

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC XÃ HỘI VÀ NHÂN VĂN**  
**KHOA THƯ VIỆN – THÔNG TIN HỌC**

\*\*\*\*\*



# **ĐỒ ÁN KẾT THÚC MÔN**

## **Môn: Cơ sở dữ liệu 2**

**Giảng viên hướng dẫn:** *TS. Trần Đình Anh Huy*

**Nhóm sinh viên thực hiện:**

*Nguyễn Mai Xuân Linh - 2256210028*

*Nguyễn Khải Tín - 2256210051*

*Lê Trọng Phúc An - 2256210001*

*Lưu Hải Phú - 2256210046*

**Ngành học:** Thông tin – Thư viện

**Khóa:** 2022-2026

# MỤC LỤC

Danh mục hình ảnh .....	3
Danh mục bảng biểu.....	3
Từ điển thuật ngữ .....	3
I. THÔNG TIN CHUNG.....	4
1. Tên đề án .....	4
2. Nhóm thực hiện .....	4
II. TỔNG QUAN ĐỀ TÀI.....	4
1. Sự cần thiết của dự án.....	4
2. Hiện trạng thực tế .....	5
III. CÁC YÊU CẦU CHỨC NĂNG .....	6
1. Phân quyền chức năng.....	6
2. Các chức năng chính yếu.....	6
2.1. Gắn mã hoạt động tự động .....	6
2.2. Truy xuất hoạt động.....	6
2.3. Xuất hóa đơn .....	7
2.4. Đăng nhập từng đối tượng.....	7
IV. TÀI LIỆU THIẾT KẾ .....	7
1. Mô hình phân rã chức năng .....	7
2. Phân tích yêu cầu người dùng .....	8
2.1. Đăng nhập cho cả bệnh nhân và nhân viên của phòng khám .....	8
2.2. Quản lý hồ sơ bệnh nhân.....	8
2.2. Quản lý nhân viên.....	8
2.3. Quản lý quá trình khám và chữa bệnh.....	9
2.4. Quản lý thuốc và kho thuốc.....	9
2.5. Quản lý hóa đơn .....	9
2.6. Phân quyền và hỗ trợ người dùng.....	10
3. Mô hình quan hệ - thực thể.....	10
3.1. Xác định các thực thể .....	10
3.2. Xác định các thuộc tính của thực thể.....	12
4. Mô hình quan hệ dữ liệu.....	13
4.1. Xác định khóa chính, khóa ngoại: .....	13
4.2. Mô hình quan hệ dữ liệu (RDM).....	14
5. Mô hình mức vật lý .....	14
5.1. Thông tin về metadata .....	14

5.2.	Thông tin về khóa chính .....	17
5.3.	Thông tin về khóa ngoại .....	17
V.	CÀI ĐẶT CƠ SỞ DỮ LIỆU .....	18
1.	Cài đặt CSDL.....	18
1.1.	Tạo bảng .....	18
1.2.	Tạo khóa ngoại .....	22
2.	Query thể hiện chức năng CSDL.....	24
2.1.	Chức năng chung .....	24
2.2.	Chức năng riêng cho từng đối tượng .....	26
3.	Trigger kiểm soát dữ liệu.....	30
3.1.	Trigger tự động.....	30
3.2.	Trigger kiểm tra dữ liệu.....	33
3.3.	Trigger kiểm tra và xóa .....	37
VI.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	39
1.	Chức năng có thể phát triển thêm.....	39
2.	Hướng phát triển về quy mô.....	39
3.	Kết luận.....	39
	Danh mục tài liệu tham khảo .....	40

### Danh mục hình ảnh

Hình 1.	Mô hình phân rã chức năng cho Eclinic Management.....	8
Hình 2.	Mô hình ERD cho Eclinic Management .....	13
Hình 3.	Mô hình RDM cho Eclinic Management .....	14

### Danh mục bảng biểu

Bảng 1.	Xác định các thực thể trong sơ đồ ERD .....	12
Bảng 2.	Xác định thuộc tính của thực thể trong sơ đồ ERD .....	12
Bảng 3.	Xác định khóa chính khóa ngoại trong RDM .....	13
Bảng 4.	Thông tin về metadata.....	17
Bảng 5.	Thông tin về khóa chính .....	17
Bảng 6.	Thông tin về khóa ngoại .....	18
Bảng 7.	Trigger tự động .....	33
Bảng 8.	Trigger kiểm tra dữ liệu .....	37
Bảng 9.	Trigger kiểm tra và xóa.....	39

### Từ điển thuật ngữ

CSDL	Cơ sở dữ liệu
ERD	Sơ đồ quan hệ thực thể - Entity Relationship Model
RDM	Mô hình dữ liệu quan hệ - Relational Data Model

**ĐỒ ÁN MÔN HỌC**  
**ĐỀ TÀI: ECLINIC MANAGEMENT**  
**QUẢN LÝ HOẠT ĐỘNG PHÒNG KHÁM**

**I. THÔNG TIN CHUNG**

**1. Tên đề án**

- ECLINIC MANAGEMENT - Quản lý hoạt động khám chữa bệnh của phòng khám

**2. Nhóm thực hiện**

TT	Họ và tên	Mã số sinh viên	Nhiệm vụ	Đánh giá	Ghi chú
1	Nguyễn Mai Xuân Linh	2256210028	Nhóm trưởng Biên tập viên	10/10	- Tham dự các buổi họp - Hỗ trợ các thành viên khác thực hiện nhiệm vụ - Hoàn thành tốt nội dung được giao
2	Nguyễn Khải Tín	2256210051	Thành viên nhóm thiết kế Thành viên nhóm phát triển	10/10	- Tham dự các buổi họp - Hỗ trợ các thành viên khác thực hiện nhiệm vụ - Hoàn thành tốt nội dung được giao
3	Lưu Hải Phú	2256210046	Thành viên nhóm phát triển Phát ngôn viên	10/10	- Tham dự các buổi họp - Hỗ trợ các thành viên khác thực hiện nhiệm vụ - Hoàn thành tốt nội dung được giao
4	Lê Trọng Phúc An	2256210001	Thành viên nhóm thiết kế Biên tập viên	10/10	- Tham dự các buổi họp - Hỗ trợ các thành viên khác thực hiện nhiệm vụ - Hoàn thành tốt nội dung được giao

**II. TỔNG QUAN ĐỀ TÀI**

**1. Sự cần thiết của dự án**

- Các công tác chuyển đổi số và cải cách hành chính trong lĩnh vực y tế đã có những bước tiến đáng kể, bao gồm việc triển khai dịch vụ thanh toán không dùng tiền mặt và quản lý trên môi trường điện tử tại các cơ sở khám chữa bệnh. Các thủ tục hành chính và dịch vụ công đã được chuyển đổi sang hình thức trực tuyến toàn trình. Các bệnh viện trực thuộc Bộ Y tế và các cơ sở y tế tư nhân đều đã triển khai xây dựng hệ thống quản lý thông tin bệnh viện.
- Tuy nhiên, bên cạnh những kết quả đạt được, vẫn còn tồn tại những hạn chế cần khắc phục như tình trạng quá tải ở các bệnh viện chưa được giải quyết triệt để; số lượng hồ sơ và thông tin người bệnh chưa được kiểm soát tốt do quá tải; và những bất cập trong công tác quản trị, điều hành và phục vụ người bệnh tại một số cơ sở khám chữa bệnh chưa đạt yêu cầu. Đây cũng là những thách thức lớn của ngành y tế trong thời gian tới.
- Nhận thấy tầm quan trọng của nhu cầu hiện tại, nhóm nghiên cứu đã tiến hành thực hiện đề tài "ECLINIC MANAGEMENT" (Quản lý hoạt động phòng khám). Đề tài này được xây dựng nhằm phục vụ cho các nhu cầu số hóa, bắt kịp sự phát triển của công nghệ và khắc phục những điểm bất cập trong ngành y tế. Hệ thống quản lý phòng khám giúp quy trình khám và chữa bệnh diễn ra nhanh chóng, tiện lợi.

## **2. Hiện trạng thực tế**

- Phòng khám với một lượng khách hàng ổn định ngay từ những ngày đầu. Tuy nhiên, việc quản lý các hoạt động khám chữa bệnh tại đây đang đối mặt với nhiều thách thức do các phương thức vận hành vẫn còn mang tính chất truyền thống. Những phương pháp này không chỉ làm giảm hiệu quả trong việc theo dõi hồ sơ bệnh án, mà còn gây khó khăn trong việc tối ưu hóa quy trình làm việc và nâng cao chất lượng dịch vụ chăm sóc sức khỏe cho bệnh nhân. Điều này đặt ra nhu cầu cấp thiết trong việc hiện đại hóa áp dụng công nghệ mới để cải thiện toàn diện hệ thống quản lý của phòng khám.
- Sau khi thực hiện một số cải cách nhằm số hóa quy trình vận hành cơ bản, phòng khám đang hướng tới việc triển khai một hệ thống quản lý toàn diện để hỗ trợ bệnh nhân trong quá trình thăm khám. Hệ thống này sẽ lưu trữ các thông tin cơ bản của bệnh nhân, tạo điều kiện thuận lợi cho việc tra cứu và chuyển tiếp hồ sơ qua các giai đoạn khác nhau trong quy trình khám chữa bệnh.
- Thông qua việc lập hồ sơ bệnh nhân điện tử, mọi dữ liệu liên quan sẽ được tập trung trong hệ thống cơ sở dữ liệu của phòng khám, giúp việc truy cập trở nên dễ dàng và

nhANH chóng hơn. Hệ thống này sẽ bao gồm danh sách các bác sĩ chuyên khoa, và khi bệnh nhân hoàn tất việc đăng ký, họ sẽ được chỉ định khám bởi y tá

- Tiếp theo, bệnh nhân sẽ bước vào giai đoạn điều trị, nơi bác sĩ sẽ đưa ra chỉ định cần thiết, có thể là phẫu thuật hoặc các phương pháp điều trị khác, dựa trên tình trạng cụ thể của bệnh nhân. Quá trình này giúp đảm bảo rằng mỗi bệnh nhân nhận được sự chăm sóc chuyên nghiệp và chính xác nhất.
- Cuối cùng, hệ thống sẽ tự động cập nhật quá trình thanh toán hóa đơn của bệnh nhân khi họ đã được cấp mã bệnh nhân. Điều này không chỉ giúp giảm bớt các thủ tục hành chính phức tạp mà còn nâng cao hiệu quả của phòng khám. Nhờ đó, toàn bộ quy trình từ tiếp nhận, khám chữa bệnh đến thanh toán đều diễn ra mượt mà, mang lại trải nghiệm tốt nhất cho bệnh nhân và nâng cao chất lượng dịch vụ của phòng khám.

### **III. CÁC YÊU CẦU CHỨC NĂNG**

#### **1. Phân quyền chức năng**

- Y tá: Chịu trách nhiệm nhập, cập nhật, truy xuất, xóa thông tin bệnh nhân và lưu trữ vào hệ thống, truy xuất và cập nhật hóa đơn
- Bệnh nhân: Có quyền đăng nhập vào hệ thống để xem bệnh án cá nhân
- Bác sĩ chỉ định: Được nhập, cập nhật, truy xuất dữ liệu ở khâu khám bệnh, thông tin nhân viên đó
- Bác sĩ chuyên môn: Có quyền truy xuất dữ liệu khâu khám, có quyền nhập, cập nhật, truy xuất dữ liệu chữa bệnh, thuốc, thông tin nhân viên đó
- Quản trị viên: Toàn quyền quyết định quản lý hoạt động của hệ thống nhưng bị giới hạn và kiểm soát chặt chẽ khả năng truy xuất hồ sơ bệnh án nhằm bảo mật quyền riêng tư của bệnh nhân

#### **2. Các chức năng chính yếu**

##### **2.1. Gắn mã hoạt động tự động**

- Thông tin bệnh nhân được nhập vào hệ thống, lưu trữ trong CSDL và tự động tạo mã duy nhất cho mỗi bệnh nhân.
- Tự động gắn mã khám chỉ định cho từng mã bệnh nhân đã được nhập.

##### **2.2. Truy xuất hoạt động**

- Phân quyền chức năng bác sĩ chuyên môn có thể thực hiện truy xuất hồ sơ bệnh án bao gồm:
  - Truy xuất tên bệnh, tên thuốc theo mã bệnh nhân

- Thực hiện kê đơn thuốc bằng cách truy xuất mã thuốc dựa theo tên thuốc trong CSDL
- Truy xuất tên bệnh dựa theo mã bệnh
- Lưu trữ, cập nhật và truy xuất hồ sơ bệnh án của bệnh nhân
- Phân quyền chức năng y tá, bác sĩ có thể
  - Tra cứu hóa đơn bệnh nhân, xuất đơn thuốc của bệnh nhân tại phòng khám thông qua mã bệnh nhân
  - Tra cứu lịch sử khám thông qua thông tin các nhân của bệnh nhân ( họ và tên , ngày sinh)
  - Xem lịch sử điều trị của bệnh nhân tại phòng khám
  - Tra cứu loại bệnh bằng tên bệnh, bệnh bằng mã bệnh và ngược lại
  - Tra cứu bệnh nhân xuất viện, bác sĩ phụ trách bằng mã nhân viên
- Riêng đối tượng bác sĩ phụ trách bằng kê xuất đơn thuốc có thể
  - Tra cứu lượng thuốc trong kho bằng mã thuốc tên thuốc thông qua đó có thể kê đơn thuốc để bán thuốc cho phòng khám với điều kiện mã đơn thuốc tồn tại và số lượng thuốc bán ra không vượt quá số lượng tồn kho
- Phân quyền chức năng bệnh nhân có thể đăng nhập xem hồ sơ bệnh án cá nhân bao gồm hóa đơn và đơn thuốc , nhập mã bệnh nhân tra xuất dữ liệu, lịch sử khám của mỗi cá nhân

### **2.3. Xuất hóa đơn**

- Xuất ra hóa đơn của mỗi bệnh nhân bao gồm hóa đơn thuốc và hóa đơn khám chữa bệnh
- Mặc định mỗi bệnh nhân đăng kí khám đều có đơn giá khám là 150.000vnd cho mỗi lần đăng kí khám

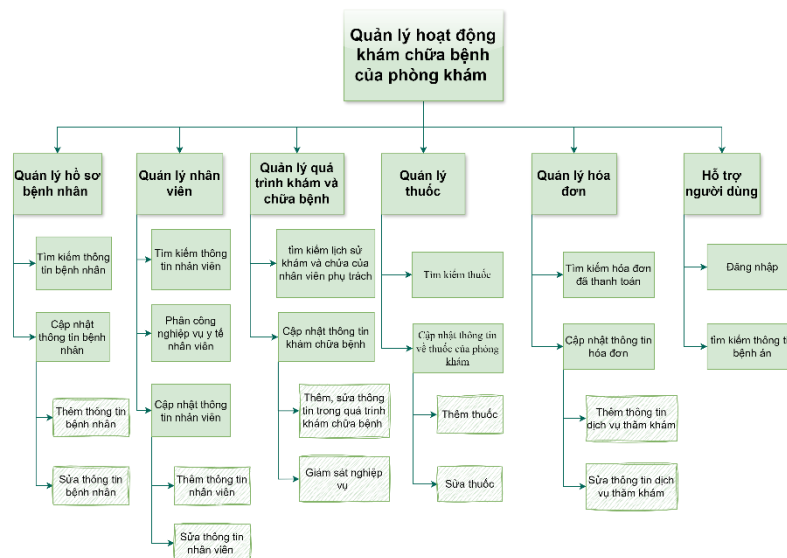
### **2.4. Đăng nhập từng đối tượng**

- Tự động mỗi đối tượng được cấp một user

## **IV. TÀI LIỆU THIẾT KẾ**

### **1. Mô hình phân rã chức năng**

- Mô hình phân rã chức năng 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. Phương pháp này được trình bày đối với ECLINIC MANAGEMENT cụ thể như sau:



Hình 1. Mô hình phân rã chức năng cho Eclinic Management

## 2. Phân tích yêu cầu người dùng

### 2.1. Đăng nhập cho cả bệnh nhân và nhân viên của phòng khám

- Người dùng, bao gồm cả bệnh nhân và nhân viên phòng khám, đều có thể đăng nhập vào trang web của phòng khám - một trang web được xây dựng trên nền tảng CSDL để sử dụng các tính năng phù hợp với vai trò và quyền hạn của họ. Bệnh nhân chỉ được phép xem và sử dụng hồ sơ bệnh án cá nhân. Bác sĩ chuyên môn có quyền xem các hồ sơ bệnh án mà họ trực tiếp khám nhằm đảm bảo bảo mật thông tin. Quản trị viên được cấp quyền xem và cập nhật thông tin nhân viên, cũng như quản lý phân quyền trong hệ thống.

### 2.2. Quản lý hồ sơ bệnh nhân

- Eclinic cung cấp chức năng tạo mã bệnh nhân chuyên biệt, bao gồm các thông tin cơ bản như họ tên, ngày sinh, giới tính, và nghề nghiệp. Hồ sơ bệnh án ban đầu này sẽ hỗ trợ việc lưu trữ và truy xuất dữ liệu bệnh nhân trong các giai đoạn tiếp theo của quá trình chăm sóc y tế. Y tá có trách nhiệm nhập thông tin tạo hồ sơ bệnh án, đăng ký khám, lưu trữ hồ sơ vào hệ thống

### 2.2. Quản lý nhân viên

- Hệ thống phải có khả năng kiểm soát và quản lý thông tin cá nhân của các nhân viên y tế (bao gồm bác sĩ chỉ định, bác sĩ chuyên môn, y tá, và quản trị viên) thông



qua các mã riêng biệt. Các mã này phân biệt chức vụ và chuyên khoa của từng nhân viên, ví dụ như khám chữa bệnh, phẫu thuật, kê đơn thuốc, xuất và thanh toán hóa đơn. Trong quá trình khám bệnh, bệnh nhân phải trải qua khám chỉ định do bác sĩ khám chỉ định phụ trách. Sau khi có kết quả khám chỉ định, tùy theo tình trạng bệnh nhân, họ sẽ được gán mã phân loại bệnh với bác sĩ có chuyên môn thích hợp để thực hiện việc khám và chữa bệnh. Bác sĩ này sẽ là người duy nhất phụ trách bệnh nhân, bao gồm việc kê thuốc, phương thức khám và các khám chỉ định.

### **2.3. Quản lý quá trình khám và chữa bệnh**

- Trong quá trình này, y tá nhập thông tin của bệnh nhân vào hệ thống. Hệ thống sẽ tự động tạo mã thông tin cho phiếu khám, đơn thuốc và hóa đơn mới. Sau đó, bác sĩ chỉ định sẽ tiếp tục khám và cập nhật thông tin chi tiết về việc khám. Điều này bao gồm việc phân loại bệnh, các phương thức khám cụ thể. Bác sĩ chỉ định cũng sẽ cập nhật thông tin về giá tiền khám trong hệ thống.
- Nếu việc điều trị bệnh đòi hỏi sự can thiệp chữa bệnh từ các bác sĩ chuyên môn khác, bác sĩ chỉ định sẽ tạo mã chữa bệnh để gán cho bệnh nhân, từ đó bác sĩ chuyên môn có thể tiếp tục điều trị bệnh nhân một cách chính xác và hiệu quả. Điều này giúp ràng buộc chặt chẽ để đảm bảo rằng mỗi bác sĩ sẽ chỉ chữa trị các trường hợp phù hợp với chuyên môn và kỹ năng của họ. Đồng thời, hệ thống cũng tự động ghi nhận và cập nhật mọi thay đổi, giúp quản lý thông tin bệnh án một cách chính xác và hiệu quả.

### **2.4. Quản lý thuốc và kho thuốc**

- Trong quá trình điều trị, bác sĩ chuyên môn cần nhập tên thuốc gắn với mã số khám bệnh tương ứng. Hệ thống sẽ tự động hiển thị mã số của thuốc, giúp bác sĩ dễ dàng lựa chọn loại thuốc phù hợp nhất cho tình trạng sức khỏe của bệnh nhân. Nếu bệnh nhân chỉ cần khám và lấy thuốc mà không cần điều trị chữa bệnh, bác sĩ chuyên môn sẽ là người kê đơn thuốc cho họ. Đảm bảo việc kê đơn thuốc được thực hiện một cách chính xác và thuận tiện.
- Hệ thống ràng buộc tổng giá đơn thuốc dựa trên giá của mỗi loại thuốc nhân với số lượng được kê trong đơn thuốc. Đồng thời, hỗ trợ việc kiểm soát số lượng thuốc bán ra và tồn kho bằng cách xác định mối quan hệ giữa số lượng thuốc trong đơn thuốc và số lượng thuốc tồn kho. Chắc chắn rằng số lượng thuốc bán ra không thể vượt quá số lượng thuốc có sẵn trong kho, giúp quản lý tồn kho tránh tình trạng thiếu hụt hoặc thừa số lượng thuốc.

### **2.5. Quản lý hóa đơn**

- Sau khi bệnh nhân hoàn thành việc thăm khám và chữa bệnh, sẽ được xuất hóa đơn điều trị cụ thể. Các thông tin liên quan bao gồm giá dịch vụ khám ban đầu (150.000 VND), các chi phí khám chi định, điều trị chữa phát sinh trong quá trình khám chữa bệnh và chi phí thuốc đã được kiểm tra, xác nhận và kết nối với mã bệnh nhân được cấp ban đầu. Tổng hợp một cách chính xác số tiền mà bệnh nhân phải chi trả. Mã quản lý hóa đơn cũng hỗ trợ việc truy xuất thông tin khi cần thiết, đảm bảo tính chính xác và quyền lợi của bệnh nhân trong quá trình khám chữa bệnh tại phòng khám.

## 2.6. Phân quyền và hỗ trợ người dùng

- Phân quyền hợp lý cho từng nghiệp vụ cụ thể, đồng thời hỗ trợ người dùng (bệnh nhân) trong việc truy xuất hồ sơ bệnh án cá nhân.

## 3. Mô hình quan hệ - thực thể

### 3.1. Xác định các thực thể

Tên thực thể	Loại thực thể	Chức năng	Ghi chú
NhanVien	Thực thể mạnh	Lưu trữ thông tin cần của nhân viên trong phòng khám	
PhongBan	Thực thể mạnh	Lưu trữ thông tin nghiệp vụ bác sĩ, y tá giúp phòng khám phân bổ nhiệm vụ	
ChuyenMon	Thực thể yếu	Nhằm phân bổ đúng chuyên môn cho từng bác sĩ	Đại diện cho chuyên môn của bác sĩ nhưng không thể được xác định độc lập mà dựa vào sự tồn tại của một thực thể bác sĩ và phòng ban
BSchidinh, BSchuyenmon	Thực thể kế thừa	Phân bổ đúng nghiệp vụ cho từng nhiệm vụ phù hợp của phòng khám	Thực thể con Yta, BacSi "kế thừa" các thuộc tính và phương thức của thực thể cha NHANVIEN và PHONGBAN, từ thực thể BacSi mở rộng ra thực thể CHUYENMON

KhamChiDinh	Thực thể mạnh	Lưu trữ thông tin y tế của bệnh nhân được bác sĩ khám chỉ định, cũng như hỗ trợ quá trình chẩn đoán và xác định liệu cần gặp bác sĩ để điều trị trực tiếp hay chỉ cần được cung cấp đơn thuốc	
ChuaBenh	Thực thể mạnh	Lưu trữ thông tin thăm khám của bác sĩ chuyên môn về các liệu pháp điều trị bệnh	
Chuabenhphuongthuc	Thực thể mạnh	Xác định phương thức chữa bệnh của bác sĩ dành cho bệnh nhân	
Khamchitiet	Thực thể liên kết	Đưa ra chẩn đoán bệnh từ việc khám chỉ định và điều trị chữa bệnh	Tạo ra để liên kết giữa các thực thể với nhau thông qua mối quan hệ.
Donthuoc	Thực thể mạnh	Tạo đơn thuốc cho từng bệnh nhân	
Benh	Thực thể mạnh	Lưu trữ mã bệnh, tên bệnh, phân loại bệnh	
Khamphuongthuc	Thực thể mạnh	Cho biết phương thức khám của khám chỉ định	
Donthuocchitiet	Thực thể liên kết	Đưa ra đơn thuốc chi tiết cho từng đơn thuốc và loại thuốc	Liên kết giữa hai thực thể DONTUOC và THUOC thông qua mối quan hệ nhiều nhiều
Thuoc	Thực thể mạnh	Hỗ trợ quản lý dược phẩm cho phòng khám	
Benhnhan	Thực thể mạnh	Lưu trữ hồ sơ thông tin bệnh án của bệnh nhân	

DangNhap	Thực thể mạnh	Chứa thông tin cá nhân bệnh nhân, hồ sơ bệnh án, hóa đơn thanh toán	
Hoadon	Thực thể mạnh	Lưu trữ thông tin hóa đơn của từng bệnh nhân	

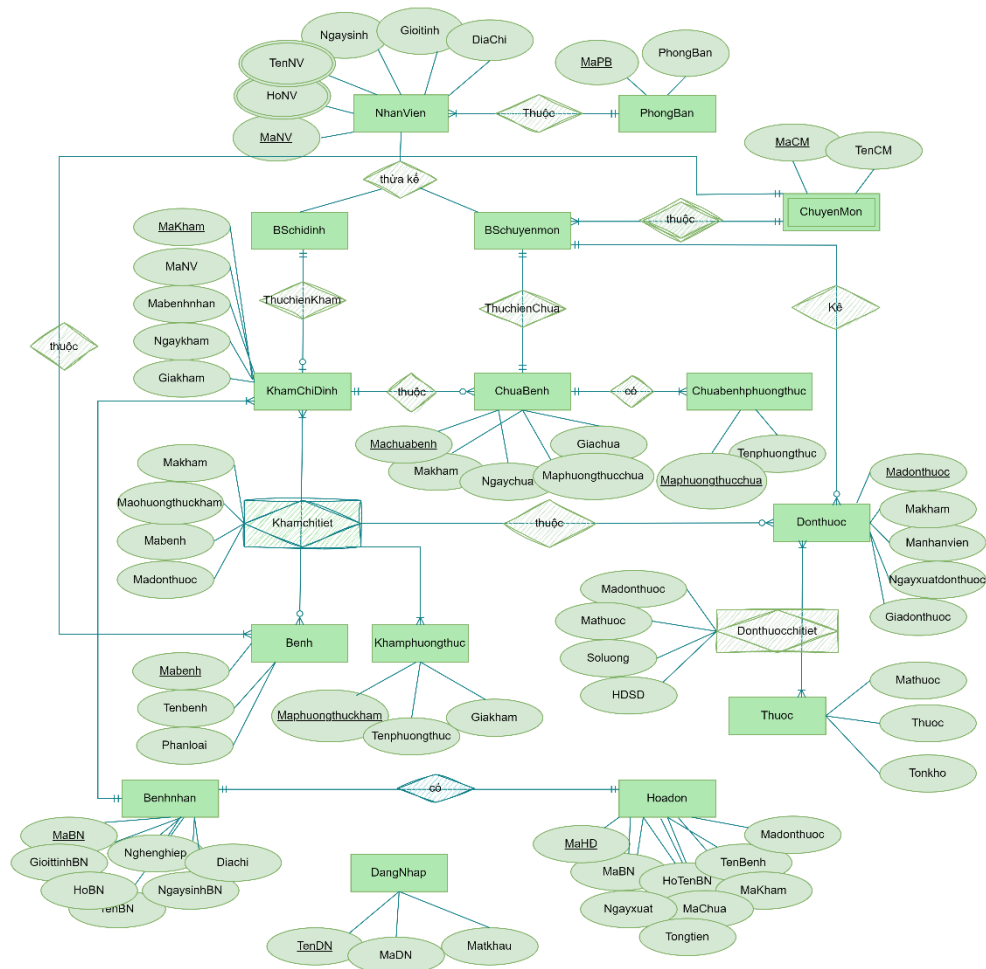
Bảng 1. Xác định các thực thể trong sơ đồ ERD

### 3.2. Xác định các thuộc tính của thực thể

Tên thực thể	Tên thuộc tính	Tên thực thể	Tên thuộc tính
Benh	Mabenh	Hoadon	MaHĐ
	Tenbenh		MaBN
	Phanloai		HoTenBN
Benhnhan	MaBN		TenBenh
	HoBN		MaKham
	TenBN		MaChua
	NgaysinhBN		Madonthuoc
	GioitinhBN		Ngayxuat
	Nghenghiiep		Tongtien
	Diachi	Khamchidinh	Makham
ChuaBenh	Machuabenh		Mabenhnhan
	Makham		MaNV
	Ngaychua		Ngaykham
	Maphuongthucchua		Giakham
ChuyenMon	Giachua	NhanVien	MaNV
	MaCM		TenNV
Donthuoc	TenCM		HoNV
	Madonthuoc		Ngaysinh
	Makham		Gioitinh
	Manhanvien		Diachi
	Ngayxuatdonthuoc	PhongBan	MaPB
DangNhap	Giadonthuoc		PhongBan
	MaDn	Thuoc	Mathuoc
	TenDN		Thuoc
	Matkhau		Tonkho

Bảng 2. Xác định thuộc tính của thực thể trong sơ đồ ERD

### 3.2.1. Xác định các quan hệ



Hình 2. Mô hình ERD cho Eclinic Management

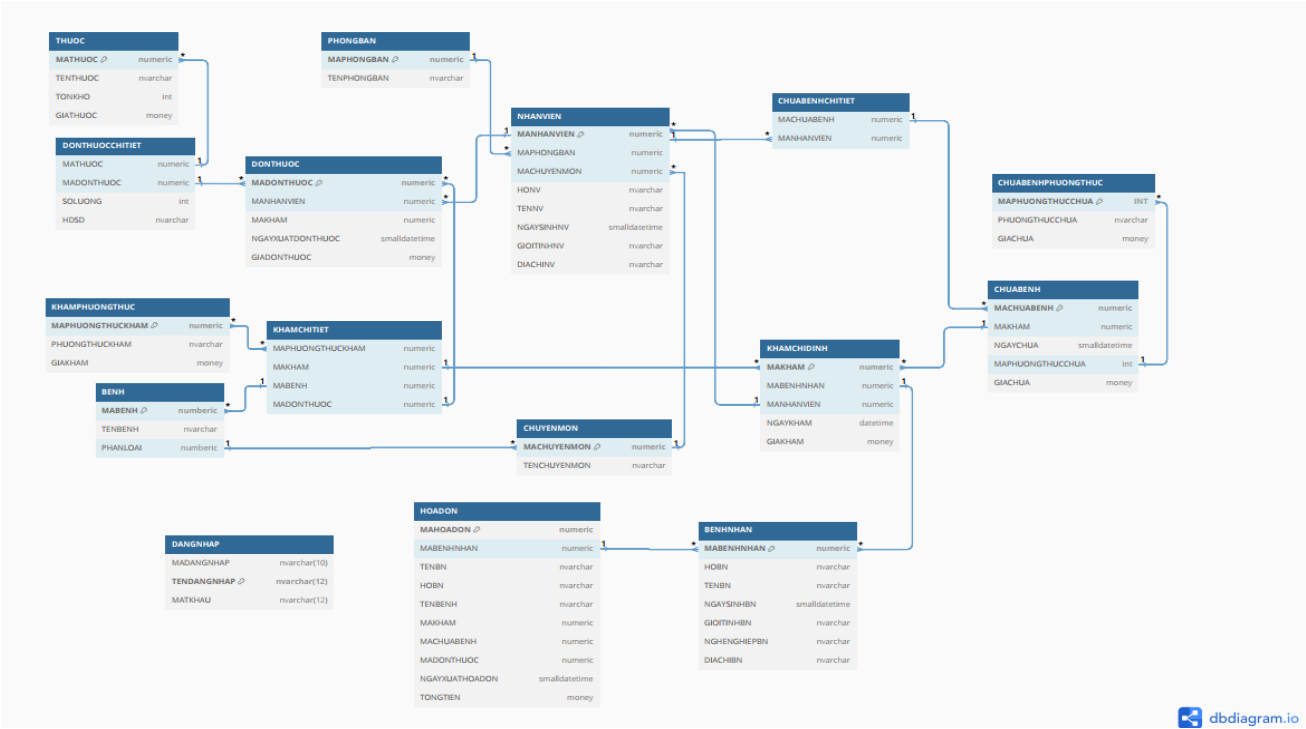
#### 4. Mô hình quan hệ dữ liệu

#### 4.1. Xác định khóa chính, khóa ngoại:

Tên bảng	Khóa chính	Khóa ngoại
BENH	MABENH	
NHANVIEN	MANHANVIEN	MAPHONGBAN, MACHUYENMON
BENH	MABENH	PHANLOAI
CHUABENH	MACHUABENH	MAKHAM, MAPHUONGTHUCCHUA
DONTHUOC	MADONTHUOC	MAKHAM
HOADON	MAHOADON	MABENH
KHAMCHIDINH	MAKHAM	MABENH, MANHANVIEN
PHONGBAN	MAPHONGBAN	
THUOC	MATHUOC	
DANGNHAP	TENDANGNHAP	
KHAMPHUONGTHUC	MAKHAMPHUONGTHUC	
CHUABENHPHUONGTHUC	MACHUABENHPHUONGTHUC	

*Bảng 3. Xác định khóa chính khóa ngoại trong RDM*

4.2. Mô hình quan hệ dữ liệu (RDM)



Hình 3. Mô hình RDM cho Eclinc Management

5. Mô hình mức vật lý

5.1. Thông tin về metadata

Tên bảng	Tên thuộc tính	Kiểu dữ liệu	Tham chiếu	Giá trị mặc định	NOT NULL
BENHNHAN	MABENHNHAN	[numeric](8, 0)			TRUE
	TENBN	[nvarchar](100)			
	HOBN	[nvarchar](100)			
	NGAYSINHBN	[smalldatetime]			
	GIOITINHBN	[nvarchar](4)			
	NGHENGHIEP	[nvarchar](100)			
	DIACHIBN	[nvarchar](100)			
BENH	MABENH	[numeric](3, 0)			TRUE
	TENBENH	[nvarchar](100)			
	PHANLOAI	[numeric](3, 0)			
KHAMCHIDINH	MAKHAM	[numeric](8, 0)			TRUE
	MABENHNHAN	[numeric](8, 0)	BENHNHAN		

	MANHANVIEN	[numeric](8, 0)	NHANVIE N		
	NGAYKHAM	[datetime]		Ngayhie ntai	
	GIAKHAM	[money]			
KHAMCHIT IET	MAPHUONGTHUC KHAM	[numeric](3, 0)			
	MAKHAM	[numeric](8, 0)	KHAMCH IDINH		
	MABENH	[numeric](3, 0)	BENH		
	MADONTHUOC	[numeric](8, 0)	DONTHU OC		
KHAMPHU ONGTHUC	MAPHUONGTHUC KHAM	[numeric](3, 0)			TRUE
	PHUONGTHUCKH AM	[nvarchar](100)			
	GIAKHAM	[money]			
CHUABENH	MACHUABENH	[numeric](8, 0)			TRUE
	MAKHAM	[numeric](8, 0)	KHAM		
	NGAYCHUA	[smalldatetime]			
	PHUONGTHUCCHU A	INT			
	GIACHUA	[money]			
CHUABENH CHITIET	MACHUABENH	[numeric](8, 0)	CHUABE NH		
	MANHANVIEN	[numeric](8, 0)	NHANVIE N		
CHUABENH PHUONGTH UC	MAPHUONGTHUC CHUA	INT			
	PHUONGTHUCCHU A	[nvarchar](100)			
	GIACHUA	[money]			
NHANVIEN	MANHANVIEN	[numeric](8, 0)			

	MAPHONGBAN	[numeric](3, 0)	PHONGBAN		
	MACHUYENMON	[numeric](3, 0)	CHUYENMON		
	TENNV	[nvarchar](100)			
	HONV	[nvarchar](100)			
	NGAYSINHNHANV IEN	[smalldatetime]			
	GIOITINHNHANVI EN	[nvarchar](4)			
	DIACHINV	[nvarchar](100)			
CHUYEMON	MACHUYENMON	[numeric](3, 0)			TRUE
	TENCHUYENMON	[nvarchar](100)			
PHONGBAN	MAPHONGBAN	[numeric](3, 0)			TRUE
	TENPHONGBAN	[nvarchar](100)			
THUOC	MATHUOC	[numeric](6, 0)			TRUE
	THUOC	[nvarchar](100)			
	TONKHO	[int]			
	GIATHUOC	[money]			
HOADON	MAHOADON	[numeric](8, 0)			TRUE
	MABENHNHAN	[numeric](8, 0)	BENHNHAN		
	TENBN	[nvarchar](100)			
	HOBN	[nvarchar](100)			
	TENBENH	[nvarchar](100)			
	MAKHAM		KHAM		
	MACHUABENH	[numeric](8, 0)	CHUABENH		
	MADONTHUOC	[numeric](8, 0)	DONTHUOC		
	NGAYXUATHOADON	[smalldatetime]			
	TONGTIEN	[money]			



DONTHUOC	MADONTHUOC	[numeric](8, 0)			TRUE
	MAKHAM	[numeric](8, 0)	KHAM		
	MANHANVIEN	[numeric](8, 0)	NHANVIE N		
	NGAYXUATHOAD ON	[smalldatetime]			
	GIADONTHUOC	[money]			
DONTHUOC CHITIET	MATHUOC	[numeric](6, 0)	THUOC		
	MADONTHUOC	[numeric](8, 0)	DONTHU OC		
	SOLUONG	[int]		0	
	HUONGDANSUDU NG	[nvarchar](100)			
DANGNHAP	MADANGNHAP	[numeric](8,0)			TRUE
	TENDANGNHAP	[nvarchar](12)			TRUE
	MATKHAU	[nvarchar](12)			TRUE

Bảng 4. Thông tin về metadata

## 5.2. Thông tin về khóa chính

Tên bảng	Tên Khóa	Thuộc tính khóa
BENHNHAN	pk_MaBenhnhan	MABENHNHAN
BENH	pk_MaBenh	MABENH
KHAMCHIDINH	pk_MaKham	MAKHAM
KHAMPHUONGTHUC	pk_MaPhuongthucKham	MAPHUONGTHUCKHAM
CHUABENH	pk_MaChuaBenh	MACHUABENH
CHUABENHPHUONGTHUC	pk_MaPhuongthucChua	MAPHUONGTHUCCHUA
NHANVIEN	pk_MaNhanvien	MANHANVIEN
CHUYENMON	pk_MaChuyenmon	MACHUYENMON
PHONGBAN	pk_MaPhongban	MAPHONGBAN
THUOC	pk_MaThuoc	MATHUOC
HOADON	pk_MaHoadon	MAHOADON
DONTHUOC	pk_MaDonthuoc	MADONTHUOC
DANGNHAP	pk_TenDangnhap	TENDANGNHAP

Bảng 5. Thông tin về khóa chính

## 5.3. Thông tin về khóa ngoại

Tên Khóa	Bảng chứa khóa	Thuộc tính	Bảng được tham chiếu	Khóa chính
FK_KHAMCHITIET_KHAMPHUONGTHUC	KHAMCHITIET	[MAPHUONGTHUCKHAM]	KHAMPHUONGTHUC	[MAPHUONGTHUCKHAM]
FK_KHAMCHITIET_KHAM	KHAMCHITIET	[MAKHAM]	KHAMCHIDINH	[MAKHAM]
FK_KHAMCHIDINH_BENHNHAN	KHAMCHIDINH	[MABENHNHAN]	BENHNHAN	[MABENHNHAN]
FK_KHAMCHITIET_BENH	KHAMCHITIET	[MABENH]	BENH	[MABENH]
FK_CHUABENH_KHAMCHIDINH	CHUABENH	[MAKHAM]	KHAMCHIDINH	[MAKHAM]
FK_CHUABENHCHITIET_CHUABENH	CHUABENHCHITIET	[MACHUABENH]	CHUABENH	[MACHUABENH]
FK_CHUABENHCHITIET_NHANVIEN	CHUABENHCHITIET	[MANHANVIEN]	NHANVIEN	[MANHANVIEN]
FK_CHUABENH_CHUABENHPHUONGTHUC	CHUABENH	[MAPHUONGTHUCCHUA]	CHUABENHPHUONGTHUC	[MAPHUONGTHUCCHUA]
FK_NHANVIEN_PHONGBAN	NHANVIEN	[MAPHONGBAN]	PHONGBAN	[MAPHONGBAN]
FK_NHANVIEN_CHUYENMON	NHANVIEN	[MACHUYENMON]	CHUYENMON	[MACHUYENMON]
FK_KHAMCHIDINH_NHANVIEN	KHAMCHIDINH	[MANHANVIEN]	NHANVIEN	[MANHANVIEN]
FK_HOADON_BENHNHAN	HOADON	[MABENHNHAN]	BENHNHAN	[MABENHNHAN]
FK_DONTHUOC_NHANVIEN	DONTHUOC	[MANHANVIEN]	NHANVIEN	[MANHANVIEN]
FK_KHAMCHITIET_DONTHUOC	KHAMCHITIET	[MADONTHUOC]	DONTHUOC	[MADONTHUOC]
FK_DONTHUOCCHITIET_DONTHUOC	DONTHUOCCHITIET	[MADONTHUOC]	DONTHUOC	[MADONTHUOC]
FK_DONTHUOCCHITIET_THUOC	DONTHUOCCHITIET	[MATHUOC]	THUOC	[MATHUOC]

Bảng 6. Thông tin về khóa ngoại

## V. CÀI ĐẶT CƠ SỞ DỮ LIỆU

### 1. Cài đặt CSDL

#### 1.1. Tạo bảng

--TẠO BẢNG BỆNH NHÂN  
SET ANSI\_NULLS ON

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BENHNHAN](
    [MABENHNHAN] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
    [TENBN] [nvarchar](100) NULL,
    [HOBN] [nvarchar](100) NULL,
    [NGAYSINHBN] [smalldatetime] NULL,
    [GIOITINHBN] [nvarchar](4) NULL,
    [NGHENGHIEP] [nvarchar](100) NULL,
    [DIACHIBN] [nvarchar](100) NULL
    CONSTRAINT [PK_BENHNHAN] PRIMARY KEY NONCLUSTERED
(
    [MABENHNHAN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG BỆNH
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BENH](
    [MABENH] [numeric](3, 0) IDENTITY(100,1) NOT NULL,
    [TENBENH] [nvarchar](100) NULL,
    [PHANLOAI] [numeric](3, 0) NULL
    CONSTRAINT [PK_BENH] PRIMARY KEY NONCLUSTERED
(
    [MABENH] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG KHÁM CHỈ ĐỊNH
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[KHAMCHIDINH](
    [MAKHAM] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
    [MABENHNHAN] [numeric](8, 0) NULL,
    [MANHANVIEN] [numeric](8, 0) NULL,
    [NGAYKHAM] [datetime] NULL,
    [GIAKHAM] [money] NULL
    CONSTRAINT [PK_KHAM] PRIMARY KEY NONCLUSTERED
(
    [MAKHAM] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG CHI TIẾT KHÁM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[KHAMCHITIET](
    [MAPHUONGTHUCKHAM] [numeric](3, 0) NULL,
    [MAKHAM] [numeric](8, 0) NULL,
    [MABENH] [numeric](3, 0) NULL,
    [MADONTHUOC] [numeric](8, 0) NULL,
) ON [PRIMARY]
GO
--TẠO BẢNG PHƯƠNG THỨC KHÁM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[KHAMPHUONGTHUC](
    [MAPHUONGTHUCKHAM] [numeric](3, 0) IDENTITY(100,1) NOT NULL,

```

```

        [PHUONGTHUCKHAM] [nvarchar](100) NULL,
        [GIAKHAM] [money] NULL
    CONSTRAINT [PK_PHUONGTHUCKHAM] PRIMARY KEY NONCLUSTERED
(
    [MAPHUONGTHUCKHAM] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-- TẠO BẢNG CHỮA BỆNH
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CHUABENH](
    [MACHUABENH] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
    [MAKHAM] [numeric](8, 0) NULL,
    [NGAYCHUA] [smalldatetime] NULL,
    [MAPHUONGTHUCCHUA] INT NULL,
    [GIACHUA] [money] NULL
    CONSTRAINT [PK_CHUABENH] PRIMARY KEY NONCLUSTERED
(
    [MACHUABENH] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG CHỮA BỆNH CHI TIẾT
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CHUABENHCHITIET](
    [MACHUABENH] [numeric](8, 0) NULL,
    [MANHANVIEN] [numeric](8, 0) NULL
) ON [PRIMARY]
GO
--TẠO BẢNG PHƯƠNG THỨC CHỮA BỆNH
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CHUABENHPHUONGTHUC](
    [MAPHUONGTHUCCHUA] INT NOT NULL,
    [PHUONGTHUCCHUA] [nvarchar](100) NULL,
    [GIACHUA] [money] NULL
    CONSTRAINT [PK_PHUONGTHUCCHUABENH] PRIMARY KEY NONCLUSTERED
(
    [MAPHUONGTHUCCHUA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG NHÂN VIÊN PHÒNG KHÁM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[NHANVIEN](
    [MANHANVIEN] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
    [MAPHONGBAN] [numeric](3, 0) NULL,
    [MACHUYENMON] [numeric](3, 0) NULL,
    [TENNV] [nvarchar](100) NULL,
    [HONV] [nvarchar](100) NULL,
    [NGAYSINHNV] [smalldatetime] NULL,
    [GIOITINHNV] [nvarchar](4) NULL,
    [DIACHINV] [nvarchar](100) NULL,
    CONSTRAINT [PK_NHANVIEN] PRIMARY KEY NONCLUSTERED
(

```

```

[MANHANVIEN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
--TẠO BẢNG CHUYÊN MÔN
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CHUYENMON](
[MACHUYENMON] [numeric](3, 0) IDENTITY(100,1) NOT NULL,
[TENCHUYENMON] [nvarchar](100) NULL,
CONSTRAINT [PK_CHUYENMON] PRIMARY KEY NONCLUSTERED
(
[MACHUYENMON] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG PHÒNG BAN
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PHONGBAN](
[MAPHONGBAN] [numeric](3, 0) IDENTITY(100,1) NOT NULL,
[TENPHONGBAN] [nvarchar](100) NULL,
CONSTRAINT [PK_PHONGBAN] PRIMARY KEY NONCLUSTERED
(
[MAPHONGBAN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
-- TẠO BẢNG THUỐC
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[THUOC](
[MATHUOC] [numeric](6, 0) IDENTITY(100000,1) NOT NULL,
[THUOC] [nvarchar](100) NULL,
[TONKHO] [int] NULL,
[GIATHUOC] [money] NULL,
CONSTRAINT [PK_THUOC] PRIMARY KEY NONCLUSTERED
(
[MATHUOC] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG HÓA ĐƠN
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HOADON](
[MAHOADON] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
[MABENHNHAN] [numeric](8, 0) NULL,
[TENBN] [nvarchar](100) NULL,
[HOBN] [nvarchar](100) NULL,
[TENBENH] [nvarchar](100) NULL,
[MAKHAM] [numeric](8, 0) NULL,
[MACHUABENH] [numeric](8, 0) NULL,
[MADONTHUOC] [numeric](8, 0) NULL,
[NGAYXUATHOADON] [smalldatetime] NULL,
[TONGTIEN] [money] NULL,
CONSTRAINT [PK_HOADON] PRIMARY KEY NONCLUSTERED

```

```

(
    [MAHOADON] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG ĐƠN THUỐC
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DONTHUOC](
    [MADONTHUOC] [numeric](8, 0) IDENTITY(10000000,1) NOT NULL,
    [MAKHAM] [numeric](8, 0) NULL,
    [MANHANVIEN] [numeric](8, 0) NULL,
    [NGAYXUATDONTHUOC] [smalldatetime] NULL,
    [GIADONTHUOC] [money] NULL
CONSTRAINT [PK_DONTHUOC] PRIMARY KEY NONCLUSTERED
(
    [MADONTHUOC] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
--TẠO BẢNG ĐƠN THUỐC CHI TIẾT
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DONTHUOCCHITIET](
    [MATHUOC] [numeric](6, 0) NULL,
    [MADONTHUOC] [numeric](8, 0) NULL,
    [SOLUONG] [int] NULL,
    [HDSO] [nvarchar](100) NULL
) ON [PRIMARY]
GO
--TẠO BẢNG ĐĂNG NHẬP
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DANGNHAP](
    [MADANGNHAP] [numeric](8, 0) NULL,
    [TENDANGNHAP] [NVARCHAR] (12) NOT NULL,
    [MATKHAU] [NVARCHAR] (12) NOT NULL
CONSTRAINT [PK_DANGNHAP] PRIMARY KEY NONCLUSTERED
(
    [TENDANGNHAP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

## 1.2. Tạo khóa ngoại

```

-- ForeignKey [FK_KHAMCHITIET_KHAMPHUONGTHUC]
ALTER TABLE [dbo].[KHAMCHITIET] WITH CHECK ADD CONSTRAINT
[FK_KHAMCHITIET_KHAMPHUONGTHUC] FOREIGN KEY([MAPHUONGTHUCKHAM] )
REFERENCES [dbo].[KHAMPHUONGTHUC] ([MAPHUONGTHUCKHAM] )
GO
ALTER TABLE [dbo].[KHAMCHITIET] CHECK CONSTRAINT [FK_KHAMCHITIET_KHAMPHUONGTHUC]
GO
-- ForeignKey [FK_KHAMCHITIET_KHAMCHIDINH]
ALTER TABLE [dbo].[KHAMCHITIET] WITH CHECK ADD CONSTRAINT [FK_KHAMCHITIET_KHAMCHIDINH]
FOREIGN KEY([MAKHAM] )
REFERENCES [dbo].[KHAMCHIDINH] ([MAKHAM] )
GO
ALTER TABLE [dbo].[KHAMCHITIET] CHECK CONSTRAINT [FK_KHAMCHITIET_KHAMCHIDINH]
GO
-- ForeignKey [FK_KHAMCHIDINH_BENHNHAN]

```

```

ALTER TABLE [dbo].[KHAMCHIDINH] WITH CHECK ADD CONSTRAINT [FK_KHAMCHIDINH_BENHNHAN]
FOREIGN KEY([MABENHNHAN] )
REFERENCES [dbo].[BENHNHAN] ([MABENHNHAN] )
GO
ALTER TABLE [dbo].[KHAMCHIDINH] CHECK CONSTRAINT [FK_KHAMCHIDINH_BENHNHAN]
GO
-- ForeignKey [FK_KHAMCHITIET_BENH]
ALTER TABLE [dbo].[KHAMCHITIET] WITH CHECK ADD CONSTRAINT [FK_KHAMCHITIET_BENH]
FOREIGN KEY([MABENH] )
REFERENCES [dbo].[BENH] ([MABENH] )
GO
ALTER TABLE [dbo].[KHAMCHITIET] CHECK CONSTRAINT [FK_KHAMCHITIET_BENH]
GO
-- ForeignKey [FK_CHUABENH_KHAMCHIDINH]
ALTER TABLE [dbo].[CHUABENH] WITH CHECK ADD CONSTRAINT [FK_CHUABENH_KHAMCHIDINH]
FOREIGN KEY([MAKHAM] )
REFERENCES [dbo].[KHAMCHIDINH] ([MAKHAM] )
GO
ALTER TABLE [dbo].[CHUABENH] CHECK CONSTRAINT [FK_CHUABENH_KHAMCHIDINH]
GO
-- ForeignKey [FK_CHUABENHCHITIET_CHUABENH]
ALTER TABLE [dbo].[CHUABENHCHITIET] WITH CHECK ADD CONSTRAINT
[FK_CHUABENHCHITIET_CHUABENH] FOREIGN KEY([MACHUABENH] )
REFERENCES [dbo].[CHUABENH] ([MACHUABENH] )
GO
ALTER TABLE [dbo].[CHUABENHCHITIET] CHECK CONSTRAINT [FK_CHUABENHCHITIET_CHUABENH]
GO
-- ForeignKey [FK_CHUABENHCHITIET_NHANVIEN]
ALTER TABLE [dbo].[CHUABENHCHITIET] WITH CHECK ADD CONSTRAINT
[FK_CHUABENHCHITIET_NHANVIEN] FOREIGN KEY([MANHANVIEN] )
REFERENCES [dbo].[NHANVIEN] ([MANHANVIEN] )
GO
ALTER TABLE [dbo].[CHUABENHCHITIET] CHECK CONSTRAINT [FK_CHUABENHCHITIET_NHANVIEN]
GO
-- ForeignKey [FK_CHUABENH_CHUABENHPHUONGTHUC]
ALTER TABLE [dbo].[CHUABENH] WITH CHECK ADD CONSTRAINT
[FK_CHUABENH_CHUABENHPHUONGTHUC] FOREIGN KEY([MAPHUONGTHUCCHUA])
REFERENCES [dbo].[CHUABENHPHUONGTHUC] ([MAPHUONGTHUCCHUA])
GO
ALTER TABLE [dbo].[CHUABENH] CHECK CONSTRAINT [FK_CHUABENH_CHUABENHPHUONGTHUC]
GO
-- ForeignKey [FK_NHANVIEN_PHONGBAN]
ALTER TABLE [dbo].[NHANVIEN] WITH CHECK ADD CONSTRAINT [FK_NHANVIEN_PHONGBAN] FOREIGN
KEY([MAPHONGBAN])
REFERENCES [dbo].[PHONGBAN] ([MAPHONGBAN])
GO
ALTER TABLE [dbo].[NHANVIEN] CHECK CONSTRAINT [FK_NHANVIEN_PHONGBAN]
GO
-- ForeignKey [FK_NHANVIEN_CHUYENMON]
ALTER TABLE [dbo].[NHANVIEN] WITH CHECK ADD CONSTRAINT [FK_NHANVIEN_CHUYENMON] FOREIGN
KEY([MACHUYENMON])
REFERENCES [dbo].[CHUYENMON] ([MACHUYENMON])
GO
ALTER TABLE [dbo].[NHANVIEN] CHECK CONSTRAINT [FK_NHANVIEN_CHUYENMON]
GO
-- ForeignKey [FK_KHAMCHIDINH_NHANVIEN]
ALTER TABLE [dbo].[KHAMCHIDINH] WITH CHECK ADD CONSTRAINT [FK_KHAMCHIDINH_NHANVIEN]
FOREIGN KEY([MANHANVIEN] )
REFERENCES [dbo].[NHANVIEN] ([MANHANVIEN] )
GO
ALTER TABLE [dbo].[KHAMCHIDINH] CHECK CONSTRAINT [FK_KHAMCHIDINH_NHANVIEN]
GO
-- ForeignKey [FK_HOADON_BENHNHAN]
ALTER TABLE [dbo].[HOADON] WITH CHECK ADD CONSTRAINT [FK_HOADON_BENHNHAN] FOREIGN
KEY([MABENHNHAN] )
REFERENCES [dbo].[BENHNHAN] ([MABENHNHAN] )
GO
ALTER TABLE [dbo].[HOADON] CHECK CONSTRAINT [FK_HOADON_BENHNHAN]
GO
-- ForeignKey [FK_DONTHUOC_NHANVIEN]

```

```

ALTER TABLE [dbo].[DONTHUOC] WITH CHECK ADD CONSTRAINT [FK_DONTHUOC_NHANVIEN] FOREIGN
KEY([MANHANVIEN])
REFERENCES [dbo].[NHANVIEN] ([MANHANVIEN])
GO
ALTER TABLE [dbo].[DONTHUOC] CHECK CONSTRAINT [FK_DONTHUOC_NHANVIEN]
GO
-- ForeignKey [FK_KHAMCHITIET_DONTHUOC]
ALTER TABLE [dbo].[KHAMCHITIET] WITH CHECK ADD CONSTRAINT [FK_KHAMCHITIET_DONTHUOC]
FOREIGN KEY([MADONTHUOC])
REFERENCES [dbo].[DONTHUOC] ([MADONTHUOC])
GO
ALTER TABLE [dbo].[KHAMCHITIET] CHECK CONSTRAINT [FK_KHAMCHITIET_DONTHUOC]
GO
-- ForeignKey [FK_CHITIETDONTHUOC_DONTHUOC]
ALTER TABLE [dbo].[DONTHUOCCHITIET] WITH CHECK ADD CONSTRAINT
[FK_DONTHUOCCHITIET_DONTHUOC] FOREIGN KEY([MADONTHUOC])
REFERENCES [dbo].[DONTHUOC] ([MADONTHUOC])
GO
ALTER TABLE [dbo].[DONTHUOCCHITIET] CHECK CONSTRAINT [FK_DONTHUOCCHITIET_DONTHUOC]
GO
-- ForeignKey [FK_CHITIETDONTHUOC_THUOC]
ALTER TABLE [dbo].[DONTHUOCCHITIET] WITH CHECK ADD CONSTRAINT
[FK_DONTHUOCCHITIET_THUOC] FOREIGN KEY([MATHUOC])
REFERENCES [dbo].[THUOC] ([MATHUOC])
GO
ALTER TABLE [dbo].[DONTHUOCCHITIET] CHECK CONSTRAINT [FK_DONTHUOCCHITIET_THUOC]
GO
-- ForeignKey [FK_BENH_CHUYENMON]
ALTER TABLE [dbo].[BENH] WITH CHECK ADD CONSTRAINT [FK_BENH_CHUYENMON] FOREIGN
KEY([PHANLOAI])
REFERENCES [dbo].[CHUYENMON] ([MACHUYENMON])
GO
ALTER TABLE [dbo].[BENH] CHECK CONSTRAINT [FK_BENH_CHUYENMON]
GO

```

## 2. Query thể hiện chức năng CSDL

### 2.1. Chức năng chung

#### 2.1.1. Cho các nhóm đối tượng (y tá, bác sĩ, bệnh nhân)

```

-- PROCEDURE 1: NHẬP MÃ BỆNH NHÂN TRA CỨU HÓA ĐƠN
CREATE PROCEDURE HOADON_BENHNHAN
    @MABENHNHAN numeric(8, 0)
AS
BEGIN
    SELECT *
    FROM HOADON
    WHERE MABENHNHAN = @MABENHNHAN
END
EXEC HOADON_BENHNHAN @MABENHNHAN = '10000002'
-- PROCEDURE 2: NHẬP MÃ BỆNH NHÂN XUẤT ĐƠN THUỐC
CREATE PROCEDURE DONTHUOC_BENHNHAN
    @MABENHNHAN numeric(8, 0)
AS
BEGIN
    SELECT BN.MABENHNHAN, BN.HOBN, BN.TENBN, BN.DIACHIBN, B.TENBENH, T.THUOC,
    CT.SOLUONG, CT.HDSD, (T.GIATHUOC * CT.SOLUONG) AS GIA
    FROM DONTHUOC DT
    JOIN DONTHUOCCHITIET CT ON DT.MADONTHUOC = CT.MADONTHUOC
    JOIN THUOC T ON T.MATHUOC = CT.MATHUOC
    JOIN KHAMCHIDINH KCD ON KCD.MAKHAM = DT.MAKHAM
    JOIN KHAMCHITIET KCT ON KCT.MAKHAM = KCD.MAKHAM
    JOIN BENH B ON B.MABENH = KCT.MABENH
    JOIN BENHNHAN BN ON BN.MABENHNHAN = KCD.MABENHNHAN
    WHERE BN.MABENHNHAN = @MABENHNHAN
END
EXEC DONTHUOC_BENHNHAN @MABENHNHAN = '10000000'
-- PROCEDURE 3: TRA CỨU LỊCH SỬ KHÁM BẰNG HỌ TÊN VÀ NGÀY SINH BỆNH NHÂN
CREATE PROCEDURE TRACUALICHSUKHAM
    @HOBN [nvarchar](100),
    @TENBN [nvarchar](100),

```



```

@NGAYSINH [smalldatetime]
AS
BEGIN
    SELECT BN.*, HD.TENBENH, HD.NGAYXUATHOADON, COUNT(*) AS N'SỐ LẦN KHÁM'
    FROM BENHNHAN BN
    JOIN HOADON HD ON HD.MABENHNHAN = BN.MABENHNHAN
    WHERE BN.HOBN = @HOBN AND BN.TENBN = @TENBN AND BN.NGAYSINHBN = @NGAYSINH
    GROUP BY BN.MABENHNHAN, BN.TENBN, BN.HOBN, NGAYSINHBN, GIOITINHBN, NGHENGHIEP,
    DIACHIBN, HD.TENBENH, HD.NGAYXUATHOADON
END
EXEC TRACUALICHSUKHAM @HOBN = N'Nguyễn Mai Xuân', @TENBN = N'Linh', @NGAYSINH = '2000-
01-13'
-- PROCEDURE 4: XEM BỆNH NHÂN ĐƯỢC CHỮA BỞI PHÒNG BAN NÀO, THUỘC CHUYÊN MÔN NÀO VÀ
PHƯƠNG THỨC CHỮA LÀ GÌ?
CREATE PROCEDURE THONGTINBENHNHAN
    @MABENHNHAN NUMERIC(8, 0)
AS
BEGIN
    SELECT DISTINCT
        BN.MABENHNHAN,
        BN.TENBN,
        NV.TENNV,
        PB.TENPHONGBAN,
        CM.TENCHUYENMON,
        CBPT.PHONGTHUCCHUA
    FROM
        BENHNHAN BN
        JOIN KHAMCHIDINH KCD ON KCD.MABENHNHAN = BN.MABENHNHAN
        JOIN CHUABENH CB ON CB.MAKHAM = KCD.MAKHAM
        JOIN CHUABENHPHONGTHUC CBPT ON CBPT.MAPHUONGTHUCCHUA = CB.MAPHUONGTHUCCHUA
        JOIN CHUABENHCHITIET CBCT ON CBCT.MACHUABENH = CB.MACHUABENH
        JOIN NHANVIEN NV ON NV.MANHANVIEN = CBCT.MANHANVIEN
        JOIN CHUYENMON CM ON CM.MACHUYENMON = NV.MACHUYENMON
        JOIN PHONGBAN PB ON PB.MAPHONGBAN = NV.MAPHONGBAN
    WHERE
        BN.MABENHNHAN = @MABENHNHAN
END
GO
EXEC THONGTINBENHNHAN @MABENHNHAN = 10000001

```

## 2.1.2.Cho hai đối tượng y tá, bác sĩ

```

-- PROCEDURE 5: TRA CỨU LOẠI BỆNH BẰNG TÊN BỆNH VÀ NGƯỢC LẠI
CREATE PROCEDURE TRACUU_LOAIBENH
    @PHANLOAI NVARCHAR (255),
    @MAPHANLOAI NUMERIC (3,0)
AS
BEGIN
    SELECT B.PHANLOAI, CM.TENCHUYENMON, B.TENBENH
    FROM BENH B
    JOIN CHUYENMON CM ON CM.MACHUYENMON = B.PHANLOAI
    WHERE CM.TENCHUYENMON = @PHANLOAI
    OR B.PHANLOAI = @MAPHANLOAI
END

EXEC TRACUU_LOAIBENH @PHANLOAI = N'Tiêu hóa - Gan mật', @MAPHANLOAI = NULL
-- PROCEDURE 6: TRA CỨU TÊN BỆNH BẰNG MÃ BỆNH VÀ NGƯỢC LẠI
CREATE PROCEDURE BENH_MA
    @MABENH numeric(3, 0),
    @TENBENH [nvarchar](100)
AS
BEGIN
    SELECT *
    FROM BENH
    WHERE MABENH = @MABENH
    OR TENBENH = @TENBENH
END
EXEC BENH_MA @MABENH = 103 , @TENBENH = N'Loét thực quản'
-- PROCEDURE 7: TRA CỨU BỆNH NHÂN ĐÃ XUẤT VIỆN HAY CHỮA
CREATE PROCEDURE XUATVIEN
    @TRANGTHAI NVARCHAR(255)
AS

```

```

BEGIN
    IF @TRANGTHAI = 'DA XUAT VIEN'
    BEGIN
        SELECT *
        FROM HOADON
        WHERE NGAYXUATHOADON IS NOT NULL
    END
    ELSE IF @TRANGTHAI = 'CHUA XUAT VIEN'
    BEGIN
        SELECT *
        FROM HOADON
        WHERE NGAYXUATHOADON IS NULL
    END
END
EXEC XUATVIEN @TRANGTHAI = 'DA XUAT VIEN'
-- PROCEDURE 8: TRA CỨU DANH SÁCH BỆNH NHÂN BÁC SĨ PHỤ TRÁCH BẰNG MÃ NHÂN VIÊN
CREATE PROCEDURE DANHSACH_BENHNHAN_BACSI
    @MABACSI numeric(8, 0)
AS
BEGIN
    SELECT
        HD.MABENHNHAN, HD.HOBN, HD.TENBN
    FROM HOADON HD
    JOIN CHUABENH CB ON CB.MACHUABENH = HD.MACHUABENH
    JOIN CHUABENHCHITIET CBCT ON CBCT.MACHUABENH = CB.MACHUABENH
    JOIN NHANVIEN NV ON NV.MANHANVIEN = CBCT.MANHANVIEN
    WHERE NV.MANHANVIEN = @MABACSI
END
EXEC DANHSACH_BENHNHAN_BACSI @MABACSI = 10000000

```

## 2.2. Chức năng riêng cho từng đối tượng

### 2.2.1. Cho đối tượng bác sĩ

```

-- PROCEDURE 9: TRA CỨU SỐ LƯỢNG THUỐC TRONG KHO BẰNG MÃ THUỐC HOẶC TÊN THUỐC, BÁN THUỐC
TỰ ĐỘNG NHẬP DỮ LIỆU VÀO BẢNG VỚI ĐIỀU KIỆN MÃ ĐƠN THUỐC TỒN TẠI VÀ SỐ LƯỢNG THUỐC BÁN
RA KHÔNG VƯỢT QUÁ SỐ LƯỢNG TỒN KHO
CREATE PROCEDURE BAN_THUOC
    @MADONTHUOC NUMERIC(8, 0) = NULL,
    @MATHUOC NUMERIC(6, 0),
    @TENTHUOC NVARCHAR(255),
    @SOLUONGBAN INT = 0,
    @HDSG NVARCHAR(100) = NULL
AS
BEGIN
    DECLARE @SOLUONGKHO INT;
    DECLARE @TEN_THUOC NVARCHAR (255);

    SELECT DISTINCT T.MATHUOC, T.THUOC, T.TONKHO
    FROM THUOC T
    WHERE (T.MATHUOC = @MATHUOC OR @MATHUOC IS NULL)
    AND (T.THUOC = @TENTHUOC OR @TENTHUOC IS NULL)

    IF NOT EXISTS (SELECT * FROM DONTHUOC WHERE MADONTHUOC = @MADONTHUOC OR @MADONTHUOC
    IS NULL)
    BEGIN
        RAISERROR ('MÃ ĐƠN THUỐC KHÔNG TỒN TẠI', 16, 1)
        RETURN
    END
    IF @SOLUONGBAN > 0
    BEGIN
        SELECT @SOLUONGKHO = TONKHO, @TEN_THUOC = THUOC
        FROM THUOC
        WHERE MATHUOC = @MATHUOC
        IF @SOLUONGBAN > @SOLUONGKHO
        BEGIN
            RAISERROR ('SỐ LƯỢNG THUỐC BÁN RA LỚN HƠN SỐ LƯỢNG THUỐC TRONG KHO', 16, 1)
            ROLLBACK TRANSACTION
        END
        ELSE
            BEGIN
                UPDATE THUOC

```

```

        SET TONKHO = TONKHO - @SOLUONGBAN
        WHERE MATHUOC = @MATHUOC
        INSERT DONTUOCCHITIET (MATHUOC, MADONTUOC, SOLUONG, HDSD)
        VALUES (@MATHUOC, @MADONTUOC, @SOLUONGBAN, @HDSG)
        PRINT (N'ĐÃ BÁN THUỐC VÀ CẬP NHẬT KHO THÀNH CÔNG. TÊN THUỐC: ' +
        @TEN_THUOC + N'. SỐ LƯỢNG BÁN: ' + CAST(@SOLUONGBAN AS NVARCHAR(10)) + N'.')
    END
END
END
-- TRA CỨU THUỐC
EXEC BAN_THUOC @MATHUOC = 100012, @TENTHUOC = NULL
-- BÁN THUỐC
EXEC BAN_THUOC @MADONTUOC = 10000011,
               @MATHUOC = 100012,
               @TENTHUOC = NULL,
               @SOLUONGBAN = 1,
               @HDSG = N'NHAI THUỐC'

```

### 2.2.2. Cho đối tượng bệnh nhân

```

-- PROCEDURE 10 : BỆNH NHÂN CÓ QUYỀN ĐĂNG NHẬP HỆ THỐNG ĐỂ XEM BỆNH ÁN CÁ NHÂN
--USER BENHNHAN1 - OLDPASS: "password1", NEWPASS: "123"
CREATE PROCEDURE HOSOBN
AS
BEGIN
DECLARE @TENDANGNHAP NVARCHAR(128) = SUSER_SNAME()
SELECT *
FROM dbo.BENHNHAN
WHERE MABENHNHAN = ( SELECT MADANGNHAP
                     FROM dbo.DANGNHAP
                     WHERE TENDANGNHAP = @TENDANGNHAP)
END
GO
GRANT EXECUTE ON HOSOBN TO [BENHNHAN1]
-- PROCEDURE 11 : XEM HÓA ĐƠN BỆNH NHÂN
CREATE PROCEDURE HOADONBN
AS
BEGIN
DECLARE @TENDANGNHAP NVARCHAR(128) = SUSER_SNAME()
DECLARE @MADANGNHAP NUMERIC(8, 0)
SELECT @MADANGNHAP = MADANGNHAP
FROM dbo.DANGNHAP
WHERE TENDANGNHAP = @TENDANGNHAP
SELECT *
FROM dbo.HOADON
WHERE MABENHNHAN = @MADANGNHAP
END
GO
GRANT EXECUTE ON HOADONBN TO [BENHNHAN1]
-- PROCEDURE 12 : XEM ĐƠN THUỐC BỆNH NHÂN
CREATE PROCEDURE DONTUOCBN
AS
BEGIN
DECLARE @TENDANGNHAP NVARCHAR(128) = SUSER_SNAME()
SELECT DISTINCT BN.MABENHNHAN, BN.HOBN, BN.TENBN, BN.DIACHIBN, B.TENBENH, T.THUOC,
CT.SOLUONG, CT.HSD, (T.GIATHUOC * CT.SOLUONG) AS GIA
FROM DONTUOC DT
JOIN DONTUOCCHITIET CT ON DT.MADONTUOC = CT.MADONTUOC
JOIN THUOC T ON T.MATHUOC = CT.MATHUOC
JOIN KHAMCHIDINH KCD ON KCD.MAKHAM = DT.MAKHAM
JOIN KHAMCHITIET KCT ON KCT.MAKHAM = KCD.MAKHAM
JOIN BENH B ON B.MABENH = KCT.MABENH
JOIN BENHNHAN BN ON BN.MABENHNHAN = KCD.MABENHNHAN
JOIN DANGNHAP DN ON DN.MADANGNHAP = BN.MABENHNHAN
WHERE DN.TENDANGNHAP = @TENDANGNHAP
END
GO
GRANT EXECUTE ON DONTUOCBN TO [BENHNHAN1]

```

### 2.2.3. Cho đối tượng quản trị viên

```

-- PROCEDURE 13 : TẠO LOGIN, USER TỰ ĐỘNG CHO BỆNH NHÂN VÀ NHÂN VIÊN
CREATE PROCEDURE USERS

```

```

AS
BEGIN
    DECLARE @MADANGNHAP NUMERIC(8, 0)
    DECLARE @TENDANGNHAP NVARCHAR(12)
    DECLARE @MATKHAU NVARCHAR(12)
    DECLARE @SQL NVARCHAR(MAX)
    DECLARE cur CURSOR FOR
    SELECT MADANGNHAP, TENDANGNHAP, MATKHAU FROM dbo.DANGNHAP
    OPEN cur
    FETCH NEXT FROM cur INTO @MADANGNHAP, @TENDANGNHAP, @MATKHAU
    WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @SQL = 'CREATE LOGIN [' + CONVERT(VARCHAR(12), @TENDANGNHAP) + '] WITH
PASSWORD = ''' + @MATKHAU + ''';'
        EXEC sp_executesql @SQL

        SET @SQL = 'CREATE USER [' + CONVERT(VARCHAR(12), @TENDANGNHAP) + '] FOR LOGIN
[' + CONVERT(VARCHAR(12), @TENDANGNHAP) + '];'
        EXEC sp_executesql @SQL
        FETCH NEXT FROM cur INTO @MADANGNHAP, @TENDANGNHAP, @MATKHAU
    END
    CLOSE cur
    DEALLOCATE cur
END
GO
EXEC USERS
-- PROCEDURE 14 : THỐNG KÊ DOANH THU TỪNG THÁNG THEO NĂM
CREATE PROCEDURE THONGKE_DOANHTHU
    @NAM int
AS
BEGIN
    SELECT
        ISNULL([1], 0) AS [Thang1],
        ISNULL([2], 0) AS [Thang2],
        ISNULL([3], 0) AS [Thang3],
        ISNULL([4], 0) AS [Thang4],
        ISNULL([5], 0) AS [Thang5],
        ISNULL([6], 0) AS [Thang6],
        ISNULL([7], 0) AS [Thang7],
        ISNULL([8], 0) AS [Thang8],
        ISNULL([9], 0) AS [Thang9],
        ISNULL([10], 0) AS [Thang10],
        ISNULL([11], 0) AS [Thang11],
        ISNULL([12], 0) AS [Thang12]

    FROM (
        SELECT
            MONTH(NGAYXUATHOADON) AS THANG, HD.TONGTIEN
        FROM HOADON HD
        WHERE YEAR(NGAYXUATHOADON) = @NAM
    ) AS DOANHTHU
    PIVOT
    (
        SUM(TONGTIEN)
        FOR THANG IN ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12])
    ) AS THANG
END
EXEC THONGKE_DOANHTHU @NAM = 2023
-- PROCEDURE 15 : TÍNH LƯƠNG NHÂN VIÊN BẰNG MÃ NHÂN VIÊN VÀ NĂM (LƯƠNG CƠ BẢN 10 TRIỆU +
THƯỜNG 10% TIỀN CỐ KHÁM/CHỮA/THUỐC NHÂN VIÊN ĐÓ PHỤ TRÁCH)
CREATE PROCEDURE TINH_LUONG_NHANVIEN
    @MANHANVIEN numeric(8, 0),
    @NAM int
AS
BEGIN
    DECLARE @LUONGCOBAN money
    DECLARE @THUONG money
    DECLARE @TONGLUONG money

    SET @LUONGCOBAN = 10000000.0000
    SELECT
        @THUONG = ISNULL(SUM((0.1 * ISNULL(KCD.GIAKHAM, 0)) +
(0.1 * ISNULL(CB.GIACHUA, 0)) +

```

```

        (0.1 * ISNULL(DT.GIADONTHUOC, 0))), 0)
FROM KHAMCHIDINH KCD
LEFT JOIN CHUABENH CB ON CB.MAKHAM = KCD.MAKHAM
LEFT JOIN CHUABENHCHITIET CBCT ON CBCT.MACHUABENH = CB.MACHUABENH
LEFT JOIN DONTUOC DT ON DT.MAKHAM = KCD.MAKHAM
LEFT JOIN HOADON HD ON HD.MAKHAM = KCD.MAKHAM
WHERE (KCD.MANHANVIEN = @MANHANVIEN OR CBCT.MANHANVIEN = @MANHANVIEN OR
DT.MANHANVIEN = @MANHANVIEN)
AND YEAR(HD.NGAYXUATHOADON) = @NAM
SET @TONGLUONG = @LUONGCOBAN + @THUONG
SELECT @MANHANVIEN AS MANHANVIEN, @LUONGCOBAN AS LUONGCOBAN, @THUONG AS THUONG,
@TONGLUONG AS TONGLUONG
END
EXEC TINH_LUONG_NHANVIEN @MANHANVIEN = 10000003, @NAM = 2024
-- PROCEDURE 16 : THỐNG KÊ TỶ LỆ BỆNH CỦA PHÒNG KHÁM
CREATE PROCEDURE TILE_BENH
AS
BEGIN
    SELECT TENCHUYENMON,
        COUNT(*) AS SOLUONG,
        CAST(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM BENH) AS DECIMAL(5,2)) AS TILE
    INTO #BENH
    FROM BENH
    JOIN CHUYENMON ON MACHUYENMON = PHANLOAI
    GROUP BY TENCHUYENMON

    CREATE TABLE #KETQUA (
        [Hạng mục] NVARCHAR(50),
        [Tai mũi họng] DECIMAL(5,2) NULL,
        [Tim mạch] DECIMAL(5,2) NULL,
        [Hô hấp - Dịch ứng miễn dịch lâm sàng] DECIMAL(5,2) NULL,
        [Tiêu hóa - Gan mật] DECIMAL(5,2) NULL,
        [Truyền nhiễm] DECIMAL(5,2) NULL
    )

    INSERT INTO #KETQUA ([Hạng mục], [Tai mũi họng], [Tim mạch], [Hô hấp - Dịch ứng miễn
dịch lâm sàng], [Tiêu hóa - Gan mật], [Truyền nhiễm])
    SELECT N'SỐ LƯỢNG', *
    FROM (
        SELECT TENCHUYENMON, SOLUONG
        FROM #BENH
    ) AS SL
    PIVOT (
        MAX(SOLUONG) FOR TENCHUYENMON IN ([Tai mũi họng], [Tim mạch], [Hô hấp - Dịch ứng
miễn dịch lâm sàng], [Tiêu hóa - Gan mật], [Truyền nhiễm])
    ) AS PV_SOLUONG;
    INSERT INTO #KETQUA ([Hạng mục], [Tai mũi họng], [Tim mạch], [Hô hấp - Dịch ứng miễn
dịch lâm sàng], [Tiêu hóa - Gan mật], [Truyền nhiễm])
    SELECT N'TỶ LỆ (%)', *
    FROM (
        SELECT TENCHUYENMON, TILE
        FROM #BENH
    ) AS TL
    PIVOT (
        MAX(TILE) FOR TENCHUYENMON IN ([Tai mũi họng], [Tim mạch], [Hô hấp - Dịch ứng miễn
dịch lâm sàng], [Tiêu hóa - Gan mật], [Truyền nhiễm])
    ) AS PV_TILE

    SELECT * FROM #KETQUA

    DROP TABLE #BENH
    DROP TABLE #KETQUA
END
EXEC TILE_BENH
-- PROCEDURE 17 : THỐNG KÊ TOP 10 BỆNH CÓ XÁC XUẤT MẮC PHẢI CAO NHẤT
CREATE PROCEDURE TOP_BENH
AS
BEGIN
    CREATE TABLE #TAM_BENH (
        TENBENH NVARCHAR(255),
        SOLUONG INT,

```

```

        XACXUAT DECIMAL(5, 2)
    );
    INSERT INTO #TAM_BENH (TENBENH, SOLUONG, XACXUAT)
    SELECT
        TENBENH,
        COUNT(KHAMCHITITET.MABENH) AS SOLUONG,
        CAST(COUNT(KHAMCHITITET.MABENH) * 1.0 / (SELECT COUNT(*) FROM BENH) AS
    DECIMAL(5,2)) AS TILE
    FROM BENH
    JOIN KHAMCHITITET ON BENH.MABENH = KHAMCHITITET.MABENH
    GROUP BY TENBENH

    SELECT TOP 10 TENBENH, SOLUONG, XACXUAT
    FROM #TAM_BENH
    ORDER BY XACXUAT DESC

    DROP TABLE #TAM_BENH
END
EXEC TOP_BENH
--FUNCTION 1 : TẠO FUNCTION XẾP HẠNG CÁC LOẠI THUỐC DỰA THEO SỐ LƯỢNG BÁN RA TRONG NĂM
CREATE FUNCTION XEPHANGTHUOCBAN
(
    @NAM INT
)
RETURNS TABLE
AS
RETURN
(
    SELECT DT.MATHUOC AS MaThuoc, T.Thuoc, SUM(DT.SoLuong) AS TongSoLuongBan,
    CASE
        WHEN SUM(DT.SoLuong) > 30 THEN N'THUỐC BÁN CHẠY'
        WHEN SUM(DT.SoLuong) < 5 THEN N'THUỐC BÁN Ế' ELSE N'THUỐC BÁN BÌNH THƯỜNG'
    END AS TrangThai
    FROM THUOC T
        LEFT JOIN DONTHUOCCHITITET DT ON T.MATHUOC = DT.MATHUOC
        LEFT JOIN HOADON HD ON HD.MADONTHUOC = DT.MADONTHUOC
    WHERE YEAR(HD.NGAYXUATHOADON) = @NAM
    GROUP BY DT.MATHUOC, T.Thuoc)
    SELECT * FROM XEPHANGTHUOCBAN(2023)

```

### 3. Trigger kiểm soát dữ liệu

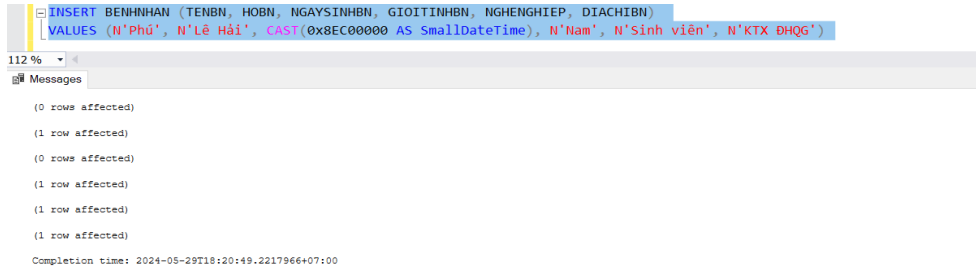
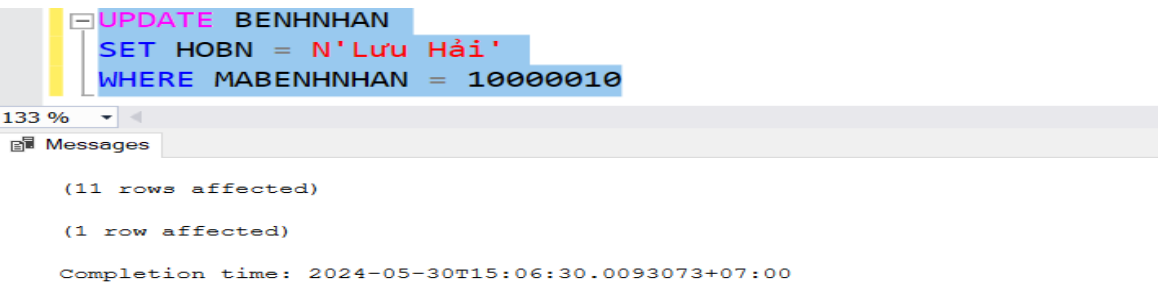
#### 3.1. Trigger tự động

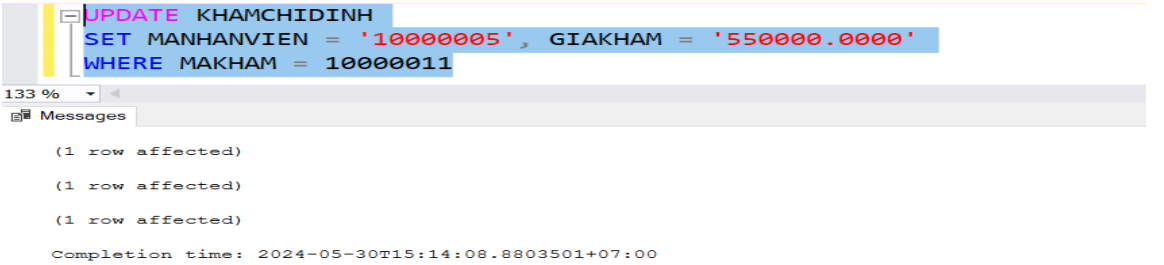
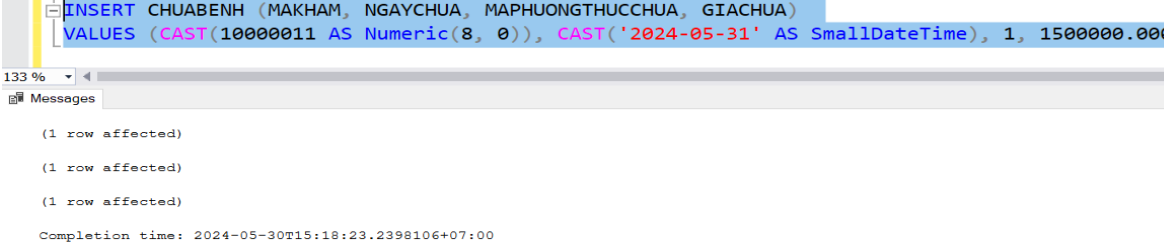
CODE

```

--TRIGGER 1: TỰ ĐỘNG TẠO MÃ KHÁM CHO BỆNH NHÂN MỚI
CREATE TRIGGER TRG_IN_KHAMCHIDINH ON BENHNHAN AFTER INSERT
AS
BEGIN
    INSERT INTO KHAMCHIDINH (MABENHNHAN, MANHANVIEN, NGAYKHAM, GIAKHAM)
    SELECT
        INSERTED.MABENHNHAN, NULL, GETDATE(), 150000.0000
    FROM INSERTED
END
GO
--TRIGGER 2: TỰ ĐỘNG TẠO MÃ ĐƠN THUỐC MỚI CHO BỆNH NHÂN MỚI
CREATE TRIGGER TRG_IN_DONTHUOC ON KHAMCHIDINH AFTER INSERT
AS
BEGIN
    INSERT INTO DONTHUOC (MAKHAM, MANHANVIEN, NGAYXUATDONTHUOC, GIADONTHUOC)
    SELECT
        INSERTED.MAKHAM, NULL, NULL, 000000.0000
    FROM INSERTED
END;
GO
--TRIGGER 3: TỰ ĐỘNG TẠO HÓA ĐƠN MỚI CHO BỆNH NHÂN MỚI
CREATE TRIGGER TRG_IN_HOADON ON BENHNHAN AFTER INSERT
AS
BEGIN
    INSERT INTO HOADON (MABENHNHAN, TENBN, HOBN, TENBENH, MAKHAM,
        MACHUABENH, MADONTHUOC, NGAYXUATHOADON, TONGTIEN)
    SELECT

```

	<pre> INSERTED.MABENHNHAN, TENBN, HOBN, NULL, NULL, NULL, NULL, 150000.0000 FROM INSERTED END; GO --TRIGGER 4: TỰ ĐỘNG CẬP NHẬT MÃ KHÁM CHỈ ĐỊNH CHO HÓA ĐƠN CỦA BỆNH NHÂN ĐÓ CREATE TRIGGER TRG_IN_HOADON_KHAMCHIDINH ON KHAMCHIDINH AFTER INSERT, UPDATE AS BEGIN     UPDATE HOADON     SET MAKHAM = INSERTED.MAKHAM     FROM INSERTED JOIN HOADON ON HOADON.MABENHNHAN = INSERTED.MABENHNHAN     WHERE HOADON.MABENHNHAN = INSERTED.MABENHNHAN END; GO </pre>
KQ	
CODE	<pre> -- TRIGGER 5: TỰ ĐỘNG CẬP NHẬT HÓA ĐƠN KHI CÓ THAY ĐỔI THÔNG TIN Ở BỆNH NHÂN CREATE TRIGGER UP_HOADON_BENHNHAN ON BENHNHAN AFTER UPDATE AS BEGIN     UPDATE HOADON     SET         HOADON.MABENHNHAN = BENHNHAN.MABENHNHAN,         HOADON.HOBN = BENHNHAN.HOBN,         HOADON.TENBN = BENHNHAN.TENBN     FROM HOADON JOIN BENHNHAN     ON HOADON.MABENHNHAN = BENHNHAN.MABENHNHAN END </pre>
KQ	
CODE	<pre> -- TRIGGER 6: TỰ ĐỘNG CẬP NHẬT TÊN BỆNH VÀ TỔNG TIỀN CHO HÓA ĐƠN CỦA BỆNH NHÂN ĐÓ CREATE TRIGGER TRG_IN_TENBENH_HOADON ON KHAMCHIDINH AFTER UPDATE AS BEGIN     UPDATE HOADON     SET         HOADON.TENBENH = B.TENBENH, HOADON.TONGTIEN = I.GIAKHAM     FROM HOADON         JOIN inserted I ON I.MABENHNHAN = HOADON.MABENHNHAN         JOIN KHAMCHITIET KCT ON KCT.MAKHAM = I.MAKHAM         JOIN BENH B ON B.MABENH = KCT.MABENH     WHERE HOADON.MABENHNHAN = I.MABENHNHAN END GO </pre>

KQ	 <pre> UPDATE KHAMCHIDINH SET MANHANVIEN = '10000005', GIAKHAM = '550000.0000' WHERE MAKHAM = 10000011 </pre> <p>133 %</p> <p>Messages</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>Completion time: 2024-05-30T15:14:08.8803501+07:00</p>
CODE	<pre> --TRIGGER 7: TỰ ĐỘNG CẬP NHẬT MÃ CHỮA BỆNH VÀO HÓA ĐƠN CHO BỆNH NHÂN ĐÓ CREATE TRIGGER TRG_IN_MACHUA_HOADON ON CHUABENH AFTER INSERT, UPDATE AS BEGIN     UPDATE HOADON     SET         HOADON.MACHUABENH = I.MACHUABENH     FROM HOADON JOIN inserted I     ON I.MAKHAM = HOADON.MAKHAM END  --TRIGGER 8: TỰ ĐỘNG CẬP NHẬT TỔNG GIÁ HÓA ĐƠN KHI Y TÁ NHẬP/CẬP NHẬT GIÁ VÀO BẢNG CHỮA BỆNH CREATE TRIGGER TRG_CHUABENH_HOADON ON CHUABENH FOR UPDATE, INSERT AS BEGIN     UPDATE HOADON     SET         HOADON.TONGTIEN = HOADON.TONGTIEN + I.GIACHUA     FROM HOADON JOIN inserted I     ON HOADON.MAKHAM = I.MAKHAM END </pre>
KQ	 <pre> INSERT CHUABENH (MAKHAM, NGAYCHUA, MAPHUONGTHUCCHUA, GIACHUA) VALUES (CAST(10000011 AS Numeric(8, 0)), CAST('2024-05-31' AS SmallDateTime), 1, 1500000.0000) </pre> <p>133 %</p> <p>Messages</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>Completion time: 2024-05-30T15:18:23.2398106+07:00</p>
CODE	<pre> --TRIGGER 9: TỰ ĐỘNG CẬP NHẬT MÃ ĐƠN THUỐC VÀO HÓA ĐƠN CHO BỆNH NHÂN ĐÓ CREATE TRIGGER TRG_IN_MADONTHUOC_HOADON ON DONTHUOC AFTER UPDATE AS BEGIN     UPDATE HOADON     SET         HOADON.MADONTHUOC = I.MADONTHUOC     FROM HOADON JOIN inserted I     ON I.MAKHAM = HOADON.MAKHAM END  --TRIGGER 10: TỰ ĐỘNG CẬP NHẬT TỔNG GIÁ HÓA ĐƠN KHI BÁC SĨ CẬP NHẬT GIÁ VÀO BẢNG ĐƠN THUỐC CREATE TRIGGER TRG_DONTHUOC_HOADON ON DONTHUOC FOR UPDATE AS BEGIN     UPDATE HOADON     SET         HOADON.TONGTIEN = HOADON.TONGTIEN + I.GIADONTHUOC     FROM HOADON JOIN inserted I     ON HOADON.MAKHAM = I.MAKHAM END </pre>

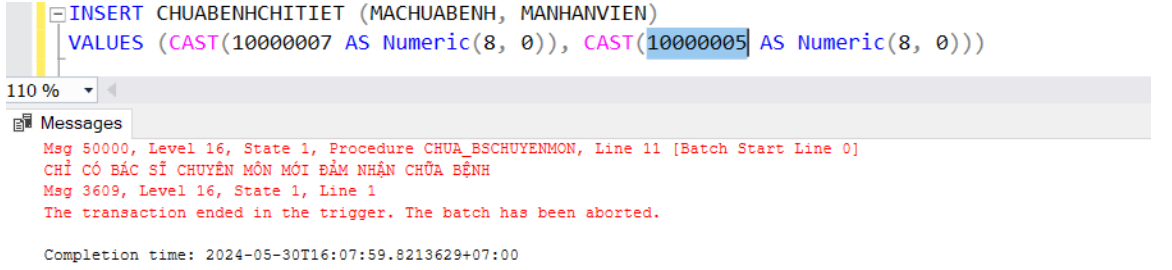
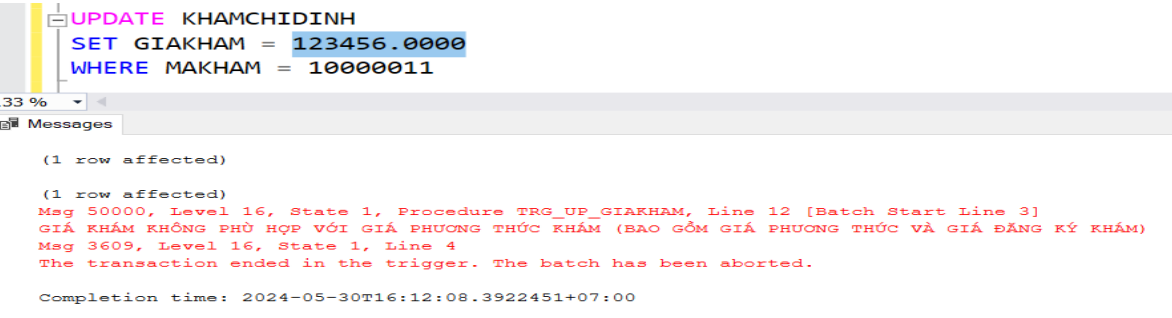


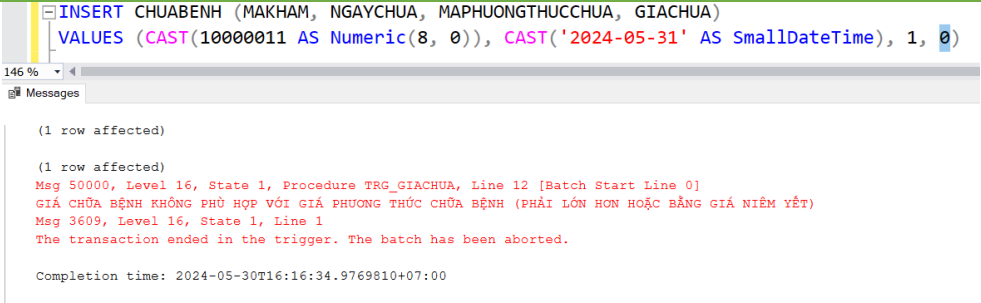
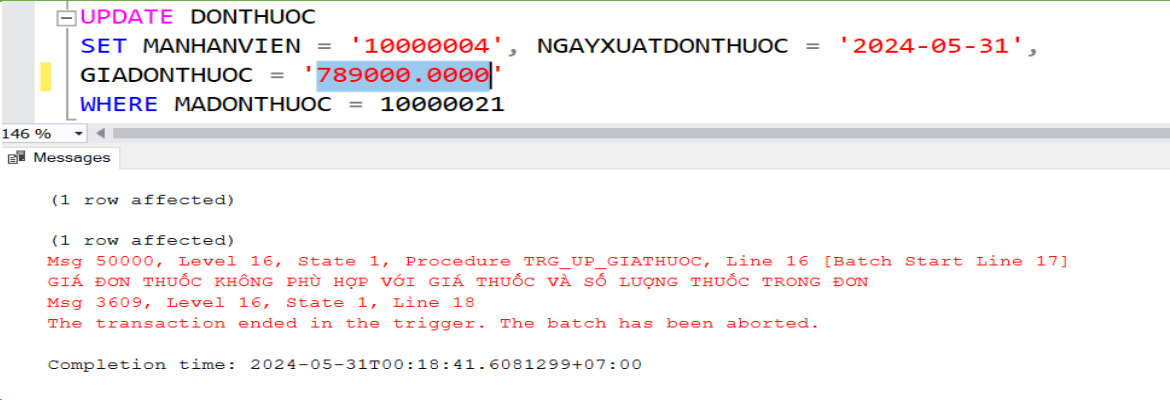
KQ	
----	--

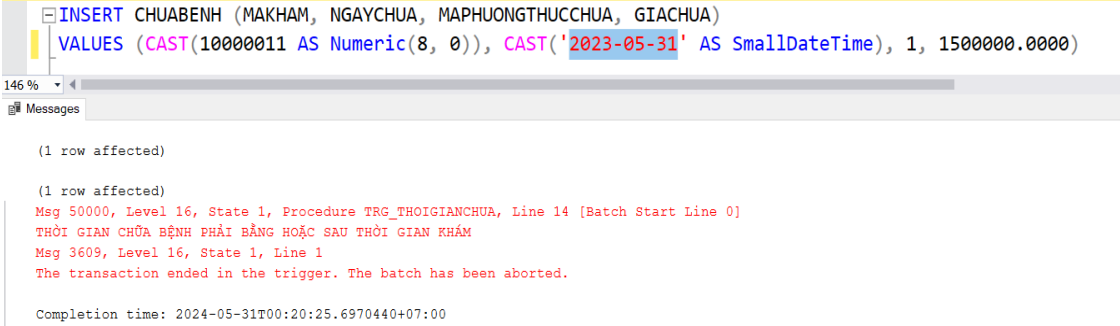
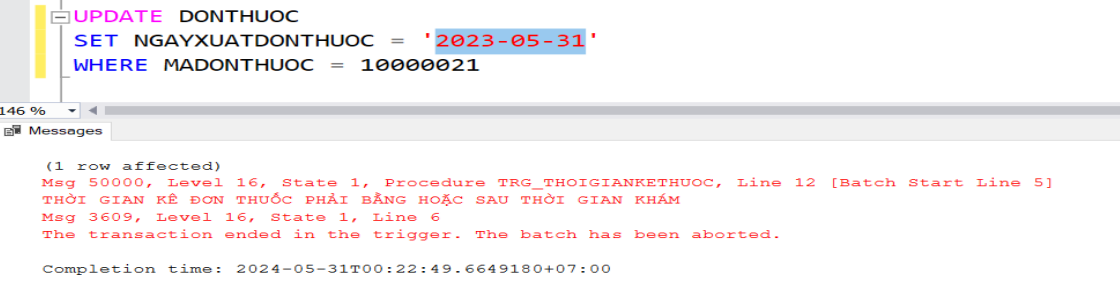
Bảng 7. Trigger tự động

3.2. Trigger kiểm tra dữ liệu

CODE	<pre>--TRIGGER 11: KIỂM TRA CHỈ CÓ BÁC SĨ CHỈ ĐỊNH MỚI ĐƯỢC PHÂN CÔNG Ở BẢNG KHÁM CREATE TRIGGER KHAM_BSCHIDINH ON KHAMCHIDINH AFTER UPDATE AS BEGIN     IF EXISTS (         SELECT *         FROM inserted I         WHERE I.MANHANVIEN &lt;&gt; 10000005     )     BEGIN         RAISERROR('CHỈ CÓ BÁC SĨ CHỈ ĐỊNH MỚI ĐẢM NHẬN VAI TRÒ KHÁM CHỈ ĐỊNH', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO</pre>
KQ	
CODE	<pre>--TRIGGER 12: KIỂM TRA CHỈ CÓ BÁC SĨ MỚI ĐƯỢC PHÂN CÔNG Ở CHỮA BỆNH CREATE TRIGGER CHUA_BSCHUYENMON ON CHUABENHCHITIET FOR INSERT, UPDATE AS BEGIN     IF EXISTS (         SELECT *         FROM inserted I         WHERE I.MANHANVIEN = 10000005         OR I.MANHANVIEN = 10000006     )     BEGIN         RAISERROR('CHỈ CÓ BÁC SĨ CHUYÊN MÔN MỚI ĐẢM NHẬN CHỮA BỆNH', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO</pre>

KQ	 <pre> INSERT CHUABENHCHITIET (MACHUABENH, MANHANVIEN) VALUES (CAST(10000007 AS Numeric(8, 0)), CAST(10000005 AS Numeric(8, 0))) </pre> <p>110 %</p> <p>Messages</p> <p>Msg 50000, Level 16, State 1, Procedure CHUA_BSCHUYENMON, Line 11 [Batch Start Line 0]    CHỈ CÓ BÁC SĨ CHUYÊN MÔN MỚI ĐẢM NHẬN CHỮA BỆNH    Msg 3609, Level 16, State 1, Line 1    The transaction ended in the trigger. The batch has been aborted.</p> <p>Completion time: 2024-05-30T16:07:59.8213629+07:00</p>
CODE	<pre> --TRIGGER 13: KIỂM TRA GIÁ KHÁM Y TÁ CẬP NHẬT PHẢI BẰNG GIÁ PHƯƠNG THỨC + GIÁ ĐĂNG KÝ KHÁM (CHỨC NĂNG CỦA Y TÁ) CREATE TRIGGER TRG_UP_GIAKHAM ON KHAMCHIDINH AFTER UPDATE AS BEGIN     IF NOT EXISTS ( SELECT * FROM inserted I                     JOIN KHAMCHITIET KCT ON I.MAKHAM = KCT.MAKHAM                     JOIN KHAMPHUONGTHUC KPT ON KCT.MAPHUONGTHUCKHAM = KPT.MAPHUONGTHUCKHAM                     WHERE I.GIAKHAM = (KPT.GIAKHAM + 150000.000)                   )     BEGIN         RAISERROR('GIÁ KHÁM KHÔNG PHÙ HỢP VỚI GIÁ PHƯƠNG THỨC KHÁM (BAO GỒM GIÁ PHƯƠNG         THỨC VÀ GIÁ ĐĂNG KÝ KHÁM)', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO </pre>
KQ	 <pre> UPDATE KHAMCHIDINH SET GIAKHAM = 123456.0000 WHERE MAKHAM = 10000011 </pre> <p>133 %</p> <p>Messages</p> <p>(1 row affected)</p> <p>(1 row affected)</p> <p>Msg 50000, Level 16, State 1, Procedure TRG_UP_GIAKHAM, Line 12 [Batch Start Line 3]    GIÁ KHÁM KHÔNG PHÙ HỢP VỚI GIÁ PHƯƠNG THỨC KHÁM (BAO GỒM GIÁ PHƯƠNG THỨC VÀ GIÁ ĐĂNG KÝ KHÁM)    Msg 3609, Level 16, State 1, Line 4    The transaction ended in the trigger. The batch has been aborted.</p> <p>Completion time: 2024-05-30T16:12:08.3922451+07:00</p>
CODE	<pre> --TRIGGER 14: KIỂM TRA GIÁ CHỮA BỆNH PHÙ HỢP VỚI GIÁ PHƯƠNG THỨC CHỮA BỆNH (CHỨC NĂNG CỦA Y TÁ VÀ BÁC SĨ) CREATE TRIGGER TRG_GIACHUA ON CHUABENH AFTER INSERT, UPDATE AS BEGIN     IF NOT EXISTS ( SELECT * FROM inserted I                     JOIN CHUABENHHPHUONGTHUC CBPT ON I.MAPHUONGTHUCCHUA                     = CBPT.MAPHUONGTHUCCHUA                     WHERE I.GIACHUA &gt;= CBPT.GIACHUA                   )     BEGIN         RAISERROR('GIÁ CHỮA BỆNH KHÔNG PHÙ HỢP VỚI GIÁ PHƯƠNG THỨC CHỮA BỆNH (PHẢI LỚN         HƠN HOẶC BẰNG GIÁ NIÊM YẾT)', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO </pre>

KQ	
CODE	<pre> --TRIGGER 15: KIỂM TRA GIÁ ĐƠN THUỐC PHÙ HỢP VỚI GIÁ THUỐC NHÂN SỐ LƯỢNG THUỐC CREATE TRIGGER TRG_UP_GIATHUOC ON DONTHUOC AFTER UPDATE AS BEGIN     IF EXISTS (         SELECT *         FROM inserted I         JOIN DONTHUOCCHITIET DTCT ON I.MADONTHUOC = DTCT.MADONTHUOC         JOIN THUOC T ON DTCT.MATHUOC = T.MATHUOC         GROUP BY I.MAKHAM, I.MADONTHUOC, I.GIADONTHUOC         HAVING I.GIADONTHUOC &lt;&gt; SUM (T.GIATHUOC * DTCT.SOLUONG)     )     BEGIN         RAISERROR('GIÁ ĐƠN THUỐC KHÔNG PHÙ HỢP VỚI GIÁ THUỐC VÀ SỐ LƯỢNG THUỐC TRONG ĐƠN', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO </pre>
KQ	
CODE	<pre> --TRIGGER 16: KIỂM TRA THỜI GIAN CHỮA BỆNH PHẢI SAU THỜI GIAN KHÁM CREATE TRIGGER TRG_THOIGIANCHUA ON CHUABENH AFTER INSERT, UPDATE AS BEGIN     IF EXISTS (         SELECT *         FROM inserted I         JOIN KHAMCHIDINH K ON K.MAKHAM = I.MAKHAM         WHERE I.NGAYCHUA &lt;= K.NGAYKHAM     )     BEGIN         RAISERROR('THỜI GIAN CHỮA BỆNH PHẢI BẰNG HOẶC SAU THỜI GIAN KHÁM', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END </pre>

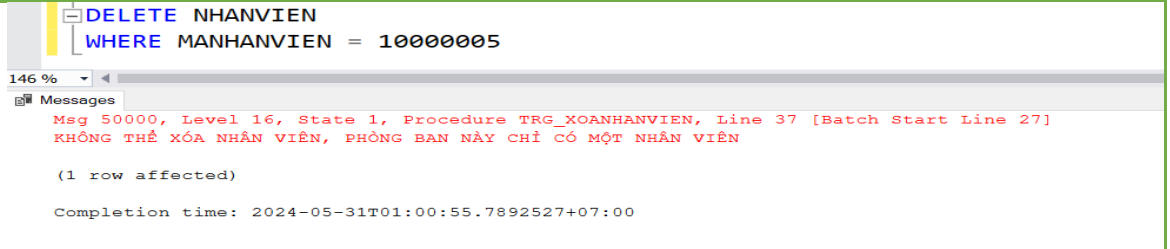
KQ	
CODE	<pre>--TRIGGER 17: KIỂM TRA THỜI GIAN KÊ THUỐC PHẢI SAU THỜI GIAN KHÁM CREATE TRIGGER TRG_THOIGIANKETHUOC ON DONTHUOC AFTER UPDATE AS BEGIN     IF EXISTS (         SELECT *         FROM inserted I         JOIN KHAMCHIDINH K ON K.MAKHAM = I.MAKHAM         WHERE I.NGAYXUATDONTHUOC &lt;= K.NGAYKHAM     )     BEGIN         RAISERROR('THỜI GIAN KÊ ĐƠN THUỐC PHẢI BẰNG HOẶC SAU THỜI GIAN KHÁM', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END</pre>
KQ	
CODE	<pre>--TRIGGER 18: KIỂM TRA BÁC SĨ THỰC HIỆN CHỮA BỆNH CÓ CHUYÊN MÔN PHÙ HỢP VỚI LOẠI BỆNH HAY KHÔNG CREATE TRIGGER TRG_CHUABENH_CHUYENMON ON CHUABENHCHITiet AFTER INSERT, UPDATE AS BEGIN     DECLARE @MANHANVIEN [numeric](8, 0),             @MACHUYENMON [numeric](3, 0)     SELECT @MANHANVIEN = MANHANVIEN FROM INSERTED     SELECT @MACHUYENMON = MACHUYENMON FROM NHANVIEN WHERE MANHANVIEN = @MANHANVIEN     IF NOT EXISTS (         SELECT *         FROM INSERTED I         JOIN CHUABENH NV ON NV.MACHUABENH = I.MACHUABENH         JOIN KHAMCHIDINH KCD ON KCD.MAKHAM = NV.MAKHAM         JOIN KHAMCHITiet KCT ON KCT.MAKHAM = KCD.MAKHAM         JOIN BENH B ON B.MABENH = KCT.MABENH         WHERE I.MANHANVIEN = @MANHANVIEN         AND B.PHANLOAI = @MACHUYENMON     )     BEGIN         RAISERROR('KHÔNG THỂ CHỈ ĐỊNH BÁC SĨ CÓ CHUYÊN MÔN KHÔNG PHÙ HỢP', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END</pre>

KQ	<pre> INSERT CHUABENHCHITIET (MACHUABENH, MANHANVIEN) VALUES (CAST(10000007 AS Numeric(8, 0)), CAST(10000002 AS Numeric(8, 0))) </pre> <p>161 %</p> <p>Messages</p> <p>Msg 50000, Level 16, State 1, Procedure TRG_CHUABENH_CHUYENMON, Line 19 [Batch Start Line 23] KHÔNG THỂ CHỈ ĐỊNH BÁC SĨ CÓ CHUYÊN MÔN KHÔNG PHÙ HỢP Msg 3609, Level 16, State 1, Line 24 The transaction ended in the trigger. The batch has been aborted.</p> <p>Completion time: 2024-05-31T00:53:46.0194418+07:00</p>
CODE	<pre> --TRIGGER 19: KIỂM TRA BÁC SĨ THỰC HIỆN KÊ THUỐC CÓ CHUYÊN MÔN PHÙ HỢP VỚI LOẠI BỆNH HAY KHÔNG CREATE TRIGGER TRG_DONTHUOC_CHUYENMON ON DONTHUOC AFTER UPDATE AS BEGIN     DECLARE @MANHANVIEN [numeric](8, 0),             @MACHUYENMON [numeric](3, 0)     SELECT @MANHANVIEN = MANHANVIEN FROM INSERTED     SELECT @MACHUYENMON = MACHUYENMON FROM NHANVIEN WHERE MANHANVIEN = @MANHANVIEN     IF NOT EXISTS (SELECT * FROM inserted I                   JOIN KHAMCHIDIINH KCD ON KCD.MAKHAM = I.MAKHAM                   JOIN KHAMCHITIET KCT ON KCT.MAKHAM = KCD.MAKHAM                   JOIN BENH B ON B.MABENH = KCT.MABENH                   WHERE I.MANHANVIEN = @MANHANVIEN                   AND B.PHANLOAI = @MACHUYENMON                  )     BEGIN         RAISERROR('KHÔNG THỂ CHỈ ĐỊNH BÁC SĨ CÓ CHUYÊN MÔN KHÔNG PHÙ HỢP', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END GO </pre>
KQ	<pre> UPDATE DONTHUOC SET MANHANVIEN = '10000002', NGÀYXUATDONTHUOC = '2024-05-31',     GIADONTHUOC = '789000.0000' WHERE MADONTHUOC = 10000021 </pre> <p>121 %</p> <p>Messages</p> <p>(1 row affected) Msg 50000, Level 16, State 1, Procedure TRG_DONTHUOC_CHUYENMON, Line 17 [Batch Start Line 17] KHÔNG THỂ CHỈ ĐỊNH BÁC SĨ CÓ CHUYÊN MÔN KHÔNG PHÙ HỢP Msg 3609, Level 16, State 1, Line 18 The transaction ended in the trigger. The batch has been aborted.</p> <p>Completion time: 2024-05-31T00:58:01.3007216+07:00</p>

Bảng 8. Trigger kiểm tra dữ liệu

### 3.3. Trigger kiểm tra và xóa

CODE	<pre> -- TRIGGER 20: KIỂM TRA VÀ XÓA NHÂN VIÊN CREATE TRIGGER TRG_XOANHANVIEN ON NHANVIEN INSTEAD OF DELETE AS BEGIN     DECLARE @MANHANVIEN numeric(8, 0);     DECLARE @MAPHONGBAN numeric(3, 0);      SELECT @MANHANVIEN = DELETED.MANHANVIEN,            @MAPHONGBAN = DELETED.MAPHONGBAN     FROM DELETED;      -- NẾU PHÒNG BAN ĐÓ CÓ NHIỀU HƠN MỘT NHÂN VIÊN     IF EXISTS (SELECT COUNT(*)               FROM NHANVIEN NV, PHONGBAN PB               WHERE PB.MAPHONGBAN = @MAPHONGBAN                     AND NV.MAPHONGBAN = PB.MAPHONGBAN                     AND NV.MANHANVIEN &lt;&gt; @MANHANVIEN               GROUP BY NV.MAPHONGBAN              )     BEGIN         RAISERROR('KHÔNG THỂ XÓA NHÂN VIÊN TRONG PHÒNG BAN CÓ NHIỀU HƠN MỘT NHÂN VIÊN', 16, 1);         ROLLBACK TRANSACTION;         RETURN;     END END </pre>
------	---

	<pre> HAVING COUNT(*) &gt; 0)  BEGIN     ALTER TABLE NHANVIEN NOCHECK CONSTRAINT FK_NHANVIEN_PHONGBAN,FK_NHANVIEN_CHUYENMON     ALTER TABLE CHUBENHCHITIET NOCHECK CONSTRAINT FK_CHUABENHCHITIET_NHANVIEN     ALTER TABLE KHAMCHIDINH NOCHECK CONSTRAINT FK_KHAMCHIDINH_NHANVIEN     ALTER TABLE DONTHUOC NOCHECK CONSTRAINT FK_DONTHUOC_NHANVIEN      DELETE FROM NHANVIEN     WHERE MANHANVIEN = @MANHANVIEN;      ALTER TABLE NHANVIEN CHECK CONSTRAINT FK_NHANVIEN_PHONGBAN, FK_NHANVIEN_CHUYENMON     ALTER TABLE CHUBENHCHITIET CHECK CONSTRAINT FK_CHUABENHCHITIET_NHANVIEN     ALTER TABLE KHAMCHIDINH CHECK CONSTRAINT FK_KHAMCHIDINH_NHANVIEN     ALTER TABLE DONTHUOC CHECK CONSTRAINT FK_DONTHUOC_NHANVIEN  END ELSE BEGIN     -- NẾU PHÒNG BAN CHỈ CÓ MỘT NHÂN VIÊN     RAISERROR ('KHÔNG THỂ XÓA NHÂN VIÊN, PHÒNG BAN NÀY CHỈ CÓ MỘT NHÂN VIÊN', 16, 1) END END GO </pre>
KQ	
CODE	<pre> -- TRIGGER 21: KIỂM TRA VÀ XÓA THÔNG TIN BỆNH NHÂN CREATE TRIGGER TRG_XOABENHNHAN ON BENHNHAN INSTEAD OF DELETE AS BEGIN     SET NOCOUNT ON;     -- KIỂM TRA THÔNG TIN BỆNH NHÂN TỪ NGÀY HOÀN THÁNH KHÁM (NGÀY XUẤT HÓA ĐƠN) TRONG     VÒNG 365 NGÀY     IF EXISTS (         SELECT *         FROM DELETED D         JOIN HOADON HD ON D.MABENHNHAN = HD.MABENHNHAN         WHERE DATEDIFF(DAY, HD.NGAYXUATHOADON, GETDATE()) &lt; 365     )     BEGIN         -- NẾU CÓ THÔNG TIN TRONG 365 NGÀY THÌ KHÔNG CHO PHÉP XÓA         RAISERROR ('KHÔNG THỂ XÓA BỆNH NHÂN. THÔNG TIN LƯU TRỮ CHƯA VƯỢT THỜI HẠN 365 NGÀY', 16, 1);     END     ELSE     BEGIN         -- NẾU KHÔNG CÓ THÔNG TIN TRONG 365 NGÀY THÌ ĐƯỢC PHÉP XÓA         ALTER TABLE KHAMCHIDINH NOCHECK CONSTRAINT FK_KHAMCHIDINH_BENHNHAN         ALTER TABLE HOADON NOCHECK CONSTRAINT FK_HOADON_BENHNHAN          DELETE FROM BENHNHAN         WHERE MABENHNHAN IN (SELECT MABENHNHAN FROM DELETED)          ALTER TABLE KHAMCHIDINH CHECK CONSTRAINT FK_KHAMCHIDINH_BENHNHAN         ALTER TABLE HOADON CHECK CONSTRAINT FK_HOADON_BENHNHAN      END END GO </pre>

KQ	<pre> DELETE BENHNHAN WHERE MABENHNHAN = 10000001 </pre>
	<p>146 %</p> <p>Messages</p> <p>Msg 50000, Level 16, State 1, Procedure TRG_XOABENHNHAN, Line 16 [Batch Start Line 27] KHÔNG THỂ XÓA BỆNH NHÂN. THÔNG TIN LƯU TRỮ CHƯA VƯỢT THỜI HẠN 365 NGÀY</p> <p>(1 row affected)</p> <p>Completion time: 2024-05-31T01:02:22.1089827+07:00</p>

Bảng 9. Trigger kiểm tra và xóa

## VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Chức năng có thể phát triển thêm

- Chức năng đặt lịch khám và hẹn tái khám, thông báo nhắc nhở lịch hẹn
  - Đặt lịch trực tuyến: cho phép bệnh nhân tự đặt và quản lý lịch hẹn qua website hoặc ứng dụng của phòng khám.
  - Thông báo nhắc nhở: gửi thông báo nhắc lịch hẹn qua sms hoặc email để giảm thiểu tình trạng bệnh nhân không đến khám.
- Chức năng xếp lịch làm việc, tích hợp tính lương và quản lý doanh thu
- Chia sẻ dữ liệu y tế: cho phép chia sẻ và cập nhật dữ liệu y tế giữa các cơ sở phòng khám, đảm bảo thông tin bệnh nhân luôn được cập nhật và đầy đủ.

### 2. Hướng phát triển về quy mô

- Mở rộng chuỗi các cơ sở phòng khám
  - Hệ thống quản lý chuỗi phòng khám: phát triển hệ thống quản lý tích hợp cho nhiều chi nhánh phòng khám, cho phép quản lý tập trung và đồng bộ hóa dữ liệu.
- Telemedicine (dịch vụ y tế từ xa)
  - Khám bệnh trực tuyến: mở rộng dịch vụ khám bệnh trực tuyến, cho phép bệnh nhân và bác sĩ tương tác qua video call, giảm thiểu nhu cầu đến trực tiếp phòng khám.
  - Theo dõi sức khỏe từ xa: cung cấp các thiết bị và ứng dụng để theo dõi sức khỏe bệnh nhân từ xa, đặc biệt là đối với các bệnh nhân mãn tính.
- Mở rộng dịch vụ chuyên sâu
  - Khám chữa bệnh chuyên sâu: mở rộng phạm vi khám chữa bệnh của phòng khám bằng cách thêm các dịch vụ y tế chuyên sâu như nha khoa, vật lý trị liệu, tâm lý học, v.v.
  - Đào tạo và phát triển: đào tạo và phát triển nhân lực chuyên môn cao để đáp ứng nhu cầu khám chữa bệnh chuyên sâu.

### 3. Kết luận

- Với cơ sở dữ liệu hiện tại, việc phát triển các chức năng và hệ thống quản lý mới sẽ giúp nâng cao hiệu quả hoạt động và chất lượng dịch vụ của phòng khám. Bằng cách mở rộng quy mô và tích hợp các công nghệ hiện đại như Telemedicine, phòng khám không chỉ đáp ứng tốt hơn nhu cầu của bệnh nhân mà còn tối ưu hóa quy trình quản lý và chăm sóc sức khỏe. Hướng đi này sẽ đảm bảo sự phát triển bền vững và nâng cao uy tín của phòng khám trong ngành y tế.

#### **Danh mục tài liệu tham khảo**

- \* Đỗ Thị Mai Hương. (n.d.). *Các hệ thống thông tin*. CNTT- Học viện kỹ thuật quân sự.
- \* IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998.
- \* IEEE Guide for Developing System Requirements Specifications," in IEEE Std 1233-1996, vol., no., pp.1-30, 22 Dec. 1996.
- \* Custom Software Requirements Specification Document Example (International Standard) - Software Development Company.
- \* IT4490:Software Design and Construction – Nguyễn Thị Thu Trang,  
trangntt@soict.hust.edu.vn.