

TRƯỜNG ĐẠI HỌC KHOA HỌC XÃ HỘI & NHÂN VĂN TP.HCM
KHOA THƯ VIỆN – THÔNG TIN HỌC



MÔN HỌC: CƠ SỞ DỮ LIỆU 2

GIẢNG VIÊN: ThS. TRẦN ĐÌNH ANH HUY

Hồ Chí Minh, tháng 5 năm 2023

MỤC LỤC

1. Tổng quan đề tài	1
1.1 Sự cần thiết của dự án	1
1.2 Hiện trạng	1
1.3 Các chức năng của dự án.....	2
1.3.1. Chức năng cơ bản của hệ thống	2
1.3.2. Chức năng cần thiết của hệ thống	3
1.3.3. Chức năng nâng cao của hệ thống.....	3
2. Phân chia nhiệm vụ	5
3. Tài liệu thiết kế	6
3.1 Tư duy thiết kế: Mô hình phân rã chức năng (BFD).....	6
3.2 Phân tích yêu cầu người dùng (nhân viên cửa hàng).....	7
3.3 Mô hình E-R	8
3.4 Mô hình mức vật lý: Từ điển dữ liệu: Phân tích các ràng buộc về kiểu dữ liệu, liên kết... ..	19
4. Cài đặt cơ sở dữ liệu	26
4.1 Cài đặt CSDL.....	26
4.2 Query thể hiện chức năng của CSDL (<i>dưới dạng stored procedure</i>)	32
4.3 Các Trigger để ngăn chặn lỗi của CSDL	41
5. Kết luận & hướng phát triển	62
5.1 Nêu các chức năng có thể phát triển thêm/cần phát triển thêm cho dự án .	62
5.2 Nêu hướng phát triển về quy mô đề tài hoặc các hệ thống quản lý khác có thể sử dụng lại CSDL vừa thiết kế.....	64
NGUỒN THAM KHẢO.....	64

DANH MỤC HÌNH

Hình 1. Mô hình phân rã chức năng của hệ thống.....	7
Hình 2. Mô hình ERD.....	13
Hình 3. Lược đồ quan hệ	19

DANH MỤC BẢNG

Bảng 1. Bảng phân chia công việc cho các thành viên trong nhóm.....	6
Bảng 2. Bảng xác định chức năng thực thể	9
Bảng 3. Bảng xác định thuộc tính của thực thể	12
Bảng 4. Bảng xác định thuộc tính của các bảng trong mô hình dữ liệu quan hệ	18
Bảng 5. Bảng xác định khóa chính, khóa ngoại của các bảng trong mô hình dữ liệu quan hệ.....	18
Bảng 6. Bảng thông tin về khóa chính	25
Bảng 7. Bảng thông tin về khóa ngoại	26

1. Tổng quan đề tài

1.1 Sự cần thiết của dự án

Trên thế giới công nghệ thông tin đang ngày càng phát triển và ứng dụng rộng rãi trong nhiều lĩnh vực, việc sử dụng cơ sở dữ liệu để quản lý thông tin và dữ liệu trở nên ngày càng phổ biến. Và đang dần trở thành điều kiện tiên quyết cho sự sống còn của mỗi doanh nghiệp.

Xây dựng một cơ sở dữ liệu cho lĩnh vực bán lẻ nói chung và cơ sở dữ liệu cho cửa hàng sách nói riêng là cần thiết vì nó giúp quản lý và lưu trữ thông tin về sản phẩm (sách), khách hàng, hoạt động kinh doanh,...

Bằng cách tạo ra một cơ sở dữ liệu, thông tin/dữ liệu của cửa hàng sẽ được quản lý, tổ chức một cách logic và khoa học trên cơ sở dữ liệu để từ đó mang lại những giá trị cho cửa hàng: quản lý các sản phẩm của cửa hàng, thông tin khách hàng, kho hàng, đơn đặt hàng,..và đặc biệt cơ sở dữ liệu này sẽ là tiền đề quan trọng trong việc đưa ra các báo cáo và thống kê về doanh thu và lượng sách bán ra, giúp cho quản lý cửa hàng ra quyết định, chiến lược kinh doanh phù hợp hơn.

Ngoài ra, việc sử dụng cơ sở dữ liệu để quản lý cửa hàng giúp tiết kiệm thời gian, chi phí và công sức trong công tác quản lý. Giúp cho việc quản lý cửa hàng trở nên dễ dàng hơn so với quản lý truyền thống.

Vì vậy, xây dựng một cơ sở dữ liệu cho một cửa hàng sách là cần thiết để tối ưu hóa quản lý cửa hàng và nâng cao hiệu quả kinh doanh.

1.2 Hiện trạng

Cửa hàng sách [A] với quy mô 16.000 đầu sách. Mỗi ngày có tới hàng trăm dữ liệu mới được sinh ra (dữ liệu khách hàng đã mua hàng, dữ liệu sách đã bán, dữ liệu làm việc của nhân viên,...). Và các dữ liệu này chỉ được quản lý bằng cách

truyền thống (lưu trữ trong excel), với số lượng thông tin nhiều như vậy thì cách lưu trữ truyền thống đã quá tải và không có khả năng tiếp tục quản lý thông tin cho cửa hàng.

Trước khi cách quản lý truyền thống bằng excel quá tải thì cách quản lý này cũng đã xuất hiện nhiều lỗi hỏng: có quá nhiều file excel lưu trữ dữ liệu dẫn đến bất tiện để tìm kiếm và tra cứu thông tin đồng thời còn rất mất thời gian, công sức; Khi quản lý thông tin trên bảng tính, thông tin thường hay bị nhầm lẫn hoặc bị mất mát dễ dàng, đặc biệt là khi số lượng sản phẩm và thông tin khách hàng lớn; Dễ xảy ra sai sót trong quá trình nhập thông tin, lưu trữ và truy xuất dữ liệu, đặc biệt là khi số lượng thông tin lớn; Lưu trữ thông tin trong excel gây mất thời gian trong việc xử lý dữ liệu và đưa ra các báo cáo thống kê.

Từ những hạn chế trên việc xây dựng một cơ sở dữ liệu để quản lý thông tin của hàng sách [A] là vô cùng hợp lý và cần được triển khai nhanh chóng để giúp ích cho việc đưa ra các quyết định kinh doanh tối ưu hơn.

1.3 Các chức năng của dự án

1.3.1. Chức năng cơ bản của hệ thống

Chức năng cơ bản là là loại chức năng bắt buộc phải có để trở thành một hệ thống. Nếu không có những chức năng cơ bản này thì hệ thống không thể tồn tại. Do đó, hệ thống quản lý thông tin cho cửa hàng sách [A] bao gồm các chức năng cơ bản sau:

- Sản phẩm được mua bởi khách hàng nào
- Sản phẩm thuộc danh mục nào và danh mục đó có được giảm giá không
- Sản phẩm được mua bởi khách hàng nào, thuộc đơn hàng nào và loại thanh toán là gì
- Sản phẩm được cung cấp bởi ai và nằm trong danh mục nào.
- Khách hàng muốn biết sản phẩm mình mua được viết bởi tác giả nào và do ai là người cung cấp

- Sản phẩm thuộc đơn hàng nào với số lượng sản phẩm là bao nhiêu với khách hàng là ai
- Khách hàng mua sản phẩm nào nhiều hơn, ít hơn để có thể ra chiến lược kinh doanh (sách nào nhập nhiều sách nào nhập ít)
- Nhập tên người dùng để xem trạng thái thanh toán và trạng thái của đơn hàng
- Số lượng sản phẩm có trong giỏ hàng của khách hàng
- Tổng doanh thu sản phẩm
- Xếp hạng danh mục được yêu thích theo số lượng sản phẩm bán được
- Doanh thu của danh mục
- Xếp hạng doanh thu của danh mục
- Sản phẩm được đánh giá bởi khách hàng nào với số điểm và nội dung là gì

1.3.2. Chức năng cần thiết của hệ thống

Chức năng cần thiết bao gồm những chức năng cơ bản và một số chức năng có khả năng hỗ trợ thêm cho người dùng tạo nên sự thuận tiện trong công việc. Nếu không có chức năng cần thiết thì hệ thống vẫn sẽ được đảm bảo là một hệ thống nhưng nó sẽ là một hệ thống không hoàn chỉnh, hoạt động không hiệu quả. Do vậy, hệ thống quản lý của cửa hàng sách [A] bao gồm các chức năng cần thiết sau:

- Sản phẩm được mua bởi khách hàng nào, với giá bao nhiêu thuộc đơn hàng nào và đơn hàng đó có được giảm giá không và địa chỉ nhận của đơn hàng
- Xác định nhà cung cấp tiềm năng theo doanh thu số lượng sản phẩm
- Dựa vào số lượng các sản phẩm bán được và độ tồn kho của sản phẩm để đưa ra chiến lược kinh doanh (ngưng nhập sách bán chậm, đưa ra các mã giá cho sách bán chậm,...)

1.3.3. Chức năng nâng cao của hệ thống

Chức năng nâng cao bao gồm cả những chức năng cơ bản, chức năng cần thiết và những chức năng mở rộng, đáp ứng những yêu cầu cao của người dùng, giúp người dùng tiết kiệm thời gian, công sức thậm chí là tiền bạc. Nếu không có chức năng nâng cao thì hệ thống vẫn sẽ hoạt động bình thường. Nhưng đối với hệ thống quản lý thông tin cho cửa hàng sách [A] nhóm muốn có các chức

năng nâng cao để nhằm cải thiện, tăng hiệu quả năng suất làm việc của người dùng (nhân viên). Chính vì vậy, các chức năng nâng của hệ thống gồm:

- Thêm một đơn hàng mới gồm chi tiết đơn hàng và chi tiết thanh toán, trong đó người dùng nhập vào mã khách hàng và các thông tin cần thiết
- Cho phép nhập vào lựa chọn và tháng, truy xuất ra khách hàng và gán xếp hàng cho khách hàng. Lựa chọn ở đây được chọn là toàn hệ thống và chọn theo tháng
- Cho phép nhập vào tên sản phẩm, mã danh mục, khoảng giá và đánh giá cùng với loại lọc và bộ lọc. Có 2 loại lọc là chính xác và tương đương, tương ứng với bộ lọc - là kết quả tìm kiếm thuần hay kết quả tìm kiếm thông qua lọc. Tương ứng với từng loại lọc sẽ ra được kết quả theo danh mục, theo đánh giá, theo khoảng giá hay theo kết hợp
- Khách hàng nhập vào tổng số tiền để mua combo, đồng thời lựa chọn các danh mục muốn mua, hệ thống đưa ra được gợi ý về các sản phẩm tương ứng
- Cập nhật giá sản phẩm dựa trên tìm kiếm tương đương hoặc tìm kiếm chính xác với mã sp và tên sp, tên danh mục và mã danh mục với hệ số $[-1,1]$
- Tính tổng doanh thu của từng sản phẩm theo mã danh mục hoặc theo tất cả trong khoảng thời gian xác định, đưa ra doanh thu tổng cho từng lựa chọn --
- Gợi ý cho khách hàng về sản phẩm sẽ mua, dựa trên lịch sử mua hàng mà chọn ra danh mục thích nhất, lấy các sản phẩm --còn lại trong danh mục đó rồi đưa ra gợi ý cho khách hàng. nếu makh là null sẽ đưa ra gợi ý cho toàn bộ khách hàng
- Thêm mới hoặc update cho bảng sản phẩm, bảng danh mục và bảng nhà cung cấp cùng với lựa chọn tương ứng, MaSP, Ma_NCC và MaDanhMuc được tạo tự động.
- Xem lịch sử khách hàng
- Cho phép khách hàng xem được các mã giảm giá hiện tại dựa trên đơn hàng, danh mục sản phẩm hay dựa trên sản phẩm
- Tính trung bình đánh giá

2. Phân chia nhiệm vụ

MSSV	Họ và tên	Nhiệm vụ	Đánh giá
2156210075	Đoàn Thiên Thiên	<ul style="list-style-type: none"> - Thành viên nhóm thiết kế - Biên tập viên 	<ul style="list-style-type: none"> - Tham dự tất cả các buổi họp nhóm; - Có tham gia làm bài cùng các thành viên khác trong nhóm; - Có hỗ trợ các thành viên khác trong việc thực hiện nhiệm vụ; - Tích cực đưa ra các ý kiến; - Hoàn thành tốt nội dung được giao
2156210125	Hoàng Xuân Quốc	<ul style="list-style-type: none"> - Thành viên nhóm phát triển - Nhóm trưởng 	<ul style="list-style-type: none"> - Tham dự tất cả các buổi họp nhóm; - Có tham gia làm bài cùng các thành viên khác trong nhóm; - Có hỗ trợ các thành viên khác trong việc thực hiện nhiệm vụ; - Tích cực đưa ra các ý kiến; - Hoàn thành tốt nội dung được giao
2156210095	Đặng Hoàng Chiến	<ul style="list-style-type: none"> - Thành viên nhóm phát triển - Thành viên nhóm thiết kế 	<ul style="list-style-type: none"> - Tham dự tất cả các buổi họp nhóm; - Có tham gia làm bài cùng các thành viên khác trong nhóm; - Có hỗ trợ các thành viên khác trong việc thực hiện nhiệm vụ; - Tích cực đưa ra các ý kiến; - Hoàn thành tốt nội dung được giao

2156210110	Bùi Văn Kiệt	<ul style="list-style-type: none"> - Phát ngôn viên - Biên tập viên 	<ul style="list-style-type: none"> - Tham dự tất cả các buổi họp nhóm; - Có tham gia làm bài cùng các thành viên khác trong nhóm; - Có hỗ trợ các thành viên khác trong việc thực hiện nhiệm vụ; - Tích cực đưa ra các ý kiến; - Hoàn thành tốt nội dung được giao
------------	--------------	---	---

Bảng 1. Bảng phân chia công việc cho các thành viên trong nhóm

3. Tài liệu thiết kế

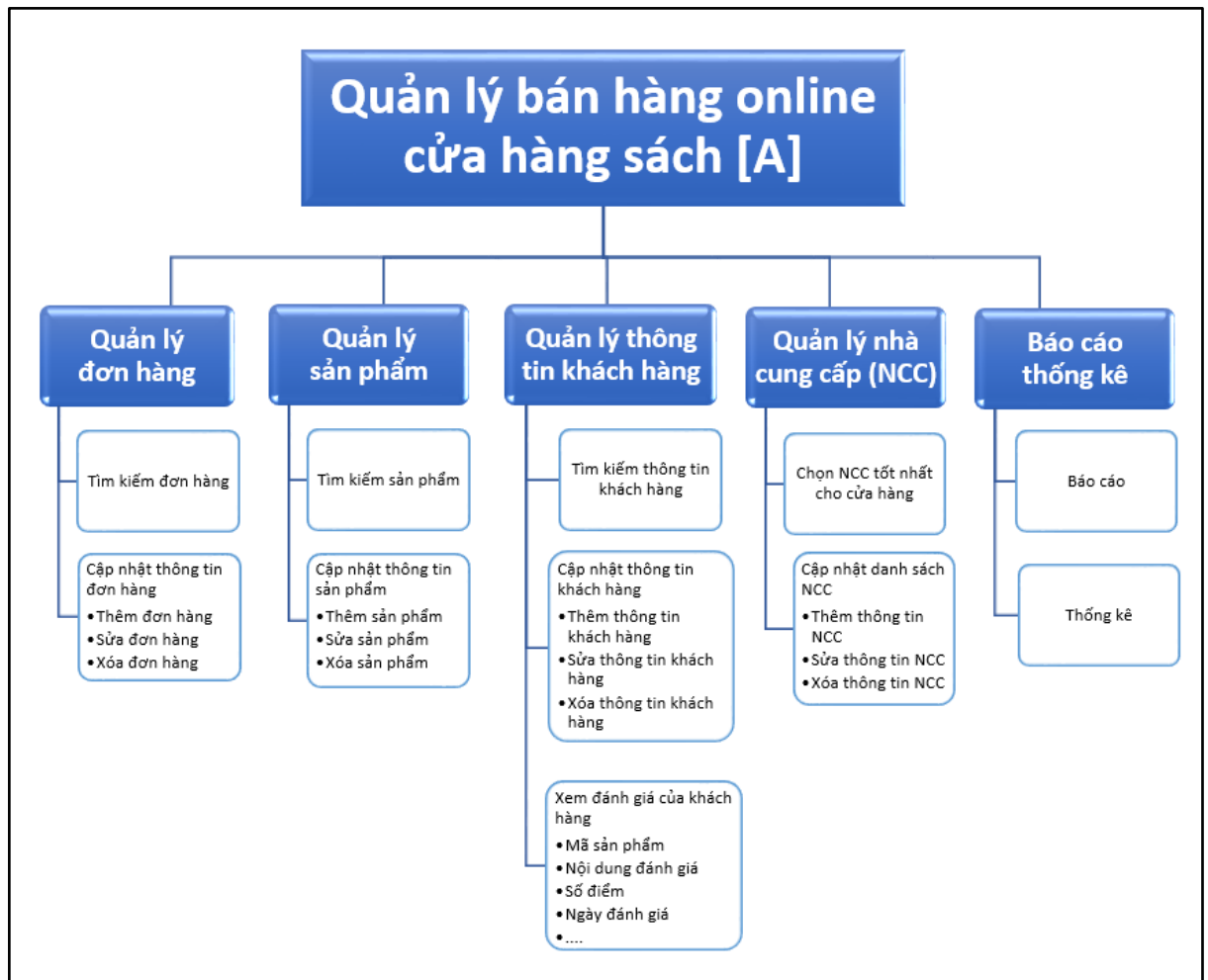
3.1 Tư duy thiết kế: Mô hình phân rã chức năng (BFD)

Với hiện trạng của cửa hàng sách [A] đã nêu ở phần 1, nhóm lựa chọn phương pháp thiết kế:

Top Down Design: Phương pháp phân rã các chức năng

Mô hình phân rã chức năng (BFD - Business Function Diagram) là công cụ biểu diễn việc phân rã một chức năng tổng hợp thành những chức năng chi tiết hơn. Số mức chia ra phụ thuộc kích cỡ và độ phức tạp của hệ thống.[1]

Phương pháp này được trình bày đối với cửa hàng [A] cụ thể như sau:



Hình 1. Mô hình phân rã chức năng của hệ thống

3.2 Phân tích yêu cầu người dùng (nhân viên cửa hàng)

- Quản lý thông tin sách: Nhân viên cần có khả năng quản lý các thông tin về sách như tên sách, tác giả, NXB, năm xuất bản, số lượng tồn kho, giá bán, vị trí trên kệ, thể loại sách và các thông tin liên quan đến sách khác. Đây là các thông tin cơ bản giúp nhân viên có thể xem và cập nhật thông tin sách trong cửa hàng.
- Quản lý thông tin khách hàng: Các thông tin về khách hàng cũng là một phần quan trọng của cơ sở dữ liệu. Nhân viên cần có khả năng quản lý các thông tin về khách hàng như tên, địa chỉ, số điện thoại, email, số lượng sách đã mua, lịch sử mua hàng, tài khoản khách hàng và các thông tin khác. Nhân viên cần có thể truy xuất thông tin khách hàng để hỗ trợ trong việc tư vấn sách, cập nhật thông

tin khách hàng mới, quản lý các chương trình khuyến mãi, tặng quà cho khách hàng.

- Quản lý kho: Nhân viên cần quản lý thông tin về số lượng sách nhập vào cửa hàng, số lượng sách bán ra, số lượng tồn kho, giá nhập sách, giá bán sách và các thông tin liên quan đến nhập/xuất sách. Đây là các thông tin cơ bản giúp nhân viên có thể kiểm soát lượng sách trong kho và có thể đưa ra quyết định mua/bán sách cho cửa hàng.
- Báo cáo doanh số: Nhân viên cần có thể tạo ra các báo cáo về doanh số sách, tồn kho, doanh số bán hàng, lợi nhuận, số lượng khách hàng, các sản phẩm bán chạy nhất, đánh giá hiệu quả của các chương trình khuyến mãi,...
- Quản lý các đơn đặt hàng: Nhân viên cần quản lý các thông tin liên quan đến các đơn đặt hàng từ khách hàng hoặc từ các nhà cung cấp sách. Nhân viên cần có khả năng xem các đơn hàng, tình trạng đơn hàng, cập nhật thông tin đơn hàng, hủy đơn hàng, tạo đơn hàng mới và thực hiện các công việc liên quan đến đơn hàng.
- Quản lý chương trình khuyến mãi: Nhân viên cần quản lý thông tin về các chương trình khuyến mãi, giảm giá sách, các ưu đãi dành cho khách hàng, các quà tặng, voucher giảm giá,... để hỗ trợ cho việc bán hàng và thu hút khách hàng đến cửa hàng.
- Quản lý thông tin nhà cung cấp sách: Nhân viên cần quản lý các thông tin về nhà cung cấp sách, tên nhà cung cấp, địa chỉ, số điện thoại, email, thông tin liên hệ, thông tin sách mà nhà cung cấp cung cấp, để tiện cho việc liên hệ, đặt mua sách và theo dõi tiến độ đặt sách.

Những nhu cầu trên, chính là cơ sở để xác định các thực thể và mối quan hệ cơ bản cần có trong cơ sở dữ liệu cửa hàng sách [A].

3.3 Mô hình E-R

- Xác định các thực thể

STT	Tên thực thể	Chức năng
1	Khách hàng	Giúp cửa hàng quản lý những thông tin cần thiết về những khách hàng
2	Nhà cung cấp	Giúp cửa hàng quản lý những thông tin cần thiết về nhà cung cấp các sản phẩm
3	Đơn hàng	Giúp cửa hàng quản lý các đơn hàng của khách hàng một cách dễ dàng hơn
4	Sản phẩm	Giúp cửa hàng quản lý các sản phẩm hiện tại và sản phẩm đang có trong kho
5	Danh mục	Giúp cửa hàng phân loại các sản phẩm thành những nhóm danh mục để dễ dàng tra cứu, tìm kiếm
6	Giảm giá đơn hàng	Giúp cửa hàng kiểm soát các loại giảm giá áp dụng cho từng đơn hàng
7	Địa chỉ nhận	Giúp cửa hàng biết được thông tin địa chỉ hàng cần giao mà khách hàng đã đặt
8	Đăng nhập	Giúp cửa hàng kiểm soát khách hàng thông qua tài khoản của từng khách hàng
9	Thanh toán	Giúp cửa hàng nắm rõ thông tin về việc thanh toán của khách hàng

Bảng 2. Bảng xác định chức năng thực thể

- Xác định thuộc tính của thực thể

Tên thực thể	Tên thuộc tính
Khách hàng	Mã khách hàng
	Tên người dùng
	Mật khẩu
	Tên khách hàng

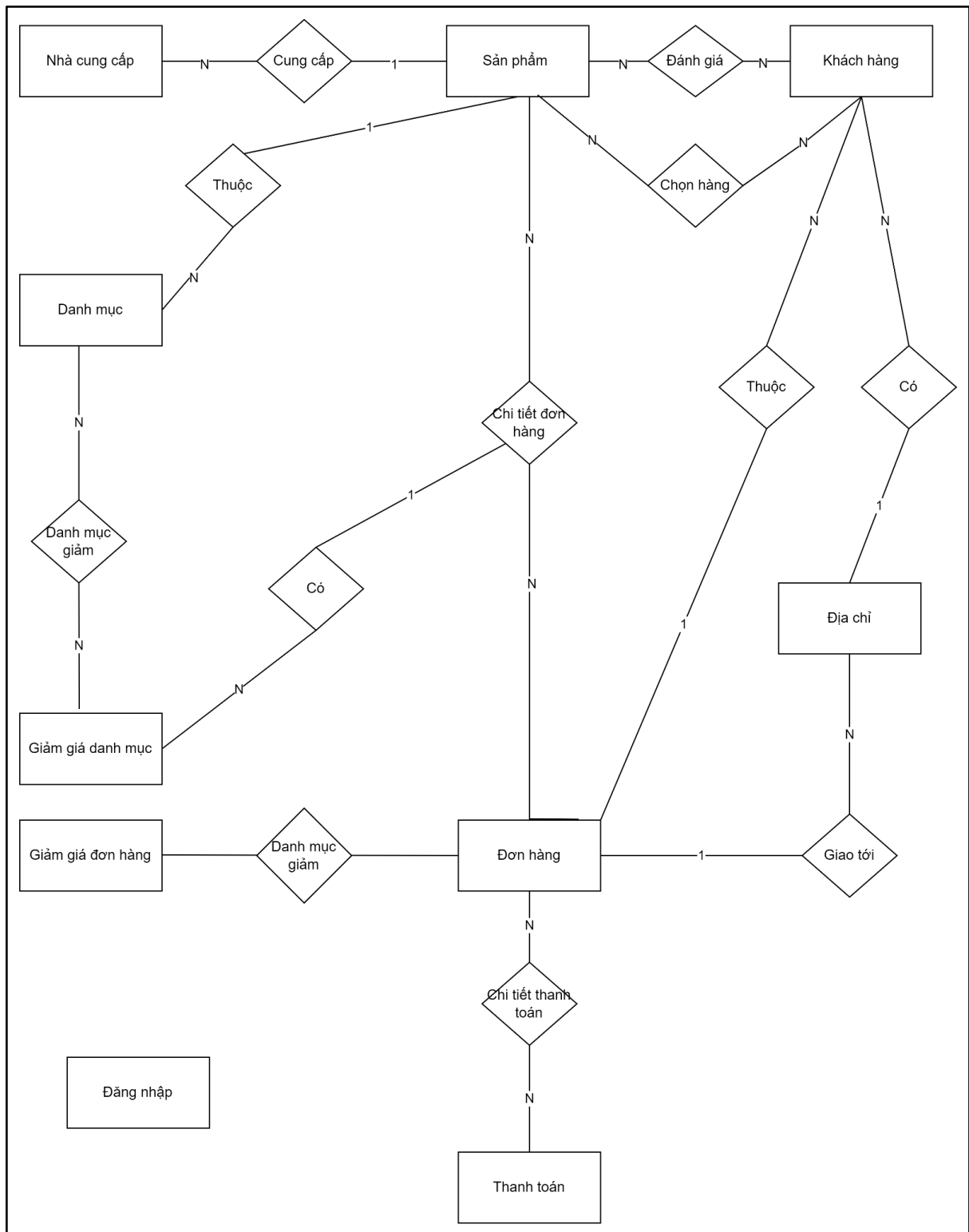
	Ngày sinh
	Giới tính
	Số điện thoại
	Email
Nhà cung cấp	Mã nhà cung cấp
	Tên nhà cung cấp
	Số điện thoại
Đơn hàng	Mã đơn hàng
	Mã khách hàng
	Mã giảm giá đơn hàng
	Mã địa chỉ
	Ngày đặt
	Ngày giao
	Tổng đơn giá
	Giảm giá đơn hàng
	Phí vận chuyển
	Thành tiền
	Trạng thái
	Thời gian tạo
	Thời gian cập nhật
Sản phẩm	Mã sản phẩm
	Mã danh mục

	Mã nhà cung cấp
	Tên sản phẩm
	Giá sản phẩm
	Tác giả
	Bìa
	Tồn kho
	Thời gian tạo
	Thời gian cập nhật
Danh mục	Mã danh mục
	Tên danh mục
Giảm giá danh mục	Mã giảm giá danh mục
	Giá trị
	Loại giá trị
	Nội dung
	Số lượng mã
	Giá trị tối thiểu
	Giảm tối đa
	Thời gian bắt đầu
	Thời gian kết thúc
	Trạng thái
	Thời gian tạo
	Thời gian cập nhật

Địa chỉ nhận	Mã địa chỉ
	Mã khách hàng
	Họ tên người nhận
	Số điện thoại
	Địa chỉ
	Quận huyện
	Thành phố
	Ghi chú
Đăng nhập	Mã đăng nhập
	Tên đăng nhập
	Mật khẩu
	Trạng thái
Thanh toán	Mã thanh toán
	Loại thanh toán

Bảng 3. Bảng xác định thuộc tính của thực thể

- Xác định các quan hệ (vẽ ERD)



Hình 2. Mô hình ERD

Vì không đủ chỗ nên nhóm em xin để mô hình ERD chi tiết tại đường link sau:
https://app.diagrams.net/?src=about#G1HVLxEQdtzOsbltYHHNt2GqZ_9BibA10c

3.4 Mô hình dữ liệu quan hệ

- Xác định các bảng và thuộc tính của từng bảng

STT	Tên bảng	Thuộc tính của bảng
1	Khách hàng	Mã khách hàng
		Tên người dùng
		Mật khẩu
		Tên khách hàng
		Ngày sinh
		Giới tính
		Số điện thoại
		Email
2	Nhà cung cấp	Mã nhà cung cấp
		Tên nhà cung cấp
		Số điện thoại
3	Đơn hàng	Mã đơn hàng
		Mã khách hàng
		Mã giảm giá đơn hàng
		Mã địa chỉ
		Ngày đặt
		Ngày giao
		Tổng đơn giá
		Giảm giá đơn hàng
		Phí vận chuyển
		Thành tiền
		Trạng thái

		Thời gian tạo
		Thời gian cập nhật
4	Chi tiết đơn hàng	Mã đơn hàng
		Mã sản phẩm
		Mã giảm giá danh mục
		Số lượng
		Đơn giá
		Lượng giảm
5	Giỏ hàng	Mã khách hàng
		Mã sản phẩm
		Tên sản phẩm
		Số lượng
		Thời gian tạo
		Thời gian cập nhật
6	Sản phẩm	Mã sản phẩm
		Mã danh mục
		Mã nhà cung cấp
		Tên sản phẩm
		Giá sản phẩm
		Tác giả
		Bìa
		Tồn kho
		Thời gian tạo
		Thời gian cập nhật

7	Đánh giá	Mã đơn hàng
		Mã sản phẩm
		Nội dung
		Số điểm
		Ngày đánh giá
		Thời gian tạo
		Thời gian cập nhật
8	Danh mục	Mã danh mục
		Tên danh mục
9	Giảm giá danh mục	Mã giảm giá danh mục
		Giá trị
		Loại giá trị
		Nội dung
		Số lượng mã
		Giá trị tối thiểu
		Giảm tối đa
		Thời gian bắt đầu
		Thời gian kết thúc
		Trạng thái
		Thời gian tạo
		Thời gian cập nhật
10	Danh mục giảm	Mã giảm giá danh mục
		Mã danh mục
11	Giảm giá đơn hàng	Mã giảm giá đơn hàng

		Giá trị
		Loại giá trị
		Nội dung
		Số lượng mã
		Giá trị tối thiểu
		Giảm tối đa
		Thời gian bắt đầu
		Thời gian kết thúc
		Trạng thái
		Thời gian tạo
		Thời gian cập nhật
12	Địa chỉ nhận	Mã địa chỉ
		Mã khách hàng
		Họ tên người nhận
		Số điện thoại
		Địa chỉ
		Quận huyện
		Thành phố
		Ghi chú
13	Đăng nhập	Mã đăng nhập
		Tên đăng nhập
		Mật khẩu
		Trạng thái
14	Thanh toán	Mã thanh toán

		Loại thanh toán
15	Chi tiết thanh toán	Mã thanh toán
		Mã đơn hàng
		Ngày thanh toán
		Trạng thái thanh toán
		Thời gian tạo
		Thời gian cập nhật

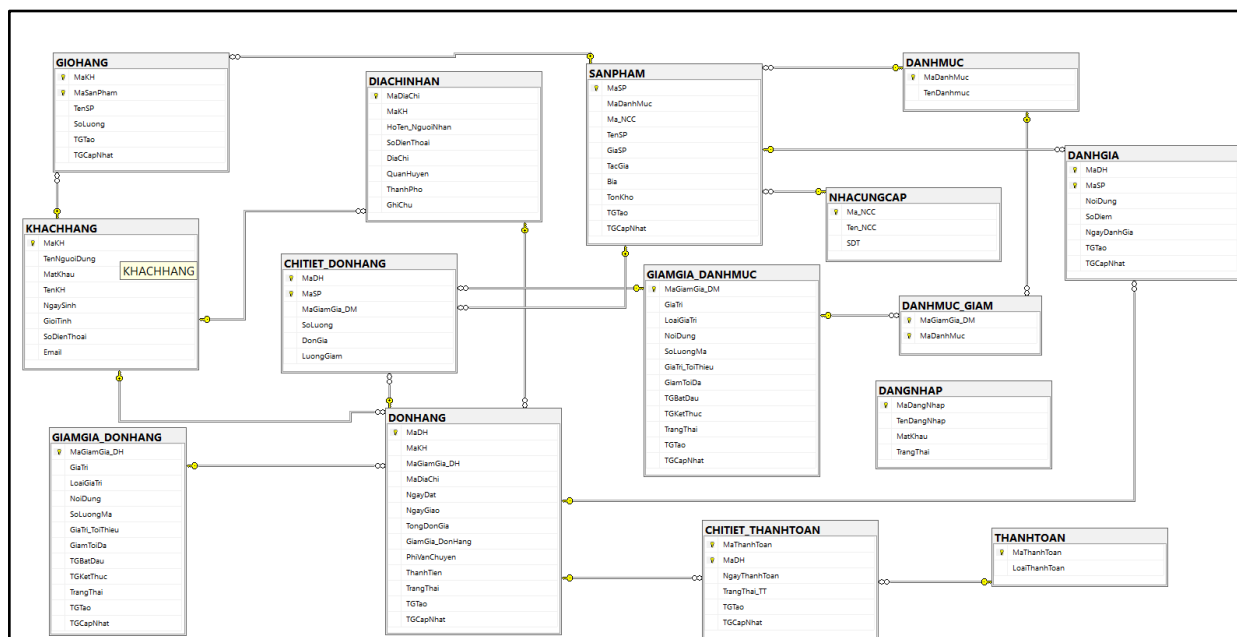
Bảng 4. Bảng xác định thuộc tính của các bảng trong mô hình dữ liệu quan hệ

- Xác định khóa chính, khóa ngoại

Tên bảng	Khoá chính	Khoá ngoại (nếu có)
SanPham	MaSP	MaDanhMuc; Ma_NCC
NhaCungCap	Ma_NCC	
DanhMuc	MaDanhMuc	
GiamGia_DonHang	MaGiamGia_DH	
GiamGia_DanhMuc	MaGiamGia_DM	
ThanhToan	MaThanhToan	
DonHang	MaDH	MaKH, MaGiamGia_DH, MaDiaChi.
DiaChiNhan	MaDiaChi	MaKH
KhachHang	MaKH	
DangNhap	MaDangNhap	

Bảng 5. Bảng xác định khóa chính, khóa ngoại của các bảng trong mô hình dữ liệu quan hệ

- Vẽ lược đồ quan hệ (Có thể code database trên SQL xong chọn view diagram -- chụp màn hình – crop ảnh).



Hình 3. Lược đồ quan hệ

3.4 Mô hình mức vật lý: Từ điển dữ liệu: Phân tích các ràng buộc về kiểu dữ liệu, liên kết...

- Thông tin về Metadata

Tên bảng	Tên thuộc tính	Kiểu dữ liệu	Tham chiếu	Giá trị mặc định	NOT NULL
KHACHHANG	MaKH	varchar(10)			TRUE
	TenNguoiDung	varchar(255)			TRUE
	MatKhau	varchar(255)			
	TenKH	varchar(255)			TRUE
	NgaySinh	datetime			
	GioiTinh	nvarchar(3)			
	SoDienThoai	char(20)			TRUE

	Email	varchar(50)			
NHA CUNG CAP	Ma_NCC	varchar(10)			TRUE
	Ten_NCC	nvarchar(255)			TRUE
	SDT	char(20)			
DON HANG	MaDH	varchar(10)			TRUE
	MaKH	varchar(10)	KHACH HANG		TRUE
	MaGiamGia_D H	varchar(10)	GIAMG IA_DO NHAN G		
	MaDiaChi	varchar(10)	DIACHI NHAN		
	NgayDat	datetime		Ngày hiện tại	TRUE
	NgayGiao	datetime			
	TongDonGia	numeric(16, 2)			
	GiamGia_Don Hang	numeric(16, 2)			
	PhiVanChuyen	numeric(16, 2)			
	ThanhTien	numeric(16, 2)			
	TrangThai	nvarchar(30)			TRUE
	TGTao	datetime			
	TGCapNhat	datetime			
CHITIET_D ONHANG	MaDH	varchar(10)	DONH ANG		
	MaSP	varchar(10)	SANPH AM		

	MaGiamGia_DM	varchar(10)	DANH MUC_GIAM		
	SoLuong	numeric(16, 2)			
	DonGia	numeric(16, 2)			
	LuongGiam	numeric(16, 2)			
GIAMGIA_DONHANG	MaGiamGia_DH	varchar(10)			TRUE
	GiaTri	numeric(16, 2)			
	LoaiGiaTri	varchar(20)			
	NoiDung	nvarchar(255)			
	SoLuongMa	int			
	GiaTri_Toithieu	numeric(16, 2)			
	GiamToiDa	numeric(16, 2)			
	TGBatDau	datetime			
	TGKetThuc	datetime			
	TrangThai	tinyint			
	TGTao	datetime		Ngày hiện tại	
	TGCapNhat	datetime			
SANPHAM	MaSP	varchar(10)			TRUE
	MaDanhMuc	varchar(10)	DANH MUC		TRUE
	Ma_NCC	varchar(10)	NHACU NGCAP		
	TenSP	nvarchar(255)			TRUE

	GiaSP	numeric(16, 2)			
	TacGia	nvarchar(255)			
	Bia	nvarchar(20)			
	TonKho	int			
	TGTao	datetime		Ngày hiện tại	
	TGCapNhat	datetime			
GIOHANG	MaKH	varchar(10)	KHACH HANG		TRUE
	MaSP	varchar(10)	SANPH AM		TRUE
	TenSP	nvarchar(255)			TRUE
	SoLuong	int			
	TGTao	datetime		Ngày hiện tại	
	TGCapNhat	datetime			
DANHMUC	MaDanhMuc	varchar(10)			TRUE
	TenDanhMuc	nvarchar(255)			TRUE
DANHMUC_ GIAM	MaGiamGia_DM	varchar(10)	GIAMG IA_DA NHMU C		TRUE
	MaDanhMuc	varchar(10)	DANH MUC		TRUE
GIAMGIA_D ANHMUC	MaGiamGia_DM	varchar(10)			TRUE
	GiaTri	numeric(16, 2)			
	LoaiGiaTri	varchar(20)			

	NoiDung	nvarchar(255)			
	SoLuongMa	int			
	GiaTri_Toithieu	numeric(16, 2)			
	GiamToiDa	numeric(16, 2)			
	TGBatDau	datetime			
	TGKetThuc	datetime			
	TrangThai	tinyint			
	TGTao	datetime			
	TGCapNhat	datetime			
THANH TOAN	MaThanhToan	varchar(10)			TRUE
	LoaiThanhToan	nvarchar(20)			TRUE
CHITIET_THANH TON	MaThanhToan	varchar(10)	THANH TOAN		TRUE
	MaDH	varchar(10)	DONH ANG		TRUE
	NgayThanhToan	datetime			
	TrangThai_TT	nvarchar(30)			TRUE
	TGTao	datetime		Ngày hiện tại	
	TGCapNhat	datetime			
DIACHINH	MaDiaChi	varchar(10)			TRUE
	MaKH	varchar(10)	KHACH HANG		TRUE
	HoTen_Người Nhân	nvarchar(255)			

	SoDienThoai	char(20)	KHACH HANG		
	DiaChi	nvarchar(1000)			TRUE
	QuanHuyen	nvarchar(1000)			TRUE
	ThanhPho	nvarchar(1000)			TRUE
	GhiChu	nvarchar(1000)			
DANH GIA	MaDH	varchar(10)	DONH ANG		TRUE
	MaSP	varchar(10)	SANPH AM		TRUE
	NoiDung	nvarchar(1000)			
	SoDiem	char(1)			TRUE
	NgayDanhGia	datetime		Ngày hiện tại	
	TGTao	datetime			
	TGCapNhat	datetime			
DANGNHAP	MaDangNhap				TRUE
	TenDangNhap				TRUE
	MatKhau				TRUE
	TrangThai				TRUE

Bảng 6. Bảng thông tin về Metadata

- Thông tin về khóa chính

STT	Tên bảng	Tên khoá	Thuộc tính khóa
1	SanPham	pk_SANPHAM	MaSP
2	NhaCungCap	pk_NHACUNGCAP	Ma_NCC
3	DanhMuc	pk_DANHMUC	MaDanhMuc

4	GiamGia_DonHang	pk_GIAMGIADONHANG	MaGiamGia_DH
5	GiamGia_DanhMuc	pk_GIAMGIADANHMUC	MaGiamGia_DM
6	ThanhToan	pk_THANHTOAN	MaThanhToan
7	DonHang	pk_DONHANG	MaDH
8	DiaChiNhan	pk_DIACHI	MaDiaChi
9	KhachHang	pk_KHACHHANG	MaKH
10	DangNhap	pk_DANGNHAP	MaDangNhap
11	GioHang	pk_GIAOHANG	

Bảng 6. Bảng thông tin về khóa chính

- Thông tin về khóa ngoại

Tên khoá	Bảng chứa khóa	Thuộc tính	Bảng được tham chiếu	Khoá chính
FK_DONHANG	DONHANG	MAKH	KHACHHANG	MAKH
FK_CHITIETDONHANG_SP	CHITIET_DONHANG	MASP	SANPHAM	MASP
FK_DANHGIA_SP	DANHGIA	MASP	SANPHAM	MASP
FK_SANPHAM_DM	SANPHAM	MADANHMUC	DANHMUC	MADANHMUC
FK_SANPHAM_NCC	SANPHAM	MA_NCC	NHACUNGCA P	MA_NCC
FK_CHITIET_DONHANG_CTDH	CHITIET_DONHANG	MADH	DONHANG	MADH
FK_CHITIET_THANHTOAN_TT	CHITIET_THANHTOAN	MaThanhToan	THANHTOAN	MaThanhToan

FK_CHITIET_THANHTOAN_DH	CHITIET_THANHTOAN	MADH	DONHANG	MADH
FK_DANHGIA_DONHANG	DANHGIA	MADH	DONHANG	MADH
FK_CHITIET_DONHANG_GG	CHITIET_DONHANG	Magiamgia_DM	GIAMGIA_DANHMUC	Magiamgia_DM
FK_GIOHANG_SP	GIOHANG	MaSanPham	SANPHAM	MaSP
FK_GIOHANG_KH	GIOHANG	MaKH	KHACHHANG	MaKH
FK_DIACHINHAN_KH	DIACHINHAN	MaKH	KHACHHANG	MaKH
FK_DONHANG_DIACHINHAN	DONHANG	MADIACHI	DIACHINHAN	MADIACHI
FK_DANHMUC_GIAMGIA	DANHMUC_GIAMGIA	MAGIAMGIA_DM	GIAMGIA_DANHMUC	MAGIAMGIA_DM
FK_DANHMUC_GIAM_DANHMUC	DANHMUC_GIAMGIA	MADANHMUC	DANHMUC	MADANHMUC
FK_DONHANG_GIAMGIA_DONHANG	DONHANG	MAGIAMGIA_DH	GIAMGIA_DONHANG	MAGIAMGIA_DH

Bảng 7. Bảng thông tin về khóa ngoại

4. Cài đặt cơ sở dữ liệu

4.1 Cài đặt CSDL

```
CREATE DATABASE CUAHANGSACH
USE CUAHANGSACH
```

//Create Table

```
CREATE TABLE SANPHAM (
    MaSP VARCHAR(10) PRIMARY KEY ,
    MaDanhMuc VARCHAR(10) NOT NULL,
    Ma_NCC VARCHAR(10),
    TenSP NVARCHAR(255) NOT NULL,
    GiaSP NUMERIC(16,2),
```

```

        TacGia NVARCHAR(255),
        Bia NVARCHAR(20),
        TonKho INT,
        TGTao DATETIME DEFAULT GETDATE(),
        TGCapNhat DATETIME DEFAULT GETDATE()
    );
CREATE TABLE NHACUNGCAP(
    Ma_NCC VARCHAR(10) PRIMARY KEY,
    Ten_NCC NVARCHAR(255) NOT NULL,
    SDT CHAR(20)
);
CREATE TABLE DANHMUC (
    MaDanhMuc VARCHAR(10) PRIMARY KEY,
    TenDanhmuc NVARCHAR(255) not null,
);
CREATE TABLE DONHANG (
    MaDH VARCHAR(10) PRIMARY KEY,
    MaKH VARCHAR(10) NOT NULL,
    MaGiamGia_DH VARCHAR(10),
    MaDiaChi VARCHAR(10) NOT NULL,
    NgayDat DATETIME NOT NULL DEFAULT GETDATE(),
    NgayGiao DATETIME,
    TongDonGia NUMERIC(16,2),
    GiamGia_DonHang NUMERIC(16,2),
    PhiVanChuyen NUMERIC(16,2),
    ThanhTien NUMERIC(16,2),
    TrangThai NVARCHAR(30) NOT NULL DEFAULT N'chờ xử lý',
    TGTao DATETIME DEFAULT GETDATE(),
    TGCapNhat DATETIME DEFAULT GETDATE()
);
CREATE TABLE DIACHINHAN(
    MaDiaChi VARCHAR(10) PRIMARY KEY,
    MaKH VARCHAR(10) NOT NULL,
    HoTen_NguoiNhan NVARCHAR(255),
    SoDienThoai CHAR(20),
    DiaChi NVARCHAR(1000) NOT NULL,
    QuanHuyen NVARCHAR(1000) NOT NULL,
    ThanhPho NVARCHAR(1000) NOT NULL,
    GhiChu NVARCHAR(1000)
);
CREATE TABLE CHITIET_DONHANG(

```

```

        MaDH VARCHAR(10) NOT NULL,
        MaSP VARCHAR(10) NOT NULL,
        MaGiamGia_DM VARCHAR(10),
        SoLuong NUMERIC(16,2) NOT NULL,
        DonGia NUMERIC(16,2) ,
        LuongGiam NUMERIC(16,2)
    );

CREATE TABLE CHITIET_THANHTOAN(
    MaThanhToan VARCHAR(10) NOT NULL ,
    MaDH VARCHAR(10) NOT NULL,
    NgayThanhToan DATETIME,
    TrangThai_TT NVARCHAR(30) NOT NULL DEFAULT N'xử lý',
    TGTao DATETIME DEFAULT GETDATE(),
    TGCapNhat DATETIME DEFAULT GETDATE()
);

CREATE TABLE KHACHHANG(
    MaKH VARCHAR(10) PRIMARY KEY,
    TenNguoiDung VARCHAR(255) NOT NULL ,
    MatKhai VARCHAR(255),
    TenKH NVARCHAR(255) NOT NULL,
    NgaySinh DATETIME,
    GioiTinh NVARCHAR(3),
    SoDienThoai CHAR(20) NOT NULL,
    Email VARCHAR(50),
);

CREATE TABLE DANHGIA(
    MaDH VARCHAR(10) NOT NULL,
    MaSP VARCHAR(10) NOT NULL,
    NoiDung NVARCHAR(1000),
    SoDiem CHAR(1) NOT NULL,
    NgayDanhGia DATETIME,
    TGTao DATETIME DEFAULT GETDATE(),
    TGCapNhat DATETIME DEFAULT GETDATE()
);

CREATE TABLE GIAMGIA_DANHMUC(
    MaGiamGia_DM VARCHAR(10) PRIMARY KEY,
    GiaTri NUMERIC(16,2),
    LoaiGiaTri VARCHAR(20),
    NoiDung NVARCHAR(255),
    SoLuongMa INT,
    GiaTri_Toithieu NUMERIC(16,2),

```

```

        GiamToiDa NUMERIC(16,2),
        TGBatDau DATETIME,
        TGKetThuc DATETIME,
        TrangThai TINYINT DEFAULT 1,
        TGTao DATETIME DEFAULT GETDATE(),
        TGCapNhat DATETIME DEFAULT GETDATE()
    );
CREATE TABLE GIAMGIA_DONHANG(
    MaGiamGia_DH VARCHAR(10) PRIMARY KEY,
    GiaTri NUMERIC(16,2),
    LoaiGiaTri VARCHAR(20),
    NoiDung NVARCHAR(255),
    SoLuongMa INT,
    GiaTri_ToiThieu NUMERIC(16,2),
    GiamToiDa NUMERIC(16,2),
    TGBatDau DATETIME,
    TGKetThuc DATETIME,
    TrangThai TINYINT DEFAULT 1,
    TGTao DATETIME DEFAULT GETDATE(),
    TGCapNhat DATETIME DEFAULT GETDATE()
);
CREATE TABLE GIOHANG(
    MaKH VARCHAR(10) NOT NULL ,
    MaSanPham VARCHAR(10) NOT NULL,
    TenSP NVARCHAR(255) NOT NULL,
    SoLuong INT,
    TGTao DATETIME DEFAULT GETDATE(),
    TGCapNhat DATETIME DEFAULT GETDATE()
);
CREATE TABLE DANHMUC_GIAM(
    MaGiamGia_DM VARCHAR(10) NOT NULL,
    MaDanhMuc VARCHAR(10) NOT NULL,
);
CREATE TABLE THANHTOAN(
    MaThanhToan VARCHAR(10) PRIMARY KEY,
    LoaiThanhToan NVARCHAR(20) NOT NULL,
);
CREATE TABLE DANGNHAP(
    MaDangNhap VARCHAR(10) PRIMARY KEY,
    TenDangNhap VARCHAR(255) NOT NULL,
    MatKhau VARCHAR(255) NOT NULL,

```



```
TrangThai NVARCHAR(30) NOT NULL  
);
```

//Create Primary Key

```
ALTER TABLE GIOHANG  
ADD CONSTRAINT PK_GIOHANG PRIMARY KEY (MAKH,MASANPHAM);  
ALTER TABLE DANHMUC_GIAM  
ADD CONSTRAINT PK_DANHMUC_GIAM PRIMARY KEY  
(MaDanhMuc,MaGiamGia_DM);  
ALTER TABLE CHITIET_THANHTOAN  
ADD CONSTRAINT PK_CHITIET_THANHTOAN PRIMARY KEY  
(MADH,MATHANHTOAN);  
ALTER TABLE CHITIET_DONHANG  
ADD CONSTRAINT PK_CHITIET_DONHANG PRIMARY KEY (MADH,MASP);  
ALTER TABLE DANHGIA  
ADD CONSTRAINT PK_DANHGIA_MADH_MASP PRIMARY KEY  
(MADH,MASP);
```

```
ALTER TABLE KHACHHANG  
ADD CONSTRAINT UN_KHACHHANG_TENNGUOIDUNG  
UNIQUE(TENNGUOIDUNG);  
ALTER TABLE DONHANG  
ADD CONSTRAINT UN_DONHANG_KHACHHANGMA UNIQUE  
(MAKH,MAGIAMGIA_DH);
```

//Create Foreign Key

```
ALTER TABLE SANPHAM  
ADD CONSTRAINT FK_SANPHAM_DM FOREIGN KEY (MADANHMUC)  
REFERENCES DANHMUC(MADANHMUC);  
ALTER TABLE SANPHAM  
ADD CONSTRAINT FK_SANPHAM_NCC FOREIGN KEY (MA_NCC)  
REFERENCES NHACUNGCAP(MA_NCC);  
ALTER TABLE CHITIET_DONHANG  
ADD CONSTRAINT FK_CHITIET_DONHANG_CTDH FOREIGN KEY (MADH)  
REFERENCES DONHANG(MADH) ON DELETE CASCADE;  
ALTER TABLE CHITIET_THANHTOAN  
ADD CONSTRAINT FK_CHITIET_THANHTOAN_TT FOREIGN KEY  
(MaThanhToan) REFERENCES THANHTOAN(MaThanhToan);
```

```

ALTER TABLE CHITIET_THANHTOAN
ADD CONSTRAINT FK_CHITIET_THANHTOAN_DH FOREIGN KEY (MADH)
REFERENCES DONHANG(MADH);
ALTER TABLE DONHANG
ADD CONSTRAINT FK_DONHANG FOREIGN KEY (MAKH) REFERENCES
KHACHHANG(MAKH);
ALTER TABLE CHITIET_DONHANG
ADD CONSTRAINT FK_CHITIETDONHANG_SP FOREIGN KEY (MASP)
REFERENCES SANPHAM(MASP);
ALTER TABLE DANHGIA
ADD CONSTRAINT FK_DANHGIA_SANPHAM FOREIGN KEY (MaSP)
REFERENCES SANPHAM(MASP);
ALTER TABLE DANHGIA
ADD CONSTRAINT FK_DANHGIA_DONHANG FOREIGN KEY (MADH)
REFERENCES DONHANG (MADH);
ALTER TABLE CHITIET_DONHANG
ADD CONSTRAINT FK_CHITIET_DONHANG_GG FOREIGN KEY
(Magiamgia_DM) REFERENCES GIAMGIA_DANHMUC (Magiamgia_DM);
ALTER TABLE GIOHANG
ADD CONSTRAINT FK_GIOHANG_SP FOREIGN KEY (MaSanPham)
REFERENCES SANPHAM (MaSP);
ALTER TABLE GIOHANG
ADD CONSTRAINT FK_GIOHANG_KH FOREIGN KEY (MaKH) REFERENCES
KHACHHANG (MaKH);
ALTER TABLE DIACHINHAN
ADD CONSTRAINT FK_DIACHINHAN_KH FOREIGN KEY (MaKH)
REFERENCES KHACHHANG(MaKH);
ALTER TABLE DONHANG
ADD CONSTRAINT FK_DONHANG_DIACHINHAN FOREIGN KEY
(MADIACHI) REFERENCES DIACHINHAN (MADIACHI);
ALTER TABLE DANHMUC_GIAM
ADD CONSTRAINT FK_DANHMUC_GIAM_GIAMGIA FOREIGN KEY
(MAGIAMGIA_DM) REFERENCES
GIAMGIA_DANHMUC(MAGIAMGIA_DM);
ALTER TABLE DANHMUC_GIAM
ADD CONSTRAINT FK_DANHMUC_GIAM_DANHMUC FOREIGN KEY
(MADANHMUC) REFERENCES DANHMUC(MADANHMUC);
ALTER TABLE DONHANG
ADD CONSTRAINT FK_DONHANG_GIAMGIA_DONHANG FOREIGN KEY
(MAGIAMGIA_DH) REFERENCES GIAMGIA_DONHANG(MAGIAMGIA_DH);

```

4.2 Query thể hiện chức năng của CSDL (dưới dạng stored procedure)

--Sản phẩm được mua bởi khách hàng nào--

```
CREATE PROCEDURE QR1
(@Masp CHAR(10) )
AS
BEGIN
    select SANPHAM.TenSP,KHACHHANG.MaKH,KHACHHANG.TenKH from
    KHACHHANG
    join GIOHANG on KHACHHANG.MaKH=GIOHANG.MaKH
    join SANPHAM on GIOHANG.MaSanPham=SANPHAM.MaSP
    WHERE MaSP=@Masp
    group by KHACHHANG.MaKH,KHACHHANG.TenKH,SANPHAM.TenSP
END
    execute QR1 C06
```

--Sản phẩm được mua bởi ai thuộc đơn hàng nào và loại thanh toán là gì--

```
CREATE PROCEDURE QR2
(@MaSP varchar(10))
AS
BEGIN
    SELECT SANPHAM.TENSP,SANPHAM.GIASP,KHACHHANG.MaKH ,
    KHACHHANG.TenKH ,
    DONHANG.MaDH,THANHTOAN.LoaiThanhToan,DONHANG.NgayDat,
    CHITIET_DONHANG.SoLuong from SANPHAM
    JOIN GIOHANG ON SANPHAM.MASP=GIOHANG.MASanpham
    JOIN KHACHHANG ON GIOHANG.MAKH=KHACHHANG.MAKH
    JOIN DONHANG on KHACHHANG.MaKH = DONHANG.MaKH
    JOIN CHITIET_THANHTOAN on
    DONHANG.MaDH=CHITIET_THANHTOAN.MaDH
    JOIN CHITIET_DONHANG ON
    DONHANG.MADH=CHITIET_DONHANG.MaDH
    JOIN THANHTOAN on
    CHITIET_THANHTOAN .Mathanhtoan=THANHTOAN.Mathanhtoan WHERE
    MaSanPham=@MaSP
    group by SANPHAM.TENSP,SANPHAM.GIASP,KHACHHANG.MaKH ,
    KHACHHANG.TenKH ,
    DONHANG.MaDH,THANHTOAN.LoaiThanhToan,DONHANG.NgayDat,
    CHITIET_DONHANG.SoLuong
END
    execute QR2 C05
```

--Sản phẩm được đánh giá bởi khách hàng nào với số điểm và nội dung là gì---

```
CREATE PROCEDURE QR3
(@MaSP varchar(10))
AS
BEGIN
```

```

SELECT KHACHHANG.MaKH ,
KHACHHANG.TenKH ,SANPHAM.TenSP,DANHGIA.Sodiem,DANHGIA.NgayD
anhgia,DANHGIA.NoiDung from KHACHHANG
join GIOHANG on KHACHHANG.MaKH=GIOHANG.MaKH
join SANPHAM on GIOHANG.MaSanPham=SANPHAM.MaSP
join DANHGIA on SANPHAM.MaSP=DanhGIA.MaSP
WHERE MaSanPham=@MaSP
group by KHACHHANG.MaKH ,
KHACHHANG.TenKH ,SANPHAM.TenSP,DANHGIA.Sodiem,DANHGIA.NgayD
anhgia,DANHGIA.NoiDung
END
execute QR3 C02

```

--SẢN PHẨM THUỘC DANH MỤC NÀO VÀ DANH MỤC ĐÓ CÓ ĐƯỢC GIẢM GIÁ KHÔNG--

```

CREATE PROCEDURE QR4
(@MaSP varchar(10))
AS
BEGIN
SELECT
SANPHAM.TenSP,DANHMUC.MADANHMUC,DANHMUC.TENDANHMUC,Gi
amgia_danhmuc.Magiamgia_dm,Giamgia_danhmuc.giatri from SANPHAM
join DANHMUC on SANPHAM.Madanhmuc=DANHMUC.Madanhmuc
join DANHMUC_GIAM on
DANHMUC.MADANHMUC=DANHMUC_GIAM.MADANHMUC
JOIN GIAMGIA_DANHMUC ON
DANHMUC_GIAM.MAGIAMGIA_DM=GIAMGIA_DANHMUC.MAGIAMGIA_
DM where MaSP=@MaSP
group by
SANPHAM.TenSP,DANHMUC.MADANHMUC,DANHMUC.TENDANHMUC,Gi
amgia_danhmuc.Magiamgia_dm,Giamgia_danhmuc.giatri
END
execute QR4 C01

```

--SẢN PHẨM ĐƯỢC MUA BỞI AI VỚI GIÁ BAO NHIÊU THUỘC ĐƠN HÀNG NÀO VÀ ĐƠN HÀNG ĐÓ CÓ ĐƯỢC GIẢM GIÁ KHÔNG VÀ ĐỊA CHỈ NHẬN CỦA ĐƠN HÀNG--

```

CREATE PROCEDURE QR5
(@MaSP varchar(10), @MaDH VARCHAR (10))
AS
BEGIN
select
KHACHHANG.MaKH,KHACHHANG.TenKH,SANPHAM.TenSP,SANPHAM.gias
p,DONHANG.MADH,DONHANG.PHIVANCHUYEN,DONHANG.MAGIAMGIA
_DH,GIAMGIA_DONHANG.GIATRI,DIACHINHAN.DiaChi,DIACHINHAN.Qua
nHuyen ,DIACHINHAN.ThanhPho from KHACHHANG
join GIOHANG on KHACHHANG.MaKH=GIOHANG.MaKH
join SANPHAM on GIOHANG.MaSanPham=SANPHAM.MaSP

```

```

JOIN DONHANG ON KHACHHANG.MAKH=DONHANG.MAKH
JOIN GIAMGIA_DONHANG ON
DONHANG.MAGIAMGIA_dh=GIAMGIA_DONHANG.MAGIAMGIA_dh
JOIN DIACHINHAN ON DONHANG.MAKH=DIACHINHAN.MAKH where
MaSP=@MaSP AND MaDH=@MaDH
group by
KHACHHANG.MaKH,KHACHHANG.TenKH,SANPHAM.TenSP,SANPHAM.gias
p,DONHANG.MADH,DONHANG.PHIVANCHUYEN,DONHANG.MAGIAMGIA
_DH,GIAMGIA_DONHANG.GIATRI,DIACHINHAN.DiaChi,DIACHINHAN.Qua
nHuyen ,DIACHINHAN.ThanhPho
END
execute QR5 C20,DH020

```

--Sản phẩm được cung cấp bởi ai và nằm trong danh mục nào--

```

CREATE PROCEDURE QR6
(@Masp varchar(10))
AS
BEGIN
SELECT
SANPHAM.TenSP,DANHMUC.TenDanhmuc,NHACUNGCAP.Ma_NCC,NHACU
NGCAP.Ten_NCC from SANPHAM
join DANHMUC on SANPHAM.MaDanhMuc=DANHMUC.MaDanhMuc
join NHACUNGCAP on SANPHAM.Ma_NCC=NHACUNGCAP.Ma_NCC
WHERE MaSP=@Masp
group
by SANPHAM.TenSP,DANHMUC.TenDanhmuc,NHACUNGCAP.Ma_NCC,NHA
CUNGCAP.Ten_NCC
END
execute QR6 C23

```

--Khách hàng muốn biết sản phẩm mình mua được viết bởi tác giả nào và do ai là người cung cấp --

```

CREATE PROC QR7
(@Masp varchar(10))
AS
BEGIN
SELECT KHACHHANG.TenKH,
SANPHAM.TenSP,SANPHAM.TacGia,NHACUNGCAP.Ten_NCC from
SANPHAM
JOIN GIOHANG ON SANPHAM.MaSP = GIOHANG.MaSanPham
JOIN KHACHHANG ON GIOHANG.MaKH=KHACHHANG.MaKH
JOIN NHACUNGCAP ON SANPHAM.Ma_NCC=NHACUNGCAP.Ma_NCC
WHERE MaSP=@Masp
GROUP BY KHACHHANG.TenKH,
SANPHAM.TenSP,SANPHAM.TacGia,NHACUNGCAP.Ten_NCC
END
EXECUTE QR7 C12

```

--Sản phẩm thuộc đơn hàng nào với số lượng sản phẩm là bao nhiêu với khách hàng là ai--

```
CREATE PROC QR8
(@MaSP varchar(10))
AS
BEGIN
    SELECT SANPHAM.TenSP,
    CHITIET_DONHANG.MaDH,CHITIET_DONHANG.SoLuong,KHACHHANG.Ten
    KH,KHACHHANG.MaKH FROM SANPHAM
    JOIN CHITIET_DONHANG ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
    JOIN GIOHANG ON SANPHAM.MaSP=GIOHANG.MaSanPham
    JOIN KHACHHANG ON GIOHANG.MaKH=KHACHHANG.MaKH WHERE
    MaSanPham=@MaSP
    GROUP BY SANPHAM.TenSP,
    CHITIET_DONHANG.MaDH,CHITIET_DONHANG.SoLuong,KHACHHANG.Ten
    KH ,KHACHHANG.MaKH
END
EXEC QR8 C01
```

--Khách hàng mua sản phẩm nào nhiều hơn, ít hơn để có thể ra chiến lược kinh doanh (sách nào nhập nhiều sách nào nhập ít)--

```
CREATE PROC QR9
(@MaSP varchar(10))
AS
BEGIN
    SELECT KHACHHANG.TenKH,SANPHAM.TENSP,
    sum(CHITIET_DONHANG.SOLUONG) TONGSOLUONG FROM SANPHAM
    JOIN GIOHANG ON SANPHAM.MASP=GIOHANG.MaSanPham
    JOIN KHACHHANG ON GIOHANG.MaKH=KHACHHANG.MaKH
    JOIN CHITIET_DONHANG ON
    SANPHAM.MASP=CHITIET_DONHANG.MASP
    GROUP BY
    KHACHHANG.TenKH,SANPHAM.TENSP,CHITIET_DONHANG.SOLUONG
    ORDER BY CHITIET_DONHANG.SOLUONG DESC
END
EXEC QR9 C01
```

--Dựa vào số lượng các sản phẩm bán được và độ tồn kho của sản phẩm để đưa ra chiến lược kinh doanh (ngưng nhập sách bán chậm, đưa ra các mã giá cho sách bán chậm,...)--

```
CREATE PROC QR10
(@MaSP varchar(10))
AS
BEGIN
```

```

SELECT SANPHAM.TENSP,
CHITIET_DONHANG.SOLUONG,SANPHAM.TONKHO FROM SANPHAM
JOIN CHITIET_DONHANG ON
SANPHAM.MASP=CHITIET_DONHANG.MASP
GROUP BY SANPHAM.TENSP,
CHITIET_DONHANG.SOLUONG,SANPHAM.TONKHO
ORDER BY CHITIET_DONHANG.SOLUONG DESC
END
EXEC QR10 C01

```

--Nhập tên người dùng để xem trạng thái thanh toán và trạng thái của đơn hàng--

```

CREATE PROC QR12
(@TEN varchar(255))
AS
BEGIN
SELECT KHACHHANG.TenKH,
DONHANG.MaDH,CHITIET_THANHTOAN.TrangThai_TT,
DONHANG.TrangThai FROM KHACHHANG
JOIN DONHANG ON KHACHHANG.MaKH=DONHANG.MaKH
JOIN CHITIET_THANHTOAN ON
DONHANG.MaDH=CHITIET_THANHTOAN.MaDH
WHERE TenNguoiDung=@TEN
GROUP BY KHACHHANG.TenKH,
DONHANG.MaDH,CHITIET_THANHTOAN.TrangThai_TT,DONHANG.TrangTh
ai
END
EXEC QR12 Lyys

```

--Số lượng sản phẩm có trong giỏ hàng của khách hàng--

```

CREATE PROC QR13
(@SP varchar(10))
AS
BEGIN
SELECT KHACHHANG.TenKH, GIOHANG.TenSP,GIOHANG.SoLuong FROM
KHACHHANG
JOIN GIOHANG ON KHACHHANG.MaKH=GIOHANG.MaKH
WHERE MaSanPham=@SP
GROUP BY KHACHHANG.TenKH, GIOHANG.TenSP,GIOHANG.SoLuong
END
EXEC QR13 C26

```

--Tổng doanh thu sản phẩm--

```

CREATE PROC QR16
(@SP varchar(10))
AS
BEGIN
select SUM(DONHANG.ThanhTien) TONGDOANHTHUSP FROM SANPHAM

```

```

JOIN CHITIET_DONHANG ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
Join GIOHANG ON SANPHAM.MaSP=GIOHANG.MaSanPham
JOIN DONHANG ON CHITIET_DONHANG.MaDH=DONHANG.MaDH
END
EXEC QR16 C01

```

--Xếp hạng danh mục được yêu thích theo số lượng sản phẩm bán được --

```

CREATE PROC QR17
(@SP varchar(10))
AS
BEGIN
    select SUM(CHITIET_DONHANG.SoLuong) TONGSOLUONG ,
    DANHMUC.TenDanhmuc FROM CHITIET_DONHANG
    JOIN SANPHAM ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
    JOIN DANHMUC ON SANPHAM.MaDanhMuc=DANHMUC.MaDanhMuc
    GROUP BY DANHMUC.TenDanhmuc
    ORDER BY TONGSOLUONG DESC
END
EXEC QR17 C01

```

--Doanh thu của danh mục--

```

CREATE PROC QR18
(@DMUC Nvarchar(255))
AS
BEGIN
    select SUM(DONHANG.ThanhTien) TONGDOANHTHU ,
    DANHMUC.TenDanhmuc FROM CHITIET_DONHANG
    JOIN SANPHAM ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
    JOIN DONHANG ON CHITIET_DONHANG.MaDH=DONHANG.MaDH
    JOIN DANHMUC ON SANPHAM.MaDanhMuc=DANHMUC.MaDanhMuc
    WHERE TenDanhmuc=@DMUC
    GROUP BY DANHMUC.TenDanhmuc
END
EXEC QR18 N'Sách lịch sử'
EXEC QR18 N'Doàn - Đội - Hội'
EXEC QR18 N'Sách thiếu nhi'
EXEC QR18 N'Thơ'

```

--Xếp hạng doanh thu của danh mục--

```

CREATE PROC QR19
(@DMUC Nvarchar(255))
AS
BEGIN
    select SUM(DONHANG.ThanhTien) TONGDOANHTHU ,
    DANHMUC.TenDanhmuc FROM CHITIET_DONHANG
    JOIN SANPHAM ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
    JOIN DANHMUC ON SANPHAM.MaDanhMuc=DANHMUC.MaDanhMuc

```



```

JOIN DONHANG ON CHITIET_DONHANG.MaDH=DONHANG.MaDH
GROUP BY DANHMUC.TenDanhmuc
ORDER BY TONGDOANHTHU DESC
END
EXEC QR19 N'Sách lịch sử'

```

--Nhà cung cấp tiềm năng theo doanh thu số lượng sản phẩm--

```

CREATE PROC QR20
(@MNCC varchar(10))
AS
BEGIN
    select SUM(CHITIET_DONHANG.SoLuong)
    TONGSOLUONG ,SUM(DONHANG.ThanhTien) TONGDOANHTHU ,
    DANHMUC.TenDanhmuc, NHACUNGCAP.Ten_NCC FROM
    CHITIET_DONHANG
    JOIN SANPHAM ON SANPHAM.MaSP=CHITIET_DONHANG.MaSP
    JOIN DANHMUC ON SANPHAM.MaDanhMuc=DANHMUC.MaDanhMuc
    JOIN NHACUNGCAP ON SANPHAM.Ma_NCC=NHACUNGCAP.Ma_NCC
    JOIN DONHANG ON CHITIET_DONHANG.MaDH=DONHANG.MaDH
    GROUP BY DANHMUC.TenDanhmuc , NHACUNGCAP.Ten_NCC
    ORDER BY TONGDOANHTHU DESC
END
EXEC QR20 N'KD1618'

```

--Chức năng thêm một đơn hàng mới gồm chi tiết đơn hàng và chi tiết thanh toán, trong đó người dùng nhập vào mã khách hàng và các thông tin cần thiết--

```

CREATE PROCEDURE PROC_DONHANG_CTDH_CTTT_INSERT
@MAKH VARCHAR(10),
@MAGIAMGIA_DH VARCHAR(10),
@MADIACHI VARCHAR(10),
@MATHANHTOAN VARCHAR(10),
@XMLData XML,
@LOAIKHACHHANG BIT
AS
BEGIN
    ---
    DECLARE @MAGIAMGIA_DH_DB VARCHAR(10)
    DECLARE @MADH VARCHAR(10)
    DECLARE @MOITAO BIT = 0

    WHILE @MOITAO = 0
    BEGIN
        SET @MADH = CONCAT('DH',
        SUBSTRING(CONVERT(VARCHAR(255), NEWID()), 1, 10))
    END

```

```

SET @MAGIAMGIA_DH_DB =
CONCAT('GG_DB',SUBSTRING(CONVERT(VARCHAR(255), NEWID()),
1, 10) )

IF NOT EXISTS (SELECT 1 FROM DONHANG WHERE MaDH =
@MADH)
BEGIN
    SET @MOITAO = 1
END
IF NOT EXISTS (SELECT 1 FROM GIAMGIA_DONHANG WHERE
MaGiamGia_DH = @MAGIAMGIA_DH_DB)
BEGIN
    SET @MOITAO =1
END
END
---
DECLARE @Date DATETIME = GETDATE()

-- Thiết lập giá trị mặc định cho thời gian bắt đầu giảm giá là ngày đầu tháng hiện tại
DECLARE @StartTime DATETIME = DATEADD(MONTH,
DATEDIFF(MONTH, 0, @Date), 0)

-- Thiết lập giá trị mặc định cho thời gian kết thúc giảm giá là ngày cuối tháng hiện tại
DECLARE @EndTime DATETIME = DATEADD(MONTH,
DATEDIFF(MONTH, -1, @Date), -1)

--Tạo procedure cho phép xem được đánh giá theo sản phẩm hoặc theo danh mục dựa trên
số điểm và khoảng thời gian cụ thể --
CREATE PROCEDURE PROC_TINH_TRUNG_BINH_DANH_GIA
    @LUACHON INT = 1,
    @DiemMin TINYINT = 1,
    @DiemMax TINYINT = 5,
    @TGbatdau DATETIME,
    @TGketthuc DATETIME
AS
BEGIN
    IF @LUACHON = 1
    BEGIN
        IF (@TGbatdau IS NULL AND @TGketthuc IS NULL OR (@TGbatdau IS NULL
        OR @TGketthuc IS NULL))
        BEGIN
            --Tính trung bình số điểm đánh giá theo từng sản phẩm
            SELECT SP.MaSP, SP.TenSP, AVG(DG.SoDiem*1.0) AS TrungBinhDiemDanhGia
            INTO #TrungBinhTheoSanPham
            FROM SANPHAM SP
            LEFT JOIN DANHGIA DG ON SP.MaSP = DG.MaSP
        
```

```
WHERE DG.SoDiem BETWEEN @DiemMin AND @DiemMax  
GROUP BY SP.MaSP, SP.TenSP;
```

-- In kết quả trung bình số điểm theo sản phẩm

```
SELECT * FROM #TrungBinhTheoSanPham;
```

-- Xóa bảng tạm

```
DROP TABLE IF EXISTS #TrungBinhTheoSanPham;
```

```
END
```

```
ELSE IF (@TGbatdau IS NOT NULL AND @TGketthuc IS NOT NULL)
```

```
BEGIN
```

-- Tính trung bình số điểm đánh giá theo từng sản phẩm

```
SELECT SP.MaSP, SP.TenSP, AVG(DG.SoDiem*1.0) AS TrungBinhDiemDanhGia  
INTO #TrungBinh_SPTHOIGIAN  
FROM SANPHAM SP  
LEFT JOIN DANHGIA DG ON SP.MaSP = DG.MaSP  
WHERE DG.SoDiem BETWEEN @DiemMin AND @DiemMax AND  
DG.NgayDanhGia BETWEEN @TGbatdau AND @TGketthuc  
GROUP BY SP.MaSP, SP.TenSP;
```

-- In kết quả trung bình số điểm theo sản phẩm

```
SELECT * FROM #TrungBinh_SPTHOIGIAN;
```

-- Xóa bảng tạm

```
DROP TABLE IF EXISTS #TrungBinh_SPTHOIGIAN;
```

```
END
```

```
END
```

```
ELSE IF @LUACHON = 2
```

```
BEGIN
```

```
IF (@TGbatdau IS NULL AND @TGketthuc IS NULL OR (@TGbatdau IS NULL  
OR @TGketthuc IS NULL))
```

```
BEGIN
```

-- Tính trung bình số điểm đánh giá theo từng danh mục

```
SELECT SP.MaDanhMuc, DM.TenDanhMuc, AVG(DG.SoDiem*1.0) AS  
TrungBinhDiemDanhGia  
INTO #TrungBinhTheoDanhMuc  
FROM SANPHAM SP  
LEFT JOIN DANHGIA DG ON SP.MaSP = DG.MaSP  
LEFT JOIN DANHMUC DM ON SP.MaDanhMuc = DM.MaDanhMuc  
WHERE DG.SoDiem BETWEEN @DiemMin AND @DiemMax  
GROUP BY SP.MaDanhMuc, DM.TenDanhMuc;
```

-- In kết quả trung bình số điểm theo danh mục

```
SELECT * FROM #TrungBinhTheoDanhMuc;
```

-- Xóa bảng tạm

```
DROP TABLE IF EXISTS #TrungBinhTheoDanhMuc;
```

```
END
```

```
ELSE IF (@TGbatdau IS NOT NULL AND @TGketthuc IS NOT NULL)  
BEGIN
```

```
-- Tính trung bình số điểm đánh giá theo từng danh mục
```

```
SELECT SP.MaDanhMuc, DM.TenDanhMuc, AVG(DG.SoDiem*1.0) AS
```

```
TrungBinhDiemDanhGia
```

```
INTO #TrungBinh_DANHMUC_TG
```

```
FROM SANPHAM SP
```

```
LEFT JOIN DANHGIA DG ON SP.MaSP = DG.MaSP
```

```
LEFT JOIN DANHMUC DM ON SP.MaDanhMuc = DM.MaDanhMuc
```

```
WHERE DG.SoDiem BETWEEN @DiemMin AND @DiemMax AND
```

```
DG.NgayDanhGia BETWEEN @TGbatdau AND @TGketthuc
```

```
GROUP BY SP.MaDanhMuc, DM.TenDanhMuc;
```

-- In kết quả trung bình số điểm theo danh mục

```
SELECT * FROM #TrungBinh_DANHMUC_TG;
```

-- Xóa bảng tạm

```
DROP TABLE IF EXISTS #TrungBinh_DANHMUC_TG;
```

```
END
```

```
END
```

```
END;
```

```
exec PROC_TINH_TRUNG_BINH_DANH_GIA 1,1,5,NULL,NULL
```

```
drop proc PROC_TINH_TRUNG_BINH_DANH_GIA
```

4.3 Các Trigger để ngăn chặn lỗi của CSDL

---PHẦN CHECK

--Trong bảng sản phẩm, giá sản phẩm và tồn kho >=0

```
ALTER TABLE SANPHAM
```

```
ADD CONSTRAINT CHK_SP_0 CHECK ( TONKHO >=0);
```

-- Trong bảng đơn hàng, ngày đặt <= ngày hiện tại

```
ALTER TABLE DONHANG
```

```
ADD CONSTRAINT CHK_DH_NGAYDAT CHECK (NGAYDAT<= GETDATE());
```

--Trong bảng khách hàng, giới tính chỉ có thể là [nam, nữ, khác]

```
ALTER TABLE KHACHHANG
```

```
ADD CONSTRAINT CHK_KH_GT CHECK (GIOITINH IN (N'Nam', N'Nữ'));
```

-- Trong bảng khách hàng, số điện thoại có 9-10 chữ số, mỗi chữ số nằm trong khoảng 0-9

```
ALTER TABLE KHACHHANG
ADD CONSTRAINT CHK_KH_SDT CHECK (SODIENTHOAI LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR SODIENTHOAI LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

-- Trong bảng giảm giá danh mục, loại giá trị giảm giá [GIATRI,PHANTRAM] giatri <= 100 nếu loaigiatri = PHANTRAM

```
ALTER TABLE GIAMGIA_DANHMUC
ADD CONSTRAINT CHK_GIAMGIA_DANHMUC_LOAIGIATRI CHECK ((LOAIGIATRI = 'GIATRI') OR (LOAIGIATRI='PHANTRAM' AND GIATRI<=100));
```

--Trong bảng giảm giá đơn hàng, loại giá trị giảm giá [GIATRI,PHANTRAM] và giatri <= 100 nếu loaigiatri = PHANTRAM

```
ALTER TABLE GIAMGIA_DONHANG
ADD CONSTRAINT CHK_GIAMGIA_DONHANG_LOAIGIATRI CHECK ((LOAIGIATRI = 'GIATRI') OR (LOAIGIATRI='PHANTRAM' AND GIATRI<=100));
```

-- Trong bảng đơn hàng, ngày đặt hàng phải bé hơn ngày giao

```
ALTER TABLE DONHANG
ADD CONSTRAINT CHK_DONHANG_NDNG CHECK (NGAYDAT <= NGAYGIAO);
```

-- Trong bảng đánh giá, số điểm nằm trong khoảng từ 1-5

```
ALTER TABLE DANHGIA
ADD CONSTRAINT CHK_DANHGIA_SODIEM CHECK (SODIEM IN (1,2,3,4,5));
```

-- Trong bảng đơn hàng, trạng thái nằm trong các giá trị ('chờ xử lý', 'đã xử lý', 'đang giao', 'đã giao', 'đã hủy', 'xử lý hoàn', 'đang hoàn', 'đã hoàn'))

```
ALTER TABLE DONHANG
ADD CONSTRAINT CHK_DONHANG_TRANGTHAI CHECK (TRANGTHAI IN( N'chờ xử lý', N'đã xử lý', N'đang giao', N'đã giao', N'đã hủy', N'xử lý hoàn', N'đang hoàn', N'đã hoàn'));
```

-- Trong bảng chi tiết thanh toán, trạng thái nằm trong các giá trị ('xử lý', 'thành công')

```
ALTER TABLE CHITIET_THANHTOAN
ADD CONSTRAINT CHK_TIETTIET_THANHTOAN_TRANGTHAI CHECK(TRANGTHAI_TT IN (N'xử lý', N'thành công'));
```

--Trong bảng nhà cung cấp, SĐT có 10 hoặc 11 số, nằm trong khoảng từ 0-9

```
ALTER TABLE NHACUNGCAP
ADD CONSTRAINT CHK_NHACUNGCAP_SDT CHECK (SDT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR SDT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

-- Trong bảng địa chỉ nhận, sodienthoai có 10 hoặc 11 số nằm trong khoảng từ 0-9

```
ALTER TABLE DIACHINHAN
ADD CONSTRAINT CHK_DIACHINHAN_SDT CHECK (SODIENTHOAI LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR SODIENTHOAI LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

-- Trong bảng giảm giá đơn hàng, TGbatdau<=TGketthuc

```
ALTER TABLE GIAMGIA_DONHANG
ADD CONSTRAINT CHK_GIAMGIA_DONHANG_TGBATDAU CHECK (TGBATDAU
<= TGKETTHUC);
```

-- Trong bảng giảm giá danh mục, TGbatdau <= TGketthuc

```
ALTER TABLE GIAMGIA_DANHMUC
ADD CONSTRAINT CHK_GIAMGIA_DANHMUC_TGBATDAU CHECK (TGBATDAU
<= TGKETTHUC);
```

--Trong bảng giảm giá đơn hàng, Trạng thái nằm trong 2 giá trị là 0 hoặc 1

```
ALTER TABLE GIAMGIA_DONHANG
ADD CONSTRAINT CHK_GIAMGIA_DONHANG_TRANGTHAI CHECK
(TRANGTHAI IN (0,1));
```

--Trong bảng giảm giá danh mục, Trạng thái nằm trong 2 giá trị là 0 hoặc 1

```
ALTER TABLE GIAMGIA_DANHMUC
ADD CONSTRAINT CHK_GIAMGIA_DANHMUC_TRANGTHAI CHECK
(TRANGTHAI IN (0,1));
```

-- Trong bảng đơn hàng, PhiVanchuyen >=0

```
ALTER TABLE DONHANG
ADD CONSTRAINT CHK_DONHANG_PHIVANCHUYENDUONG
CHECK(PHIVANCHUYEN >=0);
```

-- Trong bảng chitiet donhang, soluong >=1

```
ALTER TABLE CHITIET_DONHANG
ADD CONSTRAINT CHK_CHITIETDONHANG_SL CHECK (SOLUONG >=1);
```

-- Trong bảng giamgia_donhang, giatri >0, gia tri toi thieu>=0, giamtoida >0

```
ALTER TABLE GIAMGIA_DONHANG
ADD CONSTRAINT CHK_GIAMGIA_DONHANG_GIATRI_GIATRITOITHIEU CHECK
(GIATRI>0 AND GIATRI_TOITHIEU>=0 AND GIAMTOIDA >0);
```

-- Trong bảng giamgia_danhmuc, giatri>0, giatrithieu >=0, giamtoida >0

```
ALTER TABLE GIAMGIA_DANHMUC
ADD CONSTRAINT CHK_GIAMGIA_DANHMUC_GIATRI_GIATRITOITHIEU CHECK
(GIATRI>0 AND GIATRI_TOITHIEU>=0 AND GIAMTOIDA >0);
```

```
ALTER TABLE DIACHINHAN
```

```
ADD CONSTRAINT CHK_TINHTHANH_KHACHHANG CHECK (THANHPHO IN
(N'An Giang',N'Bà Rịa – Vũng Tàu',N'Bắc Giang',N'Bắc Kạn',N'Bạc Liêu',N'Bắc Ninh',N'Bến
```

Tre',N'Bình Định',N'Bình Dương',N'Bình Phước',N'Bình Thuận',N'Cà Mau',N'Cần Thơ',N'Cao Bằng',N'Dà Nẵng',N'Đắk Lắk',N'Đắk Nông',N'Điện Biên',N'Đồng Nai',N'Đồng Tháp',N'Gia Lai',N'Hà Giang',N'Hà Nam',N'Hà Nội',N'Hà Tĩnh',N'Hải Dương',N'Hải Phòng',N'Hậu Giang',N'Hòa Bình',N'Hưng Yên',N'Khánh Hòa',N'Kiên Giang',N'Kon Tum',N'Lai Châu',N'Lâm Đồng',N'Lạng Sơn',N'Lào Cai',N'Long An',N'Nam Định',N'Nghệ An',N'Ninh Bình',N'Ninh Thuận',N'Phú Thọ',N'Phú Yên',N'Quảng Bình',N'Quảng Nam',N'Quảng Ngãi',N'Quảng Ninh',N'Quảng Trị',N'Sóc Trăng',N'Sơn La',N'Tây Ninh',N'Thái Bình',N'Thái Nguyên',N'Thanh Hóa',N'Thừa Thiên Huế',N'Tiền Giang',N'Hồ Chí Minh',N'Trà Vinh',N'Tuyên Quang',N'Vĩnh Long',N'Vĩnh Phúc',N'Yên Bái')));

--PHẦN TRIGGER: TRIGGER UPDATE-TRIGGER DELETE

-- Tạo các trigger xóa các bản ghi liên quan

```
CREATE TRIGGER TR_DELETE_SANPHAM_NCC
ON NHACUNGCAP
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE SANPHAM
    FROM deleted
    WHERE SANPHAM.Ma_NCC = deleted.Ma_NCC
    DELETE NHACUNGCAP
    FROM deleted
    WHERE NHACUNGCAP.Ma_NCC = deleted.Ma_NCC
END ;

--
CREATE TRIGGER TR_DELETE_SANPHAM_DGIA
ON SANPHAM
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE DANHGIA
    FROM deleted
    WHERE DANHGIA.MaSP = deleted.MaSP
    DELETE CHITIET_DONHANG
    FROM deleted
    WHERE CHITIET_DONHANG.MaSP = deleted.MaSP
    DELETE GIOHANG
    FROM deleted
    WHERE deleted.MaSP = GIOHANG.MaSanPham
    DELETE SANPHAM
    FROM deleted
    WHERE deleted.MaSP = SANPHAM.MaSP
```

```

END;
--
CREATE TRIGGER TR_DELETE_DANHMUC_SP_GG
ON DANHMUC
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE SANPHAM
    FROM deleted
    WHERE SANPHAM.MaDanhMuc = deleted.MaDanhMuc
    DELETE DANHMUC_GIAM
    FROM deleted
    WHERE deleted.MaDanhMuc = DANHMUC_GIAM.MaDanhMuc
    DELETE DANHMUC
    FROM deleted
    WHERE deleted.MaDanhMuc = DANHMUC.MaDanhMuc
END;
--
CREATE TRIGGER TR_DELETE_GIAMGIA_DANHMUC_CTDH_DMG
ON GIAMGIA_DANHMUC
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE DANHMUC_GIAM
    FROM deleted
    WHERE deleted.MaGiamGia_DM = DANHMUC_GIAM.MaGiamGia_DM
    DELETE CHITIET_DONHANG
    FROM deleted
    WHERE deleted.MaGiamGia_DM = CHITIET_DONHANG.MaGiamGia_DM
    DELETE GIAMGIA_DANHMUC
    FROM deleted
    WHERE deleted.MaGiamGia_DM = GIAMGIA_DANHMUC.MaGiamGia_DM
END;
--
CREATE TRIGGER TR_DELETE_THANHTOAN_CTTT
ON THANHTOAN
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE CHITIET_THANHTOAN
    FROM deleted
    WHERE deleted.MaThanhToan = CHITIET_THANHTOAN.MaThanhToan
    DELETE THANHTOAN
    FROM deleted

```



```

        WHERE deleted.MaThanhToan = THANHTOAN.MaThanhToan
END;
--
CREATE TRIGGER TRIG_DELETE_DONHANG
ON DONHANG
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE CHITIET_DONHANG
    FROM deleted
    WHERE CHITIET_DONHANG.MaDH = deleted.MaDH
    DELETE CHITIET_THANHTOAN
    FROM deleted
    WHERE deleted.MaDH = CHITIET_THANHTOAN.MaDH
    DELETE DANHGIA
    FROM deleted
    WHERE DANHGIA.MaDH = deleted.MaDH
    DELETE DONHANG
    FROM deleted
    WHERE DONHANG.MaDH = deleted.MaDH
END;
--
CREATE TRIGGER TR_DELETE_GIAMGIA_DONHANG_DH
ON GIAMGIA_DONHANG
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE DONHANG
    FROM deleted
    WHERE DONHANG.MaGiamGia_DH = deleted.MaGiamGia_DH
    DELETE GIAMGIA_DONHANG
    FROM deleted
    WHERE GIAMGIA_DONHANG.MaGiamGia_DH = deleted.MaGiamGia_DH
END;
--
CREATE TRIGGER TR_DELETE_DIACHINHAN_DH
ON DIACHINHAN
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE DONHANG
    FROM deleted
    WHERE DONHANG.MaDiaChi = deleted.MaDiaChi
    DELETE DIACHINHAN

```

```

        FROM deleted
        WHERE DIACHINHAN.MaDiaChi = deleted.MaDiaChi
END;

```

```

--
CREATE TRIGGER TRIG_DELETE_KHACHHANG
ON KHACHHANG
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE DONHANG
    FROM deleted
    WHERE DONHANG.MaKH = deleted.MAKH
    DELETE GIOHANG
    FROM deleted
    WHERE GIOHANG.MaKH = deleted.MAKH
    DELETE DIACHINHAN
    FROM deleted
    WHERE deleted.MaKH = DIACHINHAN.MaKH
    DELETE KHACHHANG
    FROM deleted
    WHERE KHACHHANG.MaKH = deleted.MAKH
END;

```

--Tạo trigger, cập nhật thời gian của update trong bảng giamgia_danhmuc

```

CREATE TRIGGER TRIG_GIAMGIA_DANHMUC_TGCAPNHAT
ON GIAMGIA_DANHMUC
AFTER UPDATE, INSERT
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE GIAMGIA_DANHMUC
    SET TGCapNhat = GETDATE()
    WHERE GIAMGIA_DANHMUC.MaGiamGia_DM IN (SELECT MaGiamGia_DM
    FROM inserted)
END;

```

--Tạo trigger, cập nhật thời gian của update trong bảng giamgia_donhang

```

CREATE TRIGGER TRIG_GIAMGIA_DONHANG_TGCAPNHAT
ON GIAMGIA_DONHANG
AFTER UPDATE, INSERT
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE GIAMGIA_DONHANG
    SET TGCapNhat = GETDATE()

```

```

WHERE GIAMGIA_DONHANG.MaGiamGia_DH IN (SELECT MaGiamGia_DH
FROM inserted)
END;

```

--Tạo trigger, cập nhật thời gian của update trong bảng sản phẩm

```

CREATE TRIGGER TRIG_GIAMGIA_SANPHAM_TGCAPNHAT
ON SANPHAM
AFTER UPDATE,INSERT
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE SANPHAM
    SET TGCapNhat = GETDATE()
    WHERE SANPHAM.MaSP IN (SELECT MaSP FROM inserted)
END;

```

--Tạo trigger, cập nhật thời gian của update trong bảng giỏ hàng

```

CREATE TRIGGER TRIG_GIAMGIA_GIOHANG_TGCAPNHAT
ON GIOHANG
AFTER UPDATE , INSERT
AS BEGIN
    SET NOCOUNT ON;
    UPDATE GIOHANG
    SET TGCapNhat = GETDATE()
    WHERE GIOHANG.MaKH IN (SELECT MaKH FROM inserted) AND
    GIOHANG.MaSanPham IN (SELECT MaSanPham FROM inserted)
END;

```

--Tạo trigger, cập nhật thời gian của update trong bảng chitiet_thanhtoan

```

CREATE TRIGGER TRIG_GIAMGIA_CHITIET_THANHTOAN_TGCAPNHAT
ON CHITIET_THANHTOAN
AFTER UPDATE, INSERT
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE CHITIET_THANHTOAN
    SET TGCapNhat = GETDATE()
    WHERE CHITIET_THANHTOAN.MaDH IN (SELECT MaDH FROM inserted)
    AND CHITIET_THANHTOAN.MaThanhToan IN (SELECT MaThanhToan FROM
inserted)
END;

```

--Tạo trigger, cập nhật thời gian của update tron bảng đánh giá

```

CREATE TRIGGER TRIG_GIAMGIA_DANHGIA_TGCAPNHAT
ON DANHGIA
AFTER UPDATE , INSERT

```

```

AS
BEGIN
    SET NOCOUNT ON;
    UPDATE DANHGIA
    SET TGCapNhat = GETDATE()
    WHERE DANHGIA.MaDH IN (SELECT MaDH FROM inserted) AND
    DANHGIA.MaSP IN (SELECT MaSP FROM inserted)
END;

```

--Tạo trigger, kiểm tra tồn kho đối với bảng chi tiết đơn hàng

```

CREATE TRIGGER TRIG_CHITIET_DONHANG_SLTONKHO
ON CHITIET_DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @CHITIET TABLE
    (MASP VARCHAR(10), SOLUONG DECIMAL(16,2) )
    INSERT INTO @CHITIET
    SELECT inserted.MaSP,SUM(inserted.SoLuong) FROM inserted group by MaSP
    IF EXISTS (
    SELECT 1 FROM @CHITIET C INNER JOIN SANPHAM ON SANPHAM.MaSP =
    C.MASP
    WHERE C.SOLUONG > SANPHAM.TonKho)
    BEGIN
        DECLARE @MASP VARCHAR(10) = (SELECT TOP 1 C.MASP FROM
        @CHITIET C INNER JOIN SANPHAM ON SANPHAM.MaSP = C.MASP
        WHERE C.SOLUONG > SANPHAM.TonKho )
        DECLARE @MALOI NVARCHAR(100)= CONCAT(N'SỐ LƯỢNG CHỌN
        VƯỢT QUÁ TỒN KHO! MASP : ', @MASP)
        RAISERROR (@MALOI, 16,1)
        ROLLBACK TRANSACTION
    END
END;

```

-- Tạo trigger, kiểm tra tồn kho cho bảng GIOHANG

```

CREATE TRIGGER TRIG_GIOHANG_CHECKSLTONGKHO
ON GIOHANG
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @GIOHANG TABLE
    (MAKH VARCHAR(10), MASP VARCHAR(10), TENSP NVARCHAR(1000),
    SOLUONG INT)
    INSERT INTO @GIOHANG

```

```

SELECT inserted.MaKH, inserted.MaSanPham, inserted.TenSP, inserted.SoLuong
FROM inserted
DECLARE @TONGHANG TABLE (MASP VARCHAR(10), TONGHANG INT)
INSERT INTO @TONGHANG
SELECT inserted.MaSanPham, SUM(inserted.SoLuong) FROM inserted GROUP BY
inserted.MaSanPham
IF EXISTS
( SELECT 1 FROM @GIOHANG G INNER JOIN SANPHAM ON
SANPHAM.MaSP = G.MASP
INNER JOIN @TONGHANG T ON T.MASP = G.MASP
WHERE T.TONGHANG > SANPHAM.TonKho)
BEGIN
    RAISERROR('KHÔNG THỂ CHỌN SẢN PHẨM - ĐÃ HẾT HÀNG!', 16, 1)
    ROLLBACK TRANSACTION
    INSERT INTO GIOHANG (MaKH, MaSanPham, SoLuong, TenSP)
    SELECT G.MAKH, G.MASP, G.SOLUONG, G.TENSP
    FROM @GIOHANG G
    WHERE G.MASP IN (SELECT G.MASP FROM @GIOHANG G INNER
    JOIN SANPHAM ON SANPHAM.MaSP = G.MASP
    INNER JOIN @TONGHANG T ON T.MASP = G.MASP WHERE
    T.TONGHANG < SANPHAM.TonKho)
END
END;

```

--Tạo trigger, khi thay đổi dữ liệu cho bảng đánh giá, phải quy chiếu đến bảng đơn hàng, nếu trạng thái là đã giao thì mới được phép thay đổi dữ liệu

```

CREATE TRIGGER TRIG_DANH GIA_KIEMTRADONHANG_DAGIAO
ON DANHGIA
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @DANH GIA TABLE (
        MADH VARCHAR(10),
        MASP VARCHAR(10),
        NGAYDANH GIA DATETIME)
    INSERT INTO @DANH GIA
    SELECT inserted.MaDH, inserted.MaSP, inserted.NgayDanhGia FROM inserted
    IF EXISTS (
        SELECT 1 FROM @DANH GIA D INNER JOIN DONHANG ON
        DONHANG.MaDH = D.MADH
        WHERE DONHANG.TrangThai != N'đã giao')
    BEGIN
        DECLARE @MADH VARCHAR(10) = (SELECT TOP 1 D.MADH
        FROM @DANH GIA D INNER JOIN DONHANG ON DONHANG.MaDH
        = D.MADH
        WHERE DONHANG.TrangThai != N'đã giao')
    END

```

```

        DECLARE @MALOI NVARCHAR(100) = CONCAT(N'KHÔNG THỂ
        ĐÁNH GIÁ ĐƠN HÀNG - TRẠNG THÁI CHƯA CẬP NHẬT ! MADH: ',
        @MADH)
        RAISERROR(@MALOI,16,1)
        ROLLBACK TRANSACTION
    END
    IF EXISTS (
    SELECT 1 FROM @DANH GIA D INNER JOIN DONHANG ON
    DONHANG.MaDH = D.MADH
    WHERE D.NGAYDANH GIA < DONHANG.NgayGiao)
    BEGIN
        DECLARE @MADH2 VARCHAR(10) = (SELECT TOP 1 D.MADH FROM
        @DANH GIA D INNER JOIN DONHANG ON DONHANG.MaDH =
        D.MADH
        WHERE D.NGAYDANH GIA < DONHANG.NgayGiao)
        DECLARE @MALOI2 NVARCHAR(100) = CONCAT(N'KHÔNG THỂ
        ĐÁNH GIÁ! ĐƠN HÀNG CHƯA ĐƯỢC GIAO! MADH: ',@MADH2)
        RAISERROR(@MALOI2,16,1)
        ROLLBACK TRANSACTION
    END
END;

```

--Trong bảng CHI TIẾT THANH TOÁN và bảng đơn hàng, ngày đặt <= ngày thanh toán <= ngày giao

```

CREATE TRIGGER TR_THANH TOAN_NGAY THANH TOAN
ON CHITIET_THANH TOAN
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @THANH TOAN TABLE (
    MADH VARCHAR(10),
    MATHANH TOAN VARCHAR(10),
    NGAY THANH TOAN DATETIME)
    INSERT INTO @THANH TOAN
    SELECT inserted.MaDH,inserted.MaThanhToan,inserted.NgayThanhToan
    FROM inserted
    IF EXISTS(
    SELECT 1 FROM @THANH TOAN T INNER JOIN DONHANG ON T.MADH =
    DONHANG.MaDH
    WHERE DONHANG.NgayDat > T.NGAY THANH TOAN OR
    T.NGAY THANH TOAN > DONHANG.NgayGiao)
    BEGIN
        DECLARE @MADH VARCHAR(10) = (SELECT TOP 1 T.MADH FROM
        @THANH TOAN T INNER JOIN DONHANG ON T.MADH =
        DONHANG.MaDH

```

```

WHERE DONHANG.NgayDat > T.NGAYTHANHTOAN OR
T.NGAYTHANHTOAN > DONHANG.NgayGiao )
DECLARE @MALOI NVARCHAR(100) = CONCAT(N'Ngày thanh toán
không hợp lệ! MADH: ', @MADH)
RAISERROR(@MALOI,16,1)
ROLLBACK TRANSACTION
END
END;

```

-- Tạo trigger, trạng thái đơn hàng chỉ được cập nhật 'đang giao', 'đã giao', 'đã hủy', 'xử lý hoàn', 'đang hoàn', 'đã hoàn', chỉ khi trạng thái trong bảng chi tiết thanh toán 'thành công'

```

CREATE TRIGGER TRIG_DONHANG_TRANGTHAI_THANHCONG
ON CHITIET_THANHTOAN
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @THANHTOAN TABLE (
        MADH VARCHAR(10),
        MATHANHTOAN VARCHAR(10),
        TRANGTHAI NVARCHAR(30))
    INSERT INTO @THANHTOAN
    SELECT inserted.MaDH,inserted.MaThanhToan,DONHANG.TrangThai
    FROM inserted
    INNER JOIN DONHANG ON DONHANG.MaDH = inserted.MaDH
    IF EXISTS (
        SELECT 1 FROM @THANHTOAN T INNER JOIN inserted ON inserted.MaDH =
        T.MADH AND inserted.MaThanhToan = T.MATHANHTOAN
        WHERE (inserted.TrangThai_TT = N'Thành công' AND T.TRANGTHAI NOT IN
        (N'đang giao', N'đã giao', N'đã hủy', N'xử lý hoàn', N'đang hoàn', N'đã hoàn'))
        OR (inserted.TrangThai_TT= N'Xử lý' AND T.TRANGTHAI NOT IN ( N'chờ xử lý',
        N'đã xử lý'))))
    BEGIN
        DECLARE @MADH VARCHAR(10) = (SELECT TOP 1 T.MADH FROM
        @THANHTOAN T INNER JOIN inserted ON inserted.MaDH = T.MADH
        AND inserted.MaThanhToan = T.MATHANHTOAN
        WHERE (inserted.TrangThai_TT = N'Thành công' AND
        T.TRANGTHAI NOT IN (N'đang giao', N'đã giao', N'đã hủy', N'xử lý hoàn',
        N'đang hoàn', N'đã hoàn')) OR (inserted.TrangThai_TT= N'Xử lý' AND
        T.TRANGTHAI NOT IN ( N'chờ xử lý', N'đã xử lý'))))

        DECLARE @MALOI NVARCHAR(100) = CONCAT(N'Trạng thái đơn hàng
        chưa cập nhật đối với thanh toán! MADH : ',@MADH)
        RAISERROR(@MALOI, 16,1 )
        ROLLBACK TRANSACTION
    END
END

```

```
END
END;
```

```
-- Tạo trigger, thống kê số lượng magiamgia_donhang trong bảng đơn hàng, số lượng  
mã tương ứng phải <= số lượng mã trong bảng giảm giá danh mục, trạng thái khác 0
```

```
CREATE TRIGGER TRIG_DONHANG_SLMAGIAM
ON DONHANG
FOR INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @MAGIAMGIA TABLE (
        MADH VARCHAR(10),
        MAGIAMGIA VARCHAR(10),
        TRANGTHAI TINYINT)
    INSERT INTO @MAGIAMGIA
    SELECT
        inserted.MaDH,inserted.MaGiamGia_DH,GIAMGIA_DONHANG.TrangThai
    FROM inserted
    INNER JOIN GIAMGIA_DONHANG ON
        GIAMGIA_DONHANG.MaGiamGia_DH = inserted.MaGiamGia_DH

    DECLARE @TONGMA TABLE
    ( MAGIAMGIA VARCHAR(10),
      TONGMA DECIMAL(16,2))
    INSERT INTO @TONGMA
    SELECT inserted.MaGiamGia_DH, COUNT(inserted.MaGiamGia_DH)
    FROM inserted GROUP BY INSERTED.MaGiamGia_DH
    IF UPDATE(MAGIAMGIA_DH)
    BEGIN
        IF EXISTS (
            SELECT 1 FROM DONHANG INNER JOIN @MAGIAMGIA M ON
            M.MADH = DONHANG.MaDH
            INNER JOIN GIAMGIA_DONHANG ON
            GIAMGIA_DONHANG.MaGiamGia_DH = M.MAGIAMGIA
            INNER JOIN @TONGMA T ON T.MAGIAMGIA = M.MAGIAMGIA
            WHERE T.TONGMA > GIAMGIA_DONHANG.SoLuongMa )
        BEGIN
            RAISERROR('ĐÃ HẾT MÃ GIẢM GIÁ CHO SẢN PHẨM  
NÀY!',16,1)
            ROLLBACK TRANSACTION
        END
    END
    IF UPDATE(MAGIAMGIA_DH)
    BEGIN
        IF EXISTS (
```



```

SELECT 1 FROM DONHANG INNER JOIN @MAGIAMGIA M
ON M.MADH = DONHANG.MaDH
WHERE M.TRANGTHAI = 0)
BEGIN
    RAISERROR('MÃ GIẢM GIÁ KHÔNG HỢP LÊ !',16,1)
    ROLLBACK TRANSACTION
END
END
END;

```

-- Tạo trigger, thống kê số lượng magiamgia_danhmuc trong bảng chi tiết đơn hàng, số lượng mã tương ứng phải <= số lượng mã trong bảng giảm giá danh mục

```

CREATE TRIGGER TRIG_CHITIET_DONHANG_SLMAGIAM
ON CHITIET_DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @MAGIAMGIA TABLE (
        MADH VARCHAR(10),
        MASP VARCHAR(10),
        MAGIAMGIA VARCHAR(10),
        TRANGTHAI TINYINT)
    INSERT INTO @MAGIAMGIA
    SELECT
        inserted.MaDH,inserted.MaSP,inserted.MaGiamGia_DM,GIAMGIA_DANHMUC.T
        rangThai
    FROM inserted
    INNER JOIN GIAMGIA_DANHMUC ON
    GIAMGIA_DANHMUC.MaGiamGia_DM = inserted.MaGiamGia_DM

    DECLARE @TONGMA TABLE
    ( MAGIAMGIA VARCHAR(10),
    TONGMA DECIMAL(16,2))
    INSERT INTO @TONGMA
    SELECT inserted.MaGiamGia_DM, COUNT(inserted.MaGiamGia_DM)
    FROM inserted GROUP BY INSERTED.MaGiamGia_DM

    IF UPDATE(MAGIAMGIA_DM)
    BEGIN
        IF EXISTS (
            SELECT 1 FROM CHITIET_DONHANG INNER JOIN @MAGIAMGIA M
            ON M.MASP = CHITIET_DONHANG.MaSP AND M.MADH =
            CHITIET_DONHANG.MaDH
            INNER JOIN GIAMGIA_DANHMUC ON
            GIAMGIA_DANHMUC.MaGiamGia_DM = M.MAGIAMGIA
            INNER JOIN @TONGMA T ON T.MAGIAMGIA = M.MAGIAMGIA

```

```

WHERE T.TONGMA > GIAMGIA_DANHMUC.SoLuongMa )
BEGIN
    RAISERROR('ĐÃ HẾT MÃ GIẢM GIÁ CHO SẢN PHẨM
    NÀY!',16,1)
    ROLLBACK TRANSACTION
END
END
IF UPDATE(SOLUONG)
BEGIN
    IF EXISTS (
        SELECT 1 FROM CHITIET_DONHANG INNER JOIN @MAGIAMGIA M
        ON M.MASP = CHITIET_DONHANG.MaSP AND M.MADH =
        CHITIET_DONHANG.MaDH
        WHERE M.TRANGTHAI = 0)
    BEGIN
        RAISERROR('MÃ GIẢM GIÁ KHÔNG HỢP LÊ !',16,1)
        ROLLBACK TRANSACTION
    END
END
END;

-- Tạo trigger, trong bảng đơn hàng, nếu đơn hàng không ở trạng thái ('đã giao', 'đã hủy',
'xử lý hoàn', 'đang hoàn', 'đã hoàn') thì ngày giao phải là NULL
CREATE TRIGGER TRIG_DONHANG_TRANGTHAI_NGAYGIAO
ON DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @TRANGTHAI TABLE (
        MADH VARCHAR(10),
        TRANGTHAI NVARCHAR(30))
    INSERT INTO @TRANGTHAI
    SELECT inserted.MaDH, inserted.TrangThai FROM inserted

    IF UPDATE (TRANGTHAI)
    BEGIN
        UPDATE DONHANG
        SET NGAYGIAO = NULL
        FROM DONHANG
        INNER JOIN @TRANGTHAI T ON T.MADH = DONHANG.MaDH
        WHERE
        DONHANG.MaDH = T.MaDH
        AND T.TRANGTHAI NOT IN (N'đã giao', N'đã hủy', N'xử lý hoàn', N'đang
        hoàn', N'đã hoàn')
        PRINT 'NGAY GIAO HANG DA CAP NHAT'
    END
END

```

END;

-- Tạo trigger, kiểm tra giá trị đơn hàng đã đủ áp dụng mã giảm giá đơn hàng chưa

```
CREATE TRIGGER TRIG_DONHANG_GIAMGIADONHANG_TOITHIEU
ON DONHANG
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @TongDonGia TABLE (
        MADH VARCHAR(10),
        TONGDONGIA DECIMAL(16, 2),
        NGAYDAT DATETIME)
    INSERT INTO @TongDonGia
    SELECT inserted.MaDH, inserted.TongDonGia, inserted.NgayDat
    FROM inserted
    IF UPDATE(TONGDONGIA)
    BEGIN
        IF EXISTS (
            SELECT 1 FROM inserted
            INNER JOIN GIAMGIA_DONHANG ON
            GIAMGIA_DONHANG.MaGiamGia_DH = inserted.MaGiamGia_DH
            INNER JOIN @TongDonGia T ON T.MADH = inserted.MaDH
            WHERE T.TONGDONGIA < GIAMGIA_DONHANG.GiaTri_Toithieu)
        BEGIN
            RAISERROR(' KHÔNG ĐẠT GIÁ TRỊ ĐƠN HÀNG TỐI
            THIỂU',16,1)
            UPDATE DONHANG
            SET GiamGia_DonHang =0, MaGiamGia_DH =NULL
            FROM DONHANG

            INNER JOIN GIAMGIA_DONHANG ON
            GIAMGIA_DONHANG.MaGiamGia_DH =
            DONHANG.MaGiamGia_DH
            INNER JOIN @TongDonGia T ON T.MADH = DONHANG.MaDH
            WHERE T.TONGDONGIA <
            GIAMGIA_DONHANG.GiaTri_Toithieu
        END
    END
END;
```

END;

-- Tạo trigger, kiểm tra giá trị của giá trị đơn hàng đã đạt đơn hàng tối thiểu chưa

```
CREATE TRIGGER TRIG_CHITIET_DONHANG_KIEMTRATOITHIEU
ON CHITIET_DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
```

```

SET NOCOUNT ON;
DECLARE @CHITIET TABLE
( MADH VARCHAR(10),
  MASP VARCHAR(10),
  DONGIA_CT DECIMAL(16,2),
  MAGIAMGIA VARCHAR(10))

DECLARE @GIAMGIA TABLE
(MAGIAMGIA VARCHAR(10),
GIATRITOITHIEU DECIMAL(16,2))
INSERT INTO @CHITIET
SELECT inserted.MaDH, inserted.MaSP,inserted.SoLuong*SANPHAM.GiaSP,
inserted.MaGiamGia_DM FROM inserted INNER JOIN SANPHAM ON
SANPHAM.MaSP = inserted.MaSP
INSERT INTO @GIAMGIA
SELECT inserted.MaGiamGia_DM, GIAMGIA_DANHMUC.GiaTri_Toithieu
FROM inserted INNER JOIN GIAMGIA_DANHMUC ON
GIAMGIA_DANHMUC.MaGiamGia_DM = inserted.MaGiamGia_DM
IF EXISTS (
SELECT 1 FROM inserted INNER JOIN @CHITIET C ON C.MADH =
inserted.MaDH AND C.MASP = inserted.MaSP
INNER JOIN @GIAMGIA G ON G.MAGIAMGIA = C.MAGIAMGIA
WHERE C.DONGIA_CT < G.GIATRITOITHIEU)
BEGIN
    RAISERROR('Giá trị chưa đạt tối thiểu!',16,1)
    UPDATE CHITIET_DONHANG
    SET LuongGiam =0, MaGiamGia_DM = NULL
    FROM CHITIET_DONHANG
    INNER JOIN @CHITIET C ON C.MADH = CHITIET_DONHANG.MaDH
    AND C.MASP =CHITIET_DONHANG.MaSP
    INNER JOIN GIAMGIA_DANHMUC ON
    GIAMGIA_DANHMUC.MaGiamGia_DM = C.MAGIAMGIA
    WHERE C.DONGIA_CT < GIAMGIA_DANHMUC.GiaTri_Toithieu
END
END;
CREATE TRIGGER TRIG_DONHANG_KIEMTRADIACHINHAN
ON DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @TAMP TABLE (MAKH VARCHAR(10),MADIACHI
    VARCHAR(10))
    INSERT INTO @TAMP SELECT MaKH, MaDiaChi FROM DIACHINHAN
    UPDATE DONHANG
    SET MaDiaChi = (SELECT TOP 1 T.MADIACHI FROM inserted INNER JOIN
    @TAMP T ON T.MAKH = inserted.MaKH)
    FROM DONHANG T INNER JOIN @TAMP D ON T.MAKH = D.MAKH

```

```

INNER JOIN inserted ON inserted.MaDH = T.MaDH
WHERE inserted.MADIACHI IS NULL AND inserted.MaDH = T.MaDH
UPDATE DONHANG
SET PhiVanChuyen = ( CASE
WHEN DIACHINHAN.ThanhPho IN (N'Hà Nội',N'Hồ Chí Minh') THEN (15000)
WHEN DIACHINHAN.ThanhPho NOT IN (N'Hà Nội',N'Hồ Chí Minh') THEN
(30000)
END
FROM DONHANG INNER JOIN inserted ON inserted.MaDH =
DONHANG.MaDH
INNER JOIN DIACHINHAN ON DIACHINHAN.MaDiaChi =
DONHANG.MaDiaChi
END;

```

--Tạo trigger, khi thay đổi dữ liệu trong bảng chi tiết đơn hàng, luonggiam có hai trường hợp : nếu magiamgia_dm có loaigiatrì là GIATRI thì luonggiam = giatri mã giảm* số lượng. Nếu magiamgia_dm có loaigiatrì là PHANTRAM thì luonggiam = GiaSP*Soluong*Giatri mã giảm, luonggiam= giamtoida nếu lượng giảm lớn hơn giảm tối đa trong bảng giảm giá danh mục lượng giảm = giaSP*soluong nếu giá trị voucher >= giaSP*soluong

```

DROP TRIGGER TRIG_CHITIET_DONHANG_LUONGGIAMGIATRI
CREATE TRIGGER TRIG_CHITIET_DONHANG_LUONGGIAMGIATRI
ON CHITIET_DONHANG
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @CHITIET TABLE
    ( MADH VARCHAR(10),
      MASP VARCHAR(10),
      MADM VARCHAR(10))
    INSERT INTO @CHITIET
    SELECT inserted.MaDH, inserted.MaSP, inserted.MaGiamGia_DM FROM inserted
    INNER JOIN SANPHAM ON SANPHAM.MaSP = inserted.MaSP

    IF EXISTS (
        SELECT 1
        FROM @CHITIET G
        INNER JOIN DANHMUC_GIAM D ON G.MADM = D.MaGiamGia_DM
        INNER JOIN inserted ON inserted.MaSP = G.MASP AND inserted.MaDH =
        G.MADH
        INNER JOIN SANPHAM ON SANPHAM.MaSP = inserted.MaSP
        WHERE D.MaDanhMuc != SANPHAM.MaDanhMuc )
    BEGIN
        DECLARE @MADH VARCHAR(10)
        SELECT TOP 1 @MADH =G.MADH FROM @CHITIET G
    END

```

```

INNER JOIN DANHMUC_GIAM D ON G.MADM =
D.MaGiamGia_DM
INNER JOIN inserted ON inserted.MaSP = G.MASP AND
inserted.MaDH = G.MADH
INNER JOIN SANPHAM ON SANPHAM.MaSP = inserted.MaSP
WHERE D.MaDanhMuc != SANPHAM.MaDanhMuc
DECLARE @MALOI NVARCHAR(255) = CONCAT(N'Sản phẩm
không thuộc danh mục giảm giá MADH: ',@MADH)
RAISERROR(@MALOI, 16, 1)

END
DECLARE @GIAMGIA_DANHMUC TABLE
( MADH VARCHAR(10),
LUONGGIAM DECIMAL(16,2),
MASP VARCHAR(10),
MAGIAMGIA_DM VARCHAR(10))
DECLARE @DONGIA TABLE
( MADH VARCHAR(10),
MASP VARCHAR(10),
DONGIA_CT DECIMAL(16,2))
INSERT INTO @DONGIA
SELECT inserted.MADH, inserted.MaSP,
inserted.SoLuong*SANPHAM.GiaSP
FROM inserted
INNER JOIN SANPHAM ON SANPHAM.MaSP = inserted.MaSP

```

-- update luonggiam theo loại giá trị mã giảm giá

```

INSERT INTO @GIAMGIA_DANHMUC
SELECT INSERTED.MADH,
LUONGGIAM =
(CASE
WHEN GIAMGIA_DANHMUC.LoaiGiaTri = 'GIATRI' THEN
(GIAMGIA_DANHMUC.GiaTri * inserted.SoLuong)
WHEN GIAMGIA_DANHMUC.LoaiGiaTri = 'PHANTRAM' THEN
(GIAMGIA_DANHMUC.GiaTri * D.DONGIA_CT/100)
WHEN inserted.MaGiamGia_DM IS NULL THEN 0
ELSE 0
END), inserted.MaSP,inserted.MaGiamGia_DM
FROM inserted
INNER JOIN SANPHAM ON SANPHAM.MaSP = inserted.MaSP
LEFT JOIN GIAMGIA_DANHMUC ON GIAMGIA_DANHMUC.MaGiamGia_DM
=inserted.MaGiamGia_DM
INNER JOIN @DONGIA D ON D.MADH = inserted.MaDH AND D.MASP =
inserted.MaSP

```

-- Cập nhật luonggiam= giamtoida nếu lượng giảm lớn hơn giảm tối đa trong bảng giảm giá danh mục

```

UPDATE G
SET LUONGGIAM = GIAMGIA_DANHMUC.GiamToiDa
FROM
    @GIAMGIA_DANHMUC G
    INNER JOIN inserted ON inserted.MaDH = G.MADH AND inserted.MaSP =
    G.MASP
    INNER JOIN GIAMGIA_DANHMUC ON
    GIAMGIA_DANHMUC.MaGiamGia_DM = G.MAGIAMGIA_DM
    WHERE G.LUONGGIAM > GIAMGIA_DANHMUC.GiamToiDa

```

```

UPDATE G
SET LUONGGIAM = 0,
MAGIAMGIA_DM = NULL
FROM @GIAMGIA_DANHMUC G INNER JOIN inserted ON inserted.MaDH =
G.MADH AND G.MASP = inserted.MaSP
INNER JOIN DANHMUC_GIAM D ON G.MAGIAMGIA_DM =
D.MaGiamGia_DM
INNER JOIN SANPHAM ON SANPHAM.MaSP = G.MaSP
WHERE D.MaDanhMuc != SANPHAM.MaDanhMuc AND D.MaGiamGia_DM =
G.MAGIAMGIA_DM

```

--Cập nhật lượng giảm = giaSP*soluong nếu giá trị voucher >= giaSP*soluong

```

UPDATE G
SET LuongGiam= D.DONGIA_CT
FROM @GIAMGIA_DANHMUC G
INNER JOIN GIAMGIA_DANHMUC ON GIAMGIA_DANHMUC.MaGiamGia_DM =
G.MAGIAMGIA_DM
INNER JOIN @DONGIA D ON D.MADH= G.MADH
INNER JOIN inserted ON inserted.MaDH = G.MADH AND inserted.MaSP = G.MASP
WHERE GIAMGIA_DANHMUC.GiaTri> D.DONGIA_CT

```

```

UPDATE CHITIET_DONHANG
SET LuongGiam = G.LUONGGIAM,
MaGiamGia_DM = G.MAGIAMGIA_DM
FROM CHITIET_DONHANG
INNER JOIN inserted ON inserted.MADH = CHITIET_DONHANG.MaDH AND
inserted.MaSP = CHITIET_DONHANG.MaSP
INNER JOIN @GIAMGIA_DANHMUC G ON G.MADH = CHITIET_DONHANG.MaDH
AND G.MASP = CHITIET_DONHANG.MaSP

```

```

UPDATE CHITIET_DONHANG
SET DonGia =D.DONGIA_CT - G.LUONGGIAM
FROM CHITIET_DONHANG
INNER JOIN inserted ON inserted.MaDH = CHITIET_DONHANG.MaDH AND
CHITIET_DONHANG.MASP = inserted.MaSP

```

```

INNER JOIN @GIAMGIA_DANHMUC G ON G.MADH = CHITIET_DONHANG.MaDH
AND CHITIET_DONHANG.MaSP= G.MASP
INNER JOIN @DONGIA D ON D.MADH = CHITIET_DONHANG.MaDH AND D.MASP
= CHITIET_DONHANG.MaSP

```

```

UPDATE DONHANG
SET TONGDONGIA = (
    SELECT SUM(D.DONGIA_CT - G.LUONGGIAM)
    FROM @GIAMGIA_DANHMUC G
    INNER JOIN @DONGIA D ON G.MADH = D.MADH AND G.MASP = D.MASP
    WHERE G.MADH = DONHANG.MaDH AND D.MADH = DONHANG.MaDH),
    TGCapNhat = GETDATE()
FROM DONHANG
INNER JOIN inserted ON inserted.MaDH = DONHANG.MaDH

```

```

DECLARE @TongDonGia TABLE (
    MADH VARCHAR(10),
    TONGDONGIA DECIMAL(16, 2))
INSERT INTO @TongDonGia
SELECT DONHANG.MaDH, DONHANG.TongDonGia
FROM DONHANG

```

```

DECLARE @GiamGia_DonHang TABLE (
    MADH VARCHAR(10),
    GIAMGIA_DONHANG DECIMAL(16, 2),
    MAGIAMGIA_DH VARCHAR(10))

```

```

INSERT INTO @GiamGia_DonHang
SELECT DONHANG.MaDH, GIAMGIA_DONHANG = (
    CASE
        WHEN GIAMGIA_DONHANG.LoaiGiaTri = 'PHANTRAM' THEN
            (T.TONGDONGIA * GIAMGIA_DONHANG.GiaTri / 100)
        WHEN GIAMGIA_DONHANG.LoaiGiaTri = 'GIATRI' THEN
            (GIAMGIA_DONHANG.GiaTri)
        ELSE 0
    END), DONHANG.MaGiamGia_DH
FROM DONHANG
INNER JOIN inserted ON inserted.MADH = DONHANG.MaDH
LEFT JOIN GIAMGIA_DONHANG ON
    GIAMGIA_DONHANG.MAGIAMGIA_DH =
    DONHANG.MAGIAMGIA_DH
INNER JOIN @TongDonGia T ON T.MADH = DONHANG.MaDH

```

```

UPDATE G
SET GIAMGIA_DONHANG = GIAMGIA_DONHANG.GiamToiDa
FROM @GiamGia_DonHang G
INNER JOIN DONHANG ON DONHANG.MaDH = G.MADH

```



```

INNER JOIN GIAMGIA_DONHANG ON
GIAMGIA_DONHANG.MaGiamGia_DH = G.MAGIAMGIA_DH
WHERE G.GiamGia_DonHang > GIAMGIA_DONHANG.GiamToiDa

```

```

UPDATE G
SET GIAMGIA_DONHANG = T.TongDonGia
FROM @GiamGia_DonHang G
INNER JOIN DONHANG ON DONHANG.MaDH = G.MADH
INNER JOIN GIAMGIA_DONHANG ON
GIAMGIA_DONHANG.MaGiamGia_DH = DONHANG.MAGIAMGIA_DH
INNER JOIN @TongDonGia T ON DONHANG.MaDH = T.MADH
WHERE GIAMGIA_DONHANG.GiaTri > T.TongDonGia

```

```

UPDATE DONHANG
SET GiamGia_DonHang = G.GIAMGIA_DONHANG
FROM DONHANG
INNER JOIN @GiamGia_DonHang G ON G.MADH = DONHANG.MaDH

```

```

UPDATE DONHANG
SET ThanhTien = T.TongDonGia + DONHANG.PhiVanChuyen -
G.GIAMGIA_DONHANG
FROM DONHANG
INNER JOIN @TongDonGia T ON T.MADH = DONHANG.MaDH
INNER JOIN @GiamGia_DonHang G ON G.MADH = DONHANG.MaDH

```

END;

5. Kết luận & hướng phát triển

5.1 Nêu các chức năng có thể phát triển thêm/cần phát triển thêm cho dự án

Cơ sở dữ liệu nhóm xây dựng là cơ sở dữ liệu đáp ứng nhu cầu cơ bản của một cơ sở dữ liệu cho cửa hàng sách trực tuyến. Cơ sở dữ liệu chỉ mới đáp ứng yêu cầu ở mức sơ khởi, chức năng cơ bản, tuy nhiên nhóm nhận thấy có một số chức năng thể phát triển thêm giúp cơ sở dữ liệu tối ưu hơn, đáp ứng nhu cầu người dùng tốt hơn:

- Chức năng gợi ý sản phẩm cho khách hàng dựa trên cơ sở lịch sử tìm kiếm và lịch sử mua hàng, tạo ra trải nghiệm mua hàng mang tính cá nhân hoá.
- Dựa trên những giao dịch (nhập hàng, bán hàng,...) đưa ra những báo cáo thống kê và phân tích. Giúp chủ, quản lý cửa hàng có thể nhìn nhận được tình hình thực tế của cửa hàng từ đó đưa ra những quyết định kinh doanh hợp lý và tối ưu nhất.
- Phân cấp nhân viên, người quản lý: đối tượng sử dụng có thể tiếp cận thông tin phù hợp với vị trí công việc của mình nhất.

- Tính năng bảo mật của cơ sở dữ liệu với một số chức năng như sau:
 - Xác thực (Authentication): Chức năng này đảm bảo rằng người dùng chỉ có thể truy cập vào cơ sở dữ liệu sau khi được xác thực. Bằng cách yêu cầu tên đăng nhập và mật khẩu, hệ thống xác thực người dùng và đảm bảo rằng chỉ những người dùng có quyền truy cập mới được phép vào.
 - Quản lý quyền truy cập (Access Control): Chức năng này quản lý quyền truy cập của người dùng vào dữ liệu trong cơ sở dữ liệu. Bạn có thể thiết lập các vai trò và quyền hạn khác nhau cho từng người dùng, đảm bảo rằng họ chỉ có thể truy cập và thao tác với dữ liệu mà họ được ủy quyền.
 - Mã hóa dữ liệu (Data Encryption): Mã hóa dữ liệu là quá trình chuyển đổi thông tin thành dạng không đọc được, gọi là mã hóa, để ngăn chặn truy cập trái phép và đảm bảo tính bảo mật của dữ liệu. Cơ sở dữ liệu nên hỗ trợ việc mã hóa dữ liệu trong quá trình lưu trữ và truyền thông giữa máy chủ và người dùng.
 - Giám sát và ghi nhật ký (Monitoring and Logging): Chức năng này giúp theo dõi các hoạt động truy cập cơ sở dữ liệu và ghi lại thông tin chi tiết về những hoạt động đó. Điều này cho phép phát hiện các hành vi nghi ngờ, theo dõi lịch sử truy cập và hỗ trợ trong việc phân tích sự việc khi có sự cố xảy ra.
 - Kiểm tra rào cản (Barrier Testing): Cơ sở dữ liệu cần được kiểm tra để xác định các rào cản bảo mật có thể tồn tại và đánh giá mức độ rủi ro. Kiểm tra bao gồm việc tìm kiếm lỗ hổng bảo mật, thiết lập kiểm tra xác thực và xác nhận các cấu hình an ninh.
 - Sao lưu và phục hồi (Backup and Recovery): Chức năng này đảm bảo rằng dữ liệu trong cơ sở dữ liệu được sao lưu định kỳ và có khả năng khôi phục sau khi xảy ra sự cố. Việc sao lưu dữ liệu đảm bảo rằng bạn có thể khôi phục dữ liệu trong trường hợp mất mát hoặc hỏng hóc.
 - Kiểm tra bảo mật (Security Auditing): Kiểm tra bảo mật là một quá trình đánh giá toàn diện về bảo mật của cơ sở dữ liệu. Nó bao gồm việc xác định và đánh giá các lỗ hổng bảo mật, kiểm tra việc tuân thủ các chính sách bảo mật và xác định các vấn đề tiềm ẩn có thể gây nguy hiểm cho hệ thống.
 - Bảo vệ từ chối dịch vụ (Denial-of-Service Protection): Để đối phó với các cuộc tấn công từ chối dịch vụ (DoS) và tấn công từ chối dịch vụ phân tán (DDoS),

cơ sở dữ liệu cần có các biện pháp bảo vệ chống lại các cuộc tấn công này. Điều này bao gồm việc cấu hình hệ thống để chịu được tải cao, sử dụng bộ lọc mạng và hạn chế số lượng yêu cầu từ một nguồn duy nhất

5.2 Nêu hướng phát triển về quy mô đề tài hoặc các hệ thống quản lý khác có thể sử dụng lại CSDL vừa thiết kế

Với quy mô là một cơ sở dữ liệu cho một cửa hàng bán sách trực tuyến giả định, đề tài có những hạn chế về thời gian, tìm lược. Một cơ sở dữ liệu nói riêng và hệ thống thương mại điện tử nói chung cần rất nhiều yếu tố để có thể vận hành trơn tru (hệ thống bảo trì và quản lý vận hành chuyên nghiệp, phần mềm ứng dụng và giao diện người dùng, đội ngũ chuyên gia,..và đặc biệt quan trọng là yếu tố bảo mật...). Nếu đề tài có thể ứng dụng vào thực tế cần phải đáp ứng những yếu tố cơ bản nêu trên để có thể vận hành tốt hệ thống. Về việc mở rộng quy mô của đề tài, đề tài được thực hiện nhằm mục đích xây dựng một cơ sở dữ liệu cho cửa hàng sách trực tuyến, đề tài có thể mở rộng với việc xây dựng cơ sở dữ liệu cho hệ thống cửa hàng sách (trên địa bàn thành phố Hồ Chí Minh hoặc có thể mở rộng hơn là trên phạm vi cả nước). Để có thể mở rộng từ cơ sở dữ liệu cho một cửa hàng sách sang cơ sở dữ liệu cho một chuỗi hệ thống cửa hàng sách (chi nhánh của cửa hàng và các cửa hàng sách khác)...

Cơ sở dữ liệu cho một cửa hàng sách trực tuyến, một số hệ thống khác có thể sử dụng cơ sở dữ liệu này như: cơ sở dữ liệu cho cửa hàng đồ ăn trực tuyến, cơ sở dữ liệu cho cửa hàng điện thoại di động, cơ sở dữ liệu cho cửa hàng giày, dép, quần áo,..Tuy nhiên phải có sự thay đổi linh hoạt các thực thể, chức năng, truy vấn,...sao cho phù hợp nhất. Như đã nói ở trên cơ sở dữ liệu có hạn chế về quy mô (chỉ dành cho một cửa hàng trực tuyến với một loại/dòng sản phẩm và những chức năng cơ bản)

NGUỒN THAM KHẢO

- [1] Nguyễn, P. Đ. L. (2023, May). *Phân tích thiết kế hệ thống thông tin. Chương 1 - Tổng quan về hệ thống thông tin*. Trường Đại học Công nghệ Thông tin - Khoa Hệ thống Thông tin; Trường Đại học Công nghệ Thông tin - Khoa Hệ thống Thông tin.