

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TIẾN ĐỘ ĐỒ ÁN
Cơ Sở Dữ Liệu Nâng Cao
21HTTT1

Đồ án

Quản lý phòng khám nha khoa

Giáo viên hướng dẫn

Hồ Thị Hoàng Vy
Nguyễn Trần Minh Thư



Ipsu

Thành viên

Lý Nhật Hào
Huỳnh Sơn Hà
Lê Văn Dương
Nguyễn Văn Tuấn Kiệt
Hoàng Trần Thông

NIÊN KHOÁ 2022 - 2023

LỜI CẢM ƠN

Để hoàn thành được bài báo cáo này, chúng em đã nhận được sự giúp đỡ rất nhiều từ phía thầy cô giảng viên, trợ giảng và bạn bè. Nay “**NHÓM 04**” xin gửi lời cảm ơn chân thành và sâu sắc nhất đến giảng viên môn Cơ sở dữ liệu nâng cao lớp 21HTTT1, Khoa Công nghệ thông tin :

- Giảng viên **Hồ Thị Hoàng Vy**
- Giảng viên **Nguyễn Trần Minh Thư**

Các Cô đã đồng hành cùng nhóm, đã luôn quan tâm, hướng dẫn và truyền đạt, cung cấp kiến thức, tài liệu và các thủ thuật cần thiết để nhóm có thể hoàn thành bài báo cáo!

Trong quá trình thực hiện đồ án không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được nhiều ý kiến đóng góp từ các Cô giáo và bạn bè để đồ án ngày càng hoàn thiện hơn!

Xin chân thành cảm ơn



THÀNH VIÊN NHÓM BÁO CÁO

Huỳnh Sơn Hà	Lý Nhật Hào	Nguyễn Văn Tuấn Kiệt	Lê Văn Dương	Hoàng Trần Thông
--------------	-------------	----------------------	--------------	------------------



MỤC LỤC

LỜI CẢM ƠN	2
MỤC LỤC	3
CHƯƠNG I: Giới Thiệu	3
1.1. Đôi nét về Thông tin thành viên:	3
1.2. Tổng quát yêu cầu đề án:	4
CHƯƠNG II: Ý TƯỞNG ĐỒ ÁN VÀ PHÂN CÔNG CÔNG VIỆC CỦA CÁC THÀNH VIÊN	4
2.1. Đôi nét về chủ đề của đề án:	4
2.2. Phân Công Công Việc Và Kế Hoạch Phát Triển:	5
2.3. KẾ HOẠCH PHÁT TRIỂN ĐỒ ÁN CHI TIẾT	5
CHƯƠNG III: CHI TIẾT MÔ TẢ QUÁ TRÌNH	7
3.1. Các loại người dùng của hệ thống:	7
3.2. Quy trình chi tiết:	8
CHƯƠNG IV: DANH SÁCH CHỨC NĂNG VÀ TẦN SUẤT GIAO DỊCH	9
4.1. Chức năng dùng chung:	9
4.2. Chức năng của quản trị viên:	10
4.3. Chức năng của nhân viên:	10
4.4. Chức năng của nha sĩ:	11
CHƯƠNG V: RÀNG BUỘC	11
5.1. Ràng Buộc Toàn Vẹn	11
5.2. BUSINESS RULE:	14
CHƯƠNG VI: LƯỢC ĐỒ ER VÀ RDB	15
6.1. Lược đồ ER	15
6.2. Lược đồ RDB	16
CHƯƠNG VII: CHI TIẾT THIẾT KẾ CƠ SỞ DỮ LIỆU MỨC VẬT LÍ	17
7.1. Lựa chọn kiểu dữ liệu phù hợp khi cài đặt vật lí:	17
7.2. Các phương án cải thiện hiệu suất được áp dụng cho cơ sở dữ liệu:	21
7.3. Cài đặt dữ liệu lớn cho cơ sở dữ liệu	24
CHƯƠNG VIII: Các kịch bản sử dụng ứng dụng	27
8.1 Kịch bản 1: Nhân viên tiếp nhận yêu cầu đặt Lịch hẹn từ Bệnh nhân	27
8.2 Kịch bản 2: Nhân viên hoặc Quản trị viên tra cứu Thông tin chi tiết thanh toán cho Bệnh nhân	27
8.3 Kịch bản 3: Nhân viên hoặc Quản trị viên tạo Kế hoạch điều trị cho Bệnh nhân	28
8.4: Kịch bản 4: Quản trị viên tìm kiếm thuốc.	29
8.5 Kịch bản 5: Người dùng đăng nhập vào hệ thống.	30
CHƯƠNG IX: Cài đặt và hiệu quả của các phương pháp gia tăng hiệu suất truy vấn	30
9.1 Index trên bảng TaiKhoan (TenTaiKhoan, HoTen, SDT)	31
9.2 Index trên bảng Thuoc (TenThuoc)	36
9.3 Index trên bảng Lịch hẹn (ID_NhaSi)	39
9.4 Index trên bảng Thông tin chi tiết thanh toán (ID_HoSo)	42
9.5 Index trên bảng Kế hoạch điều trị (ID_ThanhToan, ID_HoSo)	43
9.6 Tổng kết Index	45

CHƯƠNG I: Giới Thiệu

1.1. Đôi nét về Thông tin thành viên:

Nhóm thực hiện đồ án có mã số nhóm là **NHÓM 04** và bao gồm 5 thành viên:

- **Huỳnh Sơn Hà**
 - MSSV: 21127035.
 - Email: hsha21@clc.fitus.edu.vn
- **Lý Nhật Hào**
 - MSSV: 2112041.
 - Email: lnhao21@clc.fitus.edu.vn
- **Nguyễn Văn Tuấn Kiệt**
 - MSSV: 21127331.
 - Email: nvtkiet21@clc.fitus.edu.vn
- **Lê Văn Dương**
 - MSSV: 21127500.
 - Email: lvduong21@clc.fitus.edu.vn
- **Hoàng Trần Thông**
 - MSSV: 21127695.
 - Email: htthong21@clc.fitus.edu.vn

1.2. Tổng quát yêu cầu đồ án:

Đồ án cuối kỳ phải đáp ứng ít nhất các yêu cầu sau:

- Giai đoạn 1: Mô tả chi tiết quy trình, dữ liệu, ràng buộc liên quan, danh sách chức năng cần xây dựng kèm tần suất giao dịch tương ứng. Thiết kế dữ liệu mức quan niệm và logic.

- Giai đoạn 2: Bảng thiết kế cơ sở dữ liệu mức vật lý theo đề xuất của nhóm, các đề xuất về cải thiện hiệu quả truy vấn dựa trên thiết kế đề xuất, source code của project mô phỏng các chức năng, source script tạo csdl, query, index, partition....

CHƯƠNG II: Ý TƯỞNG ĐỒ ÁN VÀ PHÂN CÔNG CÔNG VIỆC CỦA CÁC THÀNH VIÊN

2.1. Đôi nét về chủ đề của đồ án:

Hiện nay, nhu cầu làm đẹp, chăm sóc sức khỏe răng miệng của người dân ngày càng gia tăng. Đồng thời, cùng với sự phát triển công nghệ mạnh mẽ những năm gần đây, nhiều phòng khám nha khoa đã bắt đầu chuyển sang sử dụng các phần mềm quản lý để thay thế phương pháp thủ công. Nhờ đó, việc quản lý phòng khám nha khoa trở nên hiệu quả, dễ dàng, mang lại trải nghiệm tốt, chuyên nghiệp và tiện lợi hơn cho cả các y bác sĩ lẫn khách hàng đến khám, chữa bệnh. Cụ thể, việc quản lý hồ sơ bệnh nhân, quản lý cuộc hẹn, quản lý nhân viên, thống kê, tất cả đều được thực hiện trên máy tính. Hỗ trợ bệnh nhân đặt lịch hẹn với phòng khám một cách nhanh chóng, dễ dàng, có thể trao đổi trực tiếp với bác sĩ thông qua chức năng nhắn tin trên ứng dụng, xem hồ sơ khám chữa bệnh của bản thân. Hỗ trợ giải đáp thắc mắc của khách hàng về thông tin phòng khám, giới thiệu phòng khám...

2.2. Phân Công Công Việc Và Kế Hoạch Phát Triển:

HUỖNH SƠN HÀ (20%) :

- Đảm nhận vai trò mô tả chi tiết quy trình, dữ liệu ràng buộc liên quan, thiết kế dữ liệu mức quan niệm và logic.
- Cài đặt giao tác và chức năng liên quan.

LÝ NHẬT HÀO (20%) :

- Đảm nhận vai trò đề xuất bảng thiết kế dữ liệu mức vật lý.
- Tìm hiểu và cài đặt partition, giao tác.

NGUYỄN VĂN TUẤN KIỆT (20%) :

- Đảm nhận vai trò tham gia thiết kế mức logic.
- Cài đặt ứng dụng.
- Cài đặt truy vấn và viết giao tác.

LÊ VĂN DƯƠNG (20%):

- Cài đặt phương pháp tăng hiệu suất truy vấn như partition, index.

- Tạo dữ liệu lớn để test ứng dụng.
- Viết giao tác

HOÀNG TRẦN THÔNG (20%):

- Cài đặt cơ sở dữ liệu mức vật lý.
- Viết các câu truy vấn, cài đặt giao tác chính, tìm hiểu index
- Viết báo cáo.

2.3. KẾ HOẠCH PHÁT TRIỂN ĐỒ ÁN CHI TIẾT

NO.	Công việc cần làm	Thành viên chịu trách nhiệm	Thời hạn
	Tìm hiểu chi tiết quy trình nghiệp vụ, ràng buộc, danh sách chức năng	Cả nhóm	28/10/2023 - 7/11/2023
	Thiết kế cơ sở dữ liệu mức quan niệm.	Huỳnh Sơn Hà, Lý Nhật Hào	08/11/2023 - 14/11/2023
	Thiết kế cơ sở dữ liệu mức logic	Lê Văn Dương, Hoàng Trần Thông, Nguyễn Văn Tuấn Kiệt	15/11/2023 - 20/11/2023
	Thiết kế cơ sở dữ liệu mức vật lí	Huỳnh Sơn Hà, Hoàng Trần Thông	21/11/2023 - 30/11/2023
	Cài đặt cơ sở dữ liệu mức vật lí	Lê Văn Dương, Hoàng Trần Thông	01/12/2023 - 04/12/2023

	Cài đặt giao tác, các câu truy vấn	Cả nhóm	05/12/2023 - 10/12/2023
	Tìm hiểu, cài đặt index, partition	Cả nhóm	11/12/2023 - 15/12/2023
	Thiết kế cài đặt ứng dụng	Nguyễn Văn Tuấn Kiệt	01/12/2023 - 10/12/2023
	Kết nối cơ sở dữ liệu với ứng dụng, testing	Cả nhóm	11/12/2023 - 18/12/2023
	Viết báo cáo	Cả nhóm	19/12/2023 - 23/12/2023

CHƯƠNG III: CHI TIẾT MÔ TẢ QUÁ TRÌNH

3.1. Các loại người dùng của hệ thống:

- **Quản Trị Viên:** là người dùng có quyền hạn cao nhất của ứng dụng, có thể sử dụng tất cả chức năng mà ứng dụng cung cấp. Quản trị viên có thể thay đổi nhân sự, lịch trình, thủ tục,... Mỗi một Quản Trị Viên sẽ có một **ID_QTV** khác nhau và duy nhất.
- **Nhân Viên:** có chức năng sắp xếp lịch hẹn giữa bệnh nhân và nha sĩ, theo dõi các yêu cầu của bệnh nhân. Mỗi một nhân viên sẽ có một **ID_NV** duy nhất. Nhân Viên **được quản lý bởi Quản Trị Viên**.
- **Nha Sĩ:** có chức năng chỉnh sửa thông tin bệnh án, sơ đồ nha chu, tình trạng răng hàm, hồ sơ điều trị,... của Bệnh Nhân. Mỗi Nha Sĩ sẽ có nhiều **Lịch Bác Sĩ** bao gồm các thông tin: **ID_NhaSi** và Thời Gian Trông. Mỗi nha sĩ có một **ID_NhaSi** duy nhất. Nha Sĩ **được quản lý bởi Quản Trị Viên**.
- **Bệnh Nhân:** có chức năng đăng ký lịch hẹn với Nha Sĩ để khám chữa răng. Mỗi Bệnh Nhân có một **ID_BenhNhan** duy nhất.

3.2. Quy trình chi tiết:

- **B1:** Bệnh Nhân liên hệ và đặt lịch hẹn với nhân viên phòng khám / tự đặt trên website.
- **B2:** Phòng khám xác nhận lịch hẹn qua điện thoại/ email/tin nhắn văn bản. Khi lịch hẹn được xác nhận, Nhân Viên phụ trách sẽ tạo ra một **Lịch Hẹn**. Lịch Hẹn có một **ID_LichHen** duy nhất, có **ID_NhaSi** của Nha sĩ phụ trách và Nha sĩ trợ khám, **ID_BenhNhan** của Bệnh nhân khám chữa bệnh, thời gian hẹn và tình trạng của Bệnh Nhân. Bệnh Nhân đến điều trị ở **Phòng Khám** theo lịch hẹn. Phòng Khám bao gồm một mã **ID_PhongKham** duy nhất và thông tin địa chỉ, số điện thoại và email của phòng khám.
- **B3:** Khám theo lịch hẹn. Bệnh nhân gặp bác sĩ/ chuyên gia theo thời gian lịch hẹn sau đó kiểm tra sức khỏe răng miệng. Bệnh Nhân sẽ có một **Hồ Sơ Bệnh Nhân** bao gồm một **ID_HoSo** duy nhất và thông tin tổng quan sức khỏe và ghi chú chống chỉ định thuốc của bệnh nhân. Dựa trên kết quả kiểm tra, nha sĩ có thể đưa ra lời khuyên hoặc chỉ định điều trị. Mỗi một Hồ Sơ Bệnh Nhân sẽ có nhiều **Kế hoạch Điều Trị** dựa vào chỉ định của bác sĩ.
- **B4:** Bác sĩ trình bày phương án điều trị (thời gian, mục tiêu, chi phí dự kiến). Mỗi một **Kế Hoạch Điều Trị** sẽ có một **ID_KHDieuTri** duy nhất và các thông tin như mô tả kế hoạch, ngày điều trị, ghi chú, trạng thái, phí điều trị. Kế Hoạch Điều Trị sẽ có thông tin của Nha sĩ phụ

trách và Nha sĩ phụ khám. Cùng với đó là thông tin của **Răng** và **Mặt Răng** mà Bệnh Nhân khám chữa. **Răng** sẽ có một **ID_Rang** duy nhất và tên răng cùng với phí khám. **Mặt Răng** có **ID_MatRang** và tên của mặt răng đó.

- **B5**: Thanh toán dịch vụ khám hoặc điều trị. Đặt lịch hẹn cho lần khám tiếp theo nếu cần thiết. Mỗi một **Hồ Sơ Bệnh Nhân** sẽ có nhiều **Thông Tin Chi Tiết Thanh Toán**. Một **Thông Tin Chi Tiết Thanh Toán** sẽ có một **ID_ThanhToan** duy nhất cùng với các thông tin ngày giao dịch, người thanh toán, tổng tiền, tiền đã trả, tiền thối, loại thanh toán, ghi chú. **Thông tin chi tiết thanh toán** sẽ có nhiều **Kế hoạch điều trị**. Kế hoạch điều trị sẽ có các **Đơn Thuốc** đi kèm. **Đơn Thuốc** sẽ có số lượng thuốc cùng với thông tin của **Thuốc**. Thông tin **Thuốc** gồm **ID_Thuoc** duy nhất và tên thuốc, đơn vị tính, chỉ định, số lượng tồn kho, ngày hết hạn. đơn giá. **Thuốc** được quản lý bởi **Quản Trị Viên**.

CHƯƠNG IV: DANH SÁCH CHỨC NĂNG VÀ TẦN SUẤT GIAO DỊCH

4.1. Chức năng dùng chung:

ID	Chức năng	Mô tả	Tần suất (lần/ngày)
GN1	Đăng nhập	Đăng nhập vào hệ thống	1000/ngày
GN2	Xem thông tin tài khoản	Thông tin email, sđt, dc,	500/ngày
GN3	Xem danh sách nha sĩ	Xem danh sách các nha sĩ và lịch làm việc tương ứng	100/ngày
GN4	Theo dõi hồ sơ bệnh nhân	Tra cứu thông tin hồ sơ khám bệnh	300/ngày

4.2. Chức năng của quản trị viên:

ID	Chức năng	Mô tả	Tần suất (lần/ngày)
AD1	Đăng ký tài khoản	Quản trị viên tạo tài khoản cho nha sĩ, nhân viên	20/ngày
AD2	Quản lý hồ sơ bệnh nhân	Tra cứu, thêm, xóa, sửa hồ sơ bệnh nhân	300/ngày
AD3	Quản lý tài khoản	Xem, thêm, xóa, sửa thông tin tài khoản của nhân viên và nha sĩ	50/ngày
AD4	Quản lý thuốc	Xem, thêm, xóa, sửa thông tin các loại thuốc (chỉ định, giá, ...)	50/ngày
AD5	Thống kê	Tạo báo cáo về lịch hẹn, điều trị	50/ngày

		của từng nha sĩ	
AD6	Quản lý cuộc hẹn	Xem, thêm, xóa, sửa lịch hẹn của bệnh nhân, nha sĩ	300/ngày
AD7	Quản lý lịch làm việc	Thêm, xóa, sửa, cập nhật lịch làm việc cho nha sĩ	100/ngày

4.3. Chức năng của nhân viên:

ID	Chức năng	Mô tả	Tần suất (lần/ngày)
ST1	Xem thông tin tài khoản	Thông tin email, sđt, dc,	200/ngày
ST2	Quản lý cuộc hẹn	Xem, thêm, xóa, sửa lịch hẹn của bệnh nhân, nha sĩ	300/ngày
ST3	Tiếp nhận yêu cầu	Tiếp nhận yêu cầu hẹn lịch của bệnh nhân	300/ngày

4.4. Chức năng của nha sĩ:

DT1	Quản lý hồ sơ bệnh nhân	Xem hồ sơ của bệnh nhân, danh sách thanh toán,	100/ngày
DT2	Cập nhật thông tin sức khỏe răng miệng	Cập nhật tình trạng sức khỏe răng miệng vào hồ sơ bệnh nhân.	300/ngày
DT3	Quản lý kế hoạch điều trị	Xem, thêm, cập nhật kế hoạch điều trị của bệnh nhân	300/ngày

CHƯƠNG V: RÀNG BUỘC

5.1. Ràng Buộc Toàn Vẹn

- Ràng buộc về Giới tính trong Tài Khoản phải là một trong 2 giá trị: Nam, Nữ.

Bối cảnh: Tài Khoản

Bảng tầm ảnh hưởng:

	T	X	S
Tài Khoản	+	-	+(Giới Tính)

- Ràng buộc về Trạng Thái Kế Hoạch DT phải là một trong 3 giá trị: màu xanh dương, màu xanh lá, màu vàng.

Bối cảnh: Kế Hoạch DT

Bảng tầm ảnh hưởng:

	T	X	S
Kế hoạch DT	+	-	+(TRẠNG THÁI)

- Ràng buộc về ID Mặt Răng phải là một trong 6 giá trị: L, F, D, M, T, R.

Bối cảnh: Mặt Răng

Bảng tầm ảnh hưởng:

	T	X	S
Mặt Răng	+	-	+(ID Mặt Răng)

- Ràng buộc về Loại thanh toán trong Thông tin chi tiết thanh toán phải là một trong 4 giá trị: Tiền mặt, quẹt thẻ, momo, chuyển khoản.

Bối cảnh: Thông tin chi tiết thanh toán

Bảng tầm ảnh hưởng:

	T	X	S
Thông tin chi tiết thanh toán	+	-	+(Loại thanh toán)

- Ràng buộc về Thời gian hẹn khi đặt lịch hẹn: Thời gian khám không thể nằm trong quá khứ:

Bối cảnh: **Lịch Hẹn**

Bảng tầm ảnh hưởng:

	T	X	S
Lịch Hẹn	+	-	+(Thoigianhen)

- **Ràng buộc về Tình Trạng của lịch hẹn:** Tình trạng là một trong 2 giá trị: Cuộc hẹn mới hoặc tái khám.

Bối cảnh: **Lịch Hẹn**

Bảng tầm ảnh hưởng:

	T	X	S
Lịch Hẹn	+	-	+(Tình trạng)

- **Ràng buộc về ngày Thanh Toán của Thông Tin Thanh Toán:**

Ngày Thanh Toán của khách hàng phải nằm trong tương lai.

Bối cảnh: **Thông Tin Thanh Toán**

Bảng tầm ảnh hưởng:

	T	X	S
Thông Tin Thanh Toán	+	-	+(Ngaythanhtoan)

+ **Không thể đặt lịch hẹn cho một nha sĩ không có thời gian rảnh:**

Bối cảnh: **Lịch Hẹn, Lịch Bác Sĩ**

Bảng tầm ảnh hưởng:

	T	X	S
Lịch Bác Sĩ	+	-	+(Thoigiantrong)
Lịch Hẹn	+	-	+(Thoigianhen)

- **Ràng buộc về Ngày DT trong Kế Hoạch DT phải nằm trong tương lai**

Bối cảnh: **Kế Hoạch DT**

Bảng tầm ảnh hưởng:

	T	X	S
Kế Hoạch DT	+	-	+(NgàyDT)

- **Ràng buộc về ngày hết hạn của thuốc:**

Ngày hết hạn của thuốc không thể nằm trong quá khứ.

Bối cảnh: **Thuốc**

Bảng tầm ảnh hưởng:

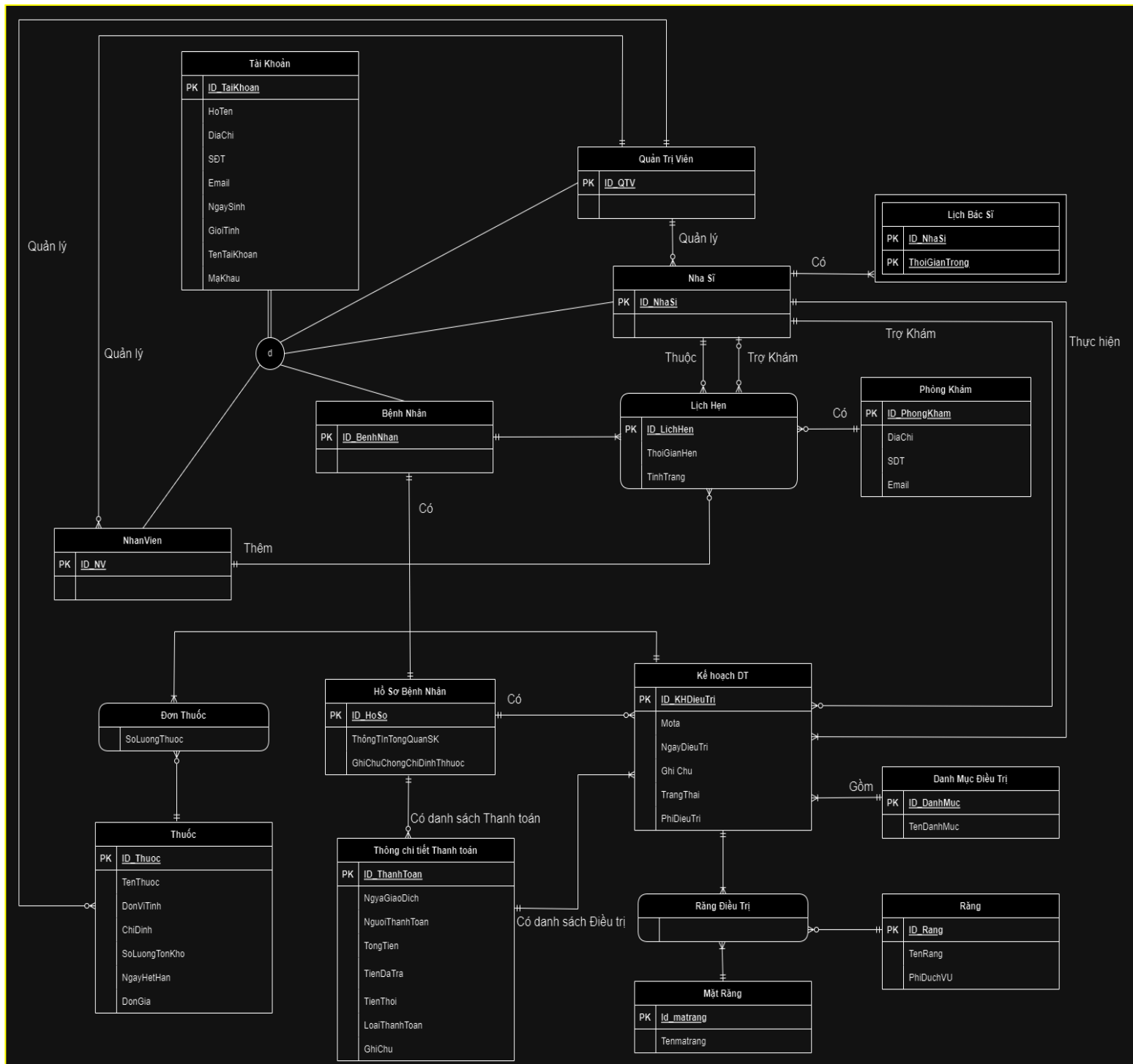
	T	X	S
Thuốc	+	-	+(NgàyHetHan)

5.2. BUSINESS RULE:

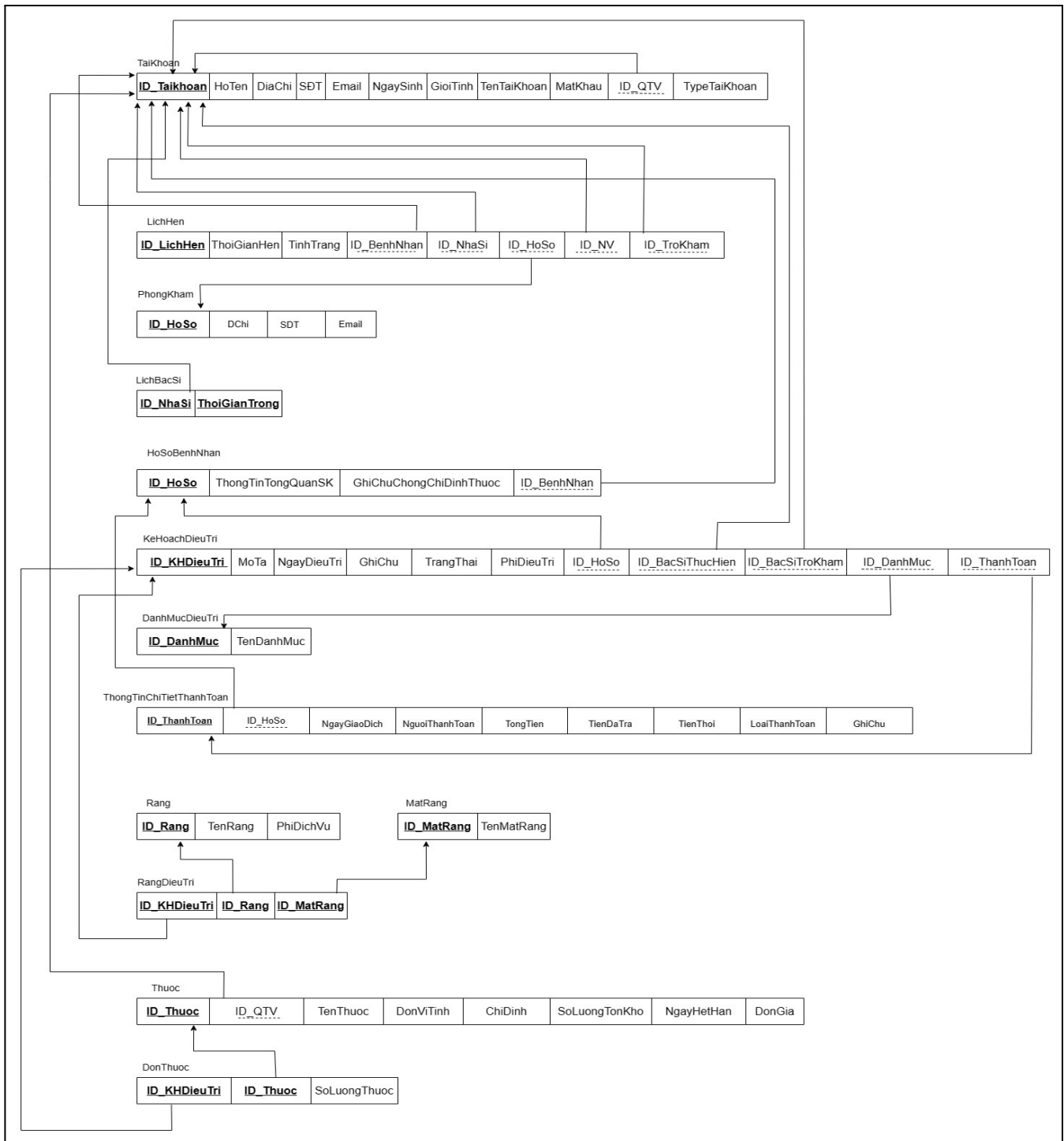
- Mỗi Quản trị viên có thể quản lý nhiều nhân viên hoặc không quản lý ai cả và mỗi nhân viên chỉ có thể được quản lý bởi 1 quản trị viên duy nhất.
- Mỗi quản trị viên có thể không quản lý hoặc quản lý nhiều thuốc và mỗi thuốc chỉ có thể được quản lý bởi 1 quản trị viên
- Quản trị viên có thể quản lý nhiều nha sĩ hoặc không quản lý ai cả và mỗi nha sĩ chỉ có thể bị quản lý bởi một quản trị viên duy nhất.
- Mỗi Nha Sĩ chỉ có một hoặc nhiều Lịch Bác Sĩ .
- Mỗi Nha Sĩ có thể không có hoặc có nhiều Lịch hẹn.
- Mỗi Lịch Hẹn thuộc một phòng khám và một bệnh nhân duy nhất.
- Mỗi Danh Mục DT gồm nhiều Kế Hoạch DT, mỗi Kế Hoạch DT điều trị nhiều răng và 1 mặt răng tương ứng với răng đó.
- Mỗi Kế Hoạch DT có nhiều Đơn Thuốc.

CHƯƠNG VI: LƯỢC ĐỒ ER VÀ RDB

6.1. Lược đồ ER



6.2. Lược đồ RDB



CHƯƠNG VII: CHI TIẾT THIẾT KẾ CƠ SỞ DỮ LIỆU

MỨC VẬT LÍ

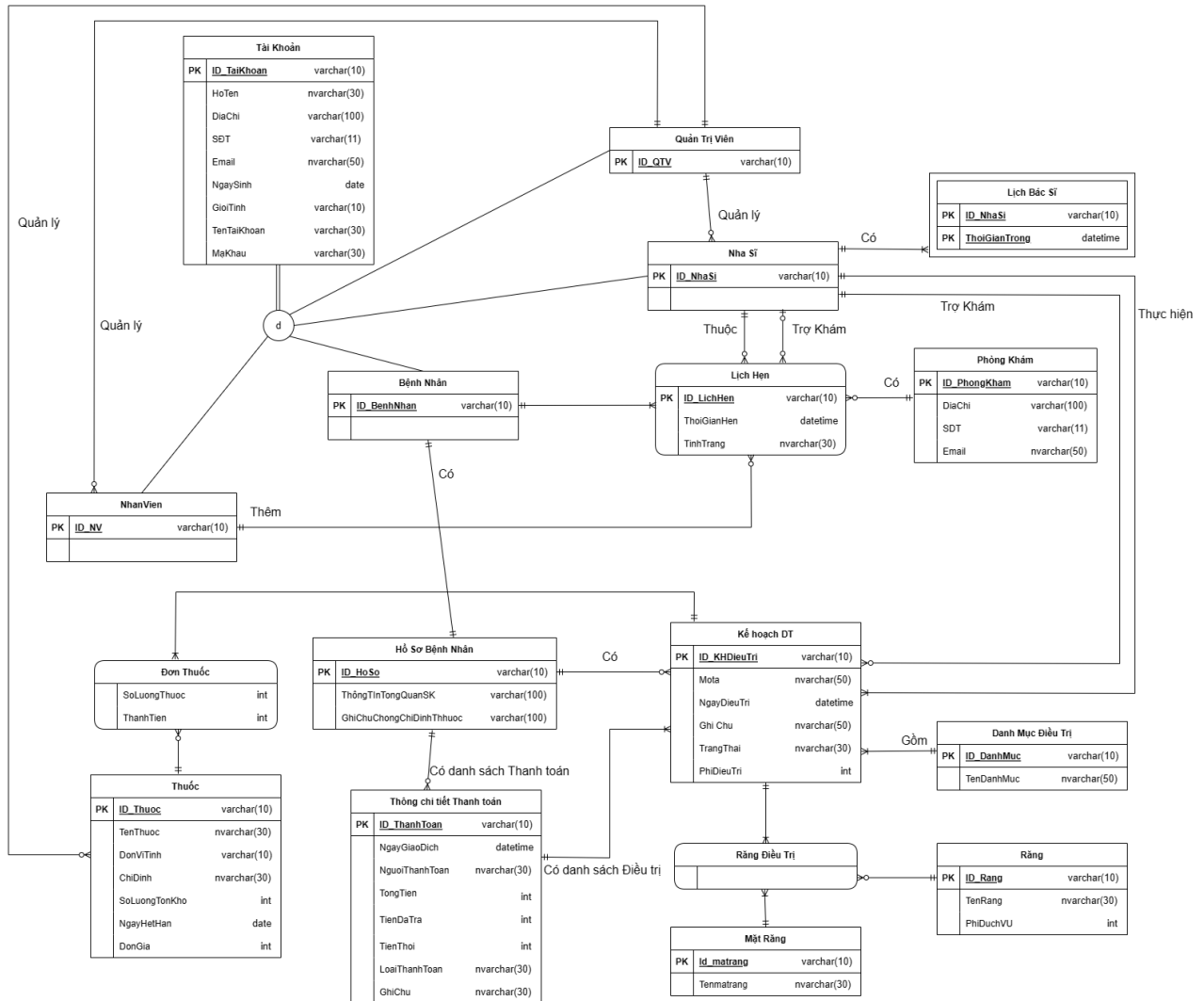
7.1. Lựa chọn kiểu dữ liệu phù hợp khi cài đặt vật lý:

- Table **TaiKhoan**(ID_TaiKhoan, HoTen, DiaChi, SDT, Email, NgaySinh, GioiTinh, TenTaiKhoan, MatKhu, ID_QTV, TypeTaiKhoan)
 - Nhóm lựa chọn cài đặt tất cả tài khoản vào table TaiKhoan, với **TypeTaiKhoan** dùng để phân biệt các vai trò với nhau (cài đặt theo model giữ supertype với TypeTaiKhoan là thuộc tính nhận biết role).
 - TypeTaiKhoan **char(2)**: 2 ký tự biểu thị vai trò ‘QT’: quản trị, ‘NS’: nha sĩ, ‘NV’: nhân viên, ‘BN’: bệnh nhân. Cài đặt mặc định chỉ có đúng 2 ký tự nên lựa chọn kiểu dữ liệu char(2).
 - ID_TaiKhoan **varchar(10)**: định dạng 2 ký tự đầu là TypeTaiKhoan + mã số tài khoản. Ví dụ Bệnh nhân có ID ‘BN12345678’. Ước tính sẽ có khoảng chục triệu bệnh nhân đến với hệ thống phòng khám nên nhóm đặt ID là varchar(10).
 - DiaChi, Email, MatKhu, TenTaiKhoan: định dạng nvarchar.
 - SDT: varchar(11) vì số điện thoại tối đa chỉ có 11 số.
 - ID_QTV varchar(10): là khóa ngoại từ chính bảng này, giá trị là NULL nếu Tài Khoản chưa có Quản Trị Viên quản lý.
 - NgaySinh: kiểu dữ liệu datetime.
- Table **PhongKham**(ID_PhongKham, DChi, SDT, Email)
 - ID_PhongKham varchar(10): khóa chính biểu thị ID của phòng khám với định dạng ‘PK’ + số thứ tự phòng khám, ví dụ ‘PK00000001’
 - SDT char(11): lựa chọn kiểu varchar(11) vì tối đa SDT chỉ có 11 số
 - DiaChi của Phòng Khám dùng kiểu dữ liệu nvarchar vì địa chỉ có dấu
 - Email phòng khám dùng varchar(30) vì không có dấu và độ dài tùy biến (ước tính độ dài khoảng 30 ký tự).
- Table **LichHen**(ID_LichHen, ThoiGianHen, TinhTrang, ID_BenhNhan, ID_NhaSi, ID_PhongKham, ID_NV, ID_TroKham)
 - ID_LichHen varchar(10): khóa chính biểu thị ID của Lịch Hẹn với định dạng ‘LH’ + số thứ tự lịch hẹn, ví dụ ‘LH00112233’
 - ThoiGianHen: kiểu dữ liệu datetime thể hiện ngày giờ hẹn
 - Tình Trạng: kiểu dữ liệu nvarchar(30) thể hiện tình trạng cuộc hẹn, ước tính độ dài tối đa 30 ký tự.

- ID_BenhNhan, ID_NhaSi, ID_NV, ID_TroKham: là khóa ngoại từ bảng TaiKhoan là ID của Bệnh Nhân, Nha Sĩ, Nhân Viên phụ trách và Nha Sĩ trợ khám của lịch hẹn đó.
- ID_PhongKham là khóa ngoại từ bảng PhongKham: ID Phòng Khám
- Table **LichBacSi(ID_NhaSi, ThoiGianTrong)**
 - ID_NhaSi varchar(10): khóa ngoại từ bảng TaiKhoan
 - ThoiGianTrong datetime: thể hiện thời gian trông của Nha Sĩ
 - ID_NhaSi, ThoiGianTrong là **2** khóa chính của bảng thể hiện Thời gian trông của Nha sĩ (mỗi Nha sĩ có nhiều thời gian trông)
- Table **HoSoBenhNhan(ID_HoSo, ThôngTinTongQuanSK, GhiChuChongChiDinhThuoc, ID_BenhNhan)**
 - Mỗi 1 Bệnh nhân chỉ có duy nhất 1 Hồ sơ Bệnh Nhân: dùng để ghi dấu tất cả lịch sử đi thăm khám của Bệnh nhân (1 hồ sơ sẽ có nhiều kế hoạch điều trị và có nhiều dịch vụ sử dụng)
 - ID_HoSo varchar(10): khóa chính của bảng, thể hiện ID của Hồ sơ có dạng 'HS' + mã số Hồ sơ, ví dụ 'HS00001234'
 - ThôngTinTongQuanSK và GhiChuChongChiDinhThuoc: nvarchar(100): ước tính hai thông tin này là dạng thông tin tổng quan (khá dài, nhiều nội dung) nên nhóm quyết định để định dạng nvarchar(100)
 - ID_BenhNhan: khóa ngoại từ bảng TaiKhoan, là ID của Bệnh nhân trong Hồ sơ
- Table **ThôngTinChiTietThanhToan(ID_ThanhToan, ID_HoSo, NgayGiaoDich, NguoiThanhToan, TongTien, TienDaTra, TienThoi, LoaiThanhToan, GhiChu)**
 - ID_ThanhToan **varchar(10)**: là khóa chính của bảng, thể hiện ID của 1 hóa đơn thanh toán, vì hóa đơn phát sinh nhiều trong ngày nên chọn kiểu dữ liệu varchar(10), định dạng 'CT' và mã của thanh toán, ví dụ: 'CT00000001'
 - ID_HoSo: khóa ngoại đến bảng HoSoBenhNhan, là Hồ sơ của Bệnh nhân được thanh toán hóa đơn
 - NgayGiaoDich datetime: ngày thanh toán tiền trả hóa đơn
 - NguoiThanhToan nvarchar(30): họ tên của người thanh toán Hóa đơn, vì là tên nên độ dài vào khoảng 30 (tối đa) và có dấu
 - TongTien int: tổng tiền của hóa đơn: mặc định = 0, được cập nhật mỗi lần phát sinh Kế hoạch điều trị thuộc về hóa đơn này và được tính bằng Tổng tiền của tất cả Kế hoạch.

- TienDaTra int: được cập nhật khi Nhân viên đưa hóa đơn cho Người thanh toán
- TienThoi int: tiền thối được tính bằng Tiền đã trả người thanh toán đưa trừ đi Tổng tiền của Chi tiết thanh toán
- Table **DanhMucDieuTri**(ID_DanhMuc, TenDanhMuc)
 - ID_DanhMuc varchar, khóa chính của bảng biểu thị cho Danh Mục của Kế hoạch điều trị, định dạng 'DM' + mã số của Danh mục
 - TenDanhMuc nvarchar(30): ước tính tên Danh mục có thể dài nên nhóm đặt TenDanhMuc là nvarchar(30)
- Table **KeHoachDieuTri**(ID_KHDieuTri, MoTa, NgayDieuTri, GhiChu, TrangThai, PhiDieuTri, ID_HoSo, ID_BacSiThucHien, ID_BacSiTroKham, ID_DanhMuc, ID_ThanhToan)
 - ID_KHDieuTri **varchar(10)** là khóa chính của bảng, biểu thị cho mã của Kế hoạch này, định dạng 'KH' + mã số, ví dụ 'KH00000001'
 - MoTa nvarchar(50), GhiChu nvarchar(50): tổng quan và ghi chú của Kế hoạch, dạng thông tin nên có thể dài nên nhóm quyết định chọn kiểu dữ liệu nvarchar(50)
 - NgayDieuTri datetime: ngày điều trị của Kế hoạch
 - PhiDieuTri int: phí điều trị của Kế hoạch, khi tạo mặc định có giá trị là 0, sẽ được cập nhật mỗi khi có phát sinh Răng điều trị hoặc Đơn thuốc thuộc về kế hoạch này và được tính bằng tổng tất cả những phí đó.
 - ID_HoSo, ID_BacSiThucHien, ID_BacSiTroKham, ID_DanhMuc, ID_ThanhToan: là khóa ngoại đến các bảng với lần lượt là ID của Hồ sơ bệnh nhân điều trị, ID Bác sĩ thực hiện chính, ID bác sĩ trợ khám, ID của Danh mục Kế hoạch thuộc về và ID của Hóa đơn thanh toán mà Kế hoạch này thuộc về.
- Table **Rang**(ID_Rang, TenRang, PhiDichVu)
 - ID_Rang: varchar, khóa chính của bảng biểu thị Mã răng, định dạng 'RA' + Mã răng
 - TenRang nvarchar(30): tên của Răng
 - PhiDichVu int: giá tiền dịch vụ của khám chữa Răng
- Table **MatRang**(ID_MatRang, TenMatRang)
 - ID_MatRang varchar, khóa chính của bảng biểu thị Mã mặt răng, định dạng 'MR' + Mã số mặt răng
 - TenMatRang nvarchar(30): tên của Mặt răng
- Table **RangDieuTri**(ID_KHDieuTri, ID_Rang, ID_MatRang, PhiDichVu)

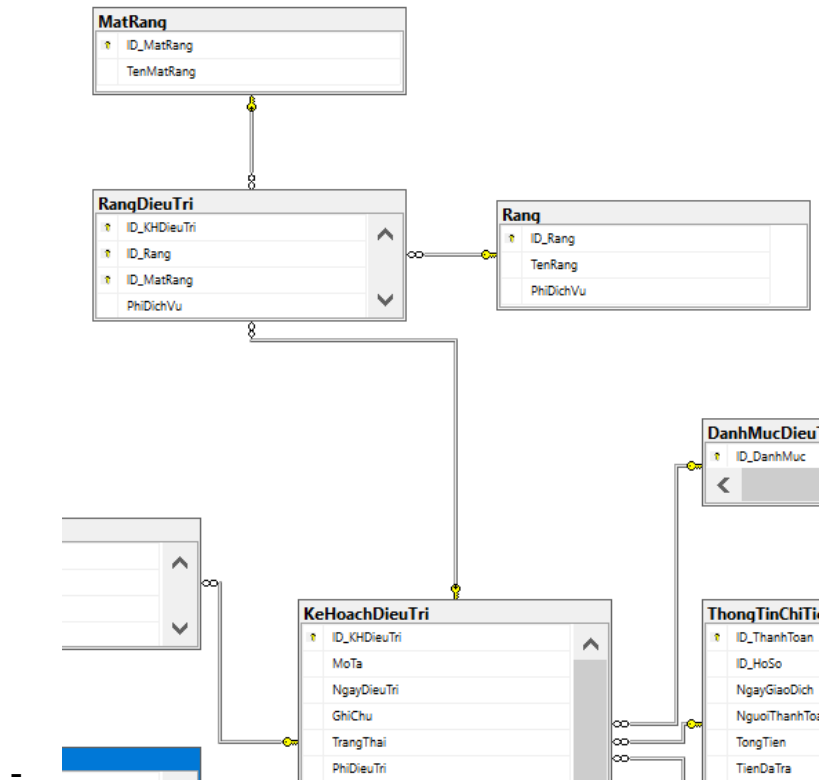
- Bảng có 3 khóa chính là ID_KHDieuTri, ID_Rang, ID_MatRang dùng để phân biệt các Răng cần điều trị trong 1 kế hoạch điều trị. Vì theo đề xuất thiết kế 1 Kế hoạch điều trị có thể điều trị được nhiều Răng
- ID_KHDieuTri, khóa ngoại tới ID của Kế hoạch điều trị của Bệnh nhân
- ID_Rang: ID của Răng khóa ngoại tới bảng Răng
- ID_MatRang: ID Mặt răng khóa ngoại tới bảng Mặt Răng
- PhiDichVu int, cũng chính là PhiDichVu của Răng điều trị
- Table **Thuoc**(ID_Thuoc, ID_QTV, TenThuoc, DonViTinh, ChiDinh, SoLuongTonKho, NgayHetHan, DonGia)
 - ID_Thuoc **varchar(10)**: khóa chính của bảng biểu thị Mã thuốc, vì có nhiều loại thuốc (có thể tăng hoặc giảm) nên lựa chọn kiểu dữ liệu varchar, định dạng ‘TH’ + mã thuốc: Ví dụ ‘TH00000001’
 - ID_QTV: khóa ngoại tới bảng TaiKhoan thể hiện Quản trị viên quản lý thuốc này
 - DonViTinh nvarchar(10): Là cách Thuốc được kê đơn, ví dụ Hộp, Vi, Viên,...
 - TenThuoc, ChiDinh: sử dụng kiểu dữ liệu nvarchar vì là thông tin của tùy loại nên có thể ngắn hoặc dài
 - SoLuongTonKho int: số lượng thuốc còn trong hệ thống
 - NgayHetHan: datetime ngày hết hạn của Thuốc
 - DonGia int: giá tiền của Thuốc
- Table **DonThuoc**(ID_KHDieuTri, ID_Thuoc, SoLuong, ThanhTien)
 - ID_KHDieuTri: khóa ngoại tới bảng Kế hoạch điều trị, biểu thị ID của Kế hoạch mà Đơn thuốc thuộc về
 - ID_Thuoc: khóa ngoại tới bảng Thuốc, biểu thị ID của Thuốc được kê đơn
 - SoLuong int: số lượng Thuốc
 - ThanhTien int: là giá trị được tính tự động bằng số lượng nhân với đơn giá của thuốc



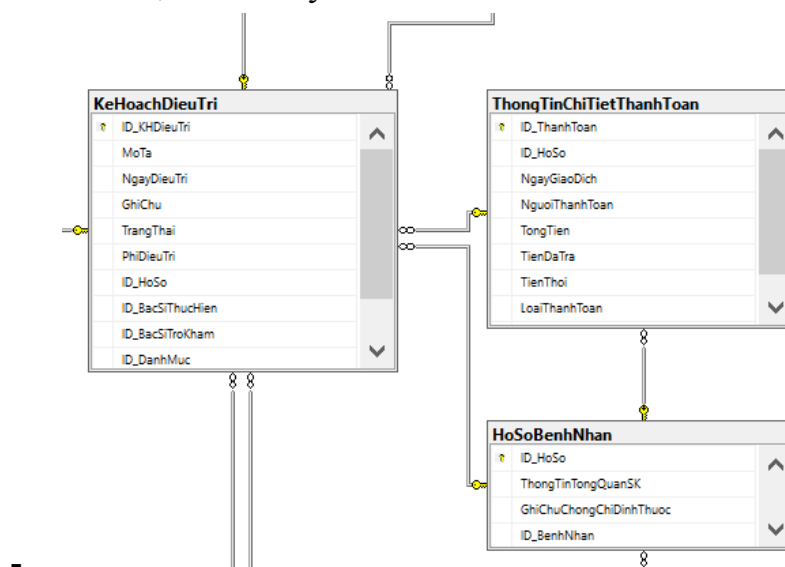
7.2. Các phương án cải thiện hiệu suất được áp dụng cho cơ sở dữ liệu:

1. Denormalization: Phi chuẩn hóa lược đồ:

- **Định nghĩa:** Là phương pháp giúp tăng hiệu suất truy vấn bằng cách giảm dạng chuẩn của một số bảng nhất định, cho phép việc trùng lặp thông tin trên nhiều bảng.
- **Lí do sử dụng:** Nhận thấy có 1 số câu truy vấn được thực hiện nhiều mà trong đó giá trị thường là kết ở nhiều bảng giống nhau mỗi lần truy vấn => cho phép bảng cần truy vấn tới chứa luôn thuộc tính ở bảng mà nó cần phải lấy giá trị từ.
- **Áp dụng** vào cơ sở dữ liệu của nhóm:



- Áp dụng vào lược đồ của nhóm, ở đây ta có bảng Rang có PhiDichVu biểu thị tiền khám của Răng và có bảng RangDieuTri cho biết Bệnh nhân khám Răng nào ở mặt răng nào, **giá điều trị** cũng là giá của Răng. Ta có Kế hoạch điều trị có giá trị tiền được cập nhật bằng cách cộng thêm **Phí dịch vụ** từ bảng Răng điều trị vào. Do đó nhóm thêm thuộc tính Phí Dịch Vụ cho bảng Răng điều trị nhằm truy vấn nhanh hơn cho thao tác UPDATE.

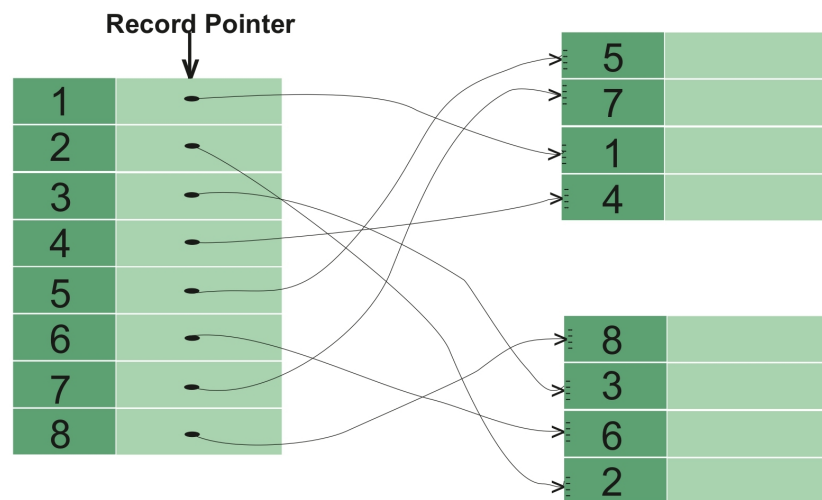


- Một áp dụng khác là ta có bảng Hồ sơ bệnh nhân có **ID_HoSo** là khóa chính. Bảng ThôngTinChiTietThanhToan có khóa ngoại ID_HoSo từ bảng này và bảng KếHoạchDieuTri lại lấy khóa

ngoại ID_ThanhToan từ bảng ThôngTinChiTietThanhToan. Tức có nghĩa là ta có thể suy ra ID_HoSo của Bệnh nhân có Kế hoạch điều trị bằng cách kết 3 bảng lại. Nhưng nhóm nhận thấy tần suất tra cứu mã Hồ sơ từ KeHoachDieuTri khá cao nên quyết định thêm khóa ID_HoSo vào luôn bảng KeHoachDieuTri.

2. Index (nonclustered index)

- **Định nghĩa:** Index (chỉ mục) là một phương pháp nhằm tăng hiệu suất câu truy vấn. Nonclustered index cũng giống như mục lục của một cuốn sách, nó là con trỏ trỏ tới cột của bảng được đánh chỉ mục.



EXAMPLE OF NON-CLUSTERED INDEX

- Non-clustered index sẽ gom nhóm cột được đánh index thành từng cụm. Khi tìm kiếm có điều kiện của cột này sẽ tìm kiếm theo cột này trước sau đó sẽ vào đó để tìm kiếm tiếp vào để lấy giá trị chính xác.
- **Nhược điểm:** tuy nhiên index cũng là con dao hai lưỡi, ta phải tốn thêm chi phí để lưu trữ index, nếu cột đó kiểu dữ liệu lớn hoặc có quá nhiều value lặp lại (chẳng hạn như cột giới tính) thì ta không nên cài index cho cột này. Chỉ nên cài index cho những bảng có giá trị vừa và nhỏ thường xuyên được truy vấn và ít giá trị NULL.
- **Áp dụng** vào cơ sở dữ liệu của nhóm: chi tiết ở chương IX

3. Partition

- **Định nghĩa:** Phân vùng, còn được gọi là phân mảnh, là một phương pháp giúp chia nhỏ một tập dữ liệu lớn thành các phân đoạn nhỏ hơn, nhằm cho phép máy tính lưu trữ và xử lý mỗi phân đoạn độc lập với

nhau. Kỹ thuật này giúp giảm tải lưu trữ trên đĩa cứng, tăng tốc độ đọc và ghi dữ liệu, và đặc biệt là tăng cường hiệu suất xử lý của các truy vấn.

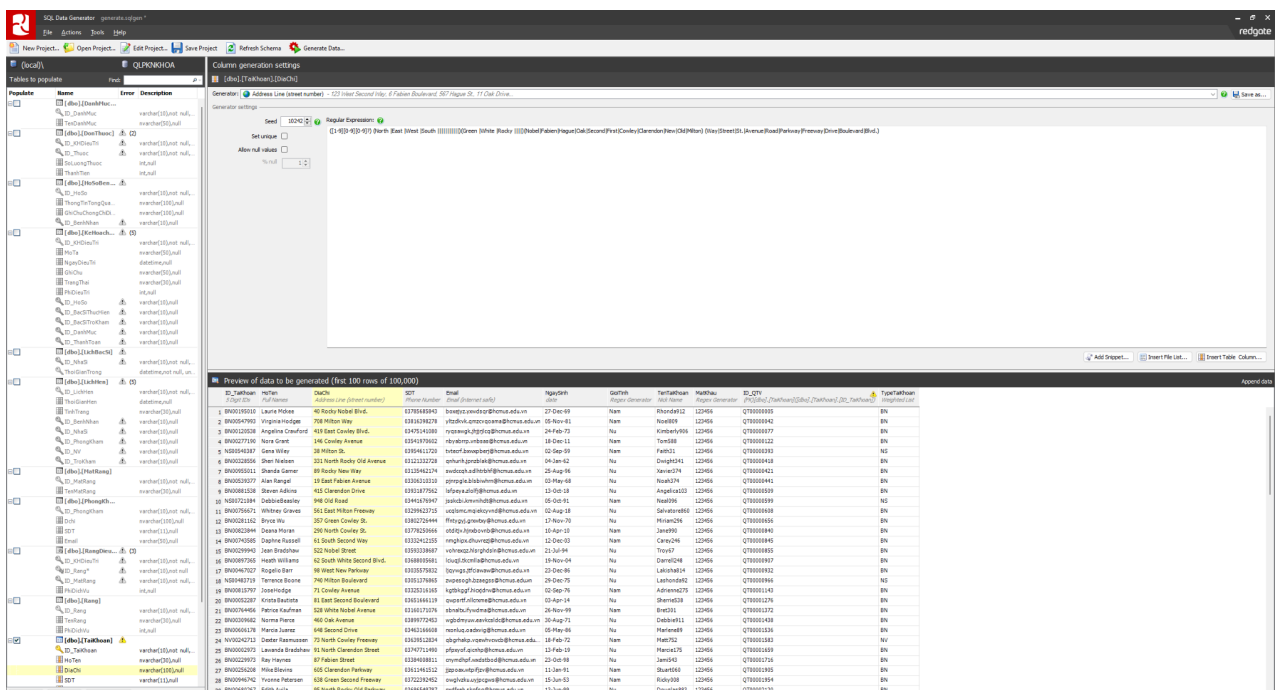
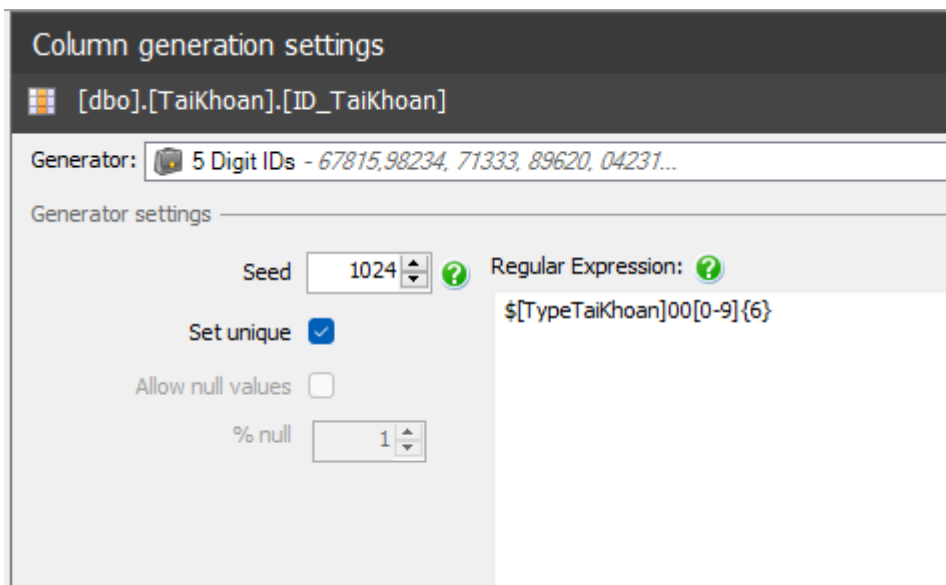
- Các loại partition:
 - **Phân vùng dọc** (Vertical Partitioning): Đây là quá trình phân chia dữ liệu bằng cách cắt dọc qua các cột trong tập dữ liệu. Mỗi phân vùng sẽ chứa các cột liên quan đến một khía cạnh cụ thể của dữ liệu. Quá trình này giúp tối ưu hóa việc truy cập vào các cột cần thiết cho các truy vấn và thao tác xử lý dữ liệu.
 - **Phân vùng ngang** (Horizontal Partitioning): Đây là quá trình phân chia dữ liệu bằng cách cắt theo hàng ngang trong tập dữ liệu, trong đó giữ nguyên các cột. Kỹ thuật này được ưa thích vì các phân vùng vẫn giữ nguyên số lượng cột và ý nghĩa của dữ liệu gốc. Ngoài ra, phân vùng ngang tương đối dễ triển khai.
 - **Nhược điểm**: Phân vùng chỉ có ý nghĩa khi các truy vấn chỉ liên quan đến một partition duy nhất. Nếu các truy vấn làm việc với dữ liệu trên nhiều partition khác nhau, hệ thống phải thực hiện quá trình ghép nối (merge) các partition lại với nhau để tạo thành tập dữ liệu gốc. Quá trình này có thể tốn nhiều chi phí hệ thống và dẫn đến việc các hàng dữ liệu được sắp xếp mà không có thứ tự. Ngoài ra, việc merge các partition để tạo thành tập dữ liệu gốc cũng có thể gây ra vấn đề về tính toàn vẹn dữ liệu. Các hàng dữ liệu có thể không còn được sắp xếp theo thứ tự ban đầu, và điều này có thể ảnh hưởng đến các truy vấn và thao tác trên tập dữ liệu ghép.
- **Cài đặt** partition ở cơ sở dữ liệu của nhóm: chi tiết ở chương IX

7.3 Cài đặt dữ liệu lớn cho cơ sở dữ liệu

Trong quá trình thực hiện generate dữ liệu lớn cho cơ sở dữ liệu quản lý phòng khám nha khoa, nhóm đã chọn và sử dụng công cụ **SQL DATA GENERATOR 4** của REDGATE, ưu điểm của công cụ này là có thể tạo ra số lượng dữ liệu cực kì lớn với kiểu dữ liệu được cài đặt trước và có thể liên kết và thêm dữ liệu trực tiếp vào database đã được cài đặt sẵn trong máy.

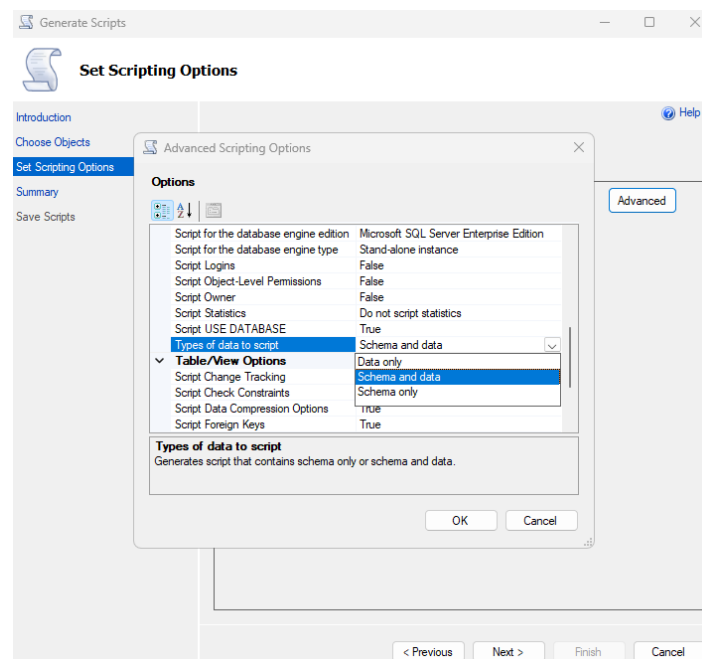
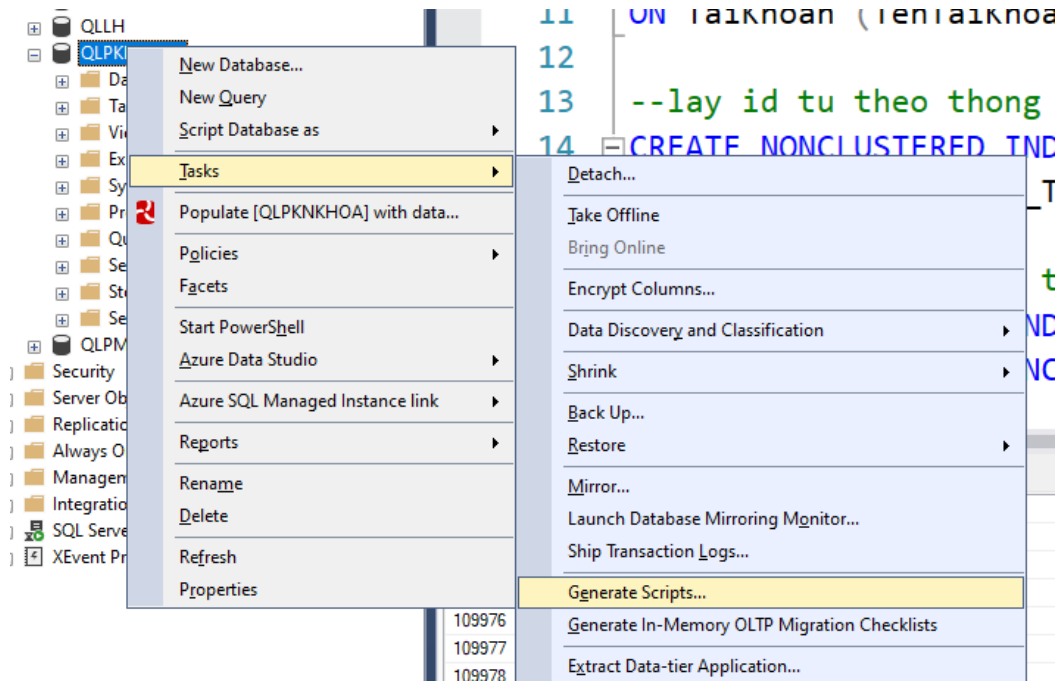


Bằng cách sử dụng các regular expression, và các công cụ có sẵn trong SQL DATA GENERATOR 4 nhóm đã tạo ra lượng dữ liệu khổng lồ với các từ ngữ chuyên ngành phù hợp cho ngành nha khoa



Nhóm đã tạo ra khoảng 100.000 dòng cho mỗi bảng, tuy nhiên do tính chất dữ liệu nên một vài bảng chỉ có vài dòng dữ liệu ví dụ như bảng Rang(4 dòng) chứa thông tin chính của 4 loại răng và MatRang(6 dòng) chứa thông tin chính của 6 mặt răng

Sau khi đã tạo thành công dữ liệu, nhóm sử dụng tính năng generate scripts của Microsoft SQL Server để tạo ra đoạn scripts tạo database và thêm dữ liệu generate nhằm mục đích dễ dàng kiểm thử và cài đặt phần mềm cho các thành viên(đoạn scripts này sẽ được đính kèm trong source code của nhóm)



CHƯƠNG VIII: Các kịch bản sử dụng ứng dụng

8.1 Kịch bản 1: Nhân viên tiếp nhận yêu cầu đặt Lịch hẹn từ Bệnh nhân

Lịch Hẹn bao gồm các thông tin ID_LichHen, ThoiGianHen ID_BenhNhan, ID_NhaSi, ID_PhongKham, ID_NV, ID_TroKham

Để tra cứu Lịch hẹn, Nhân viên có thể tra cứu từ các thông tin sau

- Tra cứu **ID Bệnh nhân**:

- Tra cứu ID_Bệnh Nhân bằng tên Bệnh Nhân hoặc Số điện thoại của Bệnh nhân từ bảng TaiKhoan:

```
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE HoTen LIKE 'Angelica%'
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan where SDT LIKE '0374%'
```

- **Tần suất** tra cứu dựa trên tên hoặc số điện thoại của bệnh nhân: 500 lần/ngày (trên tổng số 80000 hồ sơ)

- Tra cứu **ID của Nha sĩ** chính hoặc Nha sĩ trợ khám:

- Tra cứu ID_NhaSi bằng tên hoặc Số điện thoại bác sĩ (giống như trên)
- Tra cứu dựa trên loại Tài khoản

```
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE TypeTaiKhoan = 'QT'
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE ID_TaiKhoan LIKE 'QT%'
```

- **Tần suất** khoảng 4000 lần/ngày

- Tra cứu dựa trên **Thời Gian hẹn**

```
SELECT * FROM LichHen WHERE ThoiGianHen = '12/23/2023'
```

- **Tần suất** tra cứu theo thời gian hẹn: 200 lần/ngày

Đề xuất index cho Kịch bản 1:

- Table **TaiKhoan**: Họ tên, SDT, Type Tài Khoản
- Table **LichHen**: ID Nha sĩ

Bảng	Cột	Loại index
TaiKhoan	HoTen, SDT, TypeTaiKhoan	Nonclustered
LichHen	ID_NhaSi	Nonclustered

8.2 Kịch bản 2: Nhân viên hoặc Quản trị viên tra cứu Thông tin chi tiết thanh toán cho Bệnh nhân

Thông tin chi tiết thanh toán bao gồm các thông tin ID_HoSo, NgayGiaoDich, NguoiThanhToan, TongTien, TienDaTra, TienThoi, LoaiThanhToan, GhiChu
Để tra cứu Thông tin chi tiết thanh toán có thể tra cứu từ các thông tin sau

- Tra cứu **ID_HoSo**:

- Tra cứu ID_HoSo bằng ID_BenhNhan từ bảng HoSoBenhNhan:

```
SELECT * FROM HoSoBenhNhan WHERE ID_BenhNhan = 'BN00000005'  
SELECT * FROM ThongTinChiTietThanhToan WHERE ID_HoSo = 'HS00885661'
```

- **Tần suất** tra cứu dựa trên ID_HoSo: 500 lần/ngày
- Tra cứu dựa trên **Ngày giao dịch**

```
SELECT * FROM ThongTinChiTietThanhToan WHERE NgayGiaoDich = '12/22/2023'
```

- **Tần suất** tra cứu theo Ngày giao dịch: 200 lần/ngày

Đề xuất index cho Kịch bản 2:

- Table **ThongTinChiTietThanhToan**:ID_HoSo

Bảng	Cột	Loại index
ThongTinChiTietThanhToan	ID_HoSo	Nonclustered

8.3 Kịch bản 3: Nhân viên hoặc Quản trị viên tạo Kế hoạch điều trị cho Bệnh nhân

Một Kế hoạch điều trị bao gồm các thông tin MoTa, NgayDieuTri, GhiChu, TrangThai, PhiDieuTri, ID_HoSo, ID_BacSiThucHien, ID_BacSiTroKham, ID_DanhMuc, ID_ThanhToan

Để tra cứu Thông tin chi tiết thanh toán có thể tra cứu từ các thông tin sau:

- Tra cứu từ **ID_HoSo**:

- Tra cứu ID_HoSo bằng ID_BenhNhan từ bảng HoSoBenhNhan:

```
SELECT * FROM HoSoBenhNhan WHERE ID_BenhNhan = 'BN00000005'  
SELECT * FROM ThongTinChiTietThanhToan WHERE ID_HoSo = 'HS00885661'
```

- **Tần suất** tra cứu dựa trên ID_HoSo: 600 lần/ngày
- Tra cứu dựa trên **ID_BacSiThucHien**

- Tra cứu ID_BacSi từ bảng TaiKhoan

- Có thể tra cứu dựa trên Tên hoặc Số điện thoại của Bác Sĩ

```
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE HoTen LIKE 'Angelica%'  
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan where SDT LIKE '0374%'
```

- Tra cứu dựa trên loại Tài khoản của Nha sĩ: 'NS'

```
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE TypeTaiKhoan = 'QT'  
SELECT ID_TaiKhoan, HoTen FROM TaiKhoan WHERE ID_TaiKhoan LIKE 'QT%'
```

- **Tần suất** tra cứu dựa trên ID_BacSiThucHien
 - Vào khoảng 300 lần ngày (trên tổng số 80000 Kế hoạch điều trị)

- Tra cứu dựa trên **ID_ThanhToan**
 - ID_ThanhToan có thể được lấy từ truy vấn ID_HoSo hoặc Họ tên bệnh nhân trong Hồ sơ tương ứng

```
select * from KeHoachDieuTri where ID_ThanhToan = 'CT00376755'
```

- ID_ThanhToan là điều kiện khá mật thiết với ID_DieuTri vì bảng Thông tin chi tiết thanh toán để tổng hợp lại các Kế hoạch nên truy vấn Kế hoạch điều trị từ bảng này là nhanh nhất
- **Tần suất** truy vấn bảng Kế hoạch điều trị từ ID_ThanhToan vào khoảng 800 lần/ngày (trên tổng số 80000 records)

- Tra cứu dựa trên **Ngày điều trị**

```
SELECT * FROM KeHoachDieuTri WHERE NgayDieuTri = '12/22/2023'
```

- **Tần suất** tra cứu theo Ngày giao dịch: 200 lần/ngày

Đề xuất index cho Kịch bản 3:

- Table **ThôngTinChiTietThanhToan**: ID_HoSo
- Table **KeHoachDieuTri**: ID_ThanhToan, ID_HoSo

Bảng	Cột	Loại index
KeHoachDieuTri	ID_ThanhToan	Nonclustered
KeHoachDieuTri	ID_HoSo	Nonclustered
ThôngTinChiTietThanhToan	ID_HoSo	Nonclustered

8.4: Kịch bản 4: Quản trị viên tìm kiếm thuốc.

Một Thuốc bao gồm các thông tin ID_Thuoc, ID_QTV, TenThuoc, DonViTinh, ChiDinh, SoLuongTonKho, NgayHetHan, DonGia

Để tra cứu thông tin của một loại Thuốc có thể tra cứu từ các thông tin sau:

- Tra cứu từ **tên thuốc**:

```
select * from Thuoc where TenThuoc = 'XBPW24'
```

-
- **Tần suất** tra cứu dựa trên Tên thuốc: 700 lần/ ngày trên tổng số 100000 thuốc

Đề xuất cài đặt index cho Kịch bản 4:

- Table **Thuoc**: TenThuoc

Bảng	Cột	Loại index
Thuoc	TenThuoc	Nonclustered

8.5 Kịch bản 5: Người dùng đăng nhập vào hệ thống.

Thông tin tài khoản được lưu ở bảng TaiKhoan

Để đăng nhập vào hệ thống, ta cho người dùng nhập vào Tên tài khoản và mật khẩu:

```
select * from TaiKhoan where TenTaiKhoan = 'Paula96' and MatKau = 123456
```

Để dễ dàng sử dụng cho người dùng, ta cho họ nhập **Tên tài khoản** để dễ nhớ

Tần suất tra cứu dựa trên Tên tài khoản: 1000 lần/ ngày trên tổng số 100000 tài khoản.

Đề xuất cài đặt index cho Kịch bản 5:

- Table **TaiKhoan**: TenTaiKhoan

Bảng	Cột	Loại index
TaiKhoan	TenTaiKhoan	Unique

CHƯƠNG IX: Cài đặt và hiệu quả của các phương pháp gia tăng hiệu suất truy vấn

9.1 Index trên bảng TaiKhoan (TenTaiKhoan, HoTen, SDT)

- Cài đặt trên thuộc tính **TenTaiKhoan**:

```
CREATE NONCLUSTERED INDEX IX_TaiKhoan_TenTaiKhoan
ON TaiKhoan (TenTaiKhoan);
```

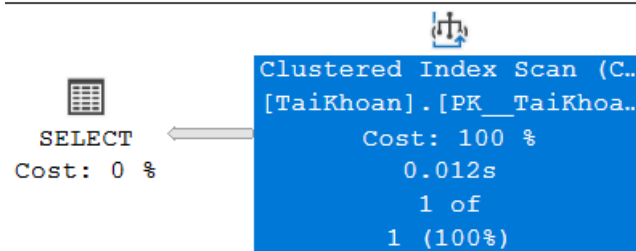
- Tìm kiếm tài khoản có tên 'Paula96': SELECT * FROM TaiKhoan WHERE TenTaiKhoan = 'Paula96'

- **Trước** khi cài index

Query 1: Query cost (relative to the batch): 100%

```
SELECT * FROM [TaiKhoan] WHERE [TenTaiKhoan]=@1
```

Missing Index (Impact 99.8041): CREATE NONCLUSTERED INDEX |

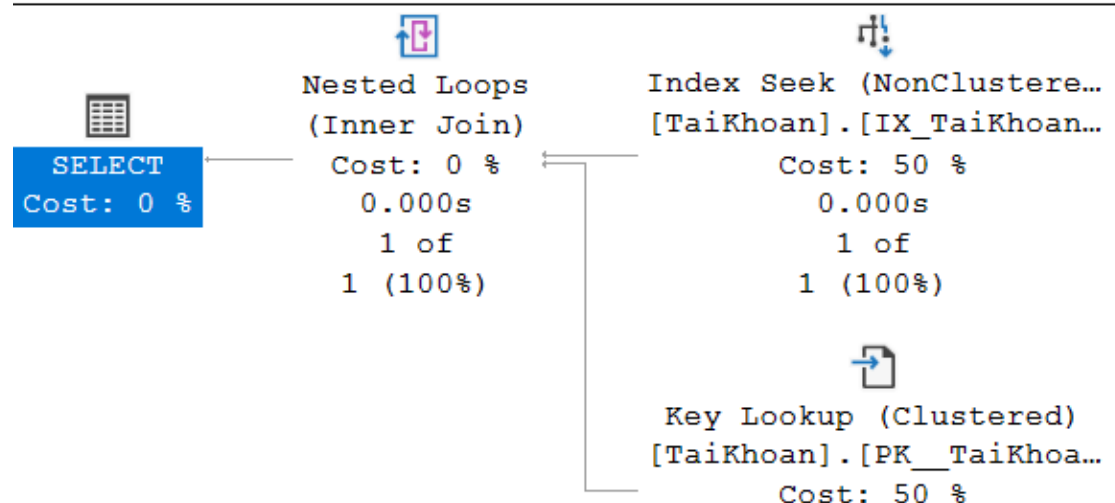


- Thời gian truy vấn là **0.012s**, hệ quản trị DBMS cũng khuyến cáo cài đặt index trên trường này

- **Sau** khi cài index

Query 1: Query cost (relative to the batch): 100%

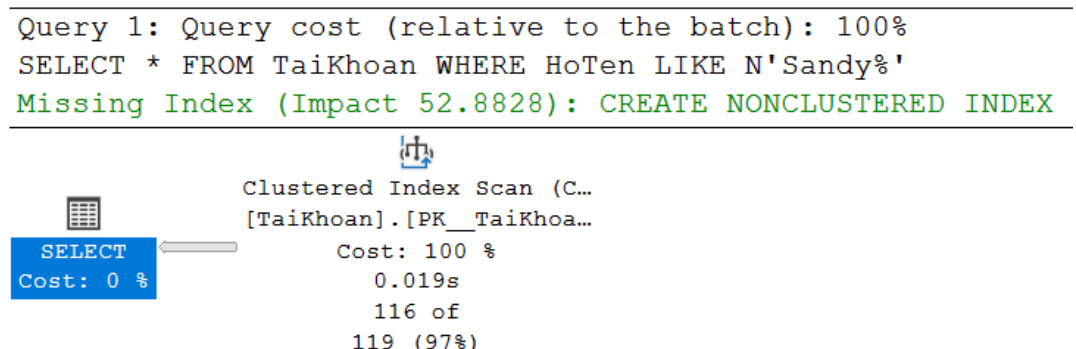
```
SELECT * FROM [TaiKhoan] WHERE [TenTaiKhoan]=@1
```



- Thời gian truy vấn **nhANH** hơn rất nhiều, gần như không tốn thời gian. Thực hiện **Index seek** trên trường TenTaiKhoan

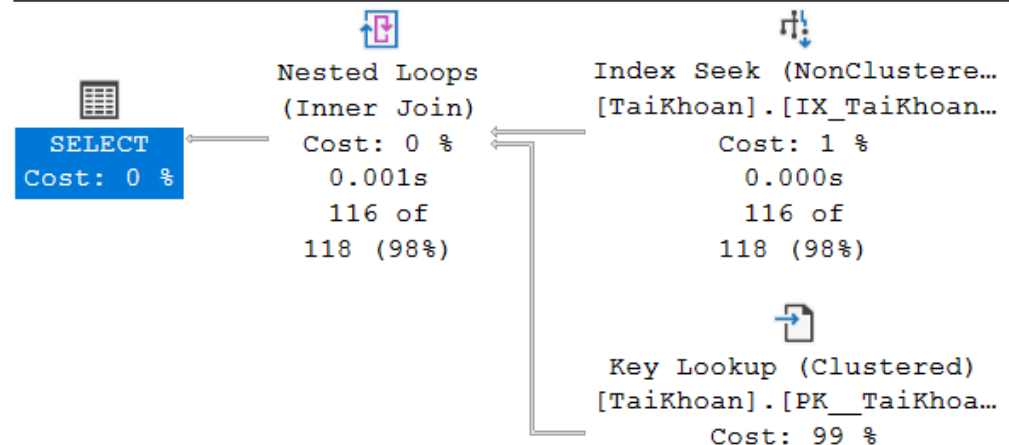
Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0032831 (50%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0032831
Estimated CPU Cost	0.0001581
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1
Estimated Row Size	26 B
Ordered	True
Node ID	1
Object	
[QLPKNKHOA].[dbo].[TaiKhoan].[IX_TaiKhoan_TenTaiKhoan]	
Output List	
[QLPKNKHOA].[dbo].[TaiKhoan].ID_TaiKhoan, [QLPKNKHOA].[dbo].[TaiKhoan].TenTaiKhoan	
Seek Predicates	
Seek Keys[1]: Prefix: [QLPKNKHOA].[dbo].[TaiKhoan].TenTaiKhoan = Scalar Operator('Paula96')	

- Cài đặt trên thuộc tính **HoTen,SDT**:
 - Truy vấn trên trường **HoTen**:
 - Câu lệnh SELECT * FROM TaiKhoan WHERE HoTen LIKE N'Sandy%'
 - Kết quả **trước** khi cài index



- Thời gian truy vấn cho câu trên khoảng **0.02s**. Với những trường đòi hỏi thông tin ở dạng LIKE như Họ Tên (thường sẽ nhập họ hoặc tên) thì cài index là hợp lí
- **Sau** khi tạo index

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM TaiKhoan WHERE HoTen LIKE N'Sandy%'
```



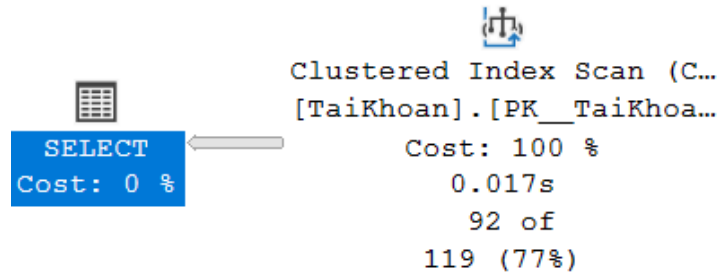
- Thời gian **nhANH hơn đÁNG KỂ**, index gom nhóm các tên lại giúp truy vẤN nhanh hơn

Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	116
Actual Number of Rows for All Executions	116
Actual Number of Batches	0
Estimated Operator Cost	0.003412 (1%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.003412
Estimated CPU Cost	0.000287
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	118.184
Estimated Number of Rows to be Read	118.184
Estimated Row Size	57 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2
Predicate	
[QLPKNKHOA].[dbo].[TaiKhoan].[HoTen] like N'Sandy%'	
Object	
[QLPKNKHOA].[dbo].[TaiKhoan].[IX_TaiKhoan_HoTen]	
Output List	
[QLPKNKHOA].[dbo].[TaiKhoan].ID_TaiKhoan, [QLPKNKHOA].[dbo].[TaiKhoan].HoTen, [QLPKNKHOA].[dbo].[TaiKhoan].SDT, [QLPKNKHOA].[dbo].[TaiKhoan].TypeTaiKhoan	
Seek Predicates	
Seek Keys[1]: Start: [QLPKNKHOA].[dbo].[TaiKhoan].HoTen >= Scalar Operator(N'Sandy'), End: [QLPKNKHOA].[dbo].[TaiKhoan].HoTen < Scalar Operator(N'SandZ')	

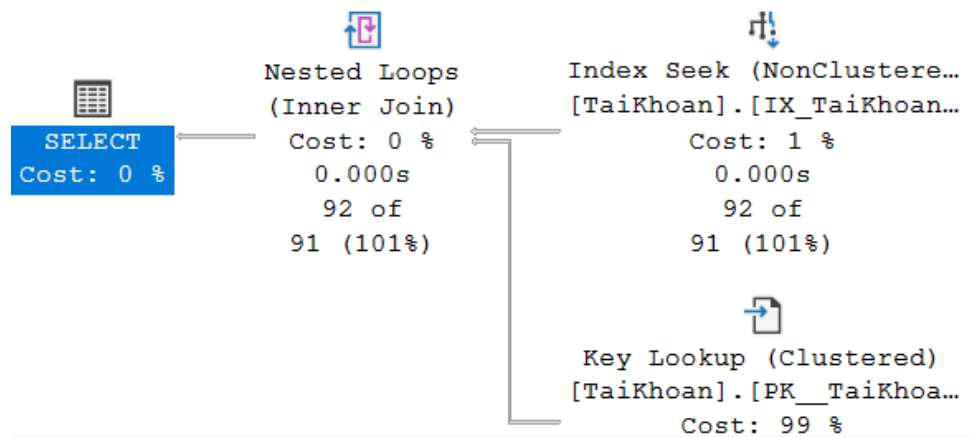
- Số dòng đọc khi **chưa** cái index là **110000** dòng, trong khi cài index là chỉ **116** dòng
- Truy vấn trên trường **SDT**
 - Câu lệnh: SELECT * FROM TaiKhoan WHERE SDT LIKE '03740%'

- Thời gian truy vấn trước khi cài index

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM TaiKhoan WHERE SDT LIKE '03740%'
Missing Index (Impact 52.8828): CREATE NONCLUSTERED



- Với kết quả khoảng 10000 record, ta tốn thời gian gần **0.02s**
- Sau khi cài đặt index



- Thời gian truy vấn gần như lập tức.

Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	92
Actual Number of Rows for All Executions	92
Actual Number of Batches	0
Estimated Operator Cost	0.0033825 (1%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0033825
Estimated CPU Cost	0.0002575
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	91.3242
Estimated Number of Rows to be Read	91.3242
Estimated Row Size	23 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2
Predicate	
[QLPKNKHOA].[dbo].[TaiKhoan].[SDT] like '03740%'	
Object	
[QLPKNKHOA].[dbo].[TaiKhoan].[IX_TaiKhoan_SDT]	
Output List	
[QLPKNKHOA].[dbo].[TaiKhoan].ID_TaiKhoan, [QLPKNKHOA].[dbo].[TaiKhoan].SDT	
Seek Predicates	
Seek Keys[1]: Start: [QLPKNKHOA].[dbo].[TaiKhoan].SDT >= Scalar Operator('0374÷þ'), End: [QLPKNKHOA].[dbo].[TaiKhoan].SDT < Scalar Operator('03741')	

- Số dòng đọc chỉ là **92** trên tổng số hơn 90000 dòng

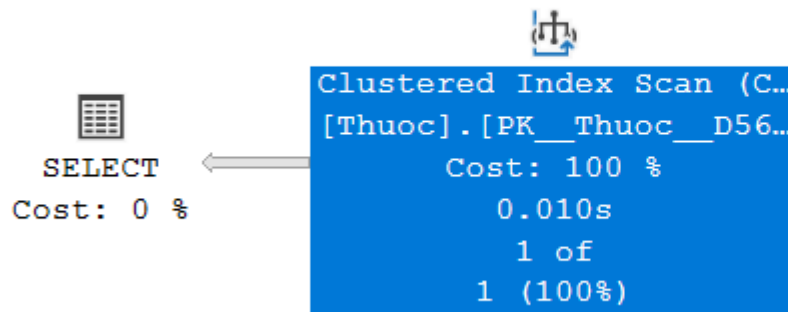
9.2 Index trên bảng Thuoc (TenThuoc)

```
-- tìm thuoc theo ten
]CREATE NONCLUSTERED INDEX IX_Thuoc_TenThuoc
ON Thuoc (TenThuoc);
```

-
- Câu truy vấn: select * from Thuoc where TenThuoc = 'XBPW24'

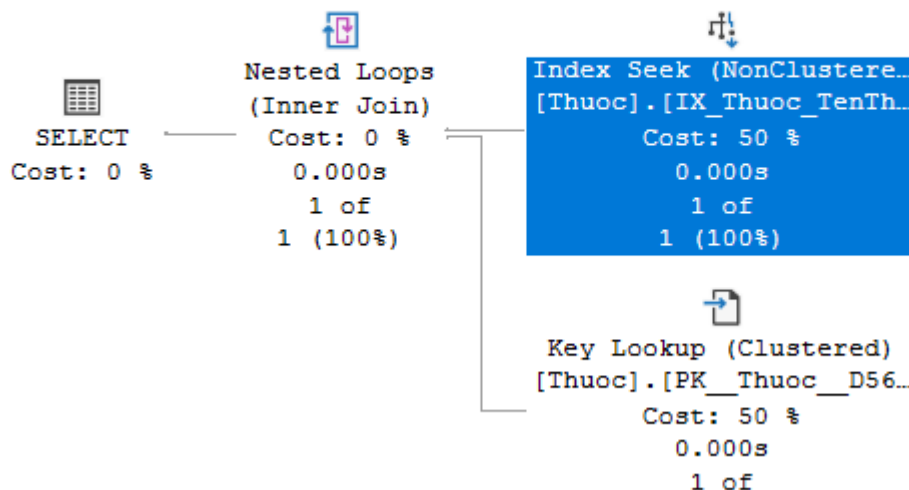
- Trước khi cài index

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [Thuoc] WHERE [TenThuoc]=@1



- Sau khi cài index trên TenThuoc

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [Thuoc] WHERE [TenThuoc]=@1



- Dễ dàng nhận thấy chỉ cost 50% (dùng Index Seek). Khi chưa cài index thời gian truy vấn cho bảng khoảng **100000** record là **0.01s**, sau khi tạo index đường như kết quả truy vấn trả về **tức thời**. Đáp ứng được hiệu suất truy vấn

hiệu quả.

Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	1
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032831 (50%)
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0032831
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1
Estimated Row Size	30 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	1
Object	
[QLPKNKHOA].[dbo].[Thuoc].[IX_Thuoc_TenThuoc]	
Output List	
[QLPKNKHOA].[dbo].[Thuoc].ID_Thuoc, [QLPKNKHOA].[dbo].[Thuoc].TenThuoc	
Seek Predicates	
Seek Keys[1]: Prefix: [QLPKNKHOA].[dbo].[Thuoc].TenThuoc = Scalar Operator(N'XBPW24')	

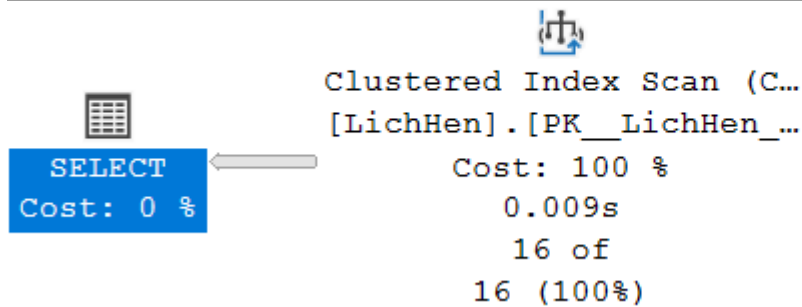
9.3 Index trên bảng Lịch hẹn (ID_NhaSi)

- Câu truy vấn:

```
select * from LịchHen WHERE ID_NhaSi = 'NS00069429'
```

- Trước khi tạo chỉ mục

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [LichHen] WHERE [ID_NhaSi]=@1



- DBMS thực hiện Clustered index scan

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	79977
Actual Number of Rows for All Executions	16
Actual Number of Batches	0
Estimated I/O Cost	0.683866
Estimated Operator Cost	0.771997 (100%)
Estimated CPU Cost	0.0881317
Estimated Subtree Cost	0.771997
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	16
Estimated Number of Rows to be Read	79977
Estimated Row Size	96 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	0

Predicate

[QLPKNKHOA].[dbo].[LichHen].[ID_NhaSi]=[@1]

Object

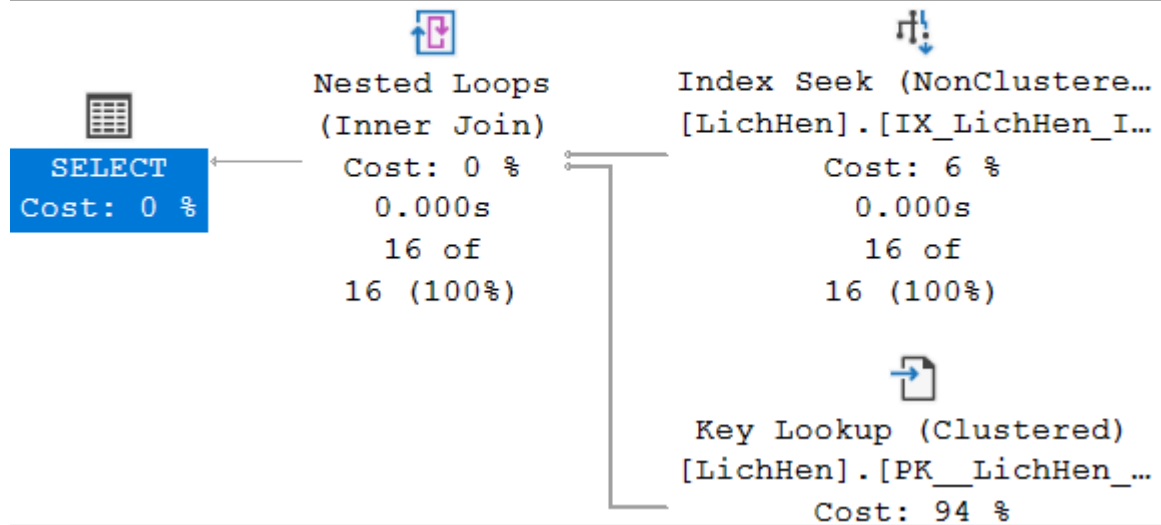
[QLPKNKHOA].[dbo].[LichHen].[PK__LichHen__FE25832C045B0B9D]

Output List

[QLPKNKHOA].[dbo].[LichHen].ID_LichHen, [QLPKNKHOA].[dbo].[LichHen].ThoiGianHen, [QLPKNKHOA].[dbo].[LichHen].TinhTrang, [QLPKNKHOA].[dbo].[LichHen].ID_BenhNhan, [QLPKNKHOA].[dbo].[LichHen].ID_NhaSi, [QLPKNKHOA].[dbo].[LichHen].ID_PhongKham, [QLPKNKHOA].[dbo].[LichHen].ID_NV, [QLPKNKHOA].[dbo].[LichHen].ID_TroKham

- Phải đọc tới gần **80000** dòng
- Sau khi cài đặt index

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [LichHen] WHERE [ID_NhaSi]=@1



Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	16
Actual Number of Rows for All Executions	16
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032996 (6%)
Estimated CPU Cost	0.0001746
Estimated Subtree Cost	0.0032996
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	16
Estimated Number of Rows to be Read	16
Estimated Row Size	28 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	1

Object

[QLPKNKHOA].[dbo].[LichHen].[IX_LichHen_ID_NhaSi]

Output List

[QLPKNKHOA].[dbo].[LichHen].ID_LichHen, [QLPKNKHOA].[dbo].[LichHen].ID_NhaSi

Seek Predicates

Seek Keys[1]: Prefix: [QLPKNKHOA].[dbo].[LichHen].ID_NhaSi = Scalar Operator('NS00069429')

- Chỉ đọc có **16 dòng**, vì ID_NhaSi là trường có lập lại nhưng không quá nhiều nên cài index cho hiệu suất tức thì

9.4 Index trên bảng Thông tin chi tiết thanh toán (ID_HoSo)

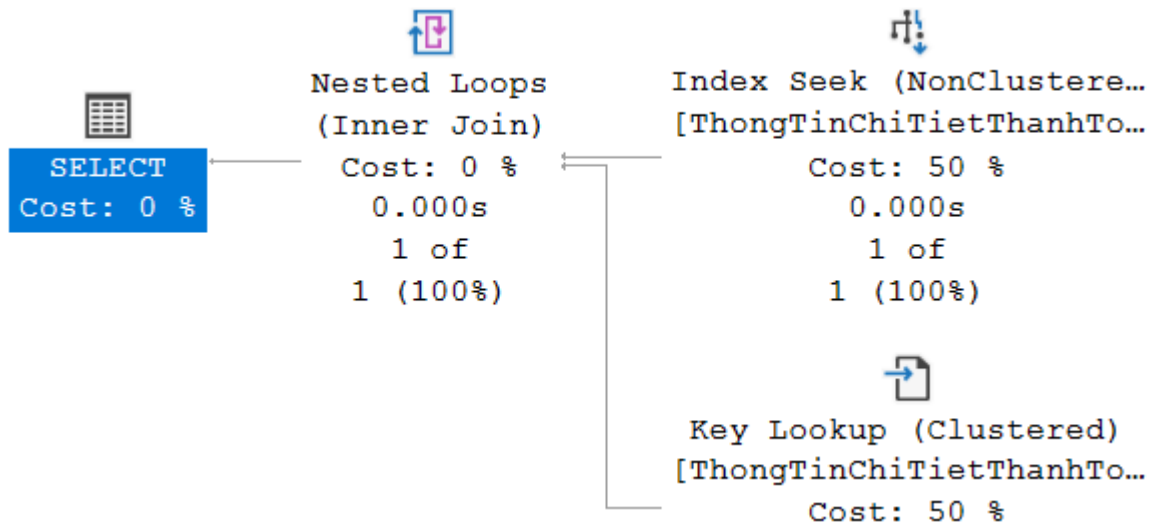
```
select * from ThongTinChiTietThanhToan where ID_HoSo = 'HS00695144'
```

-
- Trước khi tạo index

Clustered Index Scan (Clustered)	
Scanning a clustered index, entirely or only a range.	
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	79977
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.694977
Estimated Operator Cost	0.783109 (100%)
Estimated CPU Cost	0.0881317
Estimated Subtree Cost	0.783109
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	79977
Estimated Row Size	145 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	0

- Thực hiện duyệt tất cả dòng, chi phí Operator cũng khá lớn

- Sau khi cài đặt index



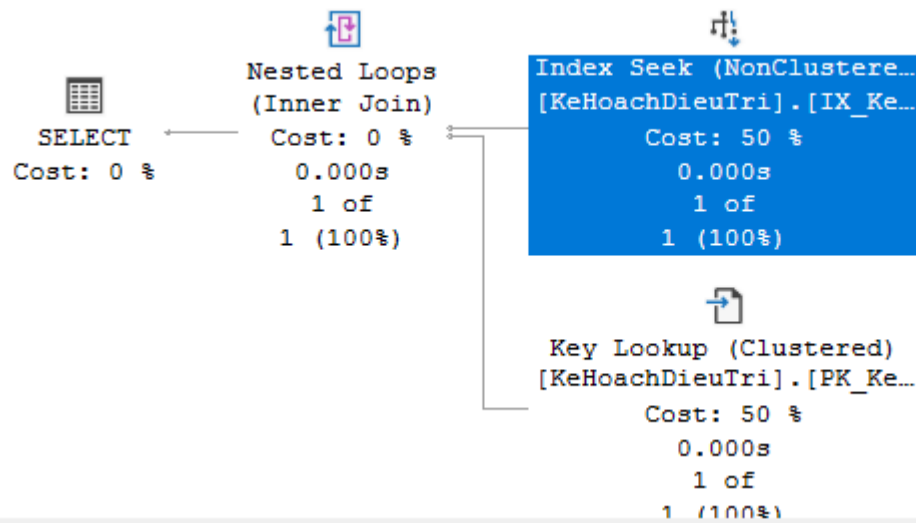
- Thời gian truy xuất nhanh hơn do index đã cost 50%, key chỉ còn tìm 50% nữa

9.5 Index trên bảng Kế hoạch điều trị (ID_ThanhToan, ID_HoSo)

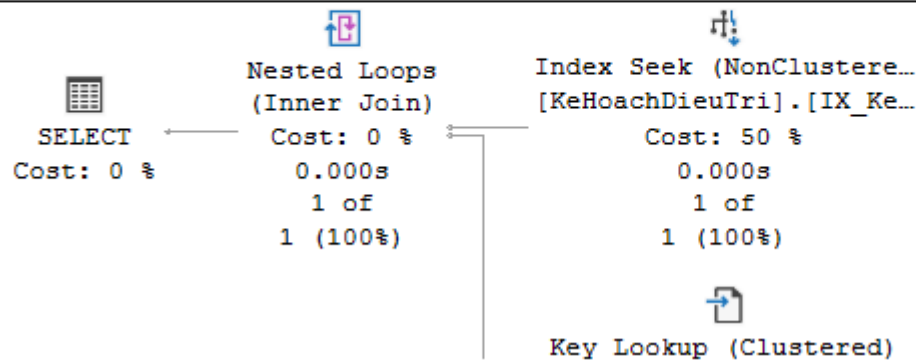
```
select * from KeHoachDieuTri where ID_ThanhToan = 'CT00961714'
select * from KeHoachDieuTri where ID_HoSo = 'HS00695144'
```

-

- Sau khi cài đặt index



Query 2: Query cost (relative to the batch): 50%
 SELECT * FROM [KeHoachDieuTri] WHERE [ID_HoSo]=@1



Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	1
Actual Number of Rows for All Executions	1
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0032831 (50%)
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0032831
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1
Estimated Row Size	35 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	1

- Index **sau** khi được cài cho thời gian truy xuất **0.000s**, đọc số dòng ít hơn thay vì duyệt trên tất cả các dòng của bảng

9.6 Tổng kết Index






- **Ưu điểm:** cho tốc độ truy xuất nhanh nếu như thuộc tính được cài được truy xuất thường xuyên
- **Nhược điểm:** tốn thêm tài nguyên để lưu trữ index, nếu như index cài đặt không cho hiệu quả thì hệ thống sẽ càng bị trì trệ.

9.7 Partition bảng RangDieuTri trên thuộc tính ID_Rang

Vì số lượng răng tồn tại trong bảng Răng chỉ là 4, hơn nữa tồn tại đến gần 100.000 dòng dữ liệu trong bảng RangDieuTri có lưu trữ giá trị ID_Rang và mỗi ID_Rang tồn tại duy nhất trong mỗi dòng.

Results		Messages	
	ID_Rang	TenRang	PhiDichVu
1	RA00437900	Răng cửa	0
2	RA00517649	Răng nan	500000
3	RA00787374	Răng nhai	1000000
4	RA00963268	Răng hàm	0

Nên nhóm đã quyết định cài đặt 5 file group với tên lần lượt là Partition_1, Partition_2, Partition_3, Partition_4, Partition_5 nhằm tối ưu việc truy vấn trên bảng RangDieuTri (trong đó Partition_1 cài đặt nonclustered index các khoá còn lại như ID_KHDieuTri, ID_MatRang, các Partition_2,3,4,5 còn lại lưu các được chia ra lưu các giá trị tương ứng với ID_Rang lần lượt là RA00437900, RA00517649, RA00787374, RA00963268)

	Partition_1.ndf	23-Dec-23 8:56 AM	SQL Server Databa...	102,400 KB
	Partition_2.ndf	22-Dec-23 12:13 PM	SQL Server Databa...	102,400 KB
	Partition_3.ndf	22-Dec-23 12:04 PM	SQL Server Databa...	102,400 KB
	Partition_4.ndf	22-Dec-23 12:13 PM	SQL Server Databa...	102,400 KB
	Partition_5.ndf	23-Dec-23 2:46 PM	SQL Server Databa...	102,400 KB

2	1	Partition_1	79977
3	2	Partition_2	19995
4	3	Partition_3	19994
5	4	Partition_4	19994
6	5	Partition_5	19994



CHƯƠNG X: Tài liệu tham khảo

<https://viblo.asia/p/su-dung-index-trong-sql-query-1ZnbRIPQR2Xo>

<https://learn.microsoft.com/vi-vn/sql/t-sql/functions/partition-transact-sql?view=sql-server-ver15>

<https://viblo.asia/p/tang-toc-performance-query-sql-voi-partitions-WAyK89XEZxX>

<https://learn.microsoft.com/vi-vn/azure/azure-sql/database/connect-query-dotnet-visual-studio?view=azuresql>

https://www.w3schools.com/sql/sql_ref_create_unique_index.asp

Tài liệu lí thuyết cô Nguyễn Trần Minh Thư

Tài liệu thực hành cô Hồ Thị Hoàng Vy