

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU PHÂN TÁN

ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÝ

KHÁCH SẠN

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Lớp
1	1771020781	Lưu Thế Hưng	30/12/2000	CNTT 17-10
2	1771020780	Nguyễn Trung Kiên	31/01/2003	CNTT 17-10

Hà Nội, năm 2025

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
ĐỀ TÀI: THIẾT KẾ CƠ SỞ DỮ LIỆU QUẢN LÝ
KHÁCH SẠN**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	1771020781	Lưu Thế Hưng	30/12/2000		
2	1771020780	Nguyễn Trung Kiên	31/01/2003		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong bối cảnh xã hội ngày càng phát triển, ngành du lịch và dịch vụ lưu trú ngày càng khẳng định vai trò quan trọng trong nền kinh tế. Cùng với đó, nhu cầu quản lý hoạt động kinh doanh khách sạn một cách khoa học, hiệu quả, chính xác và nhanh chóng trở thành yêu cầu cấp thiết. Việc ứng dụng công nghệ thông tin, đặc biệt là việc xây dựng và sử dụng hệ cơ sở dữ liệu trong quản lý khách sạn không chỉ giúp tiết kiệm thời gian, công sức mà còn nâng cao tính chuyên nghiệp và chất lượng phục vụ khách hàng.

Đề tài "Xây dựng hệ cơ sở dữ liệu quản lý khách sạn" được thực hiện nhằm xây dựng một hệ thống cơ sở dữ liệu đầy đủ, chặt chẽ, giúp cho việc lưu trữ, tra cứu và xử lý thông tin trong khách sạn trở nên dễ dàng và thuận tiện. Hệ cơ sở dữ liệu này sẽ hỗ trợ quản lý thông tin khách hàng, thông tin đặt phòng, dịch vụ đi kèm, hóa đơn thanh toán cũng như báo cáo tình hình kinh doanh của khách sạn. Qua đó, giúp ban quản lý khách sạn có cái nhìn tổng quan về hoạt động kinh doanh, tối ưu hóa việc sử dụng tài nguyên và cải tiến dịch vụ theo hướng chuyên nghiệp, hiện đại hơn.

Bên cạnh đó, hệ cơ sở dữ liệu còn giúp giảm thiểu các sai sót trong quá trình ghi chép thủ công, đồng thời tạo ra một môi trường làm việc khoa học, hiệu quả và nhất quán giữa các bộ phận trong khách sạn. Đây cũng là tiền đề quan trọng để khách sạn từng bước chuyển đổi số và áp dụng các công nghệ hiện đại vào công tác quản lý kinh doanh.

Thông qua đề tài này, nhóm không chỉ mong muốn vận dụng những kiến thức đã học về cơ sở dữ liệu vào thực tiễn mà còn hướng tới việc tạo ra một sản phẩm hữu ích, đáp ứng nhu cầu quản lý của các khách sạn hiện nay.

Do giới hạn về thời gian và kinh nghiệm thực tế, bài làm không tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự góp ý từ thầy cô và các bạn để hoàn thiện hơn nữa hệ thống này.

Xin chân thành cảm ơn!

MỤC LỤC

Lời nói đầu	3
Mục lục hình ảnh	6
Mục lục bảng	10
Chương 1. Tổng quan về đề tài	11
1.1. Giới thiệu đề tài.....	11
1.2. Mục tiêu của đề tài	11
1.3. Lý do chọn đề tài	11
1.4. Phạm vi nghiên cứu.....	12
Chương 2. Mô hình dữ liệu cơ sở quan hệ.....	13
2.1. Phân tích yêu cầu hệ thống	13
2.2. Mô tả các bảng dữ liệu	13
2.2.1. Bảng Khách hàng	13
2.2.2. Bảng Phòng	14
2.2.3. Bảng dịch vụ	14
2.2.4. Bảng đặt phòng.....	14
2.2.5. Bảng nhân viên	15
2.2.6. Bảng thanh toán.....	15
2.2.7. Bảng chi tiết hóa đơn.....	15
2.2.8. Bảng lịch sử phòng.....	16
2.3. Các ràng buộc toàn vẹn dữ liệu.....	16
2.4. Tối ưu hóa cơ sở dữ liệu.....	16
2.5. Mô hình cơ sở dữ liệu.....	17
Chương 3. Tạo cơ sở dữ liệu	18
3.1. Tạo Database.....	18
3.2. Tạo các bảng dữ liệu	18

3.3 Thêm dữ liệu vào cơ sở dữ liệu	Error! Bookmark not defined. 0
3.4. Kiểm tra dữ liệu trong cơ sở dữ liệu	23
Chương 4. Xây dựng các View	26
4.1. Khung nhìn view trong chương trình quản lý khách sạn	Error! Bookmark not defined. 6
4.2. Xây dựng các view	27
Chương 5. Xây dựng các Procedure	32
5.1. Thủ tục lưu trữ Procedure trong chương trình quản lý khách sạn	3Error! Bookmark not defined.
5.2. Xây dựng các procedure	32
Chương 6. Xây dựng các Trigger	38
4.1. Trigger trong chương trình quản lý khách sạn	38
4.2. Xây dựng các trigger	39
Chương 7. Phân quyền và bảo vệ cơ sở dữ liệu	46
7.1. Tạo Login cho Manager và Receptionist	46
7.2. Quản lý vai trò (Role).....	46
7.2.1. Tạo và xóa vai trò HotelManager.....	46
7.2.2. Tạo và xóa vai trò Receptionist.....	47
7.3. Quản lý Users.....	48
7.4. Phân quyền.....	48
7.4.1. Receptionist.....	48
7.4.2. HotelManager.....	49
7.5. Kiểm tra dữ liệu	49
7.6. Sao lưu cơ sở dữ liệu.....	50
Kết luận	51
Tài liệu tham khảo	52

MỤC LỤC HÌNH ẢNH

Hình 2.1. Mô hình cơ sở dữ liệu	17
Hình 3.1. Tạo Database HotelManagement.....	18
Hình 3.2.1. Tạo bảng Customers.....	18
Hình 3.2.2. Tạo bảng Rooms	18
Hình 3.2.3. Tạo bảng Services	19
Hình 3.2.4. Tạo bảng Bookings	19
Hình 3.2.5. Tạo bảng Employees.....	19
Hình 3.2.6. Tạo bảng Payments	19
Hình 3.2.7. Tạo bảng InvoiceDetails.....	20
Hình 3.2.8. Tạo bảng RoomHistory	20
Hình 3.3.1. Thêm dữ liệu cho bảng Customers.....	20
Hình 3.3.2. Thêm dữ liệu cho bảng Rooms	21
Hình 3.3.3. Thêm dữ liệu cho bảng Services.....	21
Hình 3.3.4. Thêm dữ liệu cho bảng Employees	21
Hình 3.3.5. Thêm dữ liệu cho bảng Bookings	21
Hình 3.3.6. Thêm dữ liệu cho bảng Payments	22
Hình 3.3.7. Thêm dữ liệu cho bảng InvoiceDetails.....	22
Hình 3.3.8. Thêm dữ liệu cho bảng RoomHistory	22
Hình 3.4.1. Bảng Customer sau khi được chèn dữ liệu	23
Hình 3.4.2. Bảng Rooms sau khi được chèn dữ liệu	23
Hình 3.4.3. Bảng Services sau khi được chèn dữ liệu	23
Hình 3.4.4. Bảng Employees sau khi được chèn dữ liệu.....	24
Hình 3.4.5. Bảng Bookings sau khi được chèn dữ liệu	24
Hình 3.4.6. Bảng Payments sau khi được chèn dữ liệu	24
Hình 3.4.7. Bảng InvoiceDetails sau khi được chèn dữ liệu..	24
Hình 3.4.8. Bảng RoomHistory sau khi được chèn dữ liệu	25

Hình 4.1. Tạo khung nhìn view vw_CurrentBookings.....	27
Hình 4.2. Kiểm tra dữ liệu khung nhìn vw_CurrentBookings.....	27
Hình 4.3. Tạo khung nhìn view vw_RoomRevenue	27
Hình 4.4. Kiểm tra dữ liệu khung nhìn vw_RoomRevenue	27
Hình 4.5. Tạo khung nhìn view vw_ServiceUsage	27
Hình 4.6 Kiểm tra dữ liệu khung nhìn vw_ServiceUsage.	28
Hình 4.7. Tạo khung nhìn view vw_EmployeeShifts.....	28
Hình 4.8. Kiểm tra dữ liệu khung nhìn vw_EmployeeShifts.....	28
Hình 4.9. Tạo khung nhìn view vw_RecentRoomHistory.....	28
Hình 4.10. Kiểm tra dữ liệu khung nhìn vw_RecentRoomHistory	28
Hình 4.11. Tạo khung nhìn view vw_CustomerBookingHistory	29
Hình 4.12. Kiểm tra dữ liệu khung nhìn vw_CustomerBookingHistory.....	29
Hình 4.13. Tạo khung nhìn view vw_TopServicesUsed	29
Hình 4.14. Kiểm tra dữ liệu khung nhìn vw_TopServicesUsed	29
Hình 4.15. Tạo khung nhìn view vw_UnpaidBookings	30
Hình 4.16. Kiểm tra dữ liệu khung nhìn vw_UnpaidBookings	30
Hình 4.17. Tạo khung nhìn view vw_EmployeeSalaryReport	30
Hình 4.18. Kiểm tra dữ liệu khung nhìn vw_EmployeeSalaryReport	31
Hình 4.19. Tạo khung nhìn view vw_AvailableRooms	31
Hình 4.20. Kiểm tra dữ liệu khung nhìn vw_AvailableRooms.....	31
Hình 5.1. Tạo proceduce sp_AddBooking.....	32
Hình 5.2. Kiểm tra dữ liệu procedure sp_AddBooking.....	32
Hình 5.3. Tạo procedure sp_CalculateInvoice	33
Hình 5.4. Kiểm tra dữ liệu procedure sp_CalculateInvoice.....	33
Hình 5.5. Tạo procedure sp_ProcessPayment.....	33
Hình 5.6. Kiểm tra dữ liệu procedure sp_ProcessPayment	33

Hình 5.7. Tạo procedure sp_AddCustomer	34
Hình 5.8. Kiểm tra dữ liệu procedure sp_AddCustomer	34
Hình 5.9. Tạo procedure sp_UpdateCustomer	34
Hình 5.10. Kiểm tra dữ liệu procedure sp_UpdateCustomer	34
Hình 5.11. Tạo procedure sp_DeleteCustomer	35
Hình 5.12. Kiểm tra dữ liệu procedure sp_DeleteCustomer	35
Hình 5.13. Tạo procedure sp_AddService	35
Hình 5.14. Kiểm tra dữ liệu procedure sp_AddService	35
Hình 5.15. Tạo procedure sp_UpdateService	36
Hình 5.16. Kiểm tra dữ liệu procedure sp_UpdateService	36
Hình 5.17. Tạo procedure sp_DeleteService	36
Hình 5.18. Kiểm tra dữ liệu procedure sp_DeleteService	36
Hình 5.19. Tạo procedure sp_GetAvailableRooms	37
Hình 5.20. Kiểm tra dữ liệu procedure sp_GetAvailableRooms	37
Hình 6.1. Tạo trigger tr_BookingInsert	39
Hình 6.2. Kiểm tra trigger tr_BookingInsert	39
Hình 6.3. Tạo trigger tr_BookingStatusUpdate	39
Hình 6.4. Kiểm tra trigger tr_BookingStatusUpdate	40
Hình 6.5. Tạo trigger tr_RoomStatusUpdate	40
Hình 6.6. Kiểm tra trigger tr_RoomStatusUpdate	41
Hình 6.7. Tạo trigger tr_CustomerDelete_CheckBookings	41
Hình 6.8. Kiểm tra trigger tr_CustomerDelete_CheckBookings	41
Hình 6.9. Tạo trigger tr_AutoCheckOut_ExpiredBookings	42
Hình 6.10. Kiểm tra trigger tr_AutoCheckOut_ExpiredBookings	42
Hình 6.11. Tạo trigger tr_LogServiceUpdate	42
Hình 6.12. Kiểm tra trigger tr_LogServiceUpdate	42

Hình 6.13. Tạo trigger tr_Service_Delete_CheckInvoice	43
Hình 6.14. Kiểm tra trigger tr_Service_Delete_CheckInvoice	43
Hình 6.15. Tạo trigger tr_UpdateRoomPrice_LogChange	43
Hình 6.16. Kiểm tra trigger tr_UpdateRoomPrice_LogChange	43
Hình 6.17. Tạo trigger tr_UpdateRoomStatusOnBooking	44
Hình 6.18. Kiểm tra trigger tr_UpdateRoomStatusOnBooking	44
Hình 6.19. Tạo trigger tr_PreventRoomDelete	45
Hình 6.20. Kiểm tra trigger tr_PreventRoomDelete	45
Hình 7.1. Tạo login ManagerLogin và ReceptionistLogin với mật khẩu	46
Hình 7.2.1. Tạo vai trò HotelManager và xóa nếu đã tồn tại	47
Hình 7.2.2. Tạo vai trò Receptionist và xóa nếu đã tồn tại	47
Hình 7.3. Tạo user và gán vào vai trò để quản lý quyền	48
Hình 7.4.1. Phân quyền cho Receptionist	48
Hình 7.4.2. Phân quyền cho HotelManager	49
Hình 7.5.1. Thông tin bảo mật	49
Hình 7.5.2. Thông tin phân quyền với Receptionist	49
Hình 7.5.3. Thông tin phân quyền với HotelManager	50
Hình 7.6. Sao lưu cơ sở dữ liệu	50

MỤC LỤC BẢNG

Bảng 2.1. Bảng khách hàng.....	13
Bảng 2.2. Bảng Phòng.....	14
Bảng 2.3. Bảng dịch vụ	14
Bảng 2.4. Bảng đặt phòng	14
Bảng 2.5. Bảng nhân viên	15
Bảng 2.6. Bảng chi tiết đơn hàng	15
Bảng 2.7. Bảng chi tiết đơn hàng	15
Bảng 2.8. Bảng lịch sử phòng	16

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu đề tài

Hệ thống quản lý khách sạn là một phần mềm giúp tự động hóa các quy trình kinh doanh trong khách sạn như đặt phòng, thanh toán, quản lý khách hàng, quản lý dịch vụ và báo cáo thống kê. Việc áp dụng một hệ quản trị cơ sở dữ liệu (DBMS) vào hệ thống quản lý khách sạn giúp nâng cao hiệu suất vận hành, giảm thiểu sai sót và tối ưu hóa trải nghiệm khách hàng.

Trong bối cảnh ngành du lịch và dịch vụ lưu trú ngày càng phát triển, nhu cầu về một hệ thống quản lý khách sạn hiệu quả ngày càng cao. Việc triển khai hệ thống này giúp khách sạn cải thiện khả năng phục vụ, tăng doanh thu và nâng cao tính chuyên nghiệp trong công tác quản lý.

1.2. Mục tiêu của đề tài

- Xây dựng cơ sở dữ liệu cho hệ thống quản lý khách sạn với các thực thể chính như khách hàng, phòng, dịch vụ, hóa đơn và nhân viên.
- Sử dụng SQL Server để thiết kế và triển khai cơ sở dữ liệu, bao gồm các thực thể, quan hệ, chỉ mục (index), khung nhìn (view), thủ tục lưu trữ (stored procedure), hàm do người dùng định nghĩa (UDF) và trigger.
- Đảm bảo tính bảo mật của dữ liệu, phân quyền người dùng và quản trị hệ thống hiệu quả.
- Xây dựng một hệ thống quản lý khách sạn có khả năng xử lý các thao tác chính như đặt phòng, thanh toán, thêm/sửa/xóa thông tin khách hàng và dịch vụ.

1.3. Lý do chọn đề tài

- Tính thực tiễn cao: Quản lý khách sạn là một ứng dụng phổ biến và có thể triển khai thực tế trong các khách sạn, nhà nghỉ và khu nghỉ dưỡng.
- Ứng dụng rộng rãi của hệ quản trị cơ sở dữ liệu: Đề tài giúp sinh viên vận dụng các kiến thức về SQL Server, bao gồm truy vấn dữ liệu, tối ưu hóa hiệu suất, bảo mật và quản trị hệ thống.
- Thử thách kỹ thuật: Việc xây dựng hệ thống này yêu cầu sinh viên áp dụng nhiều kỹ thuật khác nhau như stored procedure, trigger, view và indexing, giúp nâng cao kỹ năng lập trình SQL.
- Hỗ trợ quyết định kinh doanh: Hệ thống có thể tích hợp các báo cáo và phân tích dữ liệu để hỗ trợ khách sạn trong việc ra quyết định kinh doanh.

- Khả năng mở rộng và phát triển: Hệ thống có thể mở rộng thêm các tính năng như quản lý nhân sự, tích hợp thanh toán trực tuyến hoặc kết nối với các nền tảng đặt phòng như Booking.com hay Agoda.

1.4. Phạm vi nghiên cứu

- Đề tài tập trung vào thiết kế và triển khai cơ sở dữ liệu cho hệ thống quản lý khách sạn.
- Các chức năng chính bao gồm: đặt phòng, quản lý khách hàng, quản lý dịch vụ, hóa đơn thanh toán và báo cáo thống kê.
- Hệ thống sẽ được triển khai trên SQL Server và có thể mở rộng lên một giao diện ứng dụng để sử dụng thực tế.

CHƯƠNG 2. MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ

2.1. Phân tích yêu cầu hệ thống

Hệ thống quản lý khách sạn cần đáp ứng các chức năng chính:

- **Quản lý khách hàng:** Lưu trữ thông tin khách hàng, bao gồm họ tên, số điện thoại, email, địa chỉ, ngày sinh và giới tính.
- **Quản lý phòng:** Theo dõi danh sách phòng, bao gồm số phòng, loại phòng, giá và trạng thái phòng.
- **Quản lý dịch vụ:** Lưu trữ thông tin về các dịch vụ bổ sung mà khách hàng có thể sử dụng như bữa sáng, giặt là, thuê xe, v.v.
- **Đặt phòng:** Cho phép khách đặt phòng, kiểm tra trạng thái phòng, ghi nhận số lượng khách và thời gian lưu trú.
- **Quản lý nhân viên:** Theo dõi thông tin nhân viên khách sạn, bao gồm chức vụ, lương, ngày nhận việc và ca làm việc.
- **Thanh toán:** Xử lý các khoản thanh toán, phương thức thanh toán và chi tiết hóa đơn liên quan đến đặt phòng và dịch vụ.

2.2. Mô tả các bảng dữ liệu

Dựa trên yêu cầu hệ thống, cơ sở dữ liệu HotelManagement bao gồm các bảng sau:

2.2.1. Bảng Khách hàng (Customers)

Chứa thông tin khách hàng, có khóa chính là CustomerID.

Bảng 2.1. Bảng Khách Hàng

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	CustomerID	INT IDENTITY(1,1)	PK	
2	FullName	NVARCHAR(100)		Họ tên khách hàng
3	Phone	NVARCHAR(15)		Số điện thoại (duy nhất)
4	Email	NVARCHAR(100)		Email khách hàng (có thể NULL)
5	Address	NVARCHAR(255)		Địa chỉ khách hàng
6	DateOfBirth	DATE		Ngày sinh
7	Gender	NCHAR(1)		Giới tính (M, F, O)

2.2.2. Bảng Phòng (Rooms)

Thông tin về phòng khách sạn, khóa chính là RoomID.

Bảng 2.2. Bảng Phòng

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	RoomID	INT IDENTITY(1,1)	PK	Mã phòng
2	RoomNumber	NVARCHAR(10)		Số phòng (duy nhất)
3	RoomType	NVARCHAR(50)		Loại phòng (Single, Double, Suite)
4	Price	DECIMAL(10,2)		Giá phòng
5	Status	NVARCHAR(20)		Trạng thái phòng (Available, Booked)

2.2.3. Bảng Dịch vụ (Services)

Danh sách các dịch vụ khách sạn cung cấp, khóa chính là ServiceID.

Bảng 2.3. Bảng Dịch Vụ

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	ServiceID	INT IDENTITY(1,1)	PK	Mã dịch vụ
2	ServiceName	NVARCHAR(100)		Tên dịch vụ
3	Price	DECIMAL(10,2)		Giá dịch vụ

2.2.4. Bảng Đặt phòng (Bookings)

Ghi nhận thông tin đặt phòng, khóa chính là BookingID.

Bảng 2.4. Bảng Đặt Phòng

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	BookingID	INT IDENTITY(1,1)	PK	Mã đặt phòng
2	CustomerID	INT		Mã khách hàng (khóa ngoại từ Customers)
3	RoomID	INT		Mã phòng (khóa ngoại từ Rooms)
4	CheckInDate	DATETIME		Ngày nhận phòng
5	CheckOutDate	DATETIME		Ngày trả phòng
6	GuestCount	INT		Số lượng khách
7	Status	NVARCHAR(20)		Trạng thái đặt phòng (Pending, Checked-In, Checked-Out, Cancelled)
8	CreatedAt	DATETIME		Ngày đặt phòng

2.2.5. Bảng Nhân viên (Employees)

Quản lý thông tin nhân viên, khóa chính là EmployeeID.

Bảng 2.5. Bảng Nhân Viên

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	EmployeeID	INT IDENTITY(1,1)	PK	Mã nhân viên
2	FullName	NVARCHAR(100)		Họ tên nhân viên
3	Role	NVARCHAR(50)		Chức vụ (Quản lý, Lễ tân, Buồng phòng)
4	Salary	DECIMAL(10,2)		Lương nhân viên
5	HireDate	DATE		Ngày nhận việc
6	Shift	NVARCHAR(20)		Ca làm việc (Sáng, Chiều, Đêm)

2.2.6. Bảng Thanh toán (Payments)

Ghi nhận các khoản thanh toán cho đặt phòng, khóa chính là PaymentID.

Bảng 2.6. Bảng Thanh Toán

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	PaymentID	INT IDENTITY(1,1)	PK	Mã thanh toán
2	BookingID	INT		Mã đặt phòng (khóa ngoại từ Bookings)
3	Amount	DECIMAL(10,2)		Số tiền thanh toán
4	PaymentMethod	NVARCHAR(5)		Phương thức thanh toán (Cash, Credit Card, Debit Card, Online)
5	PaymentDate	DATETIME		Ngày thanh toán

2.2.7. Bảng Chi tiết hóa đơn (InvoiceDetails)

Ghi nhận các dịch vụ khách sử dụng trong thời gian lưu trú, khóa chính là InvoiceDetailID.

Bảng 2.7. Bảng Chi Tiết Đơn Hàng

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	InvoiceDetailID	INT IDENTITY(1,1)	PK	Mã đơn hàng
2	PaymentID	INT		Mã thanh toán (khóa ngoại từ Payments)

3	ServiceID	INT		Mã dịch vụ (khóa ngoại từ Services)
4	Quantity	INT		Số lượng dịch vụ đã sử dụng
5	UnitPrice	DECIMAL(10,2)		Giá của một đơn vị dịch vụ
6	TotalPrice	Cột tính toán (AS (Quantity * UnitPrice))		Tổng tiền (Quantity * UnitPrice)

2.2.8. Bảng lịch sử phòng (RoomHistory)

Ghi nhận các phòng đã qua sử dụng, khóa chính là HistoryID.

Bảng 2.8. Bảng RoomHistory

TT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ghi chú
1	HistoryID	INT IDENTITY(1,1)	PK	Mã lịch sử
2	RoomID	INT		Mã phòng
3	BookingID	INT		Mã đặt phòng
4	Status	NVARCHAR(50)		Trạng thái đặt phòng
5	UpdatedAt	DATETIME		Thời gian

2.3. Các ràng buộc toàn vẹn dữ liệu

- **Khóa chính (Primary Key - PK):** Đảm bảo mỗi bản ghi là duy nhất trong các bảng.
- **Khóa ngoại (Foreign Key - FK):** Liên kết giữa các bảng (ví dụ: BookingID trong Payments tham chiếu đến Bookings).
- **Ràng buộc CHECK:**
 - o Giá trị Price trong bảng Rooms và Services phải lớn hơn 0.
 - o GuestCount trong Bookings phải lớn hơn 0.
 - o Status trong Bookings chỉ nhận giá trị (Pending, Checked-In, Checked-Out, Cancelled).
 - o PaymentMethod trong Payments chỉ nhận các giá trị hợp lệ (Cash, Credit Card, Debit Card, Online).
- **Ràng buộc UNIQUE:**
 - o Phone và Email trong bảng Customers không được trùng.
 - o RoomNumber trong bảng Rooms không được trùng.
- **Ràng buộc NOT NULL:** Đảm bảo các cột quan trọng không để trống, như FullName, Phone trong Customers.

2.4. Tối ưu hóa cơ sở dữ liệu

➤ Chỉ mục (Index):

- o IX_Customers_Phone để tăng tốc tìm kiếm khách hàng theo số điện thoại.

- IX_Rooms_RoomNumber để tăng tốc truy vấn theo số phòng.
- IX_Bookings_Status để tăng hiệu suất truy vấn theo trạng thái đặt phòng.

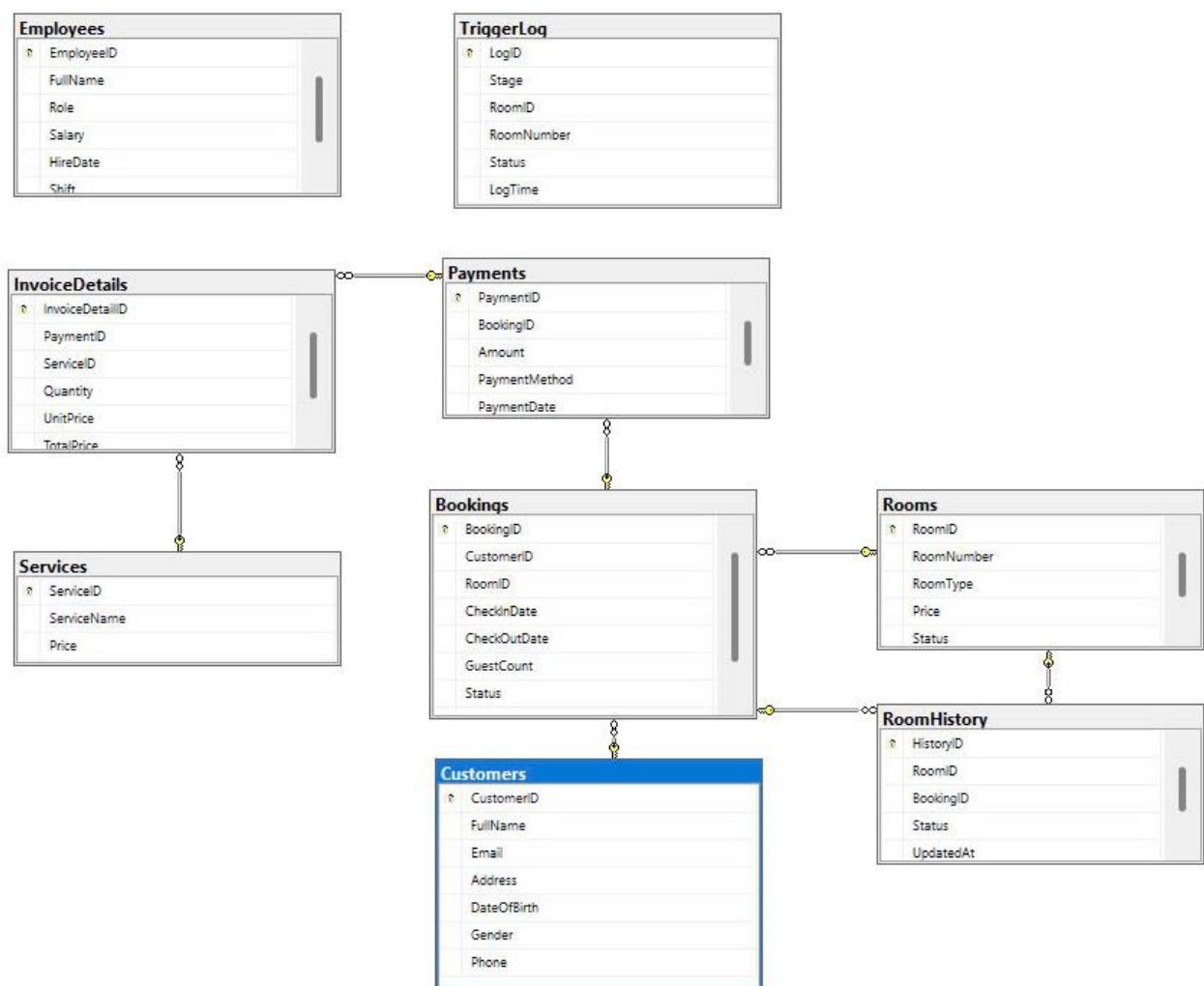
➤ **Tối ưu hiệu suất truy vấn:**

- Sử dụng chỉ mục để tăng tốc các truy vấn phổ biến.
- Tránh sử dụng SELECT * mà chỉ lấy các cột cần thiết.

➤ **Bảo mật và phân quyền:**

- Phân quyền truy cập dữ liệu, đảm bảo chỉ nhân viên có quyền mới có thể sửa đổi hoặc xóa dữ liệu nhạy cảm.
- Mã hóa dữ liệu quan trọng nếu cần.

2.5. Mô hình cơ sở dữ liệu



Hình 2.1. Mô hình cơ sở dữ liệu

CHƯƠNG 3. TẠO CƠ SỞ DỮ LIỆU

3.1. Tạo Database

```
-- Tạo cơ sở dữ liệu
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'HotelManagement')
BEGIN
    CREATE DATABASE HotelManagement;
    PRINT 'Database HotelManagement đã được tạo.';
END
ELSE
BEGIN
    PRINT 'Database HotelManagement đã tồn tại.';
END
GO
```

Hình 3.1. Tạo Database HotelManagement.

3.2. Tạo các bảng dữ liệu

Bảng Khách hàng (Customers)

```
CREATE TABLE Customers (
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    FullName NVARCHAR(100) NOT NULL,
    Phone NVARCHAR(15) UNIQUE NOT NULL,
    Email NVARCHAR(100) UNIQUE,
    Address NVARCHAR(255),
    DateOfBirth DATE,
    Gender NCHAR(1) CHECK (Gender IN (N'M', N'F', N'O'))
);
```

Hình 3.2.1. Tạo bảng Customers.

Bảng Phòng (Rooms)

```
CREATE TABLE Rooms (
    RoomID INT IDENTITY(1,1) PRIMARY KEY,
    RoomNumber NVARCHAR(10) UNIQUE NOT NULL,
    RoomType NVARCHAR(50) NOT NULL,
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
    Status NVARCHAR(20) DEFAULT N'Available'
);
```

Hình 3.2.2. Tạo bảng Rooms.

Bảng Dịch vụ (Services)

```
CREATE TABLE Services (  
    ServiceID INT IDENTITY(1,1) PRIMARY KEY,  
    ServiceName NVARCHAR(100) NOT NULL,  
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0)  
);
```

Hình 3.2.3. Tạo bảng Services.

Bảng Đặt phòng (Bookings)

```
CREATE TABLE Bookings (  
    BookingID INT IDENTITY(1,1) PRIMARY KEY,  
    CustomerID INT NOT NULL,  
    RoomID INT NOT NULL,  
    CheckInDate DATETIME NOT NULL,  
    CheckOutDate DATETIME NOT NULL,  
    GuestCount INT NOT NULL CHECK (GuestCount > 0),  
    Status NVARCHAR(20) DEFAULT N'Pending' CHECK (Status IN (N'Pending', N'Checked-In', N'Checked-Out', N'Cancelled')),  
    CreatedAt DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE,  
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID) ON DELETE CASCADE  
);
```

Hình 3.2.4. Tạo bảng Bookings.

Bảng Nhân viên (Employees)

```
CREATE TABLE Employees (  
    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,  
    FullName NVARCHAR(100) NOT NULL,  
    Role NVARCHAR(50) NOT NULL,  
    Salary DECIMAL(10,2) CHECK (Salary > 0),  
    HireDate DATE DEFAULT GETDATE(),  
    Shift NVARCHAR(20) -- Thêm cột ca làm việc  
);
```

Hình 3.2.5. Tạo bảng Employees.

Bảng Thanh toán (Payments)

```
CREATE TABLE Payments (  
    PaymentID INT IDENTITY(1,1) PRIMARY KEY,  
    BookingID INT NOT NULL,  
    Amount DECIMAL(10,2) NOT NULL CHECK (Amount > 0),  
    PaymentMethod NVARCHAR(50) CHECK (PaymentMethod IN (N'Cash', N'Credit Card', N'Debit Card', N'Online')),  
    PaymentDate DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID) ON DELETE CASCADE  
);
```

Hình 3.2.6. Tạo bảng Payments.

Bảng Chi tiết hóa đơn (InvoiceDetails)

```
CREATE TABLE InvoiceDetails (  
    InvoiceDetailID INT IDENTITY(1,1) PRIMARY KEY,  
    PaymentID INT NOT NULL,  
    ServiceID INT NOT NULL,  
    Quantity INT NOT NULL CHECK (Quantity > 0),  
    UnitPrice DECIMAL(10,2) NOT NULL,  
    TotalPrice AS (Quantity * UnitPrice),  
    FOREIGN KEY (PaymentID) REFERENCES Payments(PaymentID) ON DELETE CASCADE,  
    FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID) ON DELETE CASCADE  
);
```

Hình 3.2.7. Tạo bảng InvoiceDetails.

Bảng Lịch sử Phòng đã dùng (RoomHistory)

```
CREATE TABLE RoomHistory (  
    HistoryID INT IDENTITY(1,1) PRIMARY KEY,  
    RoomID INT NOT NULL,  
    BookingID INT NULL,  
    Status NVARCHAR(50) NOT NULL,  
    UpdatedAt DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID) ON DELETE CASCADE,  
    FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID) ON DELETE NO ACTION  
);  
GO
```

Hình 3.2.8. Tạo bảng RoomHistory.

3.3. Thêm dữ liệu vào cơ sở dữ liệu

Sau khi tạo các bảng, cần thêm dữ liệu mẫu để kiểm tra tính chính xác của hệ thống. Dưới đây là các lệnh SQL để thêm dữ liệu vào bảng **Customers**, **Rooms**, **Services**, và **Employees**.

-- Customers (13 người)

```
INSERT INTO Customers (FullName, Phone, Email, Address, DateOfBirth, Gender) VALUES  
(N'Nguyễn Văn A', N'0987654321', N'nguyenvana@example.com', N'Hà Nội', '1990-05-12', N'M'),  
(N'Trần Thị B', N'0976543210', N'tranthib@example.com', N'TP.HCM', '1995-09-23', N'F'),  
(N'Lê Minh C', N'0912345678', N'leminhc@example.com', N'Đà Nẵng', '1985-03-15', N'M'),  
(N'Phạm Thị D', N'0934567890', N'phamthid@example.com', N'Hải Phòng', '1992-11-30', N'F'),  
(N'Hoàng Văn E', N'0901234567', N'hoangvane@example.com', N'Cần Thơ', '1988-07-25', N'M'),  
(N'Vũ Thị F', N'0945678901', N'vuthif@example.com', N'Quảng Ninh', '1997-02-14', N'F'),  
(N'Đặng Văn G', N'0967890123', N'dangvang@example.com', N'Nha Trang', '1993-08-09', N'M'),  
(N'Bùi Thị H', N'0923456789', N'buithih@example.com', N'Huế', '1989-12-01', N'F'),  
(N'Trương Văn I', N'0956789012', N'truongvani@example.com', N'Vũng Tàu', '1991-06-18', N'M'),  
(N'Ngô Thị K', N'0998765432', N'ngothik@example.com', N'Bình Dương', '1994-04-22', N'F'),  
(N'Lý Văn L', N'0887654321', N'lyvanl@example.com', N'Long An', '1987-10-05', N'M'),  
(N'Mai Thị M', N'0876543219', N'maithim@example.com', N'Đồng Nai', '1996-01-17', N'F'),  
(N'Đỗ Văn N', N'0865432198', N'dovann@example.com', N'Quy Nhơn', '1990-09-28', N'M');  
GO
```

Hình 3.3.1. Thêm dữ liệu cho bảng Customers.

-- Rooms (6 phòng)

```
INSERT INTO Rooms (RoomNumber, RoomType, Price, Status) VALUES
(N'101', N'Single', 500000, N'Available'),
(N'102', N'Double', 800000, N'Available'),
(N'201', N'Suite', 1500000, N'Available'),
(N'202', N'Double', 800000, N'Available'),
(N'301', N'Single', 500000, N'Available'),
(N'302', N'Suite', 1500000, N'Available');
```

Hình 3.3.2. Thêm dữ liệu cho bảng Rooms.

-- Services (5 dịch vụ)

```
INSERT INTO Services (ServiceName, Price) VALUES
(N'Bữa sáng', 100000),
(N'Giặt là', 50000),
(N'Dịch vụ phòng', 150000),
(N'Đưa đón sân bay', 300000),
(N'Massage', 250000);
GO
```

Hình 3.3.3. Thêm dữ liệu cho bảng Services.

-- Employees (5 người, chia ca)

```
INSERT INTO Employees (FullName, Role, Salary, Shift) VALUES
(N'Nguyễn Văn H', N'Lễ tân', 8000000, N'Sáng'),
(N'Trần Thị I', N'Quản lý', 15000000, N'Chiều'),
(N'Lê Văn K', N'Bảo vệ', 6000000, N'Đêm'),
(N'Phạm Thị L', N'Buồng phòng', 7000000, N'Sáng'),
(N'Hoàng Văn M', N'Lễ tân', 8500000, N'Chiều');
GO
```

Hình 3.3.4. Thêm dữ liệu cho bảng Employees.

-- Bookings (5 phòng)

```
INSERT INTO Bookings (CustomerID, RoomID, CheckInDate, CheckOutDate, GuestCount, Status) VALUES
(1, 1, '2024-03-01 14:00:00', '2024-03-05 12:00:00', 1, N'Checked-In'),
(2, 2, '2024-03-02 15:00:00', '2024-03-06 11:00:00', 2, N'Checked-Out'),
(3, 3, '2024-03-04 13:00:00', '2024-03-07 10:00:00', 1, N'Pending'),
(4, 4, '2024-03-05 12:00:00', '2024-03-08 12:00:00', 3, N'Checked-In'),
(5, 5, '2024-03-06 16:00:00', '2024-03-09 11:00:00', 2, N'Cancelled');
GO
```

Hình 3.3.5. Thêm dữ liệu cho bảng Bookings.

-- Payments (4 phòng thanh toán)

```
INSERT INTO Payments (BookingID, Amount, PaymentMethod, PaymentDate) VALUES
(1, 2000000, N'Credit Card', '2024-03-05 12:00:00'),
(2, 3200000, N'Cash', '2024-03-06 11:00:00'),
(3, 1800000, N'Online', '2024-03-07 10:00:00'),
(4, 2550000, N'Debit Card', '2024-03-08 12:00:00');
GO
```

Hình 3.3.6. Thêm dữ liệu cho bảng Payments.

-- InvoiceDetails (4 hóa đơn)

```
INSERT INTO InvoiceDetails (PaymentID, ServiceID, Quantity, UnitPrice) VALUES
(1, 1, 2, 100000), -- PaymentID = 1, ServiceID = 1 (Bữa sáng)
(2, 2, 3, 50000), -- PaymentID = 2, ServiceID = 2 (Giặt là)
(3, 3, 1, 150000), -- PaymentID = 3, ServiceID = 3 (Dịch vụ phòng)
(4, 4, 1, 300000); -- PaymentID = 4, ServiceID = 4 (Đưa đón sân bay)
GO
```

Hình 3.3.7. Thêm dữ liệu cho bảng InvoiceDetails.

-- RoomHistory (5 phòng đã sử dụng)

```
INSERT INTO RoomHistory (RoomID, BookingID, Status, UpdatedAt) VALUES
(1, 1, N'Occupied', '2024-03-01 14:30:00'),
(2, 2, N'Cleaned', '2024-03-06 12:00:00'),
(3, 3, N'Pending', '2024-03-04 13:15:00'),
(4, 4, N'Occupied', '2024-03-05 12:20:00'),
(5, Null, N'Available', '2024-03-06 16:45:00');
GO
```

Hình 3.3.8. Thêm dữ liệu cho bảng RoomHistory.

3.4. Kiểm tra dữ liệu trong cơ sở dữ liệu

Sau khi thêm dữ liệu, có thể kiểm tra xem dữ liệu đã được chèn đúng hay chưa bằng cách sử dụng các truy vấn sau:

SELECT * FROM Customers;

	CustomerID	FullName	Phone	Email	Address	DateOfBirth	Gender
1	1	Nguyễn Văn A	0987654321	nguyenvana@example.com	Hà Nội	1990-05-12	M
2	2	Trần Thị B	0976543210	tranthib@example.com	TP.HCM	1995-09-23	F
3	3	Lê Minh C	0912345678	leminhc@example.com	Đà Nẵng	1985-03-15	M
4	4	Phạm Thị D	0934567890	phanthid@example.com	Hải Phòng	1992-11-30	F
5	5	Hoàng Văn E	0901234567	hoangvane@example.com	Cần Thơ	1988-07-25	M
6	6	Vũ Thị F	0945678901	vuthif@example.com	Quảng Ninh	1997-02-14	F
7	7	Đặng Văn G	0967890123	dangvang@example.com	Nha Trang	1993-08-09	M
8	8	Bùi Thị H	0923456789	buiythih@example.com	Huế	1989-12-01	F
9	9	Trương Văn I	0956789012	truongvani@example.com	Vũng Tàu	1991-06-18	M
10	10	Ngô Thị K	0998765432	ngothik@example.com	Bình Dương	1994-04-22	F
11	11	Lý Văn L	0887654321	lyvanl@example.com	Long An	1987-10-05	M
12	12	Mai Thị M	0876543219	maithim@example.com	Đồng Nai	1996-01-17	F
13	13	Đỗ Văn N	0865432198	dovann@example.com	Quy Nhơn	1990-09-28	M

Hình 3.4.1. Bảng Customer sau khi được chèn dữ liệu.

SELECT * FROM Rooms;

	RoomID	RoomNumber	RoomType	Price	Status
1	1	101	Single	500000.00	Available
2	2	102	Double	800000.00	Available
3	3	201	Single	550000.00	Available
4	4	202	Double	850000.00	Booked
5	5	301	Suite	1200000.00	Available
6	6	302	Single	600000.00	Occupied

Hình 3.4.2. Bảng Rooms sau khi được chèn dữ liệu.

SELECT * FROM Services;

	ServiceID	ServiceName	Price
1	1	Bữa sáng	100000.00
2	2	Giặt là	50000.00
3	3	Dịch vụ phòng	150000.00
4	4	Đưa đón sân bay	300000.00
5	5	Massage	250000.00

Hình 3.4.3. Bảng Services sau khi được chèn dữ liệu.

SELECT * FROM Employees;

	EmployeeID	FullName	Role	Salary	HireDate	Shift
1	1	Nguyễn Văn H	Lễ tân	8000000.00	2025-03-13	Sáng
2	2	Trần Thị I	Quản lý	15000000.00	2025-03-13	Chiều
3	3	Lê Văn K	Bảo vệ	6000000.00	2025-03-13	Đêm
4	4	Phạm Thị L	Buồng phòng	7000000.00	2025-03-13	Sáng
5	5	Hoàng Văn M	Lễ tân	8500000.00	2025-03-13	Chiều

Hình 3.4.4. Bảng Employees sau khi được chèn dữ liệu.

SELECT * FROM Bookings;

	BookingID	CustomerID	RoomID	CheckInDate	CheckOutDate	GuestCount	Status	CreatedAt
1	1	1	1	2024-03-01 14:00:00.000	2024-03-05 12:00:00.000	1	Checked-In	2025-03-13 22:42:43.880
2	2	2	2	2024-03-02 15:00:00.000	2024-03-06 11:00:00.000	2	Checked-Out	2025-03-13 22:42:43.880
3	3	3	3	2024-03-04 13:00:00.000	2024-03-07 10:00:00.000	1	Pending	2025-03-13 22:42:43.880
4	4	4	4	2024-03-05 12:00:00.000	2024-03-08 12:00:00.000	3	Checked-In	2025-03-13 22:42:43.880
5	5	5	5	2024-03-06 16:00:00.000	2024-03-09 11:00:00.000	2	Cancelled	2025-03-13 22:42:43.880

Hình 3.4.5. Bảng Bookings sau khi được chèn dữ liệu.

SELECT * FROM Payments;

	PaymentID	BookingID	Amount	PaymentMethod	PaymentDate
1	1	1	2000000.00	Credit Card	2024-03-05 12:00:00.000
2	2	2	3200000.00	Cash	2024-03-06 11:00:00.000
3	3	3	1800000.00	Online	2024-03-07 10:00:00.000
4	4	4	2550000.00	Debit Card	2024-03-08 12:00:00.000

Hình 3.4.6. Bảng Payments sau khi được chèn dữ liệu.

SELECT * FROM InvoiceDetails;

	PaymentID	BookingID	Amount	PaymentMethod	PaymentDate
1	1	1	2000000.00	Credit Card	2024-03-05 12:00:00.000
2	2	2	3200000.00	Cash	2024-03-06 11:00:00.000
3	3	3	1800000.00	Online	2024-03-07 10:00:00.000
4	4	4	2550000.00	Debit Card	2024-03-08 12:00:00.000

Hình 3.4.7. bảng InvoiceDetails sau khi được chèn dữ liệu.

SELECT * FROM RoomHistory;

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	1	1	1	Occupied	2024-03-01 14:30:00.000
2	2	2	2	Cleaned	2024-03-06 12:00:00.000
3	3	3	3	Pending	2024-03-04 13:15:00.000
4	4	4	4	Occupied	2024-03-05 12:20:00.000
5	5	5	5	Available	2024-03-06 16:45:00.000

Hình 3.4.8. Bảng RoomHistory sau khi được chèn dữ liệu.

Lệnh trên giúp kiểm tra dữ liệu trong các bảng Customers, Rooms, Services, Employees, Bookings, Payments, InvoiceDetails và RoomHistory để đảm bảo quá trình chèn dữ liệu diễn ra chính xác.

CHƯƠNG 4. XÂY DỰNG CÁC VIEW

4.1. Khung nhìn View trong chương trình quản lý khách sạn

1. Tóm tắt và đơn giản hóa dữ liệu phức tạp

Trong hệ thống quản lý khách sạn, dữ liệu được lưu trữ ở nhiều bảng như: Khách hàng, Đặt phòng, Phòng, Dịch vụ, Thanh toán... Việc truy vấn trực tiếp từ nhiều bảng có thể phức tạp, khó nhớ và dễ nhầm lẫn. View cho phép tạo ra một khung nhìn ảo, tập hợp thông tin cần thiết từ nhiều bảng, giúp người dùng dễ dàng truy vấn mà không cần hiểu rõ cấu trúc bên dưới. Ví dụ: Tạo view tổng hợp thông tin đặt phòng của khách, bao gồm tên khách, số phòng, ngày nhận phòng, ngày trả phòng và tổng tiền dịch vụ.

2. Tăng cường bảo mật dữ liệu

Không phải nhân viên nào cũng cần hoặc nên có quyền xem toàn bộ dữ liệu trong hệ thống. Với view, quản trị viên có thể tạo ra các khung nhìn chỉ hiển thị những cột hoặc dòng dữ liệu cần thiết cho từng bộ phận. Ví dụ: Lễ tân chỉ xem được thông tin đặt phòng, còn kế toán chỉ xem dữ liệu thanh toán. Điều này giúp hạn chế truy cập trực tiếp vào các bảng gốc, tăng cường tính bảo mật.

3. Dễ bảo trì và cập nhật

Khi thay đổi cấu trúc bảng hoặc thêm cột mới, các truy vấn hay báo cáo sử dụng view không cần phải thay đổi quá nhiều. Chỉ cần cập nhật lại view một lần, toàn bộ các báo cáo hay ứng dụng truy xuất qua view sẽ tự động cập nhật theo. Điều này giúp giảm công sức bảo trì và hạn chế sai sót.

4. Hỗ trợ tạo báo cáo nhanh chóng

Trong khách sạn, việc tạo các báo cáo tổng hợp theo ngày, tháng, quý rất phổ biến (báo cáo công suất phòng, doanh thu dịch vụ, thống kê khách lưu trú). Thay vì viết các truy vấn phức tạp nhiều lần, có thể tạo sẵn view để lấy dữ liệu nhanh chóng, phục vụ nhu cầu báo cáo tức thời.

5. Tăng hiệu suất với truy vấn phức tạp

Với các truy vấn tính toán nhiều điều kiện hoặc tham gia nhiều bảng, sử dụng view giúp tối ưu hóa hiệu suất, đặc biệt khi hệ quản trị cơ sở dữ liệu hỗ trợ view ảo (materialized view). Điều này giúp giảm tải khi hệ thống cần xử lý lượng dữ liệu lớn, đặc biệt ở các khách sạn có quy mô lớn.

4.2. Xây dựng các khung nhìn View

-- 1 Hiện thị danh sách các đặt phòng hiện tại: Để nhân viên lễ tân theo dõi danh sách khách đang ở hoặc sắp nhận phòng.

```
DROP VIEW IF EXISTS vw_CurrentBookings;
GO
CREATE VIEW vw_CurrentBookings AS
SELECT b.BookingID, c.FullName AS CustomerName, r.RoomNumber, b.CheckInDate, b.CheckOutDate, b.Status
FROM Bookings b
JOIN Customers c ON b.CustomerID = c.CustomerID
JOIN Rooms r ON b.RoomID = r.RoomID
WHERE b.Status IN (N'Pending', N'Checked-In');
GO
```

Hình 4.1. Tạo khung nhìn view vw_CurrentBookings

	BookingID	CustomerName	RoomNumber	CheckInDate	CheckOutDate	Status
1	1	Nguyễn Văn A	101	2024-03-01 14:00:00.000	2024-03-05 12:00:00.000	Checked-In
2	3	Lê Minh C	201	2024-03-04 13:00:00.000	2024-03-07 10:00:00.000	Pending
3	4	Phạm Thị D	202	2024-03-05 12:00:00.000	2024-03-08 12:00:00.000	Checked-In

Hình 4.2. Kiểm tra dữ liệu khung nhìn vw_CurrentBookings

-- 2 Thống kê doanh thu từ các phòng: Để đánh giá hiệu suất hoạt động của từng loại phòng.

```
DROP VIEW IF EXISTS vw_RoomRevenue;
GO
CREATE VIEW vw_RoomRevenue AS
SELECT r.RoomNumber, r.RoomType, COUNT(b.BookingID) AS BookingCount, SUM(DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price) AS TotalRevenue
FROM Rooms r
LEFT JOIN Bookings b ON r.RoomID = b.RoomID
GROUP BY r.RoomNumber, r.RoomType;
GO
```

Hình 4.3. Tạo khung nhìn view vw_RoomRevenue

	RoomNumber	RoomType	BookingCount	TotalRevenue
1	101	Single	1	2000000.00
2	102	Double	1	3200000.00
3	201	Single	1	1650000.00
4	202	Double	1	2550000.00
5	301	Suite	1	3600000.00
6	302	Single	0	NULL

Hình 4.4. Kiểm tra dữ liệu khung nhìn vw_RoomRevenue

-- 3 Thống kê việc sử dụng dịch vụ: Giúp quản lý khách sạn theo dõi dịch vụ nào đang được khách hàng sử dụng nhiều nhất.

```
DROP VIEW IF EXISTS vw_ServiceUsage;
GO
CREATE VIEW vw_ServiceUsage AS
SELECT s.ServiceName, SUM(id.Quantity) AS TotalQuantity, SUM(id.TotalPrice) AS TotalServiceRevenue
FROM Services s
JOIN InvoiceDetails id ON s.ServiceID = id.ServiceID
GROUP BY s.ServiceName;
GO
```

Hình 4.5. Tạo khung nhìn view vw_ServiceUsage

	ServiceName	TotalQuantity	TotalServiceRevenue
1	Bữa sáng	2	200000.00
2	Dịch vụ phòng	1	150000.00
3	Đưa đón sân bay	1	300000.00
4	Giặt là	3	150000.00

Hình 4.6. Kiểm tra dữ liệu khung nhìn vw_ServiceUsage

-- 4 Hiển thị thông tin về nhân viên: Để quản lý lịch làm việc của nhân viên.

```
DROP VIEW IF EXISTS vw_EmployeeShifts;
GO
CREATE VIEW vw_EmployeeShifts AS
SELECT EmployeeID, FullName, Role, Salary, shift
FROM Employees;
GO
```

Hình 4.7. Tạo khung nhìn view vw_EmployeeShifts

	EmployeeID	FullName	Role	Salary	Shift
1	1	Nguyễn Văn H	Lễ tân	8000000.00	Sáng
2	2	Trần Thị I	Quản lý	15000000.00	Chiều
3	3	Lê Văn K	Bảo vệ	6000000.00	Đêm
4	4	Phạm Thị L	Buồng phòng	7000000.00	Sáng
5	5	Hoàng Văn M	Lễ tân	8500000.00	Chiều

Hình 4.8. Kiểm tra dữ liệu khung nhìn vw_EmployeeShifts

-- 5 Hiển thị lịch sử sử dụng phòng trong vòng 1 tháng gần đây: Để theo dõi lịch sử thay đổi trạng thái của phòng (ví dụ: phòng nào đã được dọn dẹp, đã được đặt, v.v.).

```
DROP VIEW IF EXISTS vw_RecentRoomHistory;
GO
CREATE VIEW vw_RecentRoomHistory AS
SELECT rh.HistoryID, r.RoomNumber, b.BookingID, rh.Status, rh.UpdatedAt
FROM RoomHistory rh
JOIN Rooms r ON rh.RoomID = r.RoomID
JOIN Bookings b ON rh.BookingID = b.BookingID
WHERE rh.UpdatedAt >= DATEADD(MONTH, -1, GETDATE());
GO
```

Hình 4.9. Tạo khung nhìn view vw_RecentRoomHistory

HistoryID	RoomNumber	BookingID	Status	UpdatedAt
-----------	------------	-----------	--------	-----------

Hình 4.10. Kiểm tra dữ liệu khung nhìn vw_RecentRoomHistory

-- 6 Hiển thị lịch sử đặt phòng: Hiển thị lịch sử đặt phòng của từng khách hàng, bao gồm thông tin về số phòng, ngày nhận/trả phòng, trạng thái.

```
DROP VIEW IF EXISTS vw_CustomerBookingHistory;
GO
CREATE VIEW vw_CustomerBookingHistory AS
SELECT
    c.CustomerID,
    c.FullName AS CustomerName,
    b.BookingID,
    r.RoomNumber,
    b.CheckInDate,
    b.CheckOutDate,
    b.Status
FROM Bookings b
JOIN Customers c ON b.CustomerID = c.CustomerID
JOIN Rooms r ON b.RoomID = r.RoomID;
GO
```

Hình 4.11. Tạo khung nhìn view vw_CustomerBookingHistory

	CustomerID	CustomerName	BookingID	RoomNumber	CheckInDate	CheckOutDate	Status
1	1	Nguyễn Văn A	1	101	2024-03-01 14:00:00.000	2024-03-05 12:00:00.000	Checked-In
2	2	Trần Thị B	2	102	2024-03-02 15:00:00.000	2024-03-06 11:00:00.000	Checked-Out
3	3	Lê Minh C	3	201	2024-03-04 13:00:00.000	2024-03-07 10:00:00.000	Pending
4	4	Phạm Thị D	4	202	2024-03-05 12:00:00.000	2024-03-08 12:00:00.000	Checked-In
5	5	Hoàng Văn E	5	301	2024-03-06 16:00:00.000	2024-03-09 11:00:00.000	Cancelled

Hình 4.12. Kiểm tra dữ liệu khung nhìn vw_CustomerBookingHistory

-- 7 Thống kê các dịch vụ được sử dụng nhiều nhất: Thống kê các dịch vụ được sử dụng nhiều nhất, bao gồm tổng số lần sử dụng và doanh thu từ dịch vụ đó.

```
DROP VIEW IF EXISTS vw_TopServicesUsed;
GO
CREATE VIEW vw_TopServicesUsed AS
SELECT
    s.ServiceID,
    s.ServiceName,
    COUNT(id.InvoiceDetailID) AS UsageCount,
    SUM(id.TotalPrice) AS TotalRevenue
FROM InvoiceDetails id
JOIN Services s ON id.ServiceID = s.ServiceID
GROUP BY s.ServiceID, s.ServiceName
GO
```

Hình 4.13. Tạo khung nhìn view vw_TopServicesUsed

	ServiceID	ServiceName	UsageCount	TotalRevenue
1	1	Bữa sáng	1	200000.00
2	2	Giặt là	1	150000.00
3	3	Dịch vụ phòng	1	150000.00
4	4	Đưa đón sân bay	1	300000.00

Hình 4.14. Kiểm tra dữ liệu khung nhìn vw_TopServicesUsed

-- 8 Danh sách các phòng đặt chưa thanh toán đầy đủ: Danh sách các đặt phòng chưa thanh toán đầy đủ, giúp theo dõi và quản lý công nợ của khách hàng.

```
DROP VIEW IF EXISTS vw_UnpaidBookings;
GO
CREATE VIEW vw_UnpaidBookings AS
SELECT
    b.BookingID,
    c.FullName AS CustomerName,
    r.RoomNumber,
    SUM(p.Amount) AS TotalPaid,
    DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price AS TotalCost,
    (DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price) - SUM(p.Amount) AS AmountDue
FROM Bookings b
JOIN Customers c ON b.CustomerID = c.CustomerID
JOIN Rooms r ON b.RoomID = r.RoomID
LEFT JOIN Payments p ON b.BookingID = p.BookingID
GROUP BY b.BookingID, c.FullName, r.RoomNumber, b.CheckInDate, b.CheckOutDate, r.Price
HAVING (DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price) > SUM(p.Amount);
GO
```

Hình 4.15. Tạo khung nhìn view vw_UnpaidBookings

BookingID	CustomerName	RoomNumber	TotalPaid	TotalCost	AmountDue
-----------	--------------	------------	-----------	-----------	-----------

Hình 4.16. Kiểm tra dữ liệu khung nhìn vw_UnpaidBookings

-- 9 Báo cáo lương của nhân viên: Báo cáo lương của nhân viên, giúp quản lý theo dõi tổng chi phí nhân sự hàng tháng.

```
DROP VIEW IF EXISTS vw_EmployeeSalaryReport;
GO
CREATE VIEW vw_EmployeeSalaryReport AS
SELECT
    e.EmployeeID,
    e.FullName,
    e.Role,
    e.Shift,
    e.Salary,
    e.Salary * 12 AS AnnualSalary
FROM Employees e;
GO
```

Hình 4.17. Tạo khung nhìn view vw_EmployeeSalaryReport

	EmployeeID	FullName	Role	Shift	Salary	AnnualSalary
1	1	Nguyễn Văn H	Lễ tân	Sáng	8000000.00	96000000.00
2	2	Trần Thị I	Quản lý	Chiều	15000000.00	180000000.00
3	3	Lê Văn K	Bảo vệ	Đêm	6000000.00	72000000.00
4	4	Phạm Thị L	Buồng phòng	Sáng	7000000.00	84000000.00
5	5	Hoàng Văn M	Lễ tân	Chiều	8500000.00	102000000.00

Hình 4.18. Kiểm tra dữ liệu khung nhìn vw_EmployeeSalaryReport

-- 10 Danh sách các phòng còn trống: Danh sách các phòng còn trống để đặt phòng.

```

DROP VIEW IF EXISTS vw_AvailableRooms;
GO
CREATE VIEW vw_AvailableRooms AS
SELECT
    r.RoomID,
    r.RoomNumber,
    r.RoomType,
    r.Price,
    r.Status
FROM Rooms r
WHERE r.Status = N'Available';
GO

```

Hình 4.19. Tạo khung nhìn view vw_AvailableRooms

	RoomID	RoomNumber	RoomType	Price	Status
1	1	101	Single	500000.00	Available
2	2	102	Double	800000.00	Available
3	3	201	Single	550000.00	Available
4	5	301	Suite	1200000.00	Available

Hình 4.20. Kiểm tra dữ liệu khung nhìn vw_AvailableRooms

Những VIEW này giúp cải thiện hiệu suất truy vấn và hỗ trợ quản lý khách sạn dễ dàng hơn.

CHƯƠNG 5. XÂY DỰNG CÁC PROCEDURE

5.1. Thủ tục lưu trữ Procedure trong chương trình quản lý khách sạn

1. Tự động hóa quy trình nghiệp vụ

Xử lý đặt phòng: Procedure thực hiện kiểm tra phòng trống, ghi nhận thông tin đặt phòng và cập nhật trạng thái phòng chỉ trong một lần gọi.

Thanh toán hóa đơn: Procedure có thể tính tổng chi phí, bao gồm tiền phòng, dịch vụ và thuế, sau đó ghi nhận thanh toán.

2. Tăng tính nhất quán

Mọi thao tác nghiệp vụ phức tạp đều được đóng gói trong Procedure, đảm bảo mọi nơi trong hệ thống đều dùng chung logic, tránh sai lệch khi xử lý thủ công.

3. Tăng hiệu suất

Thực thi nhanh hơn: Procedure đã được biên dịch sẵn (pre-compiled) nên chạy nhanh hơn so với các truy vấn SQL thông thường.

Giảm tải cho ứng dụng: Toàn bộ logic xử lý được thực hiện ở tầng CSDL, ứng dụng chỉ cần gọi và nhận kết quả.

4. Tăng bảo mật

Người dùng không cần quyền truy cập trực tiếp vào các bảng dữ liệu, chỉ cần quyền thực thi Procedure, giúp bảo vệ dữ liệu nhạy cảm (doanh thu, lương nhân viên).

5.2 Xây dựng các Procedure

-- 1 Thêm đặt phòng

Chức năng:

Thêm một đặt phòng mới vào bảng Bookings.

```
DROP PROCEDURE IF EXISTS sp_AddBooking;
GO
CREATE PROCEDURE sp_AddBooking
    @CustomerID INT, @RoomID INT, @CheckInDate DATETIME, @CheckOutDate DATETIME, @GuestCount INT
AS
BEGIN
    INSERT INTO Bookings (CustomerID, RoomID, CheckInDate, CheckOutDate, GuestCount, Status)
    VALUES (@CustomerID, @RoomID, @CheckInDate, @CheckOutDate, @GuestCount, N'Pending');
    SELECT SCOPE_IDENTITY() AS NewBookingID;
    UPDATE Rooms SET Status = N'Booked' WHERE RoomID = @RoomID;
END;
GO

-- Test sp_AddBooking
EXEC sp_AddBooking 3, 5, '2024-04-01 14:00:00', '2024-04-05 12:00:00', 2;
GO
```

hình 5.1. Tạo procedure sp_AddBooking

6	6	3	5	2024-04-01 14:00:00.000	2024-04-05 12:00:00.000	2	Pending	2025-03-14 00:24:13.950
---	---	---	---	-------------------------	-------------------------	---	---------	-------------------------

Hình 5.2. Kiểm tra dữ liệu procedure sp_AddBooking

-- 2 Thêm tính hóa đơn

Chức năng:

Tính tổng tiền của hóa đơn dựa trên BookingID.

```
DROP PROCEDURE IF EXISTS sp_CalculateInvoice;
GO
CREATE PROCEDURE sp_CalculateInvoice
    @BookingID INT
AS
BEGIN
    SELECT b.BookingID, SUM(DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price) AS RoomCost,
           COALESCE(SUM(id.TotalPrice), 0) AS ServiceCost,
           SUM(DATEDIFF(DAY, b.CheckInDate, b.CheckOutDate) * r.Price) + COALESCE(SUM(id.TotalPrice), 0) AS TotalCost
    FROM Bookings b
    JOIN Rooms r ON b.RoomID = r.RoomID
    LEFT JOIN Payments p ON b.BookingID = p.BookingID
    LEFT JOIN InvoiceDetails id ON p.PaymentID = id.PaymentID
    WHERE b.BookingID = @BookingID
    GROUP BY b.BookingID, b.CheckInDate, b.CheckOutDate, r.Price;
END;
GO
```

Hình 5.3. Tạo procedure *sp_CalculateInvoice*

	BookingID	RoomCost	ServiceCost	TotalCost
1	1	2000000.00	200000.00	2200000.00

Hình 5.4. Kiểm tra dữ liệu procedure *sp_CalculateInvoice*

-- 3 Thêm phương thức thanh toán

Chức năng:

Thêm thông tin thanh toán mới vào bảng Payments dựa trên BookingID, số tiền và phương thức thanh toán.

```
DROP PROCEDURE IF EXISTS sp_ProcessPayment;
GO
CREATE PROCEDURE sp_ProcessPayment
    @BookingID INT, @Amount DECIMAL(10,2), @PaymentMethod NVARCHAR(50)
AS
BEGIN
    INSERT INTO Payments (BookingID, Amount, PaymentMethod, PaymentDate)
    VALUES (@BookingID, @Amount, @PaymentMethod, GETDATE());
    SELECT SCOPE_IDENTITY() AS NewPaymentID;
END;
GO
```

Hình 5.5. Tạo procedure *sp_ProcessPayment*

5	5	1	2000000.00	Credit Card	2025-03-14 00:31:46.470
---	---	---	------------	-------------	-------------------------

Hình 5.6. Kiểm tra procedure *sp_ProcessPayment*

-- 4 Thêm khách hàng mới

Chức năng:

Thêm một khách hàng mới vào bảng Customers với các thông tin như họ tên, số điện thoại, email, địa chỉ, ngày sinh và giới tính.

```
DROP PROCEDURE IF EXISTS sp_AddCustomer;
GO
CREATE PROCEDURE sp_AddCustomer
    @FullName NVARCHAR(100), @Phone NVARCHAR(15), @Email NVARCHAR(100), @Address NVARCHAR(255), @DateOfBirth DATE, @Gender NCHAR(1)
AS
BEGIN
    INSERT INTO Customers (FullName, Phone, Email, Address, DateOfBirth, Gender)
    VALUES (@FullName, @Phone, @Email, @Address, @DateOfBirth, @Gender);
    SELECT SCOPE_IDENTITY() AS NewCustomerID;
END;
GO
```

Hình 5.7. Tạo procedure sp_AddCustomer

14	14	Trần Văn B	0981234567	tranvanb@example.com	Hà Nội	1990-06-15	M
----	----	------------	------------	----------------------	--------	------------	---

Hình 5.8. Kiểm tra procedure sp_AddCustomer

-- 5 Cập nhật thông tin khách hàng

Chức năng:

Cập nhật thông tin khách hàng dựa trên CustomerID.

```
DROP PROCEDURE IF EXISTS sp_UpdateCustomer;
GO
CREATE PROCEDURE sp_UpdateCustomer
    @CustomerID INT, @FullName NVARCHAR(100), @Phone NVARCHAR(15), @Email NVARCHAR(100), @Address NVARCHAR(255), @DateOfBirth DATE, @Gender NCHAR(1)
AS
BEGIN
    UPDATE Customers
    SET FullName = @FullName, Phone = @Phone, Email = @Email, Address = @Address, DateOfBirth = @DateOfBirth, Gender = @Gender
    WHERE CustomerID = @CustomerID;
END;
GO
```

Hình 5.9. Tạo procedure sp_UpdateCustomer

1	1	Nguyễn Văn A Updated	0987654321	nguyenvanaupdated@example.com	Hà Nội	1990-05-12	M
---	---	----------------------	------------	-------------------------------	--------	------------	---

Hình 5.10. Kiểm tra procedure sp_UpdateCustomer

-- 6 Xóa khách hàng

Chức năng:

Xóa khách hàng khỏi hệ thống dựa trên CustomerID.

Lưu ý: Nếu khách hàng đã có dữ liệu liên quan trong Bookings hoặc Payments, có thể bị lỗi ràng buộc khóa ngoại.

```

DROP PROCEDURE IF EXISTS sp_DeleteCustomer;
GO
CREATE PROCEDURE sp_DeleteCustomer
    @CustomerID INT
AS
BEGIN
    DELETE FROM Customers WHERE CustomerID = @CustomerID;
END;
GO

```

Hình 5.11. Tạo procedure *sp_DeleteCustomer*

(1 row affected)

Completion time: 2025-03-14T00:46:33.6399123+07:00

Hình 5.12. Kiểm tra procedure *sp_DeleteCustomer*

-- 7 Thêm dịch vụ mới

Chức năng:

Thêm một dịch vụ mới vào bảng Services.

```

DROP PROCEDURE IF EXISTS sp_AddService;
GO
CREATE PROCEDURE sp_AddService
    @ServiceName NVARCHAR(100), @Price DECIMAL(10,2)
AS
BEGIN
    INSERT INTO Services (ServiceName, Price)
    VALUES (@ServiceName, @Price);
    SELECT SCOPE_IDENTITY() AS NewServiceID;
END;
GO

```

Hình 5.13. Tạo procedure *sp_AddService*

	ServiceID	ServiceName	Price
1	1	Bữa sáng	100000.00
2	2	Giặt là	50000.00
3	3	Dịch vụ phòng	150000.00
4	4	Đưa đón sân bay	300000.00
5	5	Massage	250000.00
6	6	Spa	500000.00

Hình 5.14. Kiểm tra procedure *sp_AddService*

-- 8 Cập nhật thông tin dịch vụ

Chức năng:

Cập nhật tên và giá của dịch vụ dựa trên ServiceID.

```
DROP PROCEDURE IF EXISTS sp_UpdateService;
GO
CREATE PROCEDURE sp_UpdateService
    @ServiceID INT, @ServiceName NVARCHAR(100), @Price DECIMAL(10,2)
AS
BEGIN
    UPDATE Services
    SET ServiceName = @ServiceName, Price = @Price
    WHERE ServiceID = @ServiceID;
END;
GO
```

Hình 5.15. Tạo procedure sp_UpdateService

1	1	Luxury Spa	600000.00
---	---	------------	-----------

Hình 5.16. Kiểm tra procedure sp_UpdateService

-- 9 Xóa dịch vụ

Chức năng:

Xóa một dịch vụ khỏi bảng Services dựa trên ServiceID.

```
DROP PROCEDURE IF EXISTS sp_DeleteService;
GO
CREATE PROCEDURE sp_DeleteService
    @ServiceID INT
AS
BEGIN
    DELETE FROM Services WHERE ServiceID = @ServiceID;
END;
GO
```

Hình 5.17. Tạo procedure sp_DeleteService

(1 row affected)

Completion time: 2025-03-14T11:44:22.0354941+07:00

Hình 5.18. Kiểm tra procedure sp_DeleteService

-- 10 Lấy danh sách các phòng trống

Chức năng:

Truy vấn danh sách các phòng có trạng thái Available.

```
DROP PROCEDURE IF EXISTS sp_GetAvailableRooms;
GO
CREATE PROCEDURE sp_GetAvailableRooms
AS
BEGIN
    SELECT RoomID, RoomNumber, RoomType, Price, Status
    FROM Rooms
    WHERE Status = N'Available';
END;
GO
```

Hình 5.19. Tạo procedure sp_GetAvailableRooms

	RoomID	RoomNumber	RoomType	Price	Status
1	1	101	Single	500000.00	Available
2	2	102	Double	800000.00	Available
3	3	201	Single	550000.00	Available

Hình 5.20. Kiểm tra procedure sp_GetAvailableRooms

CHƯƠNG 6. XÂY DỰNG CÁC TRIGGER

6.1. Thủ tục đặt biệt Trigger trong chương trình quản lý khách sạn

1. Tự động cập nhật trạng thái phòng (Room Status)

Khi có khách nhận phòng (Check-in), Trigger tự động đổi trạng thái phòng từ "Available" thành "Occupied".

Khi khách trả phòng (Check-out), Trigger tự động đổi trạng thái phòng thành "Available" hoặc "Cleaning".

Điều này giúp đảm bảo trạng thái phòng luôn đúng, giảm sai sót khi nhân viên quên cập nhật.

2. Ghi nhận lịch sử sử dụng phòng (Room History)

Khi có khách đặt phòng, nhận phòng hoặc hủy phòng, Trigger có thể tự động thêm bản ghi vào bảng RoomHistory để theo dõi toàn bộ lịch sử thay đổi trạng thái của từng phòng.

Điều này giúp quản lý dễ dàng tra cứu và kiểm tra lịch sử sử dụng phòng khi cần.

3. Tự động tính toán hoặc kiểm tra dữ liệu

Khi thực hiện thanh toán, Trigger có thể tự động kiểm tra tổng số tiền có đúng với chi tiết dịch vụ và tiền phòng hay không.

Khi thêm dịch vụ mới vào hóa đơn, Trigger có thể **cập nhật tổng tiền** trong hóa đơn chính (Invoice).

4. Đảm bảo tính toàn vẹn dữ liệu (Data Integrity)

Trigger giúp kiểm soát ràng buộc nghiệp vụ mà hệ quản trị không hỗ trợ trực tiếp. Ví dụ: Không cho phép xóa nhân viên nếu nhân viên đó đang phụ trách các booking chưa hoàn thành, hoặc không cho phép đặt phòng nếu ngày nhận phòng trùng với thời gian đang có khách ở.

5. Tự động thông báo hoặc ghi log sự kiện

Trigger có thể ghi lại lịch sử thao tác của nhân viên, như ai cập nhật trạng thái phòng, ai hủy booking...

Những bản ghi này rất hữu ích khi kiểm tra log hoặc phục vụ công tác kiểm tra nội bộ (Audit).

6.2. Xây dựng các Trigger

-- 1 Chèn thêm Booking

```
DROP TRIGGER IF EXISTS tr_BookingInsert;
GO
CREATE OR ALTER TRIGGER tr_BookingInsert
ON Bookings AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO RoomHistory (RoomID, BookingID, Status, UpdatedAt)
    SELECT i.RoomID, i.BookingID, N'Booked', GETDATE()
    FROM inserted i
    WHERE EXISTS (SELECT 1 FROM Rooms r WHERE r.RoomID = i.RoomID) -- Kiểm tra xem RoomID có tồn tại
    AND NOT EXISTS (SELECT 1 FROM RoomHistory rh WHERE rh.BookingID = i.BookingID);

    SELECT * FROM RoomHistory WHERE BookingID IN (SELECT BookingID FROM inserted);
END;
GO
```

Hình 6.1. Tạo trigger tr_BookingInsert

	BookingID	CustomerID	RoomID	CheckInDate	CheckOutDate	GuestCount	Status	CreatedAt
1	1	1	1	2024-03-01 14:00:00.000	2024-03-05 12:00:00.000	1	Checked-In	2025-03-14 11:21:04.327
2	2	2	2	2024-03-02 15:00:00.000	2024-03-06 11:00:00.000	2	Checked-Out	2025-03-14 11:21:04.327
3	3	3	3	2024-03-04 13:00:00.000	2024-03-07 10:00:00.000	1	Pending	2025-03-14 11:21:04.327
4	4	4	4	2024-03-05 12:00:00.000	2024-03-08 12:00:00.000	3	Checked-In	2025-03-14 11:21:04.327
5	5	5	5	2024-03-06 16:00:00.000	2024-03-09 11:00:00.000	2	Cancelled	2025-03-14 11:21:04.327
6	6	3	5	2024-04-01 14:00:00.000	2024-04-05 12:00:00.000	2	Pending	2025-03-14 11:22:16.783
7	7	2	2	2025-03-09 00:00:00.000	2025-03-11 00:00:00.000	3	Pending	2025-03-14 13:29:57.123

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	1	1	1	Occupied	2024-03-01 14:30:00.000
2	2	2	2	Cleaned	2024-03-06 12:00:00.000
3	3	3	3	Pending	2024-03-04 13:15:00.000
4	4	4	4	Occupied	2024-03-05 12:20:00.000
5	5	5	5	Available	2024-03-06 16:45:00.000
6	6	2	7	Booked	2025-03-14 13:29:57.127

	CustomerID	FullName	Phone	Email	Address	DateOfBirth	Gender
1	2	Trần Thị B	0976543210	tranthib@example.com	TP.HCM	1995-09-23	F

Hình 6.2. Kiểm tra trigger tr_BookingInsert

-- 2 Cập nhật trạng thái Booking

```
DROP TRIGGER IF EXISTS tr_BookingStatusUpdate;
GO
CREATE OR ALTER TRIGGER tr_BookingStatusUpdate
ON Bookings AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    IF UPDATE(Status)
    BEGIN
        UPDATE Rooms
        SET Status = CASE
            WHEN i.Status = N'Checked-In' THEN N'Occupied'
            WHEN i.Status IN (N'Checked-Out', N'Cancelled') THEN N'Available'
            ELSE N'Booked'
        END
        FROM Rooms r
        JOIN inserted i ON r.RoomID = i.RoomID
        WHERE EXISTS (SELECT 1 FROM Rooms WHERE RoomID = i.RoomID); -- Kiểm tra tồn tại

        SELECT r.RoomNumber, r.Status, i.Status AS BookingStatus
        FROM Rooms r
        JOIN inserted i ON r.RoomID = i.RoomID;
    END
END;
GO
```

Hình 6.3. Tạo trigger tr_BookingStatusUpdate

	RoomNumber	Status	BookingStatus				
1	102	Occupied	Checked-In				

	RoomID	RoomNumber	RoomType	Price	Status		
1	2	102	Double	800000.00	Occupied		

	CustomerID	FullName	Phone	Email	Address	DateOfBirth	Gender
1	2	Trần Thị B	0976543210	tranthib@example.com	TP.HCM	1995-09-23	F

	RoomNumber	Status	BookingStatus				
1	102	Available	Checked-Out				

	RoomID	RoomNumber	RoomType	Price	Status		
1	2	102	Double	800000.00	Available		

	RoomNumber	Status	BookingStatus				
1	102	Available	Cancelled				

	RoomID	RoomNumber	RoomType	Price	Status		
1	2	102	Double	800000.00	Available		

Hình 6.4. Kiểm tra trigger tr_BookingStatusUpdate

-- 3 Cập nhật trạng thái Room

```

DROP TRIGGER IF EXISTS tr_RoomStatusUpdate;
GO
CREATE OR ALTER TRIGGER tr_RoomStatusUpdate
ON Rooms AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    IF UPDATE(Status)
    BEGIN
        INSERT INTO RoomHistory (RoomID, BookingID, Status, UpdatedAt)
        SELECT i.RoomID, COALESCE(b.BookingID, 0), i.Status, GETDATE()
        FROM inserted i
        LEFT JOIN Bookings b
        ON i.RoomID = b.RoomID AND b.Status IN (N'Pending', N'Checked-In');

        SELECT * FROM RoomHistory WHERE RoomID IN (SELECT RoomID FROM inserted);
    END
END;
GO

```

Hình 6.5. Tạo trigger tr_RoomStatusUpdate

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	3	3	3	Pending	2024-03-04 13:15:00.000
2	7	3	3	Occupied	2025-03-14 13:55:01.910

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	3	3	3	Pending	2024-03-04 13:15:00.000
2	7	3	3	Occupied	2025-03-14 13:55:01.910

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	3	3	3	Pending	2024-03-04 13:15:00.000
2	7	3	3	Occupied	2025-03-14 13:55:01.910
3	8	3	3	Available	2025-03-14 13:55:01.920

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	3	3	3	Pending	2024-03-04 13:15:00.000
2	7	3	3	Occupied	2025-03-14 13:55:01.910
3	8	3	3	Available	2025-03-14 13:55:01.920

Hình 6.6. Kiểm tra trigger tr_RoomS

-- 4 Ngăn xóa khách hàng nếu họ có Booking đang hoạt động

```
DROP TRIGGER IF EXISTS tr_CustomerDelete_CheckBookings;
GO
CREATE OR ALTER TRIGGER tr_CustomerDelete_CheckBookings
ON Customers INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Bookings WHERE CustomerID IN (SELECT CustomerID FROM deleted))
    BEGIN
        RAISERROR (N'Không thể xóa khách hàng vì họ có đặt phòng chưa hoàn thành!', 16, 1);
        RETURN;
    END
    DELETE FROM Customers WHERE CustomerID IN (SELECT CustomerID FROM deleted);
END;
GO
```

Hình 6.7. Tạo trigger tr_CustomerDelete_CheckBookings

Không thể xóa khách hàng vì họ có đặt phòng chưa hoàn thành!

(1 row affected)

Completion time: 2025-03-14T14:14:23.4808619+07:00

Hình 6.8. Kiểm tra trigger tr_CustomerDelete_CheckBookings

-- 5 Tự động cập nhật trạng thái Checked-Out nếu ngày trả phòng đã qua

```
DROP TRIGGER IF EXISTS tr_AutoCheckOut_ExpiredBookings;
GO

CREATE OR ALTER TRIGGER tr_AutoCheckOut_ExpiredBookings
ON Bookings AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE Bookings
    SET Status = N'Checked-Out'
    WHERE CheckOutDate < GETDATE() AND Status = N'Checked-In';
END;
GO
```

Hình 6.9. Tạo trigger tr_AutoCheckOut_ExpiredBookings

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	9	3	8	Booked	2025-03-14 14:21:30.850

	BookingID	CustomerID	RoomID	CheckInDate	CheckOutDate	GuestCount	Status	CreatedAt
1	3	3	3	2024-03-04 13:00:00.000	2024-03-07 10:00:00.000	1	Pending	2025-03-14 11:21:04.327
2	8	1	3	2024-03-01 12:00:00.000	2024-03-03 12:00:00.000	2	Checked-In	2025-03-14 14:21:30.847

Hình 6.10. Kiểm tra trigger tr_AutoCheckOut_ExpiredBookings

-- 6 Lưu nhật ký khi cập nhật dịch vụ

```
DROP TRIGGER IF EXISTS trg_LogServiceUpdate;
GO
CREATE TRIGGER trg_LogServiceUpdate
ON Services
AFTER UPDATE
AS
BEGIN
    INSERT INTO ServiceLogs (ServiceID, OldPrice, NewPrice, UpdatedAt)
    SELECT d.ServiceID, d.Price, i.Price, GETDATE()
    FROM deleted d
    JOIN inserted i ON d.ServiceID = i.ServiceID;
END;
GO
```

Hình 6.11. Tạo trigger tr_LogServiceUpdate

	ServiceID	ServiceName	Price
1	1	Luxury Spa	750000.00

Hình 6.12. Kiểm tra trigger tr_LogServiceUpdate

-- 7 Ngăn xóa dịch vụ nếu dịch vụ đã từng được sử dụng trong hóa đơn

```
DROP TRIGGER IF EXISTS tr_Service_Delete_CheckInvoice;
GO
CREATE OR ALTER TRIGGER tr_Service_Delete_CheckInvoice
ON Services INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM InvoiceDetails WHERE ServiceID IN (SELECT ServiceID FROM deleted))
    BEGIN
        RAISERROR (N'Không thể xóa dịch vụ vì nó đã được sử dụng trong hóa đơn!', 16, 1);
        RETURN;
    END
    DELETE FROM Services WHERE ServiceID IN (SELECT ServiceID FROM deleted);
END;
GO
```

Hình 6.13. Tạo trigger tr_Service_Delete_CheckInvoice

Không thể xóa dịch vụ vì nó đã được sử dụng trong hóa đơn!

(1 row affected)

Hình 6.14. Kiểm tra trigger tr_Service_Delete_CheckInvoice

-- 8 Ghi lại lịch sử thay đổi giá phòng vào bảng RoomPriceHistory

```
DROP TRIGGER IF EXISTS tr_UpdateRoomPrice_LogChange;
GO
CREATE TRIGGER tr_UpdateRoomPrice_LogChange
ON Rooms AFTER UPDATE
AS
BEGIN
    IF UPDATE(Price)
    BEGIN
        INSERT INTO RoomPriceHistory (RoomID, OldPrice, NewPrice, ChangedAt)
        SELECT i.RoomID, d.Price, i.Price, GETDATE()
        FROM inserted i
        JOIN deleted d ON i.RoomID = d.RoomID;
    END
END;
GO
```

Hình 6.15. Tạo trigger tr_UpdateRoomPrice_LogChange

	HistoryID	RoomID	OldPrice	NewPrice	ChangedAt
1	1	3	550000.00	800000.00	2025-03-14 14:34:54.313

Hình 6.16. Kiểm tra trigger tr_UpdateRoomPrice_LogChange

-- 9 Tự động cập nhật trạng thái phòng khi đặt phòng

```

DROP TRIGGER IF EXISTS tr_UpdateRoomStatusOnBooking ;
GO
CREATE OR ALTER TRIGGER tr_UpdateRoomStatusOnBooking
ON Bookings
AFTER INSERT
AS
BEGIN
    UPDATE Rooms
    SET Status = N'Booked'
    WHERE RoomID IN (SELECT RoomID FROM inserted);
END;
GO

```

Hình 6.17. Tạo trigger tr_UpdateRoomStatusOnBooking

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	10	5	9	Booked	2025-03-14 14:37:24.710

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	5	5	5	Available	2024-03-06 16:45:00.000
2	10	5	9	Booked	2025-03-14 14:37:24.710
3	11	5	6	Booked	2025-03-14 14:37:24.717
4	12	5	9	Booked	2025-03-14 14:37:24.717

	NewBookingID
1	9

	HistoryID	RoomID	BookingID	Status	UpdatedAt
1	5	5	5	Available	2024-03-06 16:45:00.000
2	10	5	9	Booked	2025-03-14 14:37:24.710
3	11	5	6	Booked	2025-03-14 14:37:24.717
4	12	5	9	Booked	2025-03-14 14:37:24.717
5	13	5	6	Booked	2025-03-14 14:37:24.727
6	14	5	9	Booked	2025-03-14 14:37:24.727

	RoomID	RoomNumber	RoomType	Price	Status
1	5	301	Suite	1200000.00	Booked

Hình 6.18. Kiểm tra trigger tr_UpdateRoomStatusOnBooking

-- 10 Ngăn chặn việc xóa phòng khi đang có khách ở

```
DROP TRIGGER IF EXISTS tr_PreventRoomDelete;
GO
CREATE TRIGGER tr_PreventRoomDelete
ON Rooms
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM deleted WHERE Status = 'Booked')
    BEGIN
        RAISERROR('Không thể xóa phòng đang được đặt!', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        DELETE FROM Rooms WHERE RoomID IN (SELECT RoomID FROM deleted);
    END
END;
GO
```

Hình 6.19. Tạo trigger tr_PreventRoomDelete

(1 row affected)

(1 row affected)

Completion time: 2025-03-14T14:41:15.4679054+07:00

Hình 6.20. Kiểm tra trigger tr_PreventRoomDelete

CHƯƠNG 7. PHÂN QUYỀN VÀ BẢO VỆ CƠ SỞ DỮ LIỆU

7.1. Tạo Login cho Manager và Receptionist

Mục tiêu:

- ✓ Đảm bảo rằng ManagerLogin và ReceptionistLogin được tạo mới trên server.
- ✓ Nếu chúng đã tồn tại, phải xóa trước khi tạo mới.

Giải thích

- sys.server_principals chứa danh sách các logins của SQL Server.
- Kiểm tra nếu ManagerLogin hoặc ReceptionistLogin tồn tại thì xóa (DROP LOGIN).
- Tạo lại (CREATE LOGIN) với mật khẩu cố định.

```
IF EXISTS (SELECT * FROM sys.server_principals WHERE name = 'ManagerLogin')
BEGIN
```

```
    DROP LOGIN ManagerLogin;
    PRINT 'Da xoa login ManagerLogin.';
```

```
END
```

```
CREATE LOGIN ManagerLogin WITH PASSWORD = 'Manager123!';
PRINT 'Da tao login ManagerLogin.';
```

```
IF EXISTS (SELECT * FROM sys.server_principals WHERE name = 'ReceptionistLogin')
BEGIN
```

```
    DROP LOGIN ReceptionistLogin;
    PRINT 'Da xoa login ReceptionistLogin.';
```

```
END
```

```
CREATE LOGIN ReceptionistLogin WITH PASSWORD = 'Reception123!';
PRINT 'Da tao login ReceptionistLogin.';
```

Hình 7.1. Tạo login ManagerLogin và ReceptionistLogin với mật khẩu

7.2. Quản lý vai trò (Role)

7.2.1. Tạo và xóa vai trò HotelManager

Mục tiêu:

- ✓ Đảm bảo rằng vai trò HotelManager luôn tồn tại ở trạng thái sạch.
- ✓ Xóa các thành viên trước khi xóa vai trò.

Giải thích

- Nếu HotelManager tồn tại, tìm tất cả thành viên của nó.
- Sử dụng CURSOR để duyệt và xóa từng thành viên khỏi vai trò.
- Xóa vai trò (DROP ROLE) rồi tạo lại (CREATE ROLE).

```

IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'HotelManager' AND type = 'R')
BEGIN
    IF EXISTS (SELECT * FROM sys.database_role_members rm
                JOIN sys.database_principals dp ON rm.member_principal_id = dp.principal_id
                WHERE rm.role_principal_id = (SELECT principal_id FROM sys.database_principals WHERE name = 'HotelManager'))
    BEGIN
        DECLARE @MemberName NVARCHAR(128);
        DECLARE @SqlCommand NVARCHAR(256);
        DECLARE member_cursor CURSOR FOR
            SELECT dp.name
            FROM sys.database_role_members rm
            JOIN sys.database_principals dp ON rm.member_principal_id = dp.principal_id
            WHERE rm.role_principal_id = (SELECT principal_id FROM sys.database_principals WHERE name = 'HotelManager');
        OPEN member_cursor;
        FETCH NEXT FROM member_cursor INTO @MemberName;
        WHILE @@FETCH_STATUS = 0
        BEGIN
            SET @SqlCommand = N'ALTER ROLE HotelManager DROP MEMBER ' + QUOTENAME(@MemberName);
            EXEC sp_executesql @SqlCommand;
            PRINT 'Da xoa thanh vien ' + @MemberName + ' khoi vai tro HotelManager.';
            FETCH NEXT FROM member_cursor INTO @MemberName;
        END
        CLOSE member_cursor;
        DEALLOCATE member_cursor;
    END
    DROP ROLE HotelManager;
    PRINT 'Da xoa vai tro HotelManager.';
END
CREATE ROLE HotelManager;
PRINT 'Da tao vai tro HotelManager.';

```

Hình 7.2.1. Tạo vai trò HotelManager và xóa nếu đã tồn tại

7.2.2. Tạo và xóa vai trò Receptionist

Tương tự như trên nhưng với Receptionist

```

IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'Receptionist' AND type = 'R')
BEGIN
    IF EXISTS (SELECT * FROM sys.database_role_members rm
                JOIN sys.database_principals dp ON rm.member_principal_id = dp.principal_id
                WHERE rm.role_principal_id = (SELECT principal_id FROM sys.database_principals WHERE name = 'Receptionist'))
    BEGIN
        DECLARE @MemberName2 NVARCHAR(128);
        DECLARE @SqlCommand2 NVARCHAR(256);
        DECLARE member_cursor2 CURSOR FOR
            SELECT dp.name
            FROM sys.database_role_members rm
            JOIN sys.database_principals dp ON rm.member_principal_id = dp.principal_id
            WHERE rm.role_principal_id = (SELECT principal_id FROM sys.database_principals WHERE name = 'Receptionist');
        OPEN member_cursor2;
        FETCH NEXT FROM member_cursor2 INTO @MemberName2;
        WHILE @@FETCH_STATUS = 0
        BEGIN
            SET @SqlCommand2 = N'ALTER ROLE Receptionist DROP MEMBER ' + QUOTENAME(@MemberName2);
            EXEC sp_executesql @SqlCommand2;
            PRINT 'Da xoa thanh vien ' + @MemberName2 + ' khoi vai tro Receptionist.';
            FETCH NEXT FROM member_cursor2 INTO @MemberName2;
        END
        CLOSE member_cursor2;
        DEALLOCATE member_cursor2;
    END
    DROP ROLE Receptionist;
    PRINT 'Da xoa vai tro Receptionist.';
END
CREATE ROLE Receptionist;
PRINT 'Da tao vai tro Receptionist.';

```

Hình 7.2.2. Tạo vai trò Receptionist và xóa nếu đã tồn tại

7.3. Quản lý Users

Mục tiêu: Tạo ManagerUser và ReceptionistUser, gán vào vai trò phù hợp.

Giải thích:

- ✓ Xóa và tạo lại ManagerUser, ReceptionistUser.
- ✓ Gán (ALTER ROLE ... ADD MEMBER) vào vai trò phù hợp.

```
IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'ManagerUser')
BEGIN
    DROP USER ManagerUser;
    PRINT 'Da xoa user ManagerUser.';
END
CREATE USER ManagerUser FOR LOGIN ManagerLogin;
ALTER ROLE HotelManager ADD MEMBER ManagerUser;
PRINT 'Da tao user ManagerUser va them vao vai tro HotelManager.';

IF EXISTS (SELECT * FROM sys.database_principals WHERE name = 'ReceptionistUser')
BEGIN
    DROP USER ReceptionistUser;
    PRINT 'Da xoa user ReceptionistUser.';
END
CREATE USER ReceptionistUser FOR LOGIN ReceptionistLogin;
ALTER ROLE Receptionist ADD MEMBER ReceptionistUser;
PRINT 'Da tao user ReceptionistUser va them vao vai tro Receptionist.';
```

Hình 7.3. Tạo user và gán vào vai trò để quản lý quyền

7.4. Phân quyền

7.4.1. Receptionist

Receptionist có quyền:

- ✓ Đọc, chỉnh sửa khách hàng & đặt phòng.
- ✓ Chỉ được xem danh sách phòng.
- ✓ Thực thi stored procedures (sp_AddBooking, sp_UpdateRoomStatus).

```
GRANT SELECT, INSERT, UPDATE ON Customers TO Receptionist;
GRANT SELECT ON Rooms TO Receptionist;
GRANT SELECT, INSERT, UPDATE ON Bookings TO Receptionist;
GRANT EXECUTE ON sp_AddBooking TO Receptionist;
GRANT EXECUTE ON sp_UpdateRoomStatus TO Receptionist;
GRANT SELECT ON vw_CurrentBookings TO Receptionist;
PRINT 'Da phan quyen cho vai tro Receptionist.';
```

Hình 7.4.1. Phân quyền cho Receptionist

7.4.2. HotelManager

HotelManager có toàn quyền trên tất cả bảng.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Customers TO HotelManager;  
GRANT SELECT, INSERT, UPDATE, DELETE ON Rooms TO HotelManager;  
GRANT SELECT, INSERT, UPDATE, DELETE ON Bookings TO HotelManager;  
GRANT SELECT, INSERT, UPDATE, DELETE ON Payments TO HotelManager;  
GRANT SELECT, INSERT, UPDATE, DELETE ON Employees TO HotelManager;  
GRANT EXECUTE ON sp_AddBooking TO HotelManager;  
GRANT EXECUTE ON sp_UpdateRoomStatus TO HotelManager;  
GRANT EXECUTE ON sp_CalculateInvoice TO HotelManager;  
GRANT SELECT ON vw_RoomRevenue TO HotelManager;  
GRANT SELECT ON vw_ServiceUsage TO HotelManager;  
PRINT 'Đã phân quyền cho vai tro HotelManager.';
```

Hình 7.4.2. Phân quyền cho HotelManager

7.5 Kiểm tra dữ liệu

	LoginName	LoginType
1	ManagerLogin	SQL_LOGIN
2	ReceptionistLogin	SQL_LOGIN

	RoleName	MemberName
1	HotelManager	ManagerUser
2	Receptionist	ReceptionistUser

Hình 7.5.1. Thông tin bảo mật

	PrincipalName	PrincipalType	ObjectName	Permission
1	Receptionist	DATABASE_ROLE	Bookings	INSERT
2	Receptionist	DATABASE_ROLE	Bookings	SELECT
3	Receptionist	DATABASE_ROLE	Bookings	UPDATE
4	Receptionist	DATABASE_ROLE	sp_UpdateRoomStatus	EXECUTE
5	Receptionist	DATABASE_ROLE	vw_CurrentBookings	SELECT
6	Receptionist	DATABASE_ROLE	sp_AddBooking	EXECUTE
7	Receptionist	DATABASE_ROLE	Customers	INSERT
8	Receptionist	DATABASE_ROLE	Customers	SELECT
9	Receptionist	DATABASE_ROLE	Customers	UPDATE
10	Receptionist	DATABASE_ROLE	Rooms	SELECT

Hình 7.5.2. Thông tin phân quyền với Receptionist

	PrincipalName	PrincipalType	ObjectName	Permission
1	HotelManager	DATABASE_ROLE	Bookings	DELETE
2	HotelManager	DATABASE_ROLE	Bookings	INSERT
3	HotelManager	DATABASE_ROLE	Bookings	SELECT
4	HotelManager	DATABASE_ROLE	Bookings	UPDATE
5	HotelManager	DATABASE_ROLE	Employees	DELETE
6	HotelManager	DATABASE_ROLE	Employees	INSERT
7	HotelManager	DATABASE_ROLE	Employees	SELECT
8	HotelManager	DATABASE_ROLE	Employees	UPDATE
9	HotelManager	DATABASE_ROLE	Payments	DELETE
10	HotelManager	DATABASE_ROLE	Payments	INSERT
11	HotelManager	DATABASE_ROLE	Payments	SELECT
12	HotelManager	DATABASE_ROLE	Payments	UPDATE
13	HotelManager	DATABASE_ROLE	sp_UpdateRoomStatus	EXECUTE
14	HotelManager	DATABASE_ROLE	vw_RoomRevenue	SELECT
15	HotelManager	DATABASE_ROLE	vw_ServiceUsage	SELECT
16	HotelManager	DATABASE_ROLE	sp_AddBooking	EXECUTE
17	HotelManager	DATABASE_ROLE	sp_CalculateInvoice	EXECUTE
18	HotelManager	DATABASE_ROLE	Customers	DELETE
19	HotelManager	DATABASE_ROLE	Customers	INSERT
20	HotelManager	DATABASE_ROLE	Customers	SELECT
21	HotelManager	DATABASE_ROLE	Customers	UPDATE
22	HotelManager	DATABASE_ROLE	Rooms	DELETE
23	HotelManager	DATABASE_ROLE	Rooms	INSERT
24	HotelManager	DATABASE_ROLE	Rooms	SELECT
25	HotelManager	DATABASE_ROLE	Rooms	UPDATE

Hình 7.5.3. Thông tin phân quyền với HotelManager

7.6. Sao lưu cơ sở dữ liệu

Sao lưu cơ sở dữ liệu vào file HotelManagement_FullBackup.bak

```

IF EXISTS (SELECT * FROM sys.databases WHERE name = 'HotelManagement')
BEGIN
    BACKUP DATABASE HotelManagement
    TO DISK = 'D:\Backup\HotelManagement_FullBackup.bak'
    WITH FORMAT, INIT, NAME = 'Full Backup of HotelManagement';
    PRINT 'Sao lưu cơ sở dữ liệu HotelManagement thành công.';
END
ELSE
BEGIN
    RAISERROR ('Cơ sở dữ liệu HotelManagement không tồn tại. Không thể sao lưu.', 16, 1);
END
GO

```

Hình 7.6. Sao lưu cơ sở dữ liệu

KẾT LUẬN

Quản lý khách sạn của hệ quản trị cơ sở dữ liệu được xây dựng có thể mang lại nhiều ưu điểm nổi bật, hỗ trợ đắc lực cho công tác quản lý thông tin khách hàng, đặt phòng, dịch vụ và thanh toán. Hệ thống được thiết kế theo mô hình dữ liệu quan hệ chặt chẽ, đảm bảo tính logic, tính toàn vẹn và dễ dàng mở rộng khi có nhu cầu. Quá trình lưu trữ, truy xuất và cập nhật thông tin diễn ra nhanh chóng, giúp nhân viên tiết kiệm thời gian làm việc và giảm thiểu sai sót so với quản lý thủ công. Đồng thời, việc ứng dụng cơ sở dữ liệu còn giúp ban quản lý khách sạn theo dõi toàn bộ hoạt động kinh doanh theo thời gian thực, nắm bắt được tình trạng phòng trống, lượng khách lưu trú, doanh thu theo từng giai đoạn. Không chỉ hỗ trợ tác nghiệp hàng ngày, hệ cơ sở dữ liệu còn đóng vai trò quan trọng trong việc phục vụ báo cáo thống kê, cung cấp thông tin đầy đủ, kịp thời và chính xác để ban lãnh đạo ra quyết định hiệu quả.

Tuy nhiên, bên cạnh những ưu điểm kể trên, hệ cơ sở dữ liệu vẫn còn tồn tại một số hạn chế cần tiếp tục cải thiện trong tương lai. Hệ thống hiện tại mới chỉ đáp ứng được các yêu cầu quản lý cơ bản, chưa tích hợp các chức năng nâng cao như phân tích dữ liệu, dự đoán xu hướng kinh doanh hay tự động cá nhân hóa dịch vụ theo sở thích của từng khách hàng. Khả năng tương tác giữa người dùng và hệ thống còn đơn giản, giao diện chưa thực sự thân thiện và trực quan, gây khó khăn cho những nhân viên chưa có nhiều kinh nghiệm với công nghệ. Vấn đề bảo mật dữ liệu cũng là một thách thức, đặc biệt trong bối cảnh thông tin khách hàng ngày càng được coi là tài sản quý giá, đòi hỏi hệ thống phải có các cơ chế bảo mật tốt hơn, như mã hóa dữ liệu, phân quyền truy cập chặt chẽ và ghi log hoạt động người dùng để kiểm soát rủi ro rò rỉ thông tin.

Trong tương lai, hệ cơ sở dữ liệu quản lý khách sạn có thể phát triển theo hướng mở rộng tích hợp với các công nghệ hiện đại nhằm nâng cao chất lượng quản lý và dịch vụ. Hệ thống có thể bổ sung các tính năng phân tích dữ liệu lớn (Big Data) và trí tuệ nhân tạo (AI) để khai thác thói quen, hành vi tiêu dùng của khách, từ đó cá nhân hóa dịch vụ và gia tăng trải nghiệm cho khách lưu trú. Bên cạnh đó, hệ thống cũng có thể liên kết trực tiếp với các nền tảng đặt phòng trực tuyến, ứng dụng di động và các cổng thanh toán điện tử, tạo nên một hệ sinh thái quản lý khách sạn thông minh, tiện lợi, đáp ứng nhu cầu chuyển đổi số và nâng cao năng lực cạnh tranh của khách sạn trong môi trường kinh doanh hiện đại.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Xuân Phong (2015), *Giáo trình Hệ quản trị cơ sở dữ liệu*, NXB Giáo dục Việt Nam
- [2]. Trần Văn Lợi (2020), *Cơ sở dữ liệu và ứng dụng*, Đại học Bách khoa Hà Nội.
- [3]. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020), *Database System Concepts (7th Edition)*, McGraw-Hill Education.
- [4]. Elmasri, R., & Navathe, S. B. (2016), *Fundamentals of Database Systems (7th Edition)*, Pearson Education.
- [5]. Date, C. J. (2019), *An Introduction to Database Systems (8th Edition)*, Addison-Wesley.