốn sách phù hợp, như tên sách (title) không được để trống, độ tuổi đọc sách (reading_age) phải lớn hơn 0, giá tiền (price) của cuốn sách phải lớn hơn 0...
- Nếu cả hai điều kiện trên đều thỏa, sách sẽ được xóa cập nhật các giá trị thuộc tính mới tương ứng, còn không, thông báo lỗi tương ứng sẽ được hiện ra.

```
1 create procedure add_book (
2     in title varchar(255),
3     in reading_age int,
4     in price double,
5     in language_ varchar(255),
6     in edition varchar(255),
7     in publication_date DATE,
8     in publisher_name varchar(255),
9     in isbn varchar(13),
10    in provider_id int,
11    in quantity int,
12    out return_book_id int
13 )
14 begin
15     if title is null then
16         signal sqlstate '45000' set message_text ='Tên của sách không được để trống';
17     end if;
18     if reading_age is null then
19         signal sqlstate '45000' set message_text ='Giới hạn tuổi của sách không được để trống';
20     end if;
21     if price is null then
22         signal sqlstate '45000' set message_text ='Giá tiền của sách không được để trống';
23     end if;
24     if language_ is null then
25         signal sqlstate '45000' set message_text ='Ngôn ngữ của sách không được để trống';
26     end if;
27     if edition is null then
28         signal sqlstate '45000' set message_text ='Phiên bản của sách không được để trống';
29     end if;
30     if publication_date is null then
```



```
31      signal sqlstate '45000' set message_text ='Ngày phát hành của sách không được
32          để trống';
33      end if;
34      if publisher_name is null then
35          signal sqlstate '45000' set message_text ='Tên nhà phát hành của sách không đ
36          ược để trống';
37          end if;
38      if isbn is null then
39          signal sqlstate '45000' set message_text ='Mã ISBN của sách không được để tr
40          ng';
41          end if;
42      if quantity is null then
43          signal sqlstate '45000' set message_text ='Số lượng cuốn sách không được để
44          trống';
45          end if;
46
47      if (reading_age <= 0) then
48          signal sqlstate '45000' set message_text ='Độ tuổi giới hạn đọc sách phải lớn
49          hơn 0';
50      end if;
51      if (price <= 0) then
52          signal sqlstate '45000' set message_text ='Giá bán của sách phải lớn hơn 0';
53      end if;
54      if not (isbn regexp '^[0-9]{13}') then
55          signal sqlstate '45000' set message_text ='Mã ISBN của quyển sách phải là 13
56          ký tự chữ số';
57      end if;
58      if (provider_id <= 0) then
59          signal sqlstate '45000' set message_text ='ID của nhà cung cấp sách phải lớn
60          hơn 0';
61      end if;
62
63      insert into book value (NULL, title, reading_age, price, language_, edition,
64      publication_date, publisher_name, isbn, provider_id, quantity);
65  SELECT LAST_INSERT_ID() AS return_book_id;
66  end;
67 |
```

Listing 50: Thủ tục thêm dữ liệu vào bảng book và các bảng liên quan

Tiếp theo là procedure **update_book** dùng để cập nhật sách. Trong đó procedure sẽ thực hiện các việc sau đây:

- Kiểm tra đảm bảo rằng id cuốn sách được cập nhật là hợp lệ và có tồn tại trong cơ sở dữ liệu.
- Kiểm tra đảm bảo rằng các thuộc tính của cuốn sách phù hợp như tên sách (title) không được để trống, độ tuổi đọc sách (reading_age) phải lớn hơn 0, giá tiền (price) của cuốn sách phải lớn hơn 0,...
- Nếu cả hai điều kiện trên đều thỏa, sách sẽ được xóa cập nhật các giá trị thuộc tính mới tương ứng, còn không, thông báo lỗi tương ứng sẽ được hiện ra.

```
1  create procedure update_book (
2    in inbook_id int,
3    in title varchar(255),
4    in reading_age int,
5    in price double,
6    in language_ varchar(255),
7    in edition varchar(255),
8    in publication_date DATE,
9    in publisher_name varchar(255),
```



```
10     in isbn varchar(13),
11     in quantity int
12 )
13 begin
14     if inbook_id is null then
15         signal sqlstate '45000' set message_text = 'ID của sách không được để trống';
16     end if;
17     if title is null then
18         signal sqlstate '45000' set message_text = 'Tên của sách không được để trống';
19     end if;
20     if reading_age is null then
21         signal sqlstate '45000' set message_text = 'Giới hạn tuổi của sách không được để trống';
22     end if;
23     if price is null then
24         signal sqlstate '45000' set message_text = 'Giá tiền của sách không được để trống';
25     end if;
26     if language_ is null then
27         signal sqlstate '45000' set message_text = 'Ngôn ngữ của sách không được để trống';
28     end if;
29     if edition is null then
30         signal sqlstate '45000' set message_text = 'Phiên bản của sách không được để trống';
31     end if;
32     if publication_date is null then
33         signal sqlstate '45000' set message_text = 'Ngày phát hành của sách không được để trống';
34     end if;
35     if publisher_name is null then
36         signal sqlstate '45000' set message_text = 'Tên nhà phát hành của sách không được để trống';
37     end if;
38     if isbn is null then
39         signal sqlstate '45000' set message_text = 'Mã ISBN của sách không được để trống';
40     end if;
41     if quantity is null then
42         signal sqlstate '45000' set message_text = 'Số lượng cuốn sách không được để trống';
43     end if;
44     if (inbook_id <= 0) then
45         signal sqlstate '45000' set message_text = 'ID của sách phải lớn hơn 0';
46     end if;
47     if not (select exists (select * from book where book.book_id = inbook_id))
48     then
49         signal sqlstate '45000' set message_text = 'ID của sách không tồn tại';
50     end if;
51     if (reading_age <= 0) then
52         signal sqlstate '45000' set message_text = 'Độ tuổi giới hạn đọc sách phải lớn hơn 0';
53     end if;
54     if (price <= 0) then
55         signal sqlstate '45000' set message_text = 'Giá bán của sách phải lớn hơn 0';
56     end if;
57     if not (isbn regexp '^[0-9]{13}') then
58         signal sqlstate '45000' set message_text = 'Mã ISBN của quyển sách phải là 13 ký tự chữ số';
59     end if;
60     if (quantity < 0) then
61         signal sqlstate '45000' set message_text = 'Số lượng sách không thể là số âm';
62     end if;
63
64     update book set
65     book.title = title,
66     book.reading_age = reading_age,
67     book.price = price,
68     book.language_ = language_,
69     book.edition = edition,
       book.publication_date = publication_date,
```



```
70     book.publisher_name = publisher_name,
71     book.isbn = isbn,
72     book.quantity = quantity
73     where book.book_id = inbook_id;
74 end;
75 |
```

Listing 51: Thủ tục sửa dữ liệu trong bảng book

Tiếp theo là procedure **delete_book**. Trong đó procedure sẽ thực hiện các việc sau đây:

- Kiểm tra đảm bảo rằng id cuốn sách được xóa là hợp lệ và có tồn tại trong cơ sở dữ liệu.
- Kiểm tra đảm bảo rằng cuốn sách chưa từng được đặt mua.
- Nếu cả hai điều kiện trên đều thỏa, sách sẽ được xóa khỏi cơ sở dữ liệu, nếu không, thông báo lỗi tương ứng cho từng điều kiện kiểm tra sẽ hiện ra.

```
1 create procedure delete_book(in book_id int)
2 begin
3   Declare is_book_buy int default 0;
4   Select count(*) into is_book_buy from contain where contain.book_id=book_id;
5   if(book_id is null) then
6     signal sqlstate '45000' set message_text ='ID của sách không được để trống';
7   end if;
8   if(is_book_buy!=0) then
9     signal sqlstate '45000' set message_text ='Không thể xóa sách đã có đơn mua';
10    ;
11   end if;
12   if (book_id <= 0) then
13     signal sqlstate '45000' set message_text ='ID của sách phải lớn hơn 0';
14   end if;
15   if not (select exists (select * from book where book.book_id = book_id)) then
16     signal sqlstate '45000' set message_text ='ID của sách không tồn tại hoặc sách đã xóa từ trước.';
17   end if;
18   delete from book where book.book_id = book_id;
19
20 end;
```

Listing 52: Thủ tục xóa dữ liệu trong bảng book

4.6.2 Câu 1.2.2: Viết 2 trigger để kiểm soát các hành động INSERT, UPDATE, DELETE trên một số bảng đã tạo

Trigger đầu tiên mà nhóm chọn là để kiểm soát hành động INSERT, UPDATE, DELETE trên bảng customer.

Trước khi thêm một khách hàng mới vào bảng customer, trigger sẽ kiểm tra tính hợp lệ của mã khách hàng (customer_id), giới tính (gender) và ngày sinh (birthday).

```
1 create trigger add_customer_before
2   before insert on customer
3   for each row
4   begin
5     declare cur date;
6     set cur = curdate();
7
8     if(new.customer_id < 0) then
9       signal sqlstate '45000' set message_text ='ID người dùng không hợp lệ';
10      ;
11    end if;
```



```
11      if not (new.gender = 'F' or new.gender='M' or new.gender='O') then
12          signal sqlstate '45000' set message_text ='Giới tính chỉ có thể là
13          Nam (M), Nữ (F) hoặc khác (0)';
14      end if;
15
16      if(new.birthday >= cur) then
17          signal sqlstate '45000' set message_text ='Ngày sinh không hợp lệ';
18      end if;
19  end;
```

Listing 53: Trigger kiểm tra trước khi thêm mới khách hàng

Sau khi đã thêm khách hàng mới, trigger tiếp theo sẽ kiểm tra xem khách hàng này đã đủ 16 tuổi hay chưa, từ đó sẽ lựa chọn thêm khách hàng vào bảng adult hoặc child.

- Nếu khách hàng đã đủ 16 tuổi, thêm khách hàng vào bảng adult với số điện thoại chưa xác định ('0000000000') và tổng tiền mua sách bằng 0.
- Ngược lại, thêm khách hàng vào bảng child với mã người giám sát (guardian_id) tạm thời là mã nhân viên của hệ thống (99).

```
1  create function isAdult(customer_id int)
2      returns bool
3      deterministic
4      begin
5          declare cur_date date;
6          declare birth date;
7          declare age int;
8          declare result bool;
9
10         if(customer_id <= 0) then
11             signal sqlstate '45000' set message_text ='Mã khách hàng không hợp lệ';
12         end if;
13
14         if not (select exists (select * from customer where customer.customer_id
15 = customer_id)) then
16             signal sqlstate '45000' set message_text ='Mã khách hàng không tồn tại';
17         end if;
18
19         set cur_date = curdate();
20         select birthday into birth from customer where customer.customer_id =
21         customer_id;
22         set age = timestampdiff(year, birth, cur_date);
23
24         if (age <= 0) then
25             signal sqlstate '45000' set message_text ='Ngày sinh không hợp lệ';
26         end if;
27
28         if (age < 16) then
29             set result = false;
30         else set result = true;
31         end if;
32
33         return result;
34     end;
35
36     create trigger add_customer_after
37         after insert on customer
38         for each row
39         begin
40             if(isAdult(new.customer_id)) then
41                 insert into adult value(new.customer_id, '0000000000', 0);
42             else insert into child value(new.customer_id, 99);
43             end if;
44         end;
```

Listing 54: Trigger kiểm tra sau khi thêm mới khách hàng



Trước khi cập nhật thông tin của khách hàng, trigger sẽ kiểm tra giới tính (gender) và ngày sinh (birthday) mới có hợp lệ hay không.

```
1 create trigger update_customer_before
2   before update on customer
3   for each row
4   begin
5     declare cur date;
6     set cur = curdate();
7
8     if(new.customer_id <= 0) then
9       signal sqlstate '45000' set message_text ='ID người dùng không hợp lệ';
10    end if;
11
12    if not (new.gender ='F' or new.gender='M' or new.gender='O') then
13      signal sqlstate '45000' set message_text ='Giới tính chỉ có thể là
14      Nam (M), Nữ (F) hoặc khác (O)';
15    end if;
16
17    if(new.birthday >= cur) then
18      signal sqlstate '45000' set message_text ='Ngày sinh không hợp lệ';
19    end if;
20  end;
```

Listing 55: Trigger kiểm tra dữ liệu khách hàng trước khi cập nhật

Sau khi cập nhật thông tin khách hàng, trigger sẽ kiểm tra ngày sinh mới, nếu ngày sinh này làm khách hàng chuyển từ người lớn sang trẻ em thì cập nhật lại 2 bảng adult và child.

```
1 create trigger update_customer_after
2   after update on customer
3   for each row
4   begin
5     if(select isAdult(new.customer_id)) then
6       if not (select exists (select * from adult where adult.customer_id =
new.customer_id)) then
7         delete from child where child.customer_id = new.customer_id;
8         insert into adult value (new.customer_id, '0000000000', 0);
9       end if;
10     else
11       if not (select exists (select * from child where child.customer_id =
new.customer_id)) then
12         delete from adult where adult.customer_id = new.customer_id;
13         insert into child value (new.customer_id, 99);
14       end if;
15     end if;
16   end;
```

Listing 56: Trigger kiểm tra dữ liệu khách hàng sau khi cập nhật

Việc xóa người dùng sẽ được thực hiện khi người dùng không còn liên quan đến bất kỳ đối tượng nào khác (là khóa ngoại của một bảng khác), các trường hợp còn lại sẽ không được phép xóa người dùng.

Trigger tiếp theo nhóm lựa chọn là kiểm soát hành động INSERT, UPDATE, DELETE trên bảng rate.

Trước khi thêm một đánh giá của người dùng về một quyển sách, trigger sẽ kiểm tra xem mã người dùng và mã của sách đã hợp lệ chưa, người dùng và sách này có tồn tại trong cơ sở dữ liệu hay không, kiểm tra người dùng đã mua sách này chưa, cuối cùng sẽ kiểm tra điểm đánh giá có hợp lệ hay không.

```
1 Create Trigger assign_db.rate_only_bought_insert
2 before insert on assign_db.rate
3 for each row
4 BEGIN
5   Declare is_bought_book INT default 0;
6
7   if (new.book_id <= 0) then
8     signal sqlstate '45000' set message_text ='Mã sách không hợp lệ';
```



```
9    end if;
10
11   if (new.adult_id <= 0) then
12     signal sqlstate '45000' set message_text ='Mã khách hàng không hợp lệ';
13   end if;
14
15   if not (select exists (select * from book where book.book_id = book_id)) then
16     signal sqlstate '45000' set message_text ='Sách không tồn tại';
17   end if;
18
19   if not (select exists (select * from adult where adult.customer_id =
20     customer_id)) then
21     signal sqlstate '45000' set message_text ='Người dùng không tồn tại';
22   end if;
23
24   With book_bought AS
25     (SELECT book_id, adult_id
26      FROM confirm
27      Inner join contain on confirm.order_id=contain.order_id
28        Where book_id=new.book_id and adult_id=new.adult_id)
29
30     SELECT Count(*)
31       INTO is_bought_book
32      FROM book_bought;
33
34     if (is_bought_book<=0) then
35       signal sqlstate '45000' set message_text ='Người dùng phải mua mới được quyền
36       đánh giá điểm cuốn sách';
37     end if;
38   END;
```

Listing 57: Thủ tục thêm dữ liệu vào bảng book

Trước khi cập nhật dữ liệu trên bảng rate, trigger sẽ kiểm tra xem điểm đánh giá muôn cập nhật có hợp lệ hay không, sau đó sẽ chỉnh sửa vào bảng rate.

```
1 Create Trigger assign_db.rate_only_bought_update
2 before update on assign_db.rate
3 for each row
4 BEGIN
5   Declare is_bought_book INT default 0;
6
7   if (new.book_id <= 0) then
8     signal sqlstate '45000' set message_text ='Mã sách không hợp lệ';
9   end if;
10
11  if (new.adult_id <= 0) then
12    signal sqlstate '45000' set message_text ='Mã khách hàng không hợp lệ';
13  end if;
14
15  if not (select exists (select * from book where book.book_id = book_id)) then
16    signal sqlstate '45000' set message_text ='Sách không tồn tại';
17  end if;
18
19  if not (select exists (select * from adult where adult.customer_id =
20    customer_id)) then
21    signal sqlstate '45000' set message_text ='Người dùng không tồn tại';
22  end if;
23
24  With book_bought AS
25    (SELECT book_id, adult_id
26      FROM confirm
27      Inner join contain on confirm.order_id=contain.order_id
28        Where book_id=new.book_id and adult_id=new.adult_id)
29
30    SELECT Count(*)
31      INTO is_bought_book
32     FROM book_bought;
33
34    if (is_bought_book<=0) then
```



```
34      signal sqlstate '45000' set message_text ='Người dùng phải mua mới được quyền
35          đánh giá điểm cuốn sách';
36      end if;
37  END;
```

Listing 58: Thủ tục thêm dữ liệu vào bảng book

4.6.3 Câu 1.2.3: Viết 2 thủ tục trong đó chỉ chứa các câu truy vấn để hiển thị dữ liệu và tham số đầu vào là giá trị trong mệnh đề WHERE và/hoặc HAVING

Thủ tục đầu tiên mà nhóm chọn là lọc sách theo thể loại, giá, tên sách và thứ tự sắp xếp.

```
1  create procedure filter_book(in genres varchar(255), in price_range varchar(255),
2                               in order_by varchar(255), in book_name varchar(255))
3  begin
4      declare cur_date date;
5      set cur_date = curdate();
6
7      create table book_with_discount as select book.book_id, title, edition,
8          price, have_.discount_id, discount_value from
9          book join have_ on book.book_id = have_.book_id
10     join discount on have_.discount_id = discount.discount_id where start_date <=
11         cur_date and end_date >= cur_date;
12
13     create table book_discount as select book.book_id, book.title, book.
14         edition, book.price, max(discount_value) as max_discount from
15         book left join book_with_discount on book.book_id = book_with_discount.
16         book_id group by book.book_id;
17
18     set SQL_SAFE_UPDATES = 0;
19     update book_discount set max_discount = 0 where max_discount is null;
20     alter table book_discount add column end_price double;
21     update book_discount set end_price = price * (100 - max_discount) / 100;
22
23     if genres is not null then
24         create temporary table filter_genre as select book_discount.book_id, title,
25             edition, price, max_discount, end_price from
26             book_discount join genres_book on book_discount.book_id = genres_book
27             .book_id
28             where genres_book.genres = genres;
29     else
30         create temporary table filter_genre as
31             select book_id, title, edition, price, max_discount, end_price from
32             book_discount;
33     end if;
34
35     if price_range is not null then
36         if (price_range <=> 'low') then
37             create temporary table filter_price as select * from filter_genre
38             where filter_genre.price < 50000;
39         elseif (price_range <=> 'mid') then
40             create temporary table filter_price as select * from filter_genre
41             where filter_genre.price >= 50000 and filter_genre.price < 100000;
42         elseif (price_range <=> 'high') then
43             create temporary table filter_price as select * from filter_genre
44             where filter_genre.price >= 100000;
45         end if;
46     else create temporary table filter_price as select * from filter_genre;
47     end if;
48
49     if(book_name is not null) then
50         if order_by is not null then
51             if (order_by <=> 'titleasc') then
52                 select * from filter_price where title = book_name order by
53                     title asc;
54             elseif (order_by <=> 'titledesc') then
55
```



```
43           select * from filter_price where title = book_name order by
44           title desc;
45           elseif (order_by <=> 'priceasc') then
46               select * from filter_price where title = book_name order by
47               price asc;
48               elseif (order_by <=> 'pricedesc') then
49                   select * from filter_price where title = book_name order by
50                   price desc;
51                   end if;
52           else select * from filter_price where title = book_name;
53           end if;
54       else
55           if order_by is not null then
56               if (order_by <=> 'titleasc') then
57                   select * from filter_price order by title asc;
58               elseif (order_by <=> 'titledesc') then
59                   select * from filter_price order by title desc;
60               elseif (order_by <=> 'priceasc') then
61                   select * from filter_price order by price asc;
62               elseif (order_by <=> 'pricedesc') then
63                   select * from filter_price order by price desc;
64               end if;
65           else select * from filter_price;
66           end if;
67       end if;
68       drop table book_with_discount;
69       drop table book_discount;
70       drop table filter_genre;
71   drop table filter_price;
72 end;
```

Listing 59: Thủ tục lọc sách theo các tiêu chí riêng

Thủ tục thứ hai mà nhóm chọn là hiển thị những tác giả yêu thích của một người dùng xét theo số lượng sách đã mua (trên 1 quyển hoặc trên 5 quyển).

```
1  create procedure bought_book(in customer_id int, in limit_quantity int)
2  begin
3      if not (select exists (select * from order_ where order_.customer_id =
4          customer_id)) then
5          signal sqlstate '45000' set message_text ='Không có đơn hàng nào';
6      end if;
7
8      create table bought_table as select contain.book_id, contain.quantity,
9          title, penname from
10         order_ join contain on order_.order_id = contain.order_id
11         join book on contain.book_id = book.book_id
12         join write_ on book.book_id = write_.book_id
13         join author on write_.author_id = author.author_id where order_.
14         customer_id = customer_id and order_.status_ = "Hoàn tất";
15
16         create table bought_author as select sum(quantity) as total_quantity,
17             penname from bought_table group by penname;
18
19         create table result as select bought_author.penname, title, sum(quantity)
20             as total_quantity from
21             bought_table join bought_author on bought_table.penname = bought_author.
22             penname
23             where bought_author.total_quantity >= limit_quantity
24                 group by bought_author.penname, title
25                 order by penname asc, total_quantity desc;
26
27         select * from result;
28
29         drop table bought_table;
30         drop table bought_author;
31         drop table result;
32     end;
```



Listing 60: Thủ tục lọc ra các tác giả yêu thích

4.6.4 Câu 1.2.4 Viết 2 hàm có sử dụng câu lệnh IF, có truy vấn dữ liệu và kiểm tra tham số đầu vào

Hàm đầu tiên mà nhóm thực hiện là tính điểm đánh giá trung bình của một quyển sách bằng cách truyền vào mã của nó. Hàm sẽ kiểm tra xem mã sách có hợp lệ hay không, sách này có tồn tại không và sách này đã có đánh giá nào chưa, sau đó sẽ truy cập dữ liệu từ bảng rate để tính toán điểm trung bình của quyển sách đó.

```
1  create function calRating(book_id int)
2      returns double
3      deterministic
4      begin
5          declare avg_score double;
6
7          if(book_id <= 0) then
8              signal sqlstate '45000' set message_text ='Mã sách không hợp lệ';
9          end if;
10
11         if not (select exists (select * from book where book.book_id = book_id))
12             then
13                 signal sqlstate '45000' set message_text ='Sách không tồn tại';
14             end if;
15
16         if not (select exists (select * from rate where rate.book_id = book_id))
17             then
18                 signal sqlstate '45000' set message_text ='Sách chưa có điểm đánh giá';
19             end if;
20
21         select avg(score) into avg_score from rate where rate.book_id = book_id
22             group by book_id;
22
22         return avg_score;
23     end;
```

Listing 61: Hàm tính điểm đánh giá trung bình của một quyển sách

Hàm tiếp theo nhóm thực hiện là tính tổng tiền của một đơn hàng khi đã áp dụng promotion code và discount. Hàm sẽ dựa vào thời gian đặt hàng để xác nhận các mã giảm giá và đợt giảm giá của sách còn hiệu lực hay không. Sau đó tìm giá trị lớn nhất của mỗi loại giảm giá để áp dụng vào đơn hàng. Tính tổng tiền các quyển sách khi đã áp dụng đợt giảm giá, sau đó dùng mã giảm giá để giảm tiền của cả đơn hàng, đưa ra kết quả cuối cùng là giá tiền của đơn hàng đó.

```
1  create function cal_price_order__(a_order_id int) returns int READS SQL DATA
2 begin
3     Declare date_of_order datetime;
4     Declare code_promo_value int;
5     Declare code_max_promo_value int;
6     Declare temp_discount int;
7     Declare grand_total_of_order int;
8     Declare item_total_value int;
9     Declare discount_total_value int;
10
11    select COALESCE(promo_value,0) into code_promo_value
12        from promotion_code
13        join apply_for on apply_for.promotion_code_id=promotion_code.code_id
14        where apply_for.order_id=a_order_id;
15
16    select COALESCE(maximum_promo,0) into code_max_promo_value
17        from promotion_code
18        join apply_for on apply_for.promotion_code_id=promotion_code.code_id
19        where apply_for.order_id=a_order_id;
```



```
20
21
22
23     Select order_time into date_of_order
24     from order_
25     where order_.order_id=a_order_id;
26
27     with discount_in_time AS(
28         select discount_id, discount_value
29         from discount
30         where start_date<=date_of_order and date_of_order<=end_date),
31         book_and_quantity AS(
32             Select contain.book_id, contain.quantity, book.price
33             from contain
34             join book on contain.book_id=book.book_id
35             where order_id=a_order_id),
36
37         book_and_discount AS(
38             select book_and_quantity.book_id, book_and_quantity.quantity,
39             book_and_quantity.price, have_.discount_id
40             from book_and_quantity
41             left join have_ on book_and_quantity.book_id=have_.book_id),
42
43         book_quantity_discount_value AS(
44             Select book_id,quantity, price, discount_in_time.discount_value
45             from book_and_discount
46             left join discount_in_time on book_and_discount.discount_id=discount_in_time.
47             discount_id),
48             price_of_order AS(
49             SELECT
50                 SUM(subquery.avg_price * subquery.avg_quantity) AS item_total ,
51                 SUM(subquery.avg_price *subquery.avg_quantity* subquery.
52                 max_discount_value/100) AS discount_total ,
53                 SUM(subquery.avg_price *subquery.avg_quantity* (100-subquery.
54                 max_discount_value)/100) AS grand_total
55             FROM
56             (
57                 SELECT
58                     AVG(book_id) AS avg_book_id,
59                     AVG(quantity) AS avg_quantity ,
60                     AVG(price) AS avg_price,
61                     COALESCE(MAX(discount_value),0) AS max_discount_value
62                 FROM
63                 book_quantity_discount_value
64                 GROUP BY
65                     book_id
66             )As subquery
67         )
68
69
70     Select discount_total,item_total,grand_total into temp_discount ,
71     item_total_value , grand_total_of_order
72     from price_of_order;
73
74
75     return grand_total_of_order;
76
77 end;
```

Listing 62: Hàm tính tổng tiền của một đơn hàng

5 Đặc tả ứng dụng

5.1 Đặc tả kỹ thuật

Front-end: React.js

React.js là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng động. Với khả năng tạo các thành phần tái sử dụng, React.js giúp tối ưu hóa quá trình phát triển front-end và cung cấp trải nghiệm người dùng tốt.

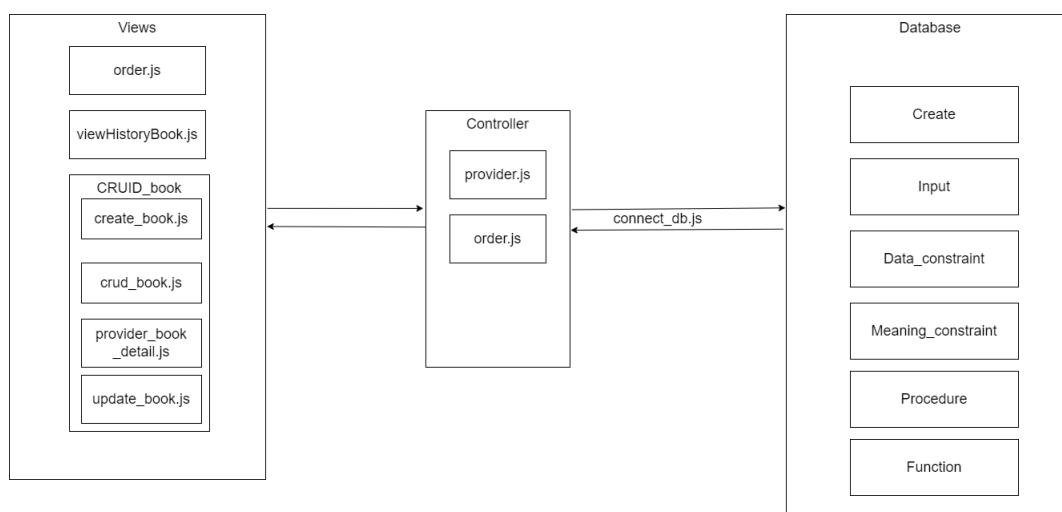
Back-end: Node.js và Express.js

Node.js là một môi trường thực thi JavaScript nhanh chóng và hiệu quả, thường được sử dụng cho phát triển server-side. Express.js là một framework Node.js nhẹ giúp xây dựng các ứng dụng web một cách dễ dàng và nhanh chóng thông qua việc cung cấp các tính năng cơ bản như routing và middleware.

Cơ Sở Dữ Liệu: MySQL qua XAMPP

MySQL là hệ quản trị cơ sở dữ liệu phổ biến được sử dụng để lưu trữ và quản lý dữ liệu. XAMPP là một bản phân phối chứa các thành phần cần thiết như Apache, MySQL, PHP, và Perl, giúp đơn giản hóa quá trình cài đặt và triển khai cơ sở dữ liệu.

5.2 Cấu trúc code



Hình 3: Cấu trúc code

Cấu trúc code được hiện thực trong bài tập lớn gồm có ba phần chính:

- **Views:** Gồm những thư mục, tệp tin hỗ trợ tạo giao diện người dùng.
 - order.js: Tạo giao diện người dùng cho giao diện đặt hàng cho khách hàng.
 - viewHistoryBook.js: Tạo giao diện người dùng cho giao diện xem tác giả yêu thích.
 - create_book.js: Tạo giao diện người dùng cho hoạt động thêm sách.
 - crud_book.js: Tạo giao diện để xem, lọc, tìm kiếm sách để phục vụ điều hướng các hoạt động thêm, xóa, sửa, xem chi tiết sách.
 - provider_book_detail.js: Tạo giao diện để nhà cung cấp (phân phối) xem chi tiết sách.
 - update_book.js: Tạo giao diện để cập nhật/ sửa thông tin chi tiết của sách.



- Controller: Gồm những thư mục, tập tin hỗ trợ gửi yêu cầu truy vấn tới cơ sở dữ liệu. Controller sẽ là trung gian để Views và Database có thể đọc, ghi, cập nhật, xử lý, validate dữ liệu. Views sẽ yêu cầu thông tin kèm tham số đầu vào cần thiết, Controller sẽ gọi truy vấn đến Database nhằm thực hiện các thao tác đọc, ghi, cập nhật, validate đầu vào, dữ liệu. Controller được kết nối với Database bằng mã nguồn tại tệp tin connect_db.js. Controller sẽ lấy kết quả trả về từ truy vấn trong database và trả về lại cho views để cập nhật hiển thị giao diện người dùng
 - order.js: Điều hướng, hỗ trợ trung gian giao tiếp giữa order.js và viewHistoryBook.js của Views với Database.
 - provider.js: Điều hướng, hỗ trợ trung gian giao tiếp giữa CRUID_book của Views với Database.
- Database: Cơ sở dữ liệu gồm các tệp tin được chia theo chức năng:
 - Create: Tệp tin có chức năng tạo các bảng trong cơ sở dữ liệu
 - Input: Tệp tin có chức năng đưa dữ liệu vào các bảng đã tạo.
 - Data_constraint: Tệp tin có chức năng tạo các trigger phục vụ cho ràng buộc dữ liệu
 - Meaning_constraint: Các tệp tin có chức năng tạo các trigger phục vụ cho ràng buộc ngữ nghĩa.
 - Procedure: Các tệp tin gồm các thủ tục trong cơ sở dữ liệu.
 - Function: Các tệp tin gồm các hàm trong cơ sở dữ liệu.

5.3 Tạo kết nối với Database

5.3.1 Kết nối đến Database

Sử dụng thư viện mysql2, tạo một module thực hiện việc kết nối đến database thông qua hàm createConnection của thư viện.

```
1 var mysql = require("mysql2")
2
3 var connect_DB = mysql.createConnection({
4     host: "localhost",
5     user: "root",
6     password: "",
7     database: "assign_db"
8 });
9
10 connect_DB.connect(function(err) {
11     if (err) throw err;
12 });
13
14 module.exports = connect_DB;
```

Listing 63: module thực hiện kết nối đến database

5.3.2 Thực hiện truy vấn để lấy/thay đổi dữ liệu từ Database

Khi muốn thực hiện câu truy vấn, gọi thủ tục ở Database, controller gọi đến module connect_DB và gọi câu truy vấn cùng với các tham số truyền vào.

```
1 const sql = 'SELECT * FROM book WHERE book_id=?'; \\câu truy vấn hoặc gọi thủ tục
2     \\cần thực hiện
3 connect_DB.query(sql, [
4     \\danh sách các tham số
5 ], function (err, result, field) {
6     if (err) {
7         \\xử lý nếu xảy ra lỗi
8     }
9     else {
10         \\Trả kết quả hoặc thông báo đến người dùng
```



10 }
11 })

Listing 64: module thực hiện kết nối đến database

5.4 Các chức năng đã hiện thực

5.4.1 Câu 5a Màn hình thêm/sửa sách

Nhà cung cấp có thể thực hiện thêm sách để bán ở trang đăng kí sách bán. Nhà cung cấp thực hiện điền các thông tin ở trong form. Sau khi hoàn tất điền bấm 'Tạo sách' để gửi yêu cầu đến hệ thống.

The screenshot shows a web page titled 'ĐĂNG KÍ BÁN SÁCH'. The form fields include:

- Tên sách: (Input field)
- Tác giả: (Input field) with buttons 'Thêm tác giả' (Add author) and 'Xóa tác giả' (Delete author).
- Độ tuổi tối thiểu: (Input field) with value '0'.
- Edition: (Input field)
- Kiểu sách: (Input field) with button 'Chọn kiểu sách' (Select book type).
- Buttons at the bottom: 'Trở lại' (Back), 'Tiếp' (Next), and 'Tạo sách' (Create book).

On the right side of the form, there is a grid of five book covers with placeholder text and images:

- NIKI: SED DO eiusmod TEMPOR incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- THE CUTTING FIRE: SED DO eiusmod TEMPOR incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Interstate 60: Episodes of the Road: DESERUNT MOLLIT ANIM ID EST LABORUM. SED DO eiusmod TEMPOR incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- INN THAT WAY: SED DO eiusmod TEMPOR incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- THE SURE THING: SED DO eiusmod TEMPOR incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Hình 4: Trang đăng kí bán sách



Tác giả:

Độ tuổi tối thiểu:

Edition:

Ánh thế loại

Kinh doanh Truyện tranh Giáo dục
 Hư cấu Sức khỏe Lịch sử
 Luật Thần thoại Y học
 Chính trị Lãng mạn Tôn giáo
 Khoa học Self-help Thể thao
 Công nghệ Du lịch Thơ ca

Kiểu sách:

Hình 5: Trang đăng ký bán sách

BookPee [Trang chủ](#) [Quản lý sách](#) [Thêm sách](#) [Đăng xuất](#)

ĐĂNG KÍ BÁN SÁCH

Ngôn ngữ:

Ngày xuất bản:

Nhà xuất bản:

Số lượng:

Giá:

ISBN:

Hình 6: Trang đăng ký bán sách

Tùy thuộc vào loại sách đã chọn là kindle, audio hay sách vật lý, hệ thống hiển thị các mục nhập tương ứng.



ĐĂNG KÍ BÁN SÁCH

Kích thước file (kb):

Số trang:

[Trở lại](#)

[Tiếp](#)

[Tạo sách](#)



Hình 7: Trang thông tin kindle

ĐĂNG KÍ BÁN SÁCH

Định dạng:

Tình trạng:

Kích thước (cm):

Cân nặng (kg):

Số trang



[Trở lại](#)

[Tiếp](#)

[Tạo sách](#)

Hình 8: Trang thông tin sách vật lý



BookPee Trang chủ Quản lý sách Thêm sách Đăng xuất

ĐĂNG KÍ BÁN SÁCH

Kích thước file (kb):

Thời lượng:

[Trở lại](#) [Tiếp](#) [Tạo sách](#)



Hình 9: Trang thông tin sách audio

Nếu thông tin bị nhập sai, hệ thống hiện thông báo yêu cầu nhà cung cấp nhập lại.

BookPee Trang chủ Quản lý sách Thêm sách Đăng xuất

ĐĂNG KÍ BÁN SÁCH

Tên sách:

Tác giả: [Thêm tác giả](#)
 [Xóa tác giả](#)

Độ tuổi tối thiểu:

Edition:

Hiển thị logo

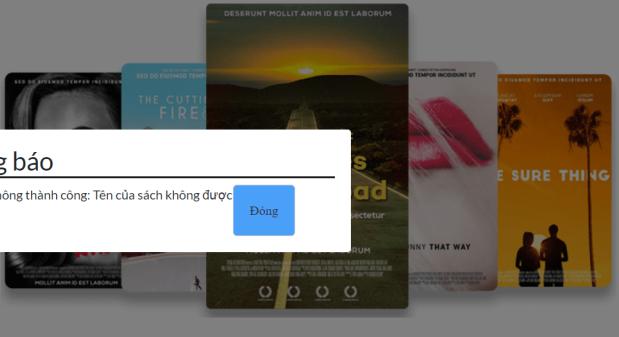
Kiểu sách:

[Trở lại](#) [Tiếp](#) [Tạo sách](#)

Thông báo

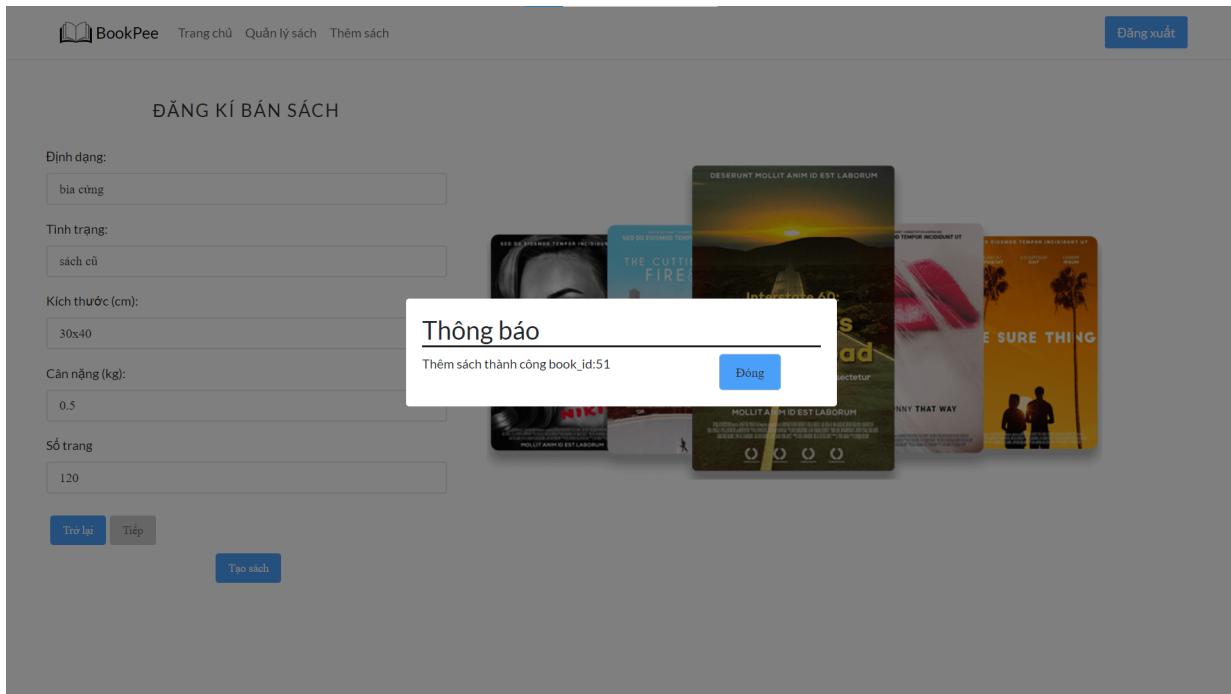
Tạo sách không thành công: Tên của sách không được để trống

[Đóng](#)



Hình 10: Tạo sách không thành công

Nếu thông tin chính xác hệ thống thông báo thành công cho nhà cung cấp.



Hình 11: Tạo sách thành công

Procedure sử dụng trong màn hình này:

```
1  create procedure add_book (
2    in title varchar(255),
3    in reading_age int,
4    in price double,
5    in language_ varchar(255),
6    in edition varchar(255),
7    in publication_date DATE,
8    in publisher_name varchar(255),
9    in isbn varchar(13),
10   in provider_id int,
11   in quantity int,
12   out return_book_id int
13 )
14 begin
15   if title is null then
16     signal sqlstate '45000' set message_text ='Tên của sách không được để trống';
17   end if;
18   if reading_age is null then
19     signal sqlstate '45000' set message_text ='Giới hạn tuổi của sách không được để trống';
20   end if;
21   if price is null then
22     signal sqlstate '45000' set message_text ='Giá tiền của sách không được để trống';
23   end if;
24   if language_ is null then
25     signal sqlstate '45000' set message_text ='Ngôn ngữ của sách không được để trống';
26   end if;
27   if edition is null then
28     signal sqlstate '45000' set message_text ='Phiên bản của sách không được để trống';
29   end if;
30   if publication_date is null then
31     signal sqlstate '45000' set message_text ='Ngày phát hành của sách không được để trống';
32   end if;
```



```
33    if publisher_name is null then
34        signal sqlstate '45000' set message_text ='Tên nhà phát hành của sách không đ
35        ược để trống';
36    end if;
37    if isbn is null then
38        signal sqlstate '45000' set message_text ='Mã ISBN của sách không được để trô
39        ng';
40    end if;
41    if quantity is null then
42        signal sqlstate '45000' set message_text ='Số lượng cuốn sách không được để
43        trống';
44    end if;
45
46    if (reading_age <= 0) then
47        signal sqlstate '45000' set message_text ='Độ tuổi giới hạn đọc sách phải lớn
48        hơn 0';
49    end if;
50    if (price <= 0) then
51        signal sqlstate '45000' set message_text ='Giá bán của sách phải lớn hơn 0';
52    end if;
53    if not (isbn regexp '^[0-9]{13}') then
54        signal sqlstate '45000' set message_text ='Mã ISBN của quyền sách phải là 13
55        ký tự chữ số';
56    end if;
57    if (provider_id <= 0) then
58        signal sqlstate '45000' set message_text ='ID của nhà cung cấp sách phải lớn
59        hơn 0';
60    end if;
61
62    if (quantity < 0) then
63        signal sqlstate '45000' set message_text ='Số lượng sách không thể là số âm';
64    end if;
65
66    insert into book value (NULL, title, reading_age, price, language_, edition,
67                            publication_date, publisher_name, isbn, provider_id, quantity);
68    SELECT LAST_INSERT_ID() AS return_book_id;
69
70
```

Listing 65: Thủ tục truyền vào các thuộc tính chung của sách và thêm vào bảng book

```
1 create procedure add\_genres\_(in genre_of_book  varchar(255), in inbook_id int )
2 begin
3     if genre_of_book not in ('Kinh doanh', 'Truyện tranh', 'Giáo dục', 'Hư cấu', 'Sức
4         khỏe', 'Lịch sử', 'Luật', 'Thần thoại', 'Y học', 'Chính trị', 'Lãng mạn', 'Tôn giáo',
5         'Khoa học', 'Self-help', 'Thể thao', 'Công nghệ', 'Du lịch', 'Thơ ca') then
6         delete from book where book.book_id=inbook_id;
7         signal sqlstate '45000' set message_text ='Thể loại không hợp lệ.
8         ';
9
10    end if;
11    insert into genres_book value(inbook_id,genre_of_book);
12
13
```

Listing 66: Thủ tục nhận vào book_id và các thể loại tương ứng để thêm vào genres,book

```
1
2 create procedure add_book_type_(in type_of_book  varchar(255), in inbook_id int,
3                                 in size int, in paper_length int,in time_ time,in format_ varchar(255), in
4                                 dimensions varchar(255), in weight double, in status_ varchar(255))
5 begin
6     if type_of_book is null then
7         delete from book where book.book_id=inbook_id;
```



```
6      signal sqlstate '45000' set message_text ='Loại sách không được để trống.';
7      end if;
8
9      if type_of_book not in ('KindleBook','AudioBook','PhysicalBook') then
10         delete from book where book.book_id=inbook_id;
11         signal sqlstate '45000' set message_text ='Loại sách không hợp lệ.';
12         end if;
13
14      if(type_of_book ="KindleBook") then
15          if paper_length is null then
16              delete from book where book.book_id=inbook_id;
17              signal sqlstate '45000' set message_text ='Số trang sách không được để trống.';
18          end if;
19          if paper_length <= 0 then
20              delete from book where book.book_id=inbook_id;
21              signal sqlstate '45000' set message_text ='Số trang sách phải lớn hơn 0.';
22          end if;
23          if size is null then
24              delete from book where book.book_id=inbook_id;
25              signal sqlstate '45000' set message_text ='Kích thước sách không được để trống.';
26          end if;
27          if size <= 0 then
28              delete from book where book.book_id=inbook_id;
29              signal sqlstate '45000' set message_text ='Kích thước sách phải lớn hơn 0.' ;
30          end if;
31
32          insert into kindle_book value( inbook_id ,size , paper_length);
33      end if;
34
35      if(type_of_book ="AudioBook") then
36          if size is null then
37              delete from book where book.book_id=inbook_id;
38              signal sqlstate '45000' set message_text ='Kích thước sách không được để trống.';
39          end if;
40          if size <= 0 then
41              delete from book where book.book_id=inbook_id;
42              signal sqlstate '45000' set message_text ='Kích thước sách phải lớn hơn 0.' ;
43          end if;
44          if time_ is null then
45              delete from book where book.book_id=inbook_id;
46              signal sqlstate '45000' set message_text ='Thời gian sách không được để trống.';
47          end if;
48          insert into audio_book value(inbook_id ,size ,time_);
49      end if;
50
51      if(type_of_book ="PhysicalBook") then
52          if format_ is null then
53              delete from book where book.book_id=inbook_id;
54              signal sqlstate '45000' set message_text ='Định dạng sách không được để trống.';
55          end if;
56
57          if dimensions is null then
58              delete from book where book.book_id=inbook_id;
59              signal sqlstate '45000' set message_text ='Kích thước cuốn sách không được để trống.';
60          end if;
61
62          if paper_length is null then
63              delete from book where book.book_id=inbook_id;
64              signal sqlstate '45000' set message_text ='Số trang sách không được để trống.';
65          end if;
66
67          if paper_length <= 0 then
```



```
68         delete from book where book.book_id=inbook_id;
69         signal sqlstate '45000' set message_text ='Số trang sách phải lớn hơn 0.';
70     end if;
71
72     if weight is null then
73         delete from book where book.book_id=inbook_id;
74         signal sqlstate '45000' set message_text ='Khối lượng sách không được để trống.';
75     end if;
76
77     if weight <= 0 then
78         delete from book where book.book_id=inbook_id;
79         signal sqlstate '45000' set message_text ='Khối lượng sách phải lớn hơn 0.';
80     end if;
81
82     if status_ is null then
83         delete from book where book.book_id=inbook_id;
84         signal sqlstate '45000' set message_text ='Trang thái sách không được để trống.';
85     end if;
86
87     insert into physical_book value(inbook_id,format_,dimensions,paper_length,
88                                     weight, status_);
89
90 end if;
91 end;
```

Listing 67: Thủ tục nhận vào thuộc tính của riêng loại sách (kindle, audio, vật lý) và thêm vào bảng tương ứng

```
1 create procedure add_write_(in inpenname varchar(255), in inbook_id int )
2 begin
3     declare get_id_author int default 0;
4     if inpenname is null then
5         delete from book where book.book_id=inbook_id;
6         signal sqlstate '45000' set message_text ='Bút danh tác giả không được để trống.';
7     end if;
8
9     if inbook_id is null then
10        delete from book where book.book_id=inbook_id;
11        signal sqlstate '45000' set message_text ='ID sách không được để trống.';
12    end if;
13
14 select author_id into get_id_author
15   from author
16  where author.penname=inpenname;
17 if get_id_author =0 then
18     insert into author value (null,inpenname);
19 --     SELECT author_id into get_id_author from author where author.penname=
20 --     inpenname;
20     SELECT LAST_INSERT_ID() into get_id_author;
21 end if;
22 if get_id_author !=0 then
23     insert into write_ value(inbook_id,get_id_author);
24 end if;
25
26 end;
```

Listing 68: Thủ tục nhận vào biệt danh tác giả và book_id để thêm vào bảng write_ và bảng author(nếu tác giả chưa có trong hệ thống)

Nhà cung cấp có thể thực hiện chỉnh sửa cuốn sách đã thêm vào.



BookPee Trang chủ Quản lý sách Thêm sách Đăng xuất

Press F11 to exit full screen

CHỈNH SỬA SÁCH

Tên sách: Fundamentals of Database system, 8th

Tác giả: Ramez Elmasri, Shamkant B. Navathe

Độ tuổi tối thiểu: 13

Edition: thường

Kiểu sách: Sách Giấy

Hình 12

Khi bấm nút 'Sửa sách', xuất hiện các thông báo thành công hoặc thất bại tương tự ở trang đăng ký bán sách.

BookPee Trang chủ Quản lý sách Thêm sách Đăng xuất

CHỈNH SỬA SÁCH

Tên sách: Fundamentals of Database system, 8th

Tác giả: Ramez Elmasri, Shamkant B. Navathe

Độ tuổi tối thiểu: 13

Edition: thường

Kiểu sách: Sách Giấy

Thông báo

Cập nhật thông tin sách thành công

Hình 13

Procedure sử dụng trong màn hình này:

```
1  create procedure show_book_info(in book_id int)
2      begin
```



```
3      if not (select exists (select * from book where book.book_id = book_id)) then
4          signal sqlstate '45000' set message_text ='ID của sách không tồn tại';
5      end if;
6
7      -- Add rating
8
9      create table rating_temp as select
10         rate.book_id,
11             count(*) as rating_quantity,
12             avg(score) as rating_score
13     from rate
14         group by rate.book_id;
15         if not (select exists(select * from rating_temp where rating_temp.book_id
16         = book_id)) then
17             create table rating_table(book_id int not null, rating_quantity int not
18             null, rating_score float not null);
19             insert into rating_table (book_id, rating_quantity, rating_score)
20             values (book_id, 0, 0);
21         else
22             create table rating_table as select * from rating_temp where rating_temp.
23             book_id = book_id;
24         end if;
25
26         -- Add genres
27
28         create table genre_book as select rating_table.book_id, rating_quantity,
29             rating_score, genres from
30             rating_table left join genres_book on rating_table.book_id = genres_book.
31             book_id;
32
33         -- Add author
34
35         create table author_book as select genre_book.book_id, rating_quantity,
36             rating_score, genres, penname from
37             genre_book left join write_ on genre_book.book_id = write_.book_id
38             join author on write_.author_id = author.author_id;
39
40         -- Add series
41
42         if not (select exists (select * from consisted where consisted.book_id =
43             book_id)) then
44             create table series_book as select author_book.book_id, rating_quantity,
45             rating_score, genres, penname from author_book;
46             alter table series_book add series_name varchar(255);
47             insert into series_book (series_name) values ('none');
48         else
49             create table series_book as select author_book.book_id, rating_quantity,
50             rating_score, genres, penname, name_ as series_name from
51             author_book join consisted on author_book.book_id = consisted.book_id
52             join series on consisted.series_id = series.series_id;
53         end if;
54
55         -- Add discount
56
57         if not (select exists (select * from have_ where have_.book_id = book_id))
58     ) then
59             create table discount_book as select series_book.book_id, rating_quantity,
60             rating_score, genres, penname, series_name from series_book;
61             alter table discount_book add max_discount int;
62             insert into discount_book (max_discount) values (0);
63             create table discount_max (book_id int not null);
64             create table discount_have (book_id int not null);
65         else
66             create table discount_have as select series_book.book_id, have_.discount_id
67             , discount_value from
68                 series_book join have_ on series_book.book_id = have_.book_id
69                 join discount on have_.discount_id = discount.discount_id;
70                 create table discount_max as select book_id, max(discount_value) as
71                 max_discount from discount_have group by book_id;
72                 create table discount_book as select series_book.book_id,
73                 rating_quantity, rating_score, genres, penname, series_name, max_discount
```



```
59      from
60          discount_max join series_book on discount_max.book_id = series_book.
61      book_id;
62  end if;
63
64      -- Add detail
65
66      select discount_book.book_id, title, genres, penname, series_name,
67      reading_age, price, language_, edition, publication_date,
68      publisher_name, isbn, book.provider_id, quantity, max_discount,
69      rating_quantity, rating_score, audio_book.size as audio_size,
70      audio_book.time_ as audio_time,
71      kindle_book.size as kindle_size,
72      kindle_book.paper_length as kindle_paper_length,
73      physical_book.format_ as physical_format,
74      physical_book.dimensions as physical_dimensions,
75      physical_book.paper_length as physical_paper_length,
76      physical_book.weight as physical_weight,
77      physical_book.status_ as physical_status,
78      provider.name_ as provider_name
79  from discount_book
80  join book on discount_book.book_id = book.book_id
81      left join assign_db.audio_book on discount_book.book_id = audio_book.
82      book_id
83      left join assign_db.kindle_book on discount_book.book_id = kindle_book.
84      book_id
85      left join assign_db.physical_book on discount_book.book_id = physical_book.
86      book_id
87      left join assign_db.provider ON book.provider_id = provider.provider_id;
88
89      drop table rating_temp;
90      drop table rating_table;
91      drop table genre_book;
92      drop table author_book;
93      drop table series_book;
94      drop table discount_have;
95      drop table discount_max;
96      drop table discount_book;
97
98  end;
```

Listing 69: Thủ tục nhận vào book_id và trả về các thông tin cho cuốn sách để hiển thị cho nhà cung cấp

```
1  create procedure update_book (
2      in inbook_id int,
3      in title varchar(255),
4      in reading_age int,
5      in price double,
6      in language_ varchar(255),
7      in edition varchar(255),
8      in publication_date DATE,
9      in publisher_name varchar(255),
10     in isbn varchar(13),
11     in quantity int
12 )
13 begin
14     if inbook_id is null then
15         signal sqlstate '45000' set message_text ='ID của sách không được để trống';
16     end if;
17     if title is null then
18         signal sqlstate '45000' set message_text ='Tên của sách không được để trống';
19     end if;
20     if reading_age is null then
21         signal sqlstate '45000' set message_text ='Giới hạn tuổi của sách không được để trống';
22     end if;
23     if price is null then
24         signal sqlstate '45000' set message_text ='Giá tiền của sách không được để trống';
25     end if;
```



```
26      if language_ is null then
27          signal sqlstate '45000' set message_text ='Ngôn ngữ của sách không được để trống';
28      end if;
29      if edition is null then
30          signal sqlstate '45000' set message_text ='Phiên bản của sách không được để trống';
31      end if;
32      if publication_date is null then
33          signal sqlstate '45000' set message_text ='Ngày phát hành của sách không được để trống';
34      end if;
35      if publisher_name is null then
36          signal sqlstate '45000' set message_text ='Tên nhà phát hành của sách không được ược để trống';
37      end if;
38      if isbn is null then
39          signal sqlstate '45000' set message_text ='Mã ISBN của sách không được để trống';
40      end if;
41      if quantity is null then
42          signal sqlstate '45000' set message_text ='Số lượng cuốn sách không được để trống';
43      end if;
44      if (inbook_id <= 0) then
45          signal sqlstate '45000' set message_text ='ID của sách phải lớn hơn 0';
46      end if;
47      if not (select exists (select * from book where book.book_id = inbook_id))
48          then
49              signal sqlstate '45000' set message_text ='ID của sách không tồn tại';
50      end if;
51      if (reading_age <= 0) then
52          signal sqlstate '45000' set message_text ='Độ tuổi giới hạn đọc sách phải lớn hơn 0';
53      end if;
54      if (price <= 0) then
55          signal sqlstate '45000' set message_text ='Giá bán của sách phải lớn hơn 0';
56      end if;
57      if not (isbn regexp '^[0-9]{13}') then
58          signal sqlstate '45000' set message_text ='Mã ISBN của quyền sách phải là 13 ký tự chữ số';
59      end if;
60      if (quantity < 0) then
61          signal sqlstate '45000' set message_text ='Số lượng sách không thể là số âm';
62      end if;
63
64      update book set
65          book.title = title,
66          book.reading_age = reading_age,
67          book.price = price,
68          book.language_ = language_,
69          book.edition = edition,
70          book.publication_date = publication_date,
71          book.publisher_name = publisher_name,
72          book.isbn = isbn,
73          book.quantity = quantity
74      where book.book_id = inbook_id;
75  end;
```

Listing 70: Thủ tục nhận vào các thuộc tính trong bảng book để cập nhật

```
1
2  create procedure update_book_type_(in type_of_book  varchar(255), in inbook_id
3                                     int, in size int, in paper_length int,in time_time,in format_
4                                     varchar(255),
5                                     in dimensions varchar(255), in weight double, in status_
6                                     varchar(255))
7 begin
8     if type_of_book is null then
9         signal sqlstate '45000' set message_text ='Loại sách không được để trống.';
10    end if;
```



```
7
8      if type_of_book not in ('KindleBook','AudioBook','PhysicalBook') then
9          signal sqlstate '45000' set message_text ='Loại sách không hợp lệ.';
10         end if;
11
12         if(type_of_book ="KindleBook") then
13             if paper_length is null then
14                 signal sqlstate '45000' set message_text ='Số trang sách không được để trống.';
15             end if;
16
17             if paper_length <= 0 then
18                 signal sqlstate '45000' set message_text ='Số trang sách phải lớn hơn 0.';
19             end if;
20
21             if size is null then
22                 signal sqlstate '45000' set message_text ='Kích thước sách không được để trống.';
23             end if;
24
25             if size <= 0 then
26                 signal sqlstate '45000' set message_text ='Kích thước sách phải lớn hơn 0.' ;
27             end if;
28
29                 update kindle_book set
30                     kindle_book.size=size,
31                     kindle_book.paper_length=paper_length
32                     where kindle_book.book_id=inbook_id;
33
34
35
36         end if;
37
38         if(type_of_book ="AudioBook") then
39             if size is null then
40                 signal sqlstate '45000' set message_text ='Kích thước sách không được để trống.';
41             end if;
42
43             if size <= 0 then
44                 signal sqlstate '45000' set message_text ='Kích thước sách phải lớn hơn 0.' ;
45             end if;
46
47             if time_ is null then
48                 signal sqlstate '45000' set message_text ='Thời gian sách không được để trống.';
49             end if;
50
51                 update audio_book set
52                     audio_book.size=size,
53                     audio_book.time_=time_
54                     where audio_book.book_id=inbook_id;
55             end if;
56
57         if(type_of_book ="PhysicalBook") then
58             if format_ is null then
59                 signal sqlstate '45000' set message_text ='Định dạng sách không được để trống.';
60             end if;
61             if dimensions is null then
62                 signal sqlstate '45000' set message_text ='Kích thước cuốn sách không được để trống.';
63             end if;
64             if paper_length is null then
65                 signal sqlstate '45000' set message_text ='Số trang sách không được để trống.';
66             end if;
67             if paper_length <= 0 then
68                 signal sqlstate '45000' set message_text ='Số trang sách phải lớn hơn 0.';
```



```
69      end if;
70      if weight is null then
71          signal sqlstate '45000' set message_text ='Khối lượng sách không được để trống.';
72      end if;
73      if weight <= 0 then
74          signal sqlstate '45000' set message_text ='Khối lượng sách phải lớn hơn 0.';
75      end if;
76      if status_ is null then
77          signal sqlstate '45000' set message_text ='Trạng thái sách không được để trống.';
78      end if;
79
80      update physical_book set
81          physical_book.format_=format_,
82          physical_book.dimensions=dimensions,
83          physical_book.paper_length=paper_length,
84          physical_book.weight =weight,
85          physical_book.status_=status_
86      where physical_book.book_id=inbook_id;
87
88      end if;
89
90 end;
```

Listing 71: Thủ tục nhận các thuộc tính riêng của loại sách để cập nhật

5.4.2 Câu 5b Màn hình hiển thị các cuốn sách của nhà cung cấp, xóa sách

The screenshot shows a web-based application for managing books. At the top, there's a navigation bar with icons for home, manage books, and add book, along with a sign-out button. Below the navigation is a search bar with a placeholder 'Search' and a note to press F11 for full screen. There are dropdown filters for category ('Thể loại') and price ('Giá'), a sorting dropdown ('Sắp xếp theo'), and buttons for filtering ('Lọc') and clearing ('Làm mới'). The main area displays a grid of book cards. Each card contains a thumbnail, book ID, title, author, price, and two buttons: 'Xóa' (Delete) and 'Chi tiết' (Details). The visible books are:

Book ID	Title	Author	Price
34	Fundamentals of Database system, 8th		145000 đ
31	Harry Potter và bảo bối tử thần		145000 đ
32	Harry Potter và bảo bối tử thần		145000 đ
28	Harry Potter và chiếc cốc lùa		145000 đ
29	Harry Potter và hoàng tử lai		145000 đ
30	Harry Potter và hoàng tử lai		145000 đ

Hình 14: Danh sách sách của nhà cung cấp

Nhà cung cấp có thể thực hiện thay đổi trên bộ lọc để tìm một cuốn sách.



The screenshot shows a search interface for a book database. At the top, there is a navigation bar with the logo, and links for 'Trang chủ', 'Quản lý sách', and 'Thêm sách'. On the right, there is a 'Đăng xuất' button. Below the navigation, there is a search bar with placeholder 'Search', a 'Search' button, and several filter options: 'Thể loại' (dropdown: Giáo dục), 'Giá' (dropdown: Từ 100.000đ trở lên), 'Sắp xếp theo' (dropdown: Từ A-Z), 'Lọc' (button), and 'Làm mới' (button). Two search results are displayed in cards:

- Book id: 33**
Tên sách: Fundamentals of Database system, 8th
Giá: 145000 đ
Xóa **Chi tiết**
- Book id: 34**
Tên sách: Fundamentals of Database system, 8th
Giá: 145000 đ
Xóa **Chi tiết**

Hình 15: Kết quả tìm kiếm bằng bộ lọc

Nhà cung cấp bấm xóa ở dưới một cuối sách để xóa cuốn sách đó ra khỏi hệ thống. Sách chỉ có thẻ xóa nếu không thuộc về đơn hàng nào.

The screenshot shows a list of books in a grid format. A confirmation dialog box is overlaid on the screen with the title 'Thông báo' and the message 'Xóa thành công'. There are 'Đóng' (Close) and 'Xóa' (Delete) buttons. The books listed are:

- Book id: 34
Tên sách: Fundamentals of Database system, 8th
Giá: 145000 đ
Xóa **Chi tiết**
- Book id: 31
Xóa **Chi tiết**
- Book id: 32
Tên sách: Harry Potter và bảo bối tử thần
Giá: 145000 đ
Xóa **Chi tiết**
- Book id: 28
Tên sách: Harry Potter và chiếc cốc lửa
Giá: 145000 đ
Xóa **Chi tiết**
- Book id: 29
Tên sách: Harry Potter và hoàng tử lai
Giá: 145000 đ
Xóa **Chi tiết**
- Book id: 30
Tên sách: Harry Potter và hoàng tử lai
Giá: 145000 đ
Xóa **Chi tiết**

Hình 16: Xóa sách thành công



The screenshot shows a list of books in the BookPee application. A modal window titled "Thông báo" (Information) is displayed in the center, containing the message "Xóa thành công" (Deleted successfully) and a blue "Đóng" (Close) button. The background grid shows book details such as Book id, Title, and Price.

Hình 17: Xóa sách có trong đơn hàng

Nhà cung cấp cũng có thể bấm vào chi tiết để xem thông tin sách.

The screenshot shows a detailed view of a book's information in the BookPee application. The book is titled "Fundamentals of Database system, 8th". The page displays the book cover, title, author (Ramez Elmasri - Shamkant B. Navathe), publisher (Nhà xuất bản trê), publication date (2013-05-10), and other details like ISBN, language, and dimensions. A sidebar on the right shows the book's availability at a specific store, indicating it is available from "Nhà xuất bản trê" with a stock level of 100.

Hình 18: Xem thông tin sách

Procedure sử dụng trong màn hình này:

```
1  create procedure show_book_info(in book_id int)
2      begin
3          if not (select exists (select * from book where book.book_id = book_id)) then
```



```
4      signal sqlstate '45000' set message_text ='ID của sách không tồn tại';
5  end if;
6
7  -- Add rating
8
9  create table rating_temp as select
10    rate.book_id,
11      count(*) as rating_quantity,
12      avg(score) as rating_score
13  from rate
14      group by rate.book_id;
15      if not (select exists(select * from rating_temp where rating_temp.book_id
16 = book_id)) then
17          create table rating_table(book_id int not null, rating_quantity int not
18 null, rating_score float not null);
19          insert into rating_table (book_id, rating_quantity, rating_score)
20 values (book_id, 0, 0);
21      else
22          create table rating_table as select * from rating_temp where rating_temp.
23 book_id = book_id;
24      end if;
25
26      -- Add genres
27
28
29      create table genre_book as select rating_table.book_id, rating_quantity,
30 rating_score, genres from
31      rating_table left join genres_book on rating_table.book_id = genres_book.
32 book_id;
33
34      -- Add author
35
36      create table author_book as select genre_book.book_id, rating_quantity,
37 rating_score, genres, penname from
38      genre_book left join write_ on genre_book.book_id = write_.book_id
39      join author on write_.author_id = author.author_id;
40
41      -- Add series
42
43
44      if not (select exists (select * from consisted where consisted.book_id =
45 book_id)) then
46          create table series_book as select author_book.book_id, rating_quantity,
47 rating_score, genres, penname from author_book;
48          alter table series_book add series_name varchar(255);
49          insert into series_book (series_name) values ('none');
50      else
51          create table series_book as select author_book.book_id, rating_quantity,
52 rating_score, genres, penname, name_ as series_name from
53      author_book join consisted on author_book.book_id = consisted.book_id
54      join series on consisted.series_id = series.series_id;
55      end if;
56
57      -- Add discount
58
59      if not (select exists (select * from have_ where have_.book_id = book_id)
60 ) then
61          create table discount_book as select series_book.book_id, rating_quantity,
62 rating_score, genres, penname, series_name from series_book;
63          alter table discount_book add max_discount int;
64          insert into discount_book (max_discount) values (0);
65          create table discount_max (book_id int not null);
66          create table discount_have (book_id int not null);
67      else
68          create table discount_have as select series_book.book_id, have_.discount_id
69 , discount_value from
70      series_book join have_ on series_book.book_id = have_.book_id
71      join discount on have_.discount_id = discount.discount_id;
72          create table discount_max as select book_id, max(discount_value) as
73 max_discount from discount_have group by book_id;
74          create table discount_book as select series_book.book_id,
75 rating_quantity, rating_score, genres, penname, series_name, max_discount
76 from
```



```
59         discount_max join series_book on discount_max.book_id = series_book.
60     book_id;
61   end if;
62
63   -- Add detail
64
65   select discount_book.book_id, title, genres, penname, series_name,
66   reading_age, price, language_, edition, publication_date,
67   publisher_name, isbn, book.provider_id, quantity, max_discount,
68   rating_quantity, rating_score, audio_book.size as audio_size,
69   audio_book.time_ as audio_time,
70   kindle_book.size as kindle_size,
71   kindle_book.paper_length as kindle_paper_length,
72   physical_book.format_ as physical_format,
73   physical_book.dimensions as physical_dimensions,
74   physical_book.paper_length as physical_paper_length,
75   physical_book.weight as physical_weight,
76   physical_book.status_ as physical_status,
77   provider.name_ as provider_name
78 from discount_book
79 join book on discount_book.book_id = book.book_id
80 left join assign_db.audio_book on discount_book.book_id = audio_book.
81 book_id
82 left join assign_db.kindle_book on discount_book.book_id = kindle_book.
83 book_id
84 left join assign_db.physical_book on discount_book.book_id = physical_book.
85 book_id
86 left join assign_db.provider ON book.provider_id = provider.provider_id;
87 drop table rating_temp;
88 drop table rating_table;
89 drop table genre_book;
90 drop table author_book;
91 drop table series_book;
92 drop table discount_have;
93 drop table discount_max;
94 drop table discount_book;
95
96 end;
```

Listing 72: Thủ tục nhận book_id và trả về các thông tin hiện tại được lưu của sách

```
1 CREATE PROCEDURE search(IN p_title VARCHAR(255), IN p_provider_id INT)
2 BEGIN
3   -- Create a temporary table
4   CREATE TEMPORARY TABLE temp_search_results AS
5   SELECT * FROM book
6   WHERE title = p_title AND provider_id = p_provider_id;
7
8   -- Return the results from the temporary table
9   SELECT * FROM temp_search_results;
10
11  DROP TEMPORARY TABLE IF EXISTS temp_search_results;
12 END
```

Listing 73: Thủ tục để tìm kiếm theo tên cuốn sách cho một provider

```
1 create procedure show_book_by_provider(in provider_id int, in genres varchar(255)
2   , in price_range varchar(255), in order_by varchar(255))
3 begin
4   if not (select exists (select * from book where book.provider_id =
5   provider_id)) then
6     signal sqlstate '45000' set message_text = 'Không có nhà cung cấp này';
7   end if;
8
9   if genres is not null then
10    create temporary table filter_genre as select book.book_id, title, edition,
11    price from
12      book join genres_book on book.book_id = genres_book.book_id
```



```
10          where genres_book.genres = genres and book.provider_id = provider_id;
11      else
12          create temporary table filter_genre as
13              select book_id, title, edition, price from book where book.
14          provider_id = provider_id;
15      end if;
16
17          if price_range is not null then
18              if (price_range <= 'low') then
19                  create temporary table filter_price as select * from filter_genre where
filter_genre.price < 50000;
20              elseif (price_range <= 'mid') then
21                  create temporary table filter_price as select * from filter_genre where
filter_genre.price > 50000 and filter_genre.price < 100000;
22              elseif (price_range <= 'high') then
23                  create temporary table filter_price as select * from filter_genre where
filter_genre.price > 100000;
24              end if;
25          else create temporary table filter_price as select * from filter_genre;
26          end if;
27
28          if order_by is not null then
29              if (order_by <= 'titleasc') then
30                  select * from filter_price order by title asc;
31              elseif (order_by <= 'titledesc') then
32                  select * from filter_price order by title desc;
33              elseif (order_by <= 'priceasc') then
34                  select * from filter_price order by price asc;
35              elseif (order_by <= 'pricedesc') then
36                  select * from filter_price order by price desc;
37              end if;
38          else select * from filter_price;
39          end if;
40
41          drop table filter_genre;
42          drop table filter_price;
43      end;
```

Listing 74: Thủ tục lấy các sách theo bộ lọc cho một provider_id

```
1  create procedure delete_book(in book_id int)
2 begin
3     Declare is_book_buy int default 0;
4     Select count(*) into is_book_buy from contain where contain.book_id=book_id;
5     if(book_id is null) then
6         signal sqlstate '45000' set message_text ='ID của sách không được để trống';
7     end if;
8     if(is_book_buy!=0) then
9         signal sqlstate '45000' set message_text ='Không thể xóa sách đã có đơn mua';
10    end if;
11
12    if (book_id <= 0) then
13        signal sqlstate '45000' set message_text ='ID của sách phải lớn hơn 0';
14    end if;
15
16    if not (select exists (select * from book where book.book_id = book_id)) then
17        signal sqlstate '45000' set message_text ='ID của sách không tồn tại hoặc sách đã xóa từ trước.';
18    end if;
19
20    delete from book where book.book_id = book_id;
21 end;
```

Listing 75: Thủ tục thực hiện xóa một cuốn sách theo book_id



5.4.3 Câu 5c Màn hình tác giả yêu thích của bạn

Khách hàng có thể xem được thống kê các tác giả yêu thích của mình, xem các sách đã mua từ các tác giả đó cùng số lượng. Các tác giả được chọn là các tác giả mà tổng số sách của tác giả đó khách hàng đã mua từ 1 trở lên hoặc từ 5 trở lên tùy theo khách hàng chọn.

Khách hàng chọn tại combo-box "Từ trên 1 cuốn" để xem tác giả mà họ đã mua tổng số sách từ tác giả đó từ 1 cuốn trở lên.

The screenshot shows a user interface for a bookstore application. At the top, there is a navigation bar with icons for BookPee, Trang chủ (Home), Đặt hàng (Order), and Tác giả yêu thích của bạn (Favorite authors). On the right side of the header is a blue button labeled Đăng xuất (Logout). Below the header, the title 'Tác giả yêu thích của bạn' is displayed in bold. A sub-instruction below it reads: 'Danh sách các tác giả yêu thích kèm các tác phẩm của họ mà bạn đã mua. Tác giả mà bạn đã mua tổng số tác phẩm của họ:'. A dropdown menu is open, showing the option 'Từ 1 cuốn trở lên' (From 1 book up to). There are two sections of book cards. The first section is titled 'Tên tác giả: Gordon Ramsey' and contains three items: 1. 'Tên sách: Gordon Ramsay Quick and Delicious: 100 Recipes to Cook in 30 Minutes or Less' (Số lượng đã mua: 1); 2. 'Tên sách: Gordon Ramsay's Home Cooking: Everything You Need to Know to Make Fabulous Food' (Số lượng đã mua: 1); 3. 'Tên sách: Gordon Ramsey's Ultimate Fit Food' (Số lượng đã mua: 1). The second section is titled 'Tên tác giả: JK Rowling' and contains five items: 1. 'Tên sách: Harry Potter và hòn đá phù thủy' (Số lượng đã mua: 2); 2. 'Tên sách: Harry Potter và hoàng tử lai' (Số lượng đã mua: 2); 3. 'Tên sách: Harry Potter và hội phượng hoàng' (Số lượng đã mua: 1); 4. 'Tên sách: Harry Potter và tù nhân ngục Azakaban' (Số lượng đã mua: 1); 5. 'Tên sách: Harry Potter và chiếc cốc lừa' (Số lượng đã mua: 1).

Hình 19: Giao diện xem các tác giả yêu thích (tổng số sách đã mua từ 1 trở lên)

This screenshot shows the same interface as Figure 19, but with a different search result. The dropdown menu in the header is set to 'Từ 5 cuốn trở lên' (From 5 books up to). The 'JK Rowling' section now contains five items, while the 'Gordon Ramsey' section is no longer visible. The items in the 'JK Rowling' section are: 1. 'Tên sách: Harry Potter và hòn đá phù thủy' (Số lượng đã mua: 2); 2. 'Tên sách: Harry Potter và hoàng tử lai' (Số lượng đã mua: 2); 3. 'Tên sách: Harry Potter và hội phượng hoàng' (Số lượng đã mua: 1); 4. 'Tên sách: Harry Potter và tù nhân ngục Azakaban' (Số lượng đã mua: 1); 5. 'Tên sách: Harry Potter và chiếc cốc lừa' (Số lượng đã mua: 1).

Hình 20: Giao diện xem các tác giả yêu thích (tổng số sách đã mua từ 1 trở lên)

Khách hàng chọn tại combo-box "Từ trên 5 cuốn" để xem tác giả mà họ đã mua tổng số sách từ tác giả đó từ 5 cuốn trở lên.



BookPee Trang chủ Đặt hàng Tác giả yêu thích của bạn Đăng xuất

Tác giả yêu thích của bạn

Danh sách các tác giả yêu thích kèm các tác phẩm của họ mà bạn đã mua. Tác giả mà bạn đã mua tổng số tác phẩm của họ:

Tù trên 5 cuốn

Tên tác giả: JK Rowling

- Tên sách: Harry Potter và hòn đá phù thủy
Số lượng đã mua: 2
- Tên sách: Harry Potter và hoàng tử lai
Số lượng đã mua: 2
- Tên sách: Harry Potter và hội phượng hoàng
Số lượng đã mua: 1
- Tên sách: Harry Potter và tù nhân ngục Azkaban
Số lượng đã mua: 1

Hình 21: Giao diện xem các tác giả yêu thích (tổng số sách đã mua từ 5 trở lên)

Tên sách: Harry Potter và chiếc cốc lửa
Số lượng đã mua: 1

Tên tác giả: Tô Hoài

- Tên sách: Dế Mèn Phiêu Lưu Ký
Số lượng đã mua: 2
- Tên sách: Kim Đồng
Số lượng đã mua: 1
- Tên sách: Ngon Cò Lau
Số lượng đã mua: 1
- Tên sách: Vũ A Dinh
Số lượng đã mua: 1

Hình 22: Giao diện xem các tác giả yêu thích (tổng số sách đã mua từ 5 trở lên)

Procedure sử dụng trong màn hình này:

```
1  create procedure bought_book(in customer_id int, in limit_quantity int)
2    begin
3      if not (select exists (select * from order_ where order_.customer_id =
4          customer_id)) then
5          signal sqlstate '45000' set message_text ='Không có đơn hàng nào';
6      end if;
7
8      create table bought_table as select contain.book_id, contain.quantity, title,
9          pename from
10         order_ join contain on order_.order_id = contain.order_id
11         join book on contain.book_id = book.book_id
12         join write_ on book.book_id = write_.book_id
```



```
11      join author on write_.author_id = author.author_id where order_.
12      customer_id = customer_id and order_.status_ = "Hoàn tất";
13
14      create table bought_author as select sum(quantity) as total_quantity ,
15      penname from bought_table group by penname;
16
17      create table result as select bought_author.penname, title, sum(quantity) as
18      total_quantity from
19      bought_table
20      join bought_author on bought_table.penname = bought_author.penname
21      where bought_author.total_quantity >= limit_quantity
22      group by bought_author.penname, title
23      order by penname asc, total_quantity desc;
24
25      select * from result;
26
27      drop table bought_table;
28      drop table bought_author;
29      drop table result;
30
31  end;
```

Listing 76: Thủ tục xuất ra tên tác phẩm, tác giả kèm số lượng mà một khách hàng (customer_id) đã mua trong đó tác giả là những tác giả có tổng số sách được mua từ khách hàng đó lớn hơn hoặc bằng một lượng nhất định (limit_quantity)

5.4.4 Màn hình thêm - Màn hình đặt sách của khách hàng:

Khách hàng khi vào màn hình có thể tìm kiếm sách theo bộ lọc (gồm tên sách, các thể loại, giá gốc, sắp xếp). Trong đó thể loại có thể là tất cả hoặc một trong các thể loại đang có trong cơ sở dữ liệu, giá được lọc theo giá gốc từ dưới 50.000đ, từ 50.000đ đến dưới 100.000đ hoặc từ 100.000đ trở lên, có thể sắp xếp từ A-Z theo tên, Z-A theo tên, giá tăng dần hoặc giá giảm dần.

The screenshot shows a search results page for books on the BookPee website. At the top, there is a navigation bar with links for 'BookPee', 'Trang chủ', 'Đặt hàng', 'Tác giả yêu thích của bạn', and 'Đăng xuất'. Below the navigation bar, there is a search bar with the placeholder 'Search' and several filter options: 'Thể loại' (All), 'Giá' (All), 'Sắp xếp theo' (Từ A-Z), 'Lọc' (Filter), and 'Làm mới' (Clear). The search results are displayed in a grid of six book cards. Each card includes a thumbnail image, the book's name, its original price, current price after discount, and its ID number. The books listed are:

- Tên sách: Chú heo giáng sinh
Giá gốc: 150000 đ
Mã: 4
Kiểu sách: Physical book
Hiện tại giảm giá còn 75000 đ
- Tên sách: Chú heo giáng sinh
Giá gốc: 90000 đ
Mã: 5
Kiểu sách: Kindle book
Hiện tại giảm giá còn 45000 đ
- Tên sách: Chú heo giáng sinh
Giá gốc: 120000 đ
Mã: 6
Kiểu sách: Physical book
Hiện tại giảm giá còn 60000 đ
- Tên sách: Dế Mèn Phiêu Lưu Ký
Giá gốc: 70000 đ
Mã: 21
Kiểu sách: Physical book
- Tên sách: Dế Mèn Phiêu Lưu Ký
Giá gốc: 120000 đ
Mã: 22
Kiểu sách: Physical book
- Tên sách: Fundamentals of Database system, 7th
Giá gốc: 150000 đ
Mã: 17
Kiểu sách: Physical book

Hình 23: Màn hình để khách hàng xem sách

Khách hàng có thể bấm vào từng cuốn sách và xem chi tiết thông qua hộp thoại được hiển thị trên màn hình. Và thêm sách vào giỏ hàng.



The screenshot shows a mobile application interface. On the left, there's a sidebar with the title "Thông tin sách". Inside the sidebar, there's a circular icon of a book. To the right of the sidebar, a book detail card is displayed for "Harry Potter và hòn đá phù thủy". The card includes the following information:

Nhà xuất bản:	Ngày xuất bản:	Phiên bản:
Nhà xuất bản trẻ	12-05-2013	thường
Mã ISBN:	Ngôn ngữ:	Kiểu sách:
0985676544681	Tiếng Việt	Physical Book
Format	Dimensions	Paper length:
bìa mềm	45x30 cm	354 trang
Status:	Weight:	
sách cũ	0.5 kg	

On the far right, there's a sidebar for "Nhà cung cấp: Nhà xuất bản trẻ" with a quantity selector set to 0 and a button to "Thêm/ cập nhật vào giỏ hàng".

Hình 24: Hộp thoại để khách hàng xem chi tiết sách và thêm vào giỏ hàng

Sau khi thêm sách vào giỏ hàng, khách hàng có thể nhấn Summary Order để xem sách trong giỏ hàng, nhập thông tin đơn hàng và nhấn đặt hàng.

The screenshot shows a summary order form. It includes fields for "Tên người nhận", "Số điện thoại người nhận", "Địa chỉ người nhận", and "Promotion code". Below these, there's a section for "Phương thức giao hàng" (Online) and "Phương thức thanh toán" (Banking). Two book items are listed in the cart:

Book ID	Số lượng mua	Tên sách	Nhà phân phối	Giá
4	2	Chú heo giáng sinh	Nhà xuất bản Kim Đồng	x 15000đ
7	2	Harry Potter và tù nhân ngục Azakaban	Nhà xuất bản trẻ	x 50000đ

At the bottom is a blue button labeled "Xác nhận đặt hàng".

Hình 25: Hộp thoại để khách hàng xem lại giỏ hàng và nhập thông tin đặt hàng

Khi giỏ hàng bao gồm sách từ nhiều nhà cung cấp khác nhau, khách hàng nhấn đặt hàng sẽ hiển thị thông báo "Chỉ đặt hàng với đơn hàng có các sản phẩm thuộc cùng một nhà cung cấp" ở dưới nút "Xác nhận đặt hàng".



The screenshot shows a summary order form with the following details:

- Tên người nhận: Hồ Trọng Nhàn
- Số điện thoại người nhận: 093452443
- Địa chỉ người nhận: 286 Lý Thường Kiệt, Quận 10, Thành phố Hồ Chí Minh
- Promotion code: (empty)
- Phương thức giao hàng: Online
- Phương thức thanh toán: Banking
- Book ID: 4, Số lượng mua: 2, Tên sách: Chú heo giáng sinh, Nhà xuất bản Kim Đồng
Kiểu sách: Physical Book
- Book ID: 7, Số lượng mua: 2, Tên sách: Harry Potter và tù nhân ngục Azakaban, Nhà xuất bản trẻ
Kiểu sách: Audio Book

At the bottom, there is a note: Chỉ đặt hàng với đơn hàng có các sản phẩm thuộc cùng một nhà cung cấp.

Xác nhận đặt hàng

Hình 26: Thông báo cho khách hàng khi giỏ hàng gồm sách từ nhiều nhà cung cấp

Khi người dùng nhập sai định dạng số điện thoại, hệ thống hiển thị thông báo "Số điện thoại phải có 10 chữ số và bắt đầu bằng 0" ở dưới nút Xác nhận đơn hàng.

The screenshot shows a summary order form with the following details:

- Tên người nhận: Hồ Trọng Nhàn
- Số điện thoại người nhận: 093452443
- Địa chỉ người nhận: 286 Lý Thường Kiệt, Quận 10, Thành phố Hồ Chí Minh
- Promotion code: (empty)
- Phương thức giao hàng: Online
- Phương thức thanh toán: Banking
- Book ID: 7, Số lượng mua: 2, Tên sách: Harry Potter và tù nhân ngục Azakaban, Nhà xuất bản trẻ
Kiểu sách: Audio Book

At the bottom, there is an error message: Số điện thoại phải có 10 chữ số và bắt đầu bằng 0.

Xác nhận đặt hàng

Hình 27: Thông báo cho khách hàng khi số điện thoại nhập sai định dạng

Khi người dùng mua lại sách bản điện tử (AudioBook/ KindleBook) đã mua từ trước đó, hệ thống sẽ hiển thị thông báo "Các cuốn sách bản điện tử chỉ được mua một lần."



Summary Order:

Tên người nhận: Hồ Trọng Nhân

Số điện thoại người nhận: 0934524432

Địa chỉ người nhận: 286 Lý Thường Kiệt, Quận 10, Thành phố Hồ Chí Minh

Promotion code:

Phương thức giao hàng: Online

Phương thức thanh toán: Banking

Book ID: 7, Số lượng mua: 2, Tên sách: Harry Potter và tù nhân ngục Azakaban, Nhà phân phối: Nhà xuất bản trẻ
Kiểu sách: Audio Book

x 50000đ

Xác nhận đặt hàng

Các cuốn sách bản điện tử chỉ được mua một lần.

Hình 28: Thông báo cho khách hàng khi khách hàng mua lại KindleBook/ AudioBook

Khi trong giỏ hàng có sách bản điện tử (KindleBook/AudioBook) được mua với số lượng lớn hơn 1, hệ thống sẽ hiển thị thông báo "Các cuốn sách bản điện tử chỉ được mua 1 quyển." ở dưới nút Xác nhận đặt hàng.

BookPee Trang chủ Đặt hàng Tác giả yêu thích của bạn Đăng xuất

Summary Order:

Tên người nhận: Hồ Trọng Nhân

Số điện thoại người nhận: 0934524432

Địa chỉ người nhận: 286 Lý Thường Kiệt, Quận 10, Thành phố Hồ Chí Minh

Promotion code:

Phương thức giao hàng: Online

Phương thức thanh toán: Banking

Book ID: 5, Số lượng mua: 2, Tên sách: Chủ heo giáng sinh, Nhà phân phối: Nhà xuất bản Kim Đồng
Kiểu sách: Kindle Book

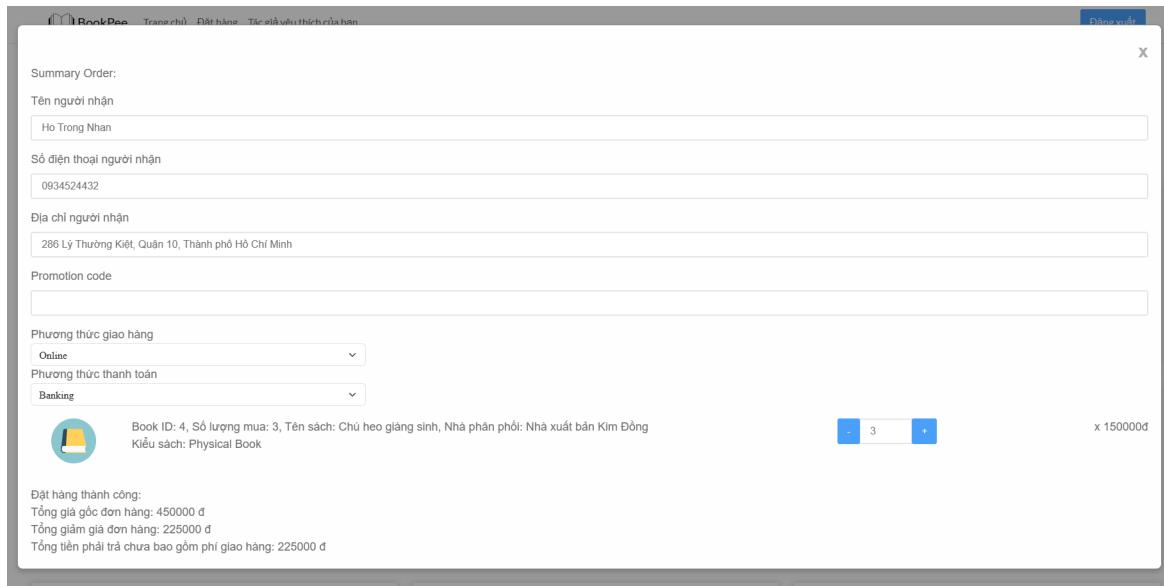
x 90000đ

Xác nhận đặt hàng

Các cuốn sách bản điện tử chỉ được mua 1 quyển.

Hình 29: Thông báo cho khách hàng khi khách hàng mua KindleBook/ AudioBook với số lượng nhiều hơn 1

Sau khi đặt hàng thành công, hệ thống sẽ hiển thị thông báo "Đặt hàng thành công" kèm theo tổng giá gốc đơn hàng, tổng giảm giá đơn hàng và tổng tiền phải trả chưa tính phí giao hàng.



Hình 30: Thông báo cho khách hàng khi đặt hàng thành công

Các procedure sử dụng trong màn hình này:

```
1  create procedure show_all_genres()
2    begin
3      select distinct genres from genres_book;
4    end;
```

Listing 77: Thủ tục xuất ra tất cả các thể loại trong hệ thống (mục đích để sử dụng thông tin làm bộ lọc sách)

```
1  create procedure filter_book(in genres varchar(255), in price_range varchar
2                               (255), in order_by varchar(255), in book_name varchar(255))
3  begin
4    declare cur_date date;
5    set cur_date = curdate();
6
7    create table book_with_discount as select book.book_id, title, edition,
8    price, have_.discount_id, discount_value from
9    book join have_ on book.book_id = have_.book_id
10   join discount on have_.discount_id = discount.discount_id where start_date <=
11     cur_date and end_date >= cur_date;
12
13   create table book_discount as select book.book_id, book.title, book.
14   edition, book.price, max(discount_value) as max_discount from
15   book left join book_with_discount on book.book_id = book_with_discount.
16   book_id group by book.book_id;
17
18   set SQL_SAFE_UPDATES = 0;
19   update book_discount set max_discount = 0 where max_discount is null;
20   alter table book_discount add column end_price double;
21   update book_discount set end_price = price * (100 - max_discount) / 100;
22
23   if genres is not null then
24     create temporary table filter_genre as select book_discount.book_id, title,
25     edition, price, 'Unknown type book' as book_type, max_discount, end_price
26     from
27       book_discount join genres_book on book_discount.book_id = genres_book
28       .book_id
29     where genres_book.genres = genres;
30   else
31     create temporary table filter_genre as
```



```
24      select book_id, title, edition, price, 'Unknown type book' as
25      book_type, max_discount, end_price from book_discount;
26      end if;
27
28      update filter_genre
29      set book_type = (
30          case
31              when book_id in (select book_id from audio_book) then 'Audio book'
32              when book_id in (select book_id from physical_book) then 'Physical
book'
33              when book_id in (select book_id from kindle_book) then 'Kindle book'
34              else 'Unknown book type'
35          end
36      );
37
38      if price_range is not null then
39          if (price_range <= 'low') then
40              create temporary table filter_price as select * from filter_genre where
filter_genre.price < 50000;
41          elseif (price_range <= 'mid') then
42              create temporary table filter_price as select * from filter_genre where
filter_genre.price > 50000 and filter_genre.price < 100000;
43          elseif (price_range <= 'high') then
44              create temporary table filter_price as select * from filter_genre where
filter_genre.price > 100000;
45          end if;
46      else create temporary table filter_price as select * from filter_genre;
47          end if;
48
49      if(book_name is not null) then
50          if order_by is not null then
51              if (order_by <= 'titleasc') then
52                  select * from filter_price where title = book_name order by title asc;
53              elseif (order_by <= 'titledesc') then
54                  select * from filter_price where title = book_name order by title desc;
55              elseif (order_by <= 'priceasc') then
56                  select * from filter_price where title = book_name order by price asc;
57              elseif (order_by <= 'pricedesc') then
58                  select * from filter_price where title = book_name order by price desc;
59              end if;
60          else select * from filter_price where title = book_name;
61              end if;
62      else
63          if order_by is not null then
64              if (order_by <= 'titleasc') then
65                  select * from filter_price order by title asc;
66              elseif (order_by <= 'titledesc') then
67                  select * from filter_price order by title desc;
68              elseif (order_by <= 'priceasc') then
69                  select * from filter_price order by price asc;
70              elseif (order_by <= 'pricedesc') then
71                  select * from filter_price order by price desc;
72              end if;
73          else select * from filter_price;
74              end if;
75      end if;
76
77      drop table book_with_discount;
78      drop table book_discount;
79      drop table filter_genre;
80      drop table filter_price;
81  end;
```

Listing 78: Thủ tục lọc sách theo tên sách, thể loại, khoảng giá, sắp xếp theo thứ tự

```
1  create procedure show_book_info(in book_id int)
2      begin
3          if not (select exists (select * from book where book.book_id = book_id)) then
4              signal sqlstate '45000' set message_text ='ID của sách không tồn tại';
```



```
5      end if;
6
7      -- Add rating
8
9      create table rating_temp as select
10         rate.book_id,
11             count(*) as rating_quantity,
12             avg(score) as rating_score
13     from rate
14         group by rate.book_id;
15         if not (select exists(select * from rating_temp where rating_temp.book_id
16 = book_id)) then
17             create table rating_table(book_id int not null, rating_quantity int not
18 null, rating_score float not null);
19                 insert into rating_table (book_id, rating_quantity, rating_score)
20 values (book_id, 0, 0);
21             else
22                 create table rating_table as select * from rating_temp where rating_temp.
23 book_id = book_id;
24             end if;
25
26             -- Add genres
27
28             create table genre_book as select rating_table.book_id, rating_quantity,
29 rating_score, genres from
30             rating_table left join genres_book on rating_table.book_id = genres_book.
31 book_id;
32
33             -- Add author
34
35             create table author_book as select genre_book.book_id, rating_quantity,
36 rating_score, genres, penname from
37             genre_book left join write_ on genre_book.book_id = write_.book_id
38             join author on write_.author_id = author.author_id;
39
40             -- Add series
41
42             if not (select exists (select * from consisted where consisted.book_id =
43 book_id)) then
44                 create table series_book as select author_book.book_id, rating_quantity,
45 rating_score, genres, penname from author_book;
46                     alter table series_book add series_name varchar(255);
47                     insert into series_book (series_name) values ('none');
48             else
49                 create table series_book as select author_book.book_id, rating_quantity,
50 rating_score, genres, penname, name_ as series_name from
51             author_book join consisted on author_book.book_id = consisted.book_id
52             join series on consisted.series_id = series.series_id;
53             end if;
54
55             -- Add discount
56
57             if not (select exists (select * from have_ where have_.book_id = book_id))
58 ) then
59                 create table discount_book as select series_book.book_id, rating_quantity,
60 rating_score, genres, penname, series_name from series_book;
61                     alter table discount_book add max_discount int;
62                     insert into discount_book (max_discount) values (0);
63                     create table discount_max (book_id int not null);
64                     create table discount_have (book_id int not null);
65             else
66                 create table discount_have as select series_book.book_id, have_.discount_id
67 , discount_value from
68                     series_book join have_ on series_book.book_id = have_.book_id
69                     join discount on have_.discount_id = discount.discount_id;
70                     create table discount_max as select book_id, max(discount_value) as
71 max_discount from discount_have group by book_id;
72                     create table discount_book as select series_book.book_id,
73 rating_quantity, rating_score, genres, penname, series_name, max_discount
74 from
75                     discount_max join series_book on discount_max.book_id = series_book.
```



```
    book_id;
60    end if;
61
62    -- Add detail
63
64        select discount_book.book_id, title, genres, penname, series_name,
65        reading_age, price, language_, edition, publication_date,
66        publisher_name, isbn, book.provider_id, quantity, max_discount,
67        rating_quantity, rating_score, audio_book.size as audio_size,
68        audio_book.time_ as audio_time,
69        kindle_book.size as kindle_size,
70        kindle_book.paper_length as kindle_paper_length,
71        physical_book.format_ as physical_format,
72        physical_book.dimensions as physical_dimensions,
73        physical_book.paper_length as physical_paper_length,
74        physical_book.weight as physical_weight,
75        physical_book.status_ as physical_status,
76        provider.name_ as provider_name
77    from discount_book
78    join book on discount_book.book_id = book.book_id
79        left join assign_db.audio_book on discount_book.book_id = audio_book.
book_id
80        left join assign_db.kindle_book on discount_book.book_id = kindle_book.
book_id
81        left join assign_db.physical_book on discount_book.book_id = physical_book.
book_id
82        left join assign_db.provider ON book.provider_id = provider.provider_id;
83        drop table rating_temp;
84        drop table rating_table;
85        drop table genre_book;
86        drop table author_book;
87        drop table series_book;
88        drop table discount_have;
89        drop table discount_max;
90        drop table discount_book;
91    end;
```

Listing 79: Thủ tục lấy thông tin một sách bất kì (kèm các thuộc tính riêng của loại sách đó, giá gốc, giá mới sau giảm)

```
1 create procedure add_order(in address varchar(255), in name_ varchar(255), in
2     phone_number varchar(255),
3     in ship_type varchar(255), in pay_method varchar(255), in customer_id int, in
4         provider_id int, out return_order_id int)
5 begin
6     declare cur_date datetime;
7         set cur_date = now();
8
9     insert into order_ (order_time, shipment_type, ship_fee, payment_method,
10         status_, address, name_, phone_number,
11         customer_id, provider_id, take_status, paid_status)
12         values (cur_date, ship_type, 15000, pay_method, 'đang giao', address,
13             name_, phone_number, customer_id, provider_id, 'chưa lấy',
14             'chưa trả');
15
16     select last_insert_id() as return_order_id;
17 end;
```

Listing 80: Thủ tục thêm đơn hàng mới vào bảng order_

```
1 create procedure add_book_to_order(in order_id int, in book_id int, in quantity
int)
2 begin
3     insert into contain_value (order_id, book_id, quantity);
4 end;
```

Listing 81: Thủ tục sách và số lượng vào bảng contain



```
1 create procedure add_promotion_code(in order_id int, in promotion_code_id int)
2 begin
3   if not (select exists(select * from promotion_code where promotion_code.code_id
4                         = promotion_code_id)) then
5     signal sqlstate '45000' set message_text ='Mã giảm giá không tồn tại';
6   end if;
7   insert into apply_for value (order_id, promotion_code_id);
8 end;
```

Listing 82: Thủ tục thêm mã giảm giá, đơn hàng áp dụng mã giảm giá vào bảng apply_for

```
1 create procedure confirm_order(in order_id int, in customer int)
2 begin
3   declare cusid int;
4   select customer_id into cusid from order_ where order_.order_id = order_id;
5
6   if not (cusid = customer) then
7     signal sqlstate '45000' set message_text ='Đơn hàng không thuộc về khách hàng
8       này';
9   end if;
10
11  insert into confirm value (order_id, customer);
12 end;
```

Listing 83: Thủ tục thêm xác nhận đơn hàng của khách hàng vào bảng confirm

```
1 create procedure delete_order(in order_id_param int)
2 begin
3   delete from assign_db.order_
4   where order_id = order_id_param;
5 end;
```

Listing 84: Thủ tục xóa một đơn hàng đã tạo

```
1 create procedure cal_price_order(in a_order_id int)
2 begin
3   Declare date_of_order datetime;
4   Declare code_promo_value int;
5   Declare code_max_promo_value int;
6   Declare temp_discount int;
7
8   select COALESCE(promo_value,0) into code_promo_value
9   from promotion_code
10  join apply_for on apply_for.promotion_code_id=promotion_code.code_id
11  where apply_for.order_id=a_order_id;
12
13  select COALESCE(maximum_promo,0) into code_max_promo_value
14  from promotion_code
15  join apply_for on apply_for.promotion_code_id=promotion_code.code_id
16  where apply_for.order_id=a_order_id;
17
18
19
20  Select order_time into date_of_order
21  from order_
22  where order_.order_id=a_order_id;
23
24  Create table discount_in_time AS
25  select discount_id, discount_value
26  from discount
27  where start_date<=date_of_order and date_of_order<=end_date;
28
29  Create table book_and_quantity AS
```



```
30  Select contain.book_id, contain.quantity, book.price
31    from contain
32    join book on contain.book_id=book.book_id
33  where order_id=a_order_id;
34
35 Create table book_and_discount AS
36   select book_and_quantity.book_id, book_and_quantity.quantity,
37   book_and_quantity.price, have_.discount_id
38   from book_and_quantity
39   left join have_ on book_and_quantity.book_id=have_.book_id;
40
41 Create table book_quantity_discount_value AS
42   Select book_id,quantity, price, discount_in_time.discount_value
43   from book_and_discount
44   left join discount_in_time on book_and_discount.discount_id=discount_in_time.
45   discount_id;
46
47   Create table price_of_order AS
48   SELECT
49     SUM(subquery.avg_price * subquery.avg_quantity) AS item_total,
50     SUM(subquery.avg_price *subquery.avg_quantity* subquery.max_discount_value/100)
51       AS discount_total,
52     SUM(subquery.avg_price *subquery.avg_quantity* (100-subquery.max_discount_value)
53       /100) AS grand_total
54   FROM
55   (
56     SELECT
57       AVG(book_id) AS avg_book_id,
58       AVG(quantity) AS avg_quantity,
59       AVG(price) AS avg_price,
60       COALESCE(MAX(discount_value),0) AS max_discount_value
61     FROM
62     book_quantity_discount_value
63     GROUP BY
64       book_id
65   )As subquery;
66
67   Select discount_total into temp_discount
68   from price_of_order;
69
70 if(code_promo_value>0) then
71   set SQL_SAFE_UPDATES = 0;
72   update price_of_order set discount_total=temp_discount+least(item_total*
73   code_promo_value,code_max_promo_value);
74   update price_of_order set grand_total=item_total-discount_total;
75   set SQL_SAFE_UPDATES = 1;
76 end if;
77
78 Select *
79   from price_of_order;
80
81 drop table discount_in_time;
82 drop table book_and_quantity;
83 drop table book_and_discount;
84 drop table book_quantity_discount_value;
85 drop table price_of_order;
86
87 end
```

Listing 85: Thủ tục tính tổng giá gốc, tổng giảm giá và tổng tiền phải trả chưa bao gồm phí giao hàng của đơn hàng