**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ PHÒNG TRỌ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1771020145 | Nguyễn Quang Đạt | 21/11/2005 | CNTT\_17-04 |
| 2 | 1771020779 | Nguyễn Đăng Huy | 08/12/2005 | CNTT\_17-04 |
| 3 | 1771020223 | Nguyễn Trương Giang | 10/11/2005 | CNTT\_17-04 |

### 

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI: QUẢN LÝ PHÒNG TRỌ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020145 | Nguyễn Quang Đạt | 21/11/2005 |  |  |
| 2 | 1771020779 | Nguyễn Đăng Huy | 08/12/2005 |  |  |
| 3 | 1771020223 | Nguyễn Trường Giang | 10/11/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 202**

**LỜI NÓI ĐẦU**

Trong thời đại công nghệ số ngày càng phát triển, dữ liệu trở thành một yếu tố quan trọng trong hầu hết các lĩnh vực kinh doanh và quản lý. Việc lưu trữ, xử lý và khai thác dữ liệu một cách hiệu quả có thể giúp tối ưu hóa quy trình làm việc, giảm thiểu sai sót và nâng cao hiệu suất quản lý. Một trong những lĩnh vực đòi hỏi sự chính xác và hệ thống hóa trong quản lý là **quản lý phòng trọ**. Khi số lượng phòng trọ và khách thuê tăng lên, việc quản lý theo cách truyền thống bằng sổ sách hoặc bảng tính Excel trở nên kém hiệu quả, dễ xảy ra sai sót và khó theo dõi thông tin chi tiết.

Hiện nay, nhiều chủ nhà trọ gặp phải các vấn đề như: khó kiểm soát tình trạng phòng trọ, theo dõi hợp đồng thuê một cách thủ công dẫn đến nhầm lẫn về thời hạn, tính toán hóa đơn chưa chính xác hoặc thất thoát dữ liệu quan trọng. Những khó khăn này cho thấy sự cần thiết của một hệ thống quản lý hiện đại, giúp tự động hóa các tác vụ và đảm bảo dữ liệu luôn được lưu trữ một cách khoa học.

**SQL (Structured Query Language)** là một trong những công cụ quan trọng giúp giải quyết bài toán quản lý dữ liệu. Với khả năng xử lý dữ liệu mạnh mẽ, SQL giúp dễ dàng tổ chức, lưu trữ và truy xuất thông tin một cách nhanh chóng. Một hệ thống quản lý phòng trọ sử dụng SQL có thể giúp chủ trọ dễ dàng kiểm tra thông tin phòng, tình trạng thuê, theo dõi hợp đồng, hóa đơn, cũng như hỗ trợ các báo cáo tài chính một cách chính xác và hiệu quả. Việc ứng dụng SQL không chỉ giúp nâng cao chất lượng quản lý mà còn tiết kiệm thời gian, công sức, giảm thiểu các sai sót không đáng có trong quá trình vận hành.

Chúng tôi sẽ trình bày chi tiết về việc xây dựng một hệ thống quản lý phòng trọ bằng SQL, bao gồm:

* **Phân tích yêu cầu của hệ thống**: Xác định các thực thể cần quản lý như phòng trọ, khách thuê, hợp đồng và hóa đơn.
* **Thiết kế cơ sở dữ liệu**: Mô hình hóa dữ liệu theo mô hình quan hệ, đảm bảo tính chặt chẽ và toàn vẹn dữ liệu.
* **Triển khai hệ thống**: Tạo các bảng dữ liệu, xây dựng các truy vấn SQL để thực hiện các thao tác như thêm, sửa, xóa và truy vấn dữ liệu.
* **Kiểm thử và đánh giá**: Đánh giá hiệu quả của hệ thống, kiểm tra tính chính xác của các truy vấn SQL, và đề xuất những cải tiến để tối ưu hóa quản lý.

Hệ thống quản lý phòng trọ bằng SQL không chỉ là một ứng dụng thực tiễn mà còn có thể mở rộng sang nhiều lĩnh vực khác như quản lý khách sạn, ký túc xá, chung cư mini hay nhà cho thuê. Đây là một bước khởi đầu quan trọng giúp tối ưu hóa công tác quản lý bằng công nghệ, góp phần nâng cao hiệu quả kinh doanh và tạo ra một quy trình làm việc chuyên nghiệp hơn.

Trong quá trình thực hiện báo cáo này, chúng tôi đã không ngừng nghiên cứu, tìm hiểu và áp dụng những kiến thức về cơ sở dữ liệu để thiết kế một hệ thống quản lý phù hợp. Tuy nhiên, do thời gian và kiến thức còn hạn chế, báo cáo có thể chưa hoàn toàn tối ưu. Chúng tôi rất mong nhận được những ý kiến đóng góp từ thầy cô và các bạn để tiếp tục hoàn thiện và phát triển hệ thống hơn nữa. Ngoài quản lý phòng trọ, SQL còn có thể ứng dụng vào những lĩnh vực nào khác? Làm thế nào để tối ưu hóa hơn nữa việc quản lý dữ liệu? Những câu hỏi này sẽ còn mở ra nhiều hướng nghiên cứu và khám phá trong tương lai.

**MỤC LỤC**

**CHƯƠNG I. TỔNG QUAN VỀ QUẢN LÝ PHÒNG TRỌ**

**[1.1. Giới thiệu chung về quản lý phòng trọ 6](#_Toc26232)**

**[1.2. Lý do chọn đề tài 10](#_Toc24471)**

**[1.3. Mục tiêu đề tài 12](#_Toc13419)**

**CHƯƠNG II. XÂY DỰNG MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ VÀ TẠO CƠ SỞ DỮ LIỆU**

**[2.2. Xây dựng các bảng 16](#_Toc11017)  
CHƯƠNG III: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG QUẢN LÝ PHÒNG TRỌ**

**[3.1. Tạo Database 20](#_Toc1741)**

**[1. Tạo Login và User 46](#_Toc6100)**

**[2. Cấp Quyền Truy Cập 46](#_Toc3638)**

**[3. Kiểm Tra Chính Sách Mật Khẩu 47](#_Toc11099)**

**[4. Sao Lưu Dữ Liệu 47](#_Toc7424)**

**CHƯƠNG IV: ỨNG DỤNG, ĐÁNH GIÁ VÀ ĐỊNH HƯỚNG TƯƠNG LAI**

**[1. Nhận xét tổng quan về hệ thống 49](#_Toc20233)**

**[2. Ưu điểm của hệ thống 49](#_Toc15734)**

**[2.1. Hoạt động tự động hóa tốt 49](#_Toc22595)**

**[2.2. Phân quyền chặt chẽ 50](#_Toc14990)**

**[2.3. Cấu trúc dữ liệu linh hoạt 50](#_Toc3269)**

**[3. Nhược điểm và hạn chế 50](#_Toc14153)**

**[3.1. Hiệu suất chưa tối ưu 50](#_Toc28409)**

**[3.2. Chưa có hệ thống Log thay đổi dữ liệu 50](#_Toc4647)**

**[3.3. Bảo mật chưa cao 51](#_Toc14865)**

**[3.4. Thiếu giao diện người dùng 51](#_Toc32099)**

**[4. Đề xuất phát triển trong tương lai 51](#_Toc29676)**

**[5. Đánh giá mức độ phù hợp với thực tế 52](#_Toc14791)**

**KẾT LUẬN............................................................................................................................54**

**CHƯƠNG I. TỔNG QUAN VỀ QUẢN LÝ PHÒNG TRỌ**

* 1. **Giới thiệu chung về quản lý phòng trọ**

Quản lý phòng trọ là một trong những công việc quan trọng đối với các chủ nhà trọ, người kinh doanh dịch vụ lưu trú hoặc các tổ chức cho thuê nhà ở. Với sự gia tăng dân số và nhu cầu thuê trọ ngày càng cao, đặc biệt tại các thành phố lớn, việc quản lý phòng trọ ngày càng trở nên phức tạp. Chủ nhà trọ không chỉ cần theo dõi danh sách các phòng, tình trạng thuê mà còn phải kiểm soát hợp đồng, hóa đơn, dịch vụ tiện ích (điện, nước, internet) và các khoản thu chi hàng tháng.

Trước đây, nhiều người quản lý phòng trọ theo cách thủ công, chẳng hạn như ghi chép vào sổ sách hoặc sử dụng các phần mềm văn phòng đơn giản như Excel. Tuy nhiên, phương pháp này dễ dẫn đến sai sót, mất mát dữ liệu, khó tìm kiếm thông tin và không đáp ứng được nhu cầu quản lý hiệu quả khi số lượng phòng và khách thuê tăng lên. Chính vì vậy, việc áp dụng công nghệ thông tin, đặc biệt là cơ sở dữ liệu SQL, vào quản lý phòng trọ đã trở thành một giải pháp tối ưu.

#### ****Những vấn đề trong quản lý phòng trọ truyền thống****

Việc quản lý phòng trọ theo phương pháp thủ công hoặc thiếu hệ thống chuyên nghiệp thường gặp phải nhiều khó khăn như:

* **Khó kiểm soát thông tin khách thuê**: Khi số lượng khách thuê nhiều, việc theo dõi từng khách hàng, hợp đồng thuê, thời hạn thanh toán dễ bị nhầm lẫn hoặc thất lạc dữ liệu.
* **Ghi chép thủ công dễ sai sót**: Việc tính toán hóa đơn điện, nước, tiền phòng hàng tháng bằng tay có thể dẫn đến sai sót, gây ảnh hưởng đến lợi nhuận của chủ trọ cũng như tạo sự không minh bạch đối với khách thuê.
* **Khó theo dõi tình trạng phòng**: Khi có nhiều phòng trọ, việc kiểm soát phòng nào đang trống, phòng nào đã có người thuê, thời gian kết thúc hợp đồng trở nên phức tạp.
* **Không có báo cáo tổng hợp**: Chủ nhà trọ thường gặp khó khăn trong việc tổng hợp doanh thu, công nợ, chi phí vận hành khi không có một hệ thống lưu trữ dữ liệu hợp lý.

#### **2.Giải pháp quản lý phòng trọ bằng công nghệ**

Để giải quyết những vấn đề trên, việc xây dựng một hệ thống quản lý phòng trọ dựa trên **cơ sở dữ liệu SQL** giúp quản lý thông tin một cách khoa học, hệ thống hóa dữ liệu và tự động hóa các quy trình quan trọng. Một số lợi ích nổi bật của việc áp dụng SQL vào quản lý phòng trọ bao gồm:

* **Lưu trữ và truy xuất thông tin nhanh chóng**: SQL giúp dễ dàng lưu trữ và truy xuất dữ liệu về phòng trọ, khách thuê, hợp đồng và hóa đơn một cách chính xác.
* **Giảm thiểu sai sót trong tính toán**: Hệ thống có thể tự động tính toán tiền phòng, điện, nước và các khoản phí khác, giảm thiểu rủi ro sai sót do con người.
* **Tối ưu hóa quy trình quản lý**: Việc cập nhật thông tin, tìm kiếm dữ liệu, theo dõi công nợ và tạo báo cáo trở nên đơn giản và hiệu quả hơn.
* **Tăng tính chuyên nghiệp**: Hệ thống quản lý giúp nâng cao trải nghiệm cho cả chủ trọ và khách thuê, tạo sự minh bạch và thuận tiện trong quá trình thuê trọ.

#### **3. Xu hướng phát triển quản lý phòng trọ trong tương lai**

Trong thời đại công nghệ số, quản lý phòng trọ không chỉ dừng lại ở việc sử dụng cơ sở dữ liệu SQL mà còn có thể phát triển thành các hệ thống quản lý hiện đại hơn, như:

* **Ứng dụng web hoặc mobile**: Hệ thống có thể được tích hợp vào website hoặc ứng dụng di động, giúp chủ trọ dễ dàng theo dõi tình trạng thuê phòng ở mọi lúc, mọi nơi.
* **Tích hợp thanh toán trực tuyến**: Việc kết hợp các cổng thanh toán điện tử giúp khách thuê có thể thanh toán tiền phòng dễ dàng hơn, hạn chế các vấn đề liên quan đến công nợ.
* **Áp dụng trí tuệ nhân tạo (AI) trong quản lý**: AI có thể được ứng dụng để dự đoán nhu cầu thuê phòng, tối ưu hóa giá thuê hoặc nhắc nhở chủ trọ và khách thuê về các khoản thanh toán sắp đến hạn.

**4. Các yếu tố cần quản lý trong quản lý phòng trọ**

Quản lý phòng trọ hiệu quả đòi hỏi phải theo dõi và xử lý nhiều yếu tố quan trọng. Dưới đây là những yếu tố chính cần được quản lý trong quản lý phòng trọ

* **Thông tin khách hàng (Người thuê)**

Quản lý thông tin người thuê là rất quan trọng để đảm bảo liên lạc dễ dàng và xác minh danh tính. Bao gồm các thông tin cơ bản như tên, địa chỉ, số điện thoại và email. Ngoài ra, các giấy tờ pháp lý như chứng minh nhân dân, hộ chiếu cũng cần được lưu trữ để bảo vệ quyền lợi cả hai bên.

* **Thông tin tài sản cho thuê**

Lưu trữ đầy đủ thông tin về bất động sản cho thuê giúp chủ nhà dễ dàng kiểm soát và quản lý tài sản. Các yếu tố cần theo dõi bao gồm địa chỉ, diện tích, mô tả chi tiết về tài sản (căn hộ, nhà phố, văn phòng), mức giá thuê, cũng như các chi phí phát sinh như tiền điện, nước, phí bảo trì.

* **Điều khoản hợp đồng**

Điều khoản hợp đồng là các cam kết chính của cả hai bên. Điều này bao gồm thời gian thuê, mức giá thuê, phương thức thanh toán (chuyển khoản, tiền mặt), và các khoản phí bổ sung. Các điều kiện bảo vệ như tiền đặt cọc cũng rất quan trọng, giúp bảo vệ tài sản của chủ nhà và đảm bảo người thuê thực hiện nghĩa vụ tài chính.

* **Thanh toán và lịch sử giao dịch**

Việc theo dõi quá trình thanh toán giúp chủ nhà kiểm soát dòng tiền, xác minh các khoản thu, và xử lý các khoản thanh toán trễ. Lịch sử thanh toán bao gồm thời gian và số tiền thanh toán, cũng như các khoản nợ tồn đọng, sẽ giúp chủ nhà giải quyết các vấn đề về tài chính dễ dàng hơn.

* **Trạng thái hợp đồng và vấn đề phát sinh**

Theo dõi tình trạng hợp đồng (đang hoạt động, đã kết thúc, gia hạn) là điều cần thiết để biết được hợp đồng nào còn hiệu lực và khi nào cần gia hạn hoặc kết thúc. Đồng thời, chủ nhà cần quản lý các vấn đề phát sinh như sửa chữa tài sản, tranh chấp về tiền thuê, hoặc vi phạm hợp đồng để đảm bảo sự ổn định trong quá trình cho thuê.

Những yếu tố này giúp quản lý hợp đồng thuê nhà một cách toàn diện, bảo vệ quyền lợi của cả chủ nhà và người thuê, đồng thời đảm bảo các vấn đề tài chính và pháp lý được xử lý đúng cách.

**5. Quy trình quản lý phòng trọ**

Quy trình quản lý phòng trọ gồm các bước từ khi ký kết hợp đồng đến khi kết thúc hoặc gia hạn hợp đồng. Mỗi bước đảm bảo sự minh bạch và hiệu quả trong quá trình cho thuê, giúp quản lý tài sản và tài chính một cách dễ dàng.

* **Ký kết hợp đồng**

Chủ nhà và người thuê cần thống nhất và ký kết hợp đồng, xác nhận các điều khoản như giá thuê, thời gian thuê, phương thức thanh toán, và các điều kiện đặc biệt. Điều này đảm bảo sự minh bạch và cam kết của cả hai bên.

* **Lưu trữ hợp đồng**

Sau khi ký kết, hợp đồng cần được lưu trữ một cách có hệ thống, dễ dàng tra cứu. Việc lưu trữ hợp đồng, các giấy tờ liên quan giúp việc theo dõi và quản lý hợp đồng trở nên thuận tiện và hiệu quả.

* **Theo dõi thanh toán**

Quá trình thanh toán của người thuê cần được theo dõi định kỳ để đảm bảo các khoản tiền được thanh toán đúng hạn. Nếu có trễ hạn, chủ nhà cần có các biện pháp nhắc nhở hoặc xử lý phù hợp.

* **Quản lý gia hạn hoặc kết thúc hợp đồng**

Khi hợp đồng sắp hết hạn, chủ nhà và người thuê cần thỏa thuận về việc gia hạn hợp đồng hoặc kết thúc hợp đồng. Các thủ tục liên quan đến việc gia hạn hoặc thanh lý hợp đồng cũng cần được thực hiện đúng quy trình.

* **Giải quyết tranh chấp (nếu có)**

Nếu có tranh chấp phát sinh trong quá trình thuê (như thanh toán trễ, hư hỏng tài sản, vi phạm hợp đồng), cần có quy trình giải quyết rõ ràng để đảm bảo quyền lợi của các bên và tránh những mâu thuẫn kéo dài.

Quy trình này giúp duy trì sự minh bạch, bảo vệ quyền lợi của các bên và quản lý hợp đồng thuê nhà một cách hiệu quả.

* 1. **Lý do chọn đề tài**

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ vào quản lý dữ liệu ngày càng trở nên quan trọng và cần thiết. Đặc biệt, trong lĩnh vực **quản lý phòng trọ**, việc theo dõi thông tin phòng, hợp đồng, khách thuê, thanh toán và các dịch vụ đi kèm như điện, nước, internet đòi hỏi một hệ thống lưu trữ và xử lý dữ liệu chính xác, hiệu quả. Nếu chỉ sử dụng các phương pháp truyền thống như ghi chép sổ sách hoặc bảng tính Excel, việc quản lý sẽ trở nên phức tạp, dễ xảy ra sai sót và mất nhiều thời gian. Chính vì vậy, việc nghiên cứu và xây dựng một hệ thống **quản lý phòng trọ sử dụng SQL** sẽ giúp tối ưu hóa quy trình làm việc, nâng cao tính chính xác và tiết kiệm thời gian cho người quản lý.

Dưới đây là những lý do cụ thể khiến chúng tôi lựa chọn đề tài này:

#### **1. Tính thực tiễn và ứng dụng cao**

Nhu cầu thuê trọ ngày càng gia tăng, đặc biệt ở các thành phố lớn, khu công nghiệp, khu vực gần trường đại học và trung tâm thương mại. Các chủ nhà trọ, ký túc xá, chung cư mini hoặc các doanh nghiệp kinh doanh dịch vụ lưu trú đều cần một hệ thống quản lý hiệu quả để theo dõi tình trạng phòng, hợp đồng thuê, hóa đơn và công nợ. Một hệ thống quản lý dựa trên **SQL** không chỉ giúp xử lý dữ liệu nhanh chóng mà còn dễ dàng mở rộng để phù hợp với các mô hình kinh doanh khác nhau.

#### **2. Giải quyết các vấn đề của phương pháp quản lý truyền thống**

Hiện nay, nhiều chủ nhà trọ vẫn sử dụng cách quản lý bằng giấy tờ hoặc file Excel, dẫn đến các vấn đề như:

* **Dữ liệu dễ bị thất lạc hoặc nhầm lẫn**: Việc ghi chép thủ công rất khó để đảm bảo tính chính xác và toàn vẹn dữ liệu, đặc biệt khi có nhiều khách thuê.
* **Khó khăn trong tra cứu và tổng hợp thông tin**: Khi cần tìm kiếm thông tin về khách thuê, phòng trọ hoặc hóa đơn, việc truy xuất dữ liệu từ sổ sách hoặc file Excel tốn nhiều thời gian và không hiệu quả.
* **Sai sót trong tính toán tiền thuê, điện nước**: Việc tính toán thủ công dễ dẫn đến nhầm lẫn, gây thất thoát tài chính hoặc tranh chấp giữa chủ trọ và khách thuê.
* **Khó mở rộng hệ thống**: Khi số lượng phòng trọ tăng lên, việc quản lý theo phương pháp cũ sẽ trở nên phức tạp, khó kiểm soát và tốn nhiều công sức.

Việc sử dụng SQL giúp giải quyết những vấn đề này bằng cách **tự động hóa việc lưu trữ và xử lý dữ liệu**, giúp hệ thống trở nên chuyên nghiệp hơn, dễ dàng truy xuất thông tin và đảm bảo tính chính xác.

#### **3. Rèn luyện kỹ năng làm việc với cơ sở dữ liệu**

SQL là một trong những ngôn ngữ quan trọng trong lĩnh vực công nghệ thông tin, được sử dụng rộng rãi trong các hệ thống quản lý dữ liệu. Thông qua đề tài này, chúng tôi có cơ hội:

* **Nâng cao kỹ năng thiết kế cơ sở dữ liệu**, từ việc phân tích yêu cầu đến xây dựng mô hình quan hệ giữa các thực thể.
* **Cải thiện khả năng viết truy vấn SQL** để thực hiện các thao tác thêm, sửa, xóa, tìm kiếm dữ liệu một cách tối ưu.
* **Tìm hiểu cách tối ưu hóa truy vấn và đảm bảo tính toàn vẹn dữ liệu**, giúp hệ thống hoạt động hiệu quả và ổn định hơn.
* **Ứng dụng SQL vào thực tế**, tạo tiền đề để phát triển các hệ thống quản lý chuyên nghiệp hơn trong tương lai.

#### **4. Hướng phát triển mở rộng**

Việc xây dựng hệ thống quản lý phòng trọ bằng SQL không chỉ dừng lại ở mức độ cơ bản mà còn có thể mở rộng thành một hệ thống quản lý toàn diện hơn, bao gồm:

* **Tích hợp với ứng dụng web hoặc di động**, giúp chủ trọ và khách thuê dễ dàng quản lý thông tin mọi lúc, mọi nơi.
* **Tích hợp tính năng thanh toán trực tuyến**, giúp khách thuê có thể thanh toán tiền phòng, điện, nước nhanh chóng và tiện lợi hơn.
* **Ứng dụng trí tuệ nhân tạo (AI) hoặc phân tích dữ liệu**, giúp dự đoán nhu cầu thuê phòng, tối ưu hóa giá thuê và nâng cao trải nghiệm của khách hàng.

#### **5. Đóng góp vào việc ứng dụng công nghệ trong đời sống**

Đề tài này không chỉ có ý nghĩa đối với những người đang kinh doanh phòng trọ mà còn là một ví dụ điển hình về cách **công nghệ có thể thay đổi cách con người làm việc và quản lý dữ liệu**. Hệ thống không chỉ giúp tối ưu hóa công việc quản lý phòng trọ mà còn có thể áp dụng cho nhiều lĩnh vực khác như quản lý khách sạn, ký túc xá, chung cư mini hoặc thậm chí là các hệ thống quản lý tài sản khác.

* 1. **Mục tiêu đề tài**

Đề tài **“Sử dụng SQL để Quản lý Phòng Trọ”** nhằm xây dựng một hệ thống quản lý dữ liệu hiệu quả, giúp chủ nhà trọ có thể theo dõi và quản lý thông tin phòng trọ, khách thuê, hợp đồng, hóa đơn một cách chính xác và thuận tiện. Hệ thống này sẽ tận dụng sức mạnh của SQL để lưu trữ, truy vấn và xử lý dữ liệu nhanh chóng, đảm bảo tính toàn vẹn và khả năng mở rộng trong tương lai.

#### **Mục tiêu cụ thể**

##### ****Xây dựng hệ thống cơ sở dữ liệu cho quản lý phòng trọ****

* Thiết kế **mô hình dữ liệu** chặt chẽ, bao gồm các bảng liên quan như: **Phòng trọ, Khách thuê, Hợp đồng, Hóa đơn, Thanh toán, Dịch vụ điện nước, Chủ trọ**, đảm bảo tính toàn vẹn và tránh trùng lặp dữ liệu.
* Định nghĩa các **ràng buộc toàn vẹn dữ liệu** (Primary Key, Foreign Key, Unique, Check,...) để đảm bảo dữ liệu được lưu trữ chính xác và hợp lý.
* Xây dựng các **mối quan hệ** giữa các thực thể để dễ dàng truy vấn thông tin.

##### ****Cung cấp các chức năng quản lý dữ liệu bằng SQL****

* Xây dựng các **truy vấn SQL** phục vụ cho các thao tác quan trọng như:
  + **Thêm, sửa, xóa** thông tin về phòng trọ, khách thuê, hợp đồng và hóa đơn.
  + **Truy xuất thông tin** phòng trọ, tình trạng thuê phòng, danh sách khách thuê theo thời gian thực.
  + **Tính toán tiền phòng, tiền điện nước, các dịch vụ phát sinh** và xuất báo cáo tổng hợp doanh thu.
  + **Theo dõi tình trạng thanh toán**, nhắc nhở khách thuê về các khoản cần đóng.
* Đảm bảo các truy vấn hoạt động hiệu quả, tối ưu hóa hiệu suất để xử lý dữ liệu nhanh chóng.

##### ****Đảm bảo tính chính xác và bảo mật dữ liệu****

* Sử dụng các **cơ chế bảo mật** trong SQL để kiểm soát quyền truy cập, chỉ cho phép người có quyền thực hiện các thao tác quan trọng.
* **Đảm bảo tính toàn vẹn dữ liệu**, tránh tình trạng dư thừa hoặc thiếu sót thông tin.
* Kiểm tra và ngăn chặn các lỗi thường gặp như nhập dữ liệu trùng, sai định dạng hoặc vi phạm ràng buộc dữ liệu.

**CHƯƠNG II. XÂY DỰNG MÔ HÌNH CƠ SỞ DỮ LIỆU QUAN HỆ VÀ TẠO CƠ SỞ DỮ LIỆU**

**2.1. Xác định các thực thể, thuộc tính và ràng buộc**

* **Bảng loại phòng**
* [MaLoaiPhong] [nchar](10)
* [TenLoaiPhong] [nvarchar](50)
* [DienTichPhong] [float]
* [DonGia] [float]
* [GhiChu] [nvarchar](50)
* [PK\_LoaiPhong]
* **Bảng khu vực**
* [MaKhuVuc] [nchar](10)
* [TenKhuVuc] [nvarchar](50)
* **[PK\_KhuVuc]**
* ****Bảng ghi nhớ****
* [ID] [nchar](10)
* [Pass] [nchar](10)
* **Bảng dịch vụ**
* [Id] [int]
* [TenDichVu] [nvarchar](50)
* [Gia] [bigint]
* **[PK\_DichVu]**
* ****Bảng đăng nhập****
* [ID] [nvarchar](50)
* [PassWord] [nchar](10)
* [PRIMARY]
* **Bảng thông tin khách hàng**
* [MaKhachTro] [nchar](10)
* [Ho] [nvarchar](50)
* [Ten] [nvarchar](50)
* [GioiTinh] [nvarchar](50)
* [AnhNhanDien] [image]
* [NgaySinh] [date]
* [CMND] [nchar](10)
* [QueQuan] [nvarchar](50)
* [NgheNghiep] [nvarchar](50)
* [MaPhong] [nchar](10)
* [GhiChu] [nvarchar](50)
* [PK\_ThongTinKhach]
* **Bảng Phòng**
* [MaPhong] [nchar](10)
* [MaLoaiPhong] [nchar](10)
* [MaKhuVuc] [nchar](10)
* [TenPhong] [nvarchar](50)
* [Day] [nchar](10)
* [TrangThai] [nvarchar](10)
* [GhiChu] [nvarchar](50)
* ****Bảng thông tin thuê phòng****
* [MaHD] [nchar](10)
* [MaKhachTro] [nchar](10)
* [MaPhong] [nchar](10)
* [NgayThue] [date]
* [TienDatCoc] [bigint]
* [GhiChu] [nvarchar](50)
* **Bảng phiếu thu**
* [MaPT] [nchar](10)
* [MaPhong] [nchar](10)
* [NgayLap] [date]
* [NgayThu] [date]
* [TienNha] [bigint]
* [SkDien] [int]
* [TienDien] [bigint]
* [SkNuoc] [int]
* [TienNuoc] [bigint]
* [TongTien] [bigint]
* [TrangThai] [nvarchar](50)
* **Ràng buộc**

****-** Khóa Chính (**Primary Key**)**:

* PK\_LoaiPhong trên [MaLoaiPhong].
* PK\_KhuVuc trên [MaKhuVuc].
* PK\_DichVu trên [Id].
* PRIMARY trên [ID] của bảng **Đăng Nhập**.
* PK\_ThongTinKhach trên [MaKhachTro].

****-** Khóa Ngoại (**Foreign Key**)**:

* [MaLoaiPhong] trong **Phòng** tham chiếu đến **Loại Phòng**.
* [MaKhuVuc] trong **Phòng** tham chiếu đến **Khu Vực**.
* [MaPhong] trong **Thông Tin Khách Hàng** và **Thông Tin Thuê Phòng** tham chiếu đến **Phòng**.
* [MaPhong] trong **Phiếu Thu** tham chiếu đến **Phòng**.

****-** Không Rỗng (**Not Null**)**:

* Các cột khóa chính và khóa ngoại.
* Các thông tin quan trọng như tên, ngày sinh, CMND, v.v.

****-** Kiểm Tra (**Check**)**:

* [GioiTinh]: "Nam" hoặc "Nữ".
* [TrangThai]: "Trống" hoặc "Đã thuê" trong **Phòng**.
* [TrangThai]: "Đã thanh toán" hoặc "Chưa thanh toán" trong **Phiếu Thu**.

****-** Duy Nhất (**Unique**)**:

* [CMND] trong **Thông Tin Khách Hàng**.

**2.2. Xây dựng các bảng**

* **Bảng loại phòng**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaLoaiPhong | NCHAR(10) | PK | Mã loại phòng |
| 2 | TenLoaiPhong | NVARCHAR(50) | NOT NULL | Tên loại phòng |
| 3 | DienTichPhong | FLOAT | NOT NULL, CHECK (DienTichPhong > 0) | Diện tích phòng |
| 4 | DonGia | FLOAT | NOT NULL, CHECK (DonGia > 0) | Đơn giá |
| 5 | GhiChu | NVARCHAR(50) | NULL | Ghi chú |

* **Bảng khu vực**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaKhuVuc | NCHAR(10) | PK | Mã khu vực |
| 2 | TenKhuVuc | NVARCHAR(50) | NOT NULL | Tên khu vực |

* **Bảng ghi nhớ**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | NCHAR(10) | PK | Mã định danh |
| 2 | Pass | NCHAR(10) | NOT NULL | Mật khẩu |

* **Bàng dịch vụ**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | Id | INT | PK | Mã dịch vụ |
| 2 | TenDichVu | NVARCHAR(50) | NOT NULL | Tên dịch vụ |
| 3 | Gia | BIGINT | NOT NULL, CHECK (Gia >= 0) | Giá dịch vụ |

* **Bảng đăng nhập**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | NVARCHAR(50) | PRIMARY KEY | Tên đăng nhập |
| 2 | PassWord | NCHAR(10) | NOT NULL | Mật khẩu |

* **Bảng thông tin khách hàng**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaKhachTro | NCHAR(10) | PK | Mã khách trọ |
| 2 | Ho | NVARCHAR(50) | NOT NULL | Họ khách trọ |
| 3 | Ten | NVARCHAR(50) | NOT NULL | Tên khách trọ |
| 4 | GioiTinh | NVARCHAR(50) | NOT NULL | Giới tính |
| 5 | AnhNhanDien | IMAGE | NULL | Ảnh nhận diện |
| 6 | NgaySinh | DATE | NOT NULL | Ngày sinh |
| 7 | CMND | NCHAR(10) | UNIQUE | CMND/CCCD |
| 8 | QueQuan | NVARCHAR(50) | NOT NULL | Quê quán |
| 9 | NgheNghiep | NVARCHAR(50) | NULL | Nghề nghiệp |
| 10 | MaPhong | NCHAR(10) | FK | Mã phòng |
| 11 | GhiChu | NVARCHAR(50) | NULL | Ghi chú |

* **Bảng phòng**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaPhong | NCHAR(10) | PK | Mã phòng |
| 2 | MaLoaiPhong | NCHAR(10) | FK | Loại phòng |
| 3 | MaKhuVuc | NCHAR(10) | FK | Mã khu vực |
| 4 | TenPhong | NVARCHAR(50) | NOT NULL | Tên phòng |
| 5 | Day | NCHAR(10) | NULL | Dãy phòng |
| 6 | TrangThai | NVARCHAR(10) | NOT NULL | Trạng thái |
| 7 | GhiChu | NVARCHAR(50) | NULL | Ghi chú |

* **Bảng thồng tin thuê phòng**

| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaHD | NCHAR(10) | PK | Mã hợp đồng |
| 2 | MaKhachTro | NCHAR(10) | FK | Mã khách trọ |
| 3 | MaPhong | NCHAR(10) | FK | Mã phòng |
| 4 | NgayThue | DATE | NOT NULL | Ngày thuê |
| 5 | TienDatCoc | BIGINT | NOT NULL, CHECK (TienDatCoc >= 0) | Tiền đặt cọc |
| 6 | GhiChu | NVARCHAR(50) | NULL | Ghi chú |

* **Bảng phiếu thu**

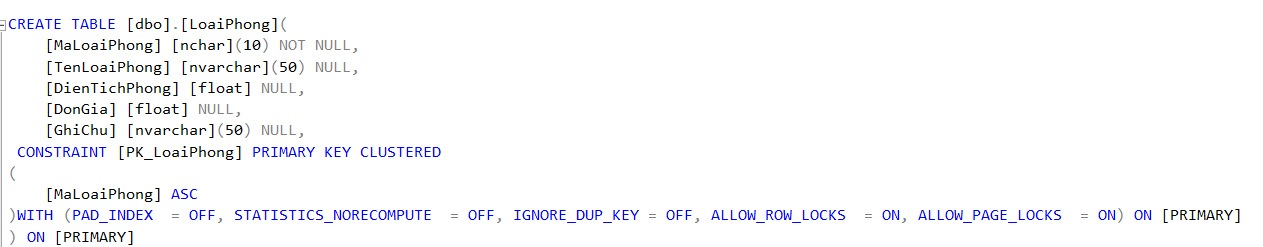
| **TT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | MaPT | NCHAR(10) | PK | Mã phiếu thu |
| 2 | MaPhong | NCHAR(10) | FK | Mã phòng |
| 3 | NgayLap | DATE | NOT NULL | Ngày lập phiếu |
| 4 | NgayThu | DATE | NULL | Ngày thu tiền |
| 5 | TienNha | BIGINT | NOT NULL, CHECK (TienNha >= 0) | Tiền nhà |
| 6 | SkDien | INT | CHECK (SkDien >= 0) | Số kWh điện |
| 7 | TienDien | BIGINT | CHECK (TienDien >= 0) | Tiền điện |
| 8 | SkNuoc | INT | CHECK (SkNuoc >= 0) | Số m³ nước |
| 9 | TienNuoc | BIGINT | CHECK (TienNuoc >= 0) | Tiền nước |
| 10 | TongTien | BIGINT | NOT NULL, CHECK (TongTien >= 0) | Tổng tiền |
| 11 | TrangThai | NVARCHAR(50) | NOT NULL | Trạng thái |

**CHƯƠNG III: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG QUẢN LÝ PHÒNG TRỌ**

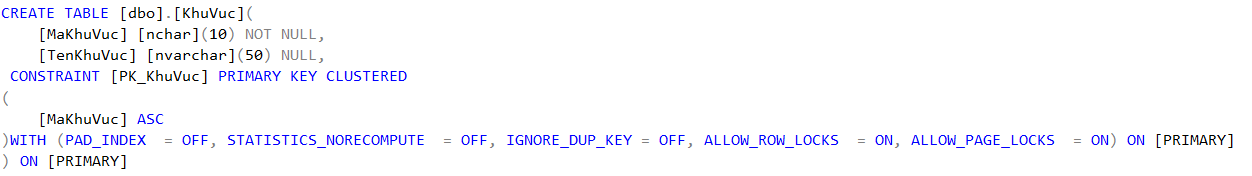
**3.1. Tạo Database**

* **Database Diagram**

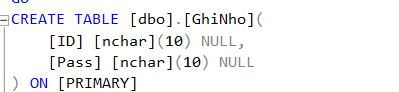
Database Diagram là một cách trực quan để thể hiện mối quan hệ giữa các bảng trong cơ sở dữ liệu. Trong hệ thống quản lý hợp đồng thuê nhà của chúng ta, các bảng chính bao gồm:



* **[dbo].[LoaiPhong]:** Dùng để lưu trữ thông tin về các loại phòng trong hệ thống quản lý khách sạn, nhà nghỉ hoặc căn hộ cho thuê.
* **Mục đích:** Lưu trữ thông tin về các loại phòng trong hệ thống quản lý khách sạn, nhà nghỉ hoặc khu căn hộ.
* **Thuộc tính**
* **MaLoaiPhong**: Mã loại phòng, kiểu dữ liệu NCHAR(10), khóa chính (**PK**).
* **TenLoaiPhong**: Tên loại phòng, kiểu dữ liệu NVARCHAR(50).
* **DienTichPhong**: Diện tích phòng, kiểu dữ liệu FLOAT.
* **DonGia**: Đơn giá thuê phòng, kiểu dữ liệu FLOAT.
* **GhiChu**: Ghi chú về loại phòng, kiểu dữ liệu NVARCHAR(50).
* **Ràng buộc**
* **MaLoaiPhong** là duy nhất và không được để trống (**Khóa chính - Primary Key**).
* **TenLoaiPhong, DienTichPhong, DonGia** không được để trống.
* **DienTichPhong, DonGia** phải lớn hơn 0.
* **MaLoaiPhong** phải tồn tại trong bảng Phong nếu có mối quan hệ ràng buộc (**Khóa ngoại - Foreign Key**).
* ****[dbo].[KhuVuc]:**** Dùng để lưu trữ thông tin về các khu vực trong hệ thống quản lý, giúp phân loại và tổ chức các phòng hoặc địa điểm liên quan.



* **Mục đích:** Lưu trữ thông tin về các khu vực trong hệ thống
* **Thuộc tính**
* **MaKhuVuc** (nchar(10), NOT NULL, Primary Key): Mã khu vực, định danh duy nhất của khu vực.
* **TenKhuVuc** (nvarchar(50), NULL): Tên của khu vực, có thể để trống.
* **Ràng buộc**
* **MaKhuVuc** là khóa chính (Primary Key), duy nhất và không được để trống.
* **TenKhuVuc** có thể để trống nhưng thường nên có giá trị để mô tả khu vực.
* **[dbo].[GhiNho]:** Dùng để lưu trữ các ghi chú hoặc thông tin quan trọng trong hệ thống.



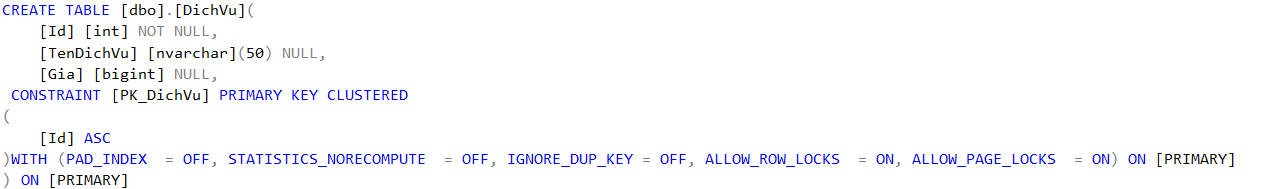
### ****Mục đích:**** Lưu trữ thông tin đăng nhập hoặc ghi nhớ tài khoản trong hệ thống.

### ****Thuộc tính:****

* **ID**: Mã định danh, kiểu dữ liệu nchar(10), cho phép giá trị NULL.
* **Pass**: Mật khẩu, kiểu dữ liệu nchar(10), cho phép giá trị NULL.

### ****Ràng buộc:****

* **ID** cần có tính duy nhất để xác định mỗi tài khoản.
* **Pass** không nên để NULL để đảm bảo mỗi tài khoản có mật khẩu.
* **[dbo].[DichVu]:** Dùng để lưu trữ thông tin về các dịch vụ trong hệ thống, bao gồm tên dịch vụ và giá cả.



### ****Mục đích:**** Lưu trữ thông tin về các dịch vụ trong hệ thống.

### ****Thuộc tính:****

* **Id**: Mã dịch vụ, kiểu dữ liệu int, không được để trống, là khóa chính (PK).
* **TenDichVu**: Tên dịch vụ, kiểu dữ liệu nvarchar(50), có thể để trống.
* **Gia**: Giá dịch vụ, kiểu dữ liệu bigint, có thể để trống.

### ****Ràng buộc:****

* **Id** là duy nhất, không được để trống và là khóa chính của bảng.
* **TenDichVu** có thể để trống nhưng nên có dữ liệu để nhận diện dịch vụ.
* **Gia** có thể để trống nhưng cần có giá trị để xác định chi phí dịch vụ.
* **[dbo].[DangNhap]:** Dùng để lưu trữ thông tin đăng nhập của người dùng, bao gồm ID và mật khẩu.



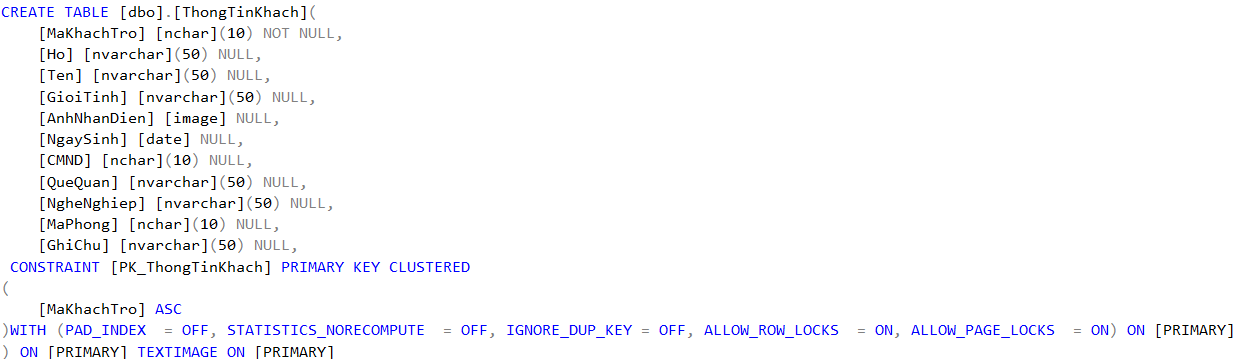
### ****Mục đích:**** Lưu trữ thông tin đăng nhập của người dùng trong hệ thống.

### ****Thuộc tính:****

* **ID**: Tên đăng nhập, kiểu dữ liệu nvarchar(50), có thể để trống.
* **PassWord**: Mật khẩu, kiểu dữ liệu nchar(10), có thể để trống.

### ****Ràng buộc:****

* **ID** nên là duy nhất để xác định mỗi người dùng.
* **PassWord** không nên để trống để đảm bảo tính bảo mật.
* Nên thêm ràng buộc NOT NULL cho cả hai cột để đảm bảo dữ liệu hợp lệ.
* Nếu có nhiều tài khoản, nên đặt ID làm **khóa chính (PK)** để tránh trùng lặp.
* **[dbo].[ThongTinKhach]:** Dùng để lưu trữ thông tin cá nhân của khách thuê, bao gồm họ tên, giới tính, ngày sinh, số CMND, nghề nghiệp, địa chỉ quê quán, số phòng đang thuê và ghi chú.



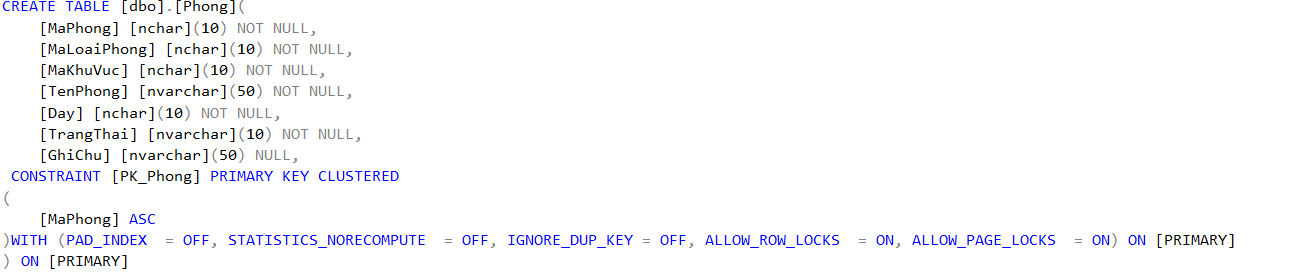
### ****Mục đích:**** Lưu trữ thông tin cá nhân của khách trọ trong hệ thống quản lý phòng trọ.

### ****Thuộc tính:****

* **MaKhachTro**: Mã khách trọ, kiểu dữ liệu nchar(10), khóa chính (PK), không được để trống.
* **Ho**: Họ của khách trọ, kiểu dữ liệu nvarchar(50), có thể để trống.
* **Ten**: Tên của khách trọ, kiểu dữ liệu nvarchar(50), có thể để trống.
* **GioiTinh**: Giới tính, kiểu dữ liệu nvarchar(50), có thể để trống.
* **AnhNhanDien**: Ảnh nhận diện, kiểu dữ liệu image, có thể để trống.
* **NgaySinh**: Ngày sinh, kiểu dữ liệu date, có thể để trống.
* **CMND**: Chứng minh nhân dân, kiểu dữ liệu nchar(10), có thể để trống.
* **QueQuan**: Quê quán, kiểu dữ liệu nvarchar(50), có thể để trống.
* **NgheNghiep**: Nghề nghiệp, kiểu dữ liệu nvarchar(50), có thể để trống.
* **MaPhong**: Mã phòng khách trọ đang ở, kiểu dữ liệu nchar(10), có thể để trống.
* **GhiChu**: Ghi chú thêm về khách trọ, kiểu dữ liệu nvarchar(50), có thể để trống.

### ****Ràng buộc:****

* **MaKhachTro** là duy nhất và không được để trống (khóa chính).
* **CMND** nên là duy nhất để tránh trùng lặp thông tin khách hàng.
* **MaPhong** có thể liên kết với bảng **Phong** để xác định phòng mà khách trọ đang ở.
* **[dbo].[Phong]:** Dùng để lưu trữ thông tin về các phòng trong hệ thống, bao gồm mã phòng, loại phòng, khu vực, tên phòng, trạng thái và các thông tin liên quan khác.



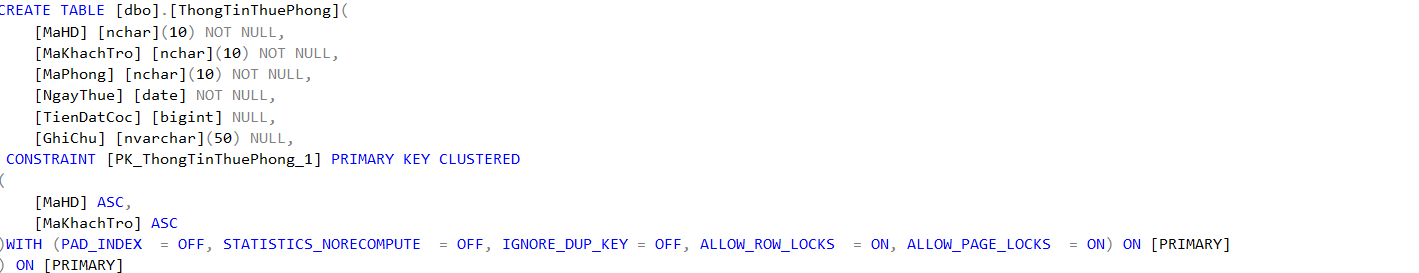
### ****Mục đích:**** Lưu trữ thông tin về các phòng trong hệ thống quản lý phòng trọ.

### ****Thuộc tính:****

* **MaPhong**: Mã phòng, kiểu dữ liệu nchar(10), khóa chính (PK), không được để trống.
* **MaLoaiPhong**: Mã loại phòng, kiểu dữ liệu nchar(10), không được để trống.
* **MaKhuVuc**: Mã khu vực, kiểu dữ liệu nchar(10), không được để trống.
* **TenPhong**: Tên phòng, kiểu dữ liệu nvarchar(50), không được để trống.
* **Day**: Dãy phòng, kiểu dữ liệu nchar(10), không được để trống.
* **TrangThai**: Trạng thái phòng (đang trống, đã thuê...), kiểu dữ liệu nvarchar(10), không được để trống.
* **GhiChu**: Ghi chú, kiểu dữ liệu nvarchar(50), có thể để trống.

### ****Ràng buộc:****

* **MaPhong** là duy nhất và không được để trống (khóa chính).
* **MaLoaiPhong** liên kết với bảng **LoaiPhong** để xác định loại phòng.
* **MaKhuVuc** liên kết với bảng **KhuVuc** để xác định vị trí phòng trong khu vực nào.
* **TrangThai** chỉ nhận các giá trị hợp lệ như "Trống", "Đã thuê", "Bảo trì"...
* **[dbo].[ThongTinThuePhong]:** Dùng để lưu trữ thông tin về quá trình thuê phòng của khách, bao gồm mã hóa đơn, mã khách thuê, mã phòng, ngày thuê, tiền đặt cọc và các thông tin liên quan khác.



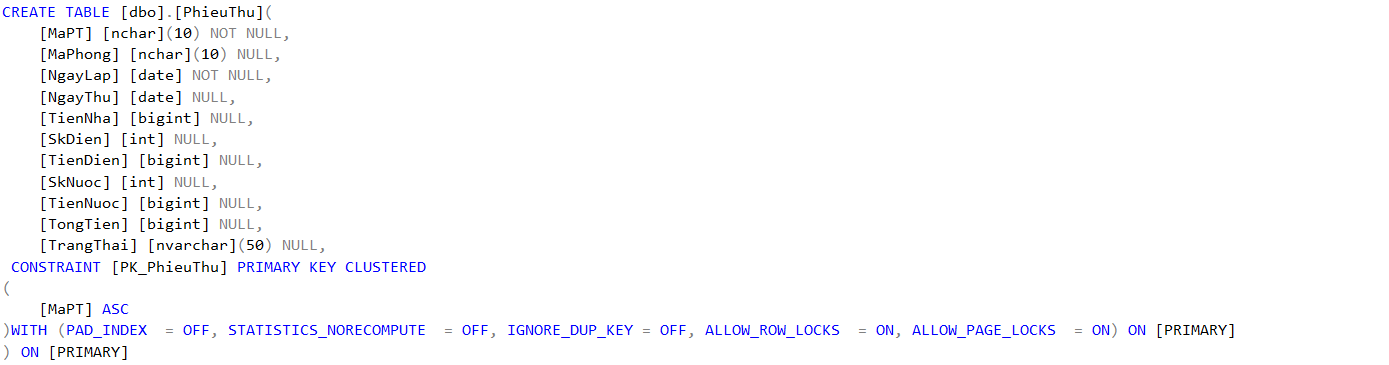
### ****Mục đích:**** Lưu trữ thông tin thuê phòng của khách trọ.

### ****Thuộc tính:****

* **MaHD**: Mã hợp đồng thuê, kiểu dữ liệu nchar(10), khóa chính (PK), không được để trống.
* **MaKhachTro**: Mã khách trọ, kiểu dữ liệu nchar(10), khóa chính (PK), không được để trống.
* **MaPhong**: Mã phòng thuê, kiểu dữ liệu nchar(10), không được để trống.
* **NgayThue**: Ngày thuê phòng, kiểu dữ liệu date, không được để trống.
* **TienDatCoc**: Tiền đặt cọc, kiểu dữ liệu bigint, có thể để trống.
* **GhiChu**: Ghi chú, kiểu dữ liệu nvarchar(50), có thể để trống.

### ****Ràng buộc:****

* **MaHD** và **MaKhachTro** là khóa chính (PK), đảm bảo mỗi khách có một hợp đồng thuê duy nhất.
* **MaKhachTro** liên kết với bảng **ThongTinKhach** để xác định thông tin khách thuê.
* **MaPhong** liên kết với bảng **Phong** để xác định phòng đã thuê.
* **NgayThue** không được để trống vì là ngày bắt đầu thuê phòng.
* **TienDatCoc** có thể để trống, tùy vào yêu cầu đặt cọc của chủ trọ.
* **[dbo].[PhieuThu]:** Dùng để lưu trữ thông tin về các phiếu thu tiền phòng, bao gồm mã phiếu thu, mã phòng, ngày lập, ngày thu, tiền nhà, tiền điện, tiền nước và trạng thái thanh toán.



### ****Mục đích:**** Lưu trữ thông tin phiếu thu tiền của khách thuê phòng.

### ****Thuộc tính:****

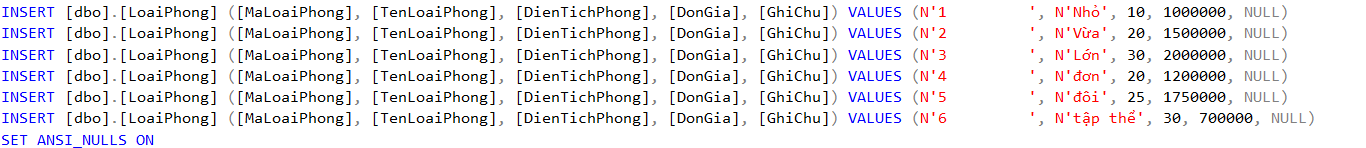
* **MaPT**: Mã phiếu thu, kiểu dữ liệu nchar(10), khóa chính (PK), không được để trống.
* **MaPhong**: Mã phòng, kiểu dữ liệu nchar(10), không được để trống.
* **NgayLap**: Ngày lập phiếu thu, kiểu dữ liệu date, không được để trống.
* **NgayThu**: Ngày thu tiền, kiểu dữ liệu date, có thể để trống.
* **TienNha**: Tiền thuê nhà, kiểu dữ liệu bigint, có thể để trống.
* **SkDien**: Số kWh điện tiêu thụ, kiểu dữ liệu int, có thể để trống.
* **TienDien**: Tiền điện, kiểu dữ liệu bigint, có thể để trống.
* **SkNuoc**: Số khối nước tiêu thụ, kiểu dữ liệu int, có thể để trống.
* **TienNuoc**: Tiền nước, kiểu dữ liệu bigint, có thể để trống.
* **TongTien**: Tổng tiền phải trả, kiểu dữ liệu bigint, có thể để trống.
* **TrangThai**: Trạng thái thanh toán, kiểu dữ liệu nvarchar(50), có thể để trống.

### ****Ràng buộc:****

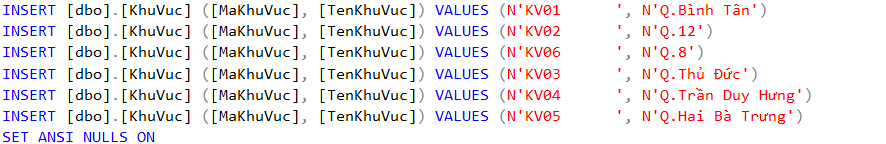
* **MaPT** là khóa chính (PK), đảm bảo mỗi phiếu thu là duy nhất.
* **MaPhong** liên kết với bảng **Phong** để xác định phòng thuê tương ứng.
* **NgayLap** không được để trống vì là ngày lập phiếu thu.
* **NgayThu** có thể để trống nếu khách chưa thanh toán.
* **TienNha**, **TienDien**, **TienNuoc**, **TongTien** có thể để trống vì có thể cập nhật sau khi tính toán.

**3.2. Chèn dữ liệu**

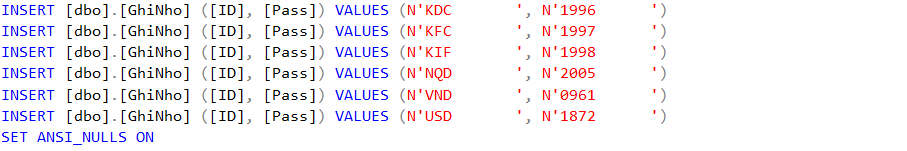
* **Bảng loại phòng**



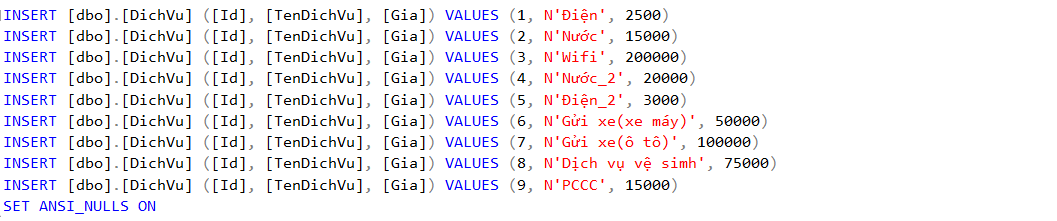
* **Bảng mã khu vực**



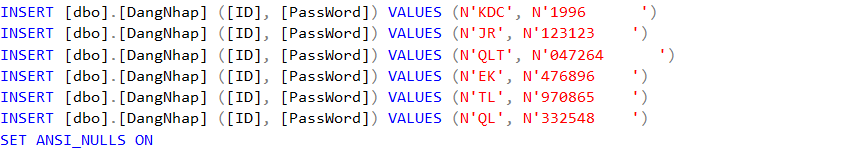
* **Bảng ghi nhớ**



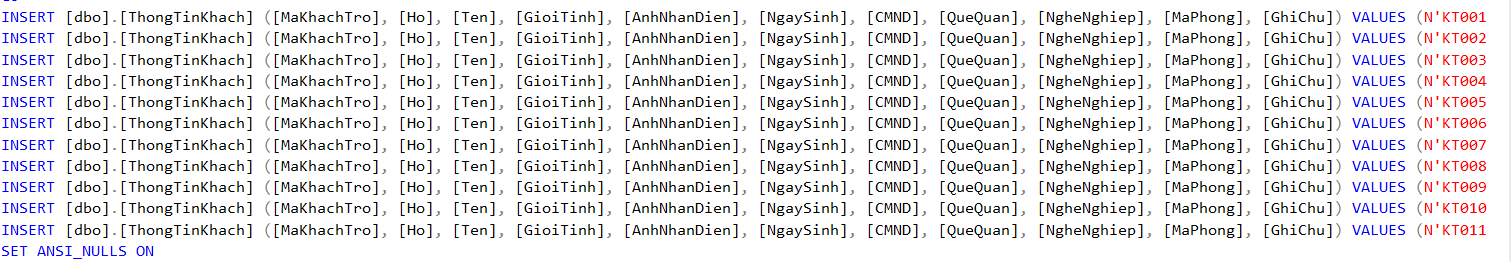
* **Bảng dịch vụ**



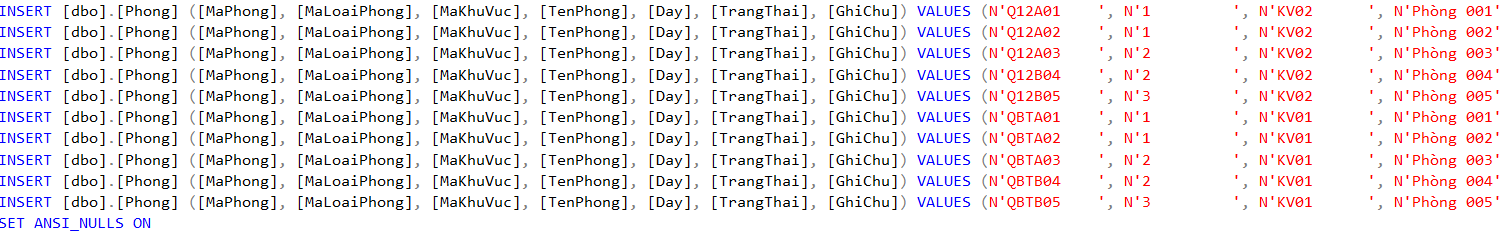
* **Bảng đăng nhập**



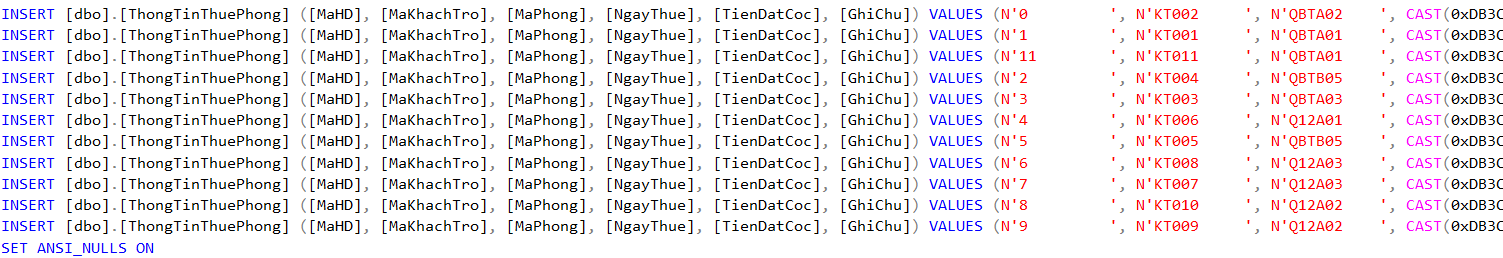
* **Bảng thông tin khách**



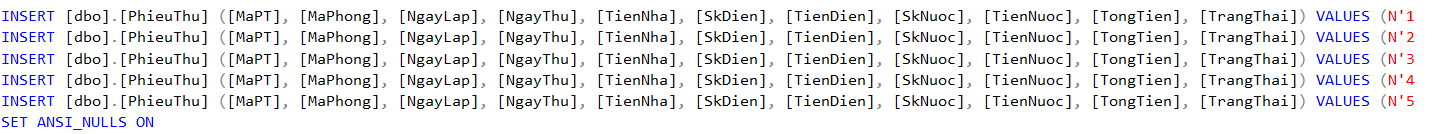
* **Bảng phòng**



* **Bảng thông tin thuê phòng**



* **Bảng phiếu thu**

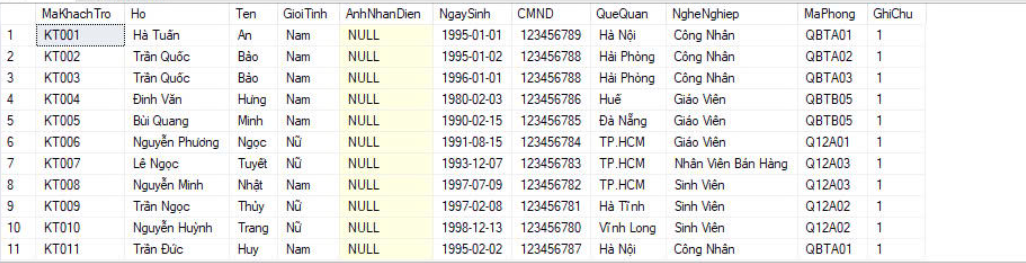


**3.3. In bảng dữ liệu**

* **Bảng thông tin khách**

SELECT \*

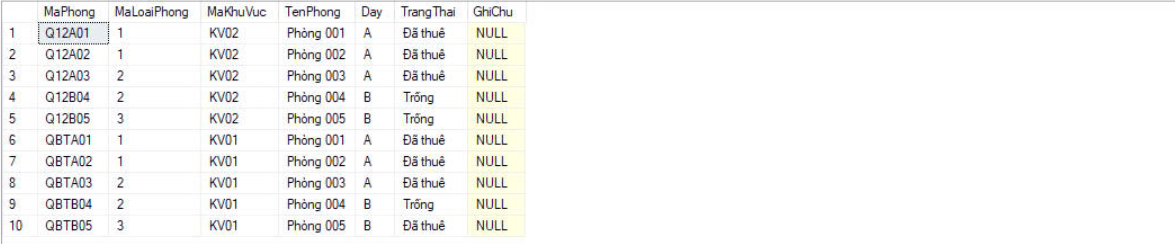
FROM dbo.ThongTinKhach;



* **Bảng phòng**

SELECT \*

FROM dbo.Phong;



* **Bảng dịch vụ**

SELECT \*

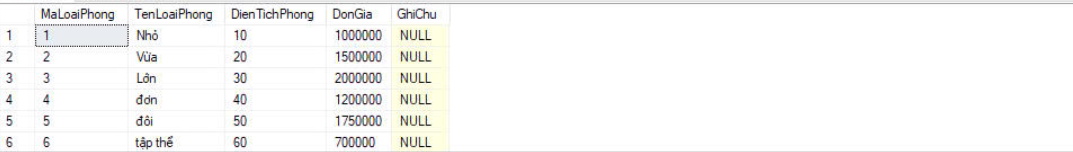
FROM dbo.DichVu;



* **Bảng loại phòng**

SELECT \*

FROM dbo.LoaiPhong;



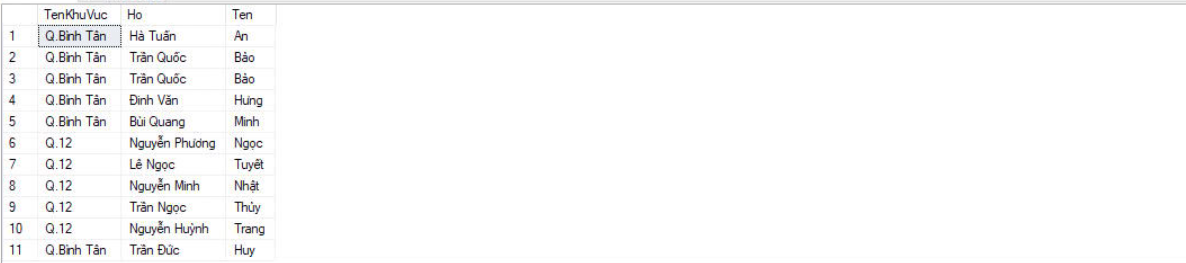
* **Bảng khu vực**

SELECT k.TenKhuVuc, t.Ho, t.Ten

FROM dbo.ThongTinKhach t

JOIN dbo.Phong p ON t.MaPhong = p.MaPhong

JOIN dbo.KhuVuc k ON p.MaKhuVuc = k.MaKhuVuc;



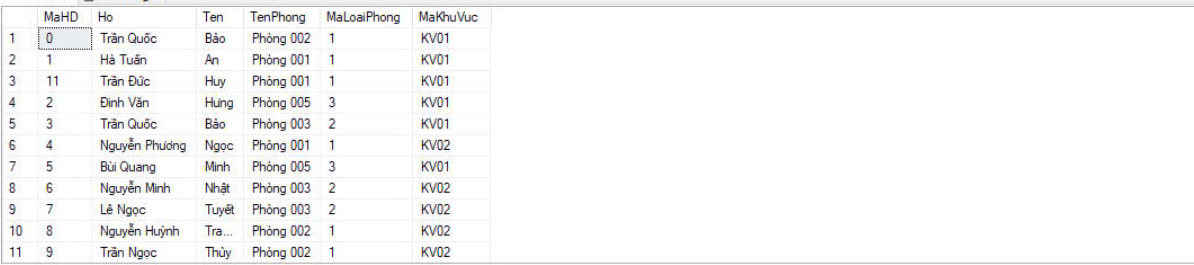
* **Bảng thông tin thuê phòng**

SELECT t.MaHD, k.Ho, k.Ten, p.TenPhong, p.MaLoaiPhong, p.MaKhuVuc

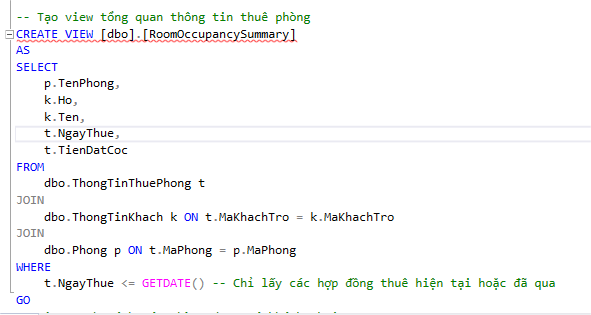
FROM dbo.ThongTinThuePhong t

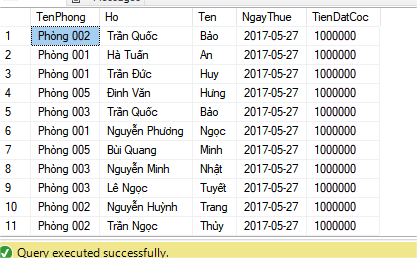
JOIN dbo.ThongTinKhach k ON t.MaKhachTro = k.MaKhachTro

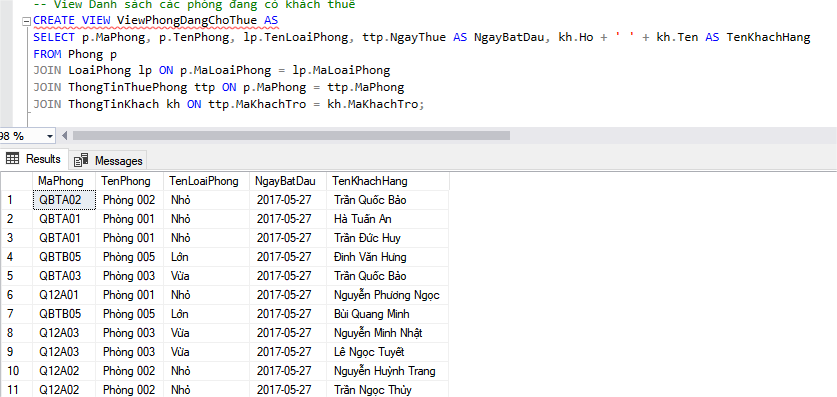
JOIN dbo.Phong p ON t.MaPhong = p.MaPhong;

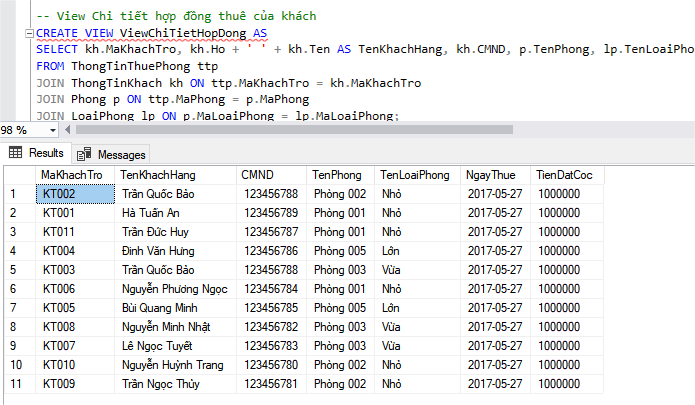


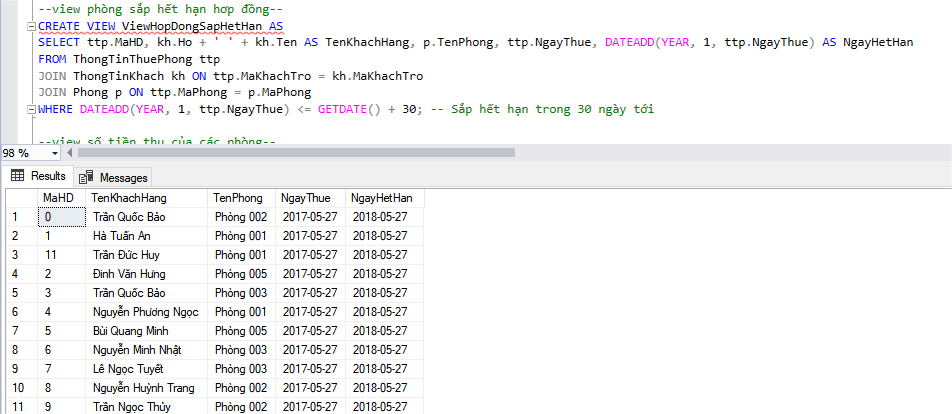
**3.4. CREATE VIEW**

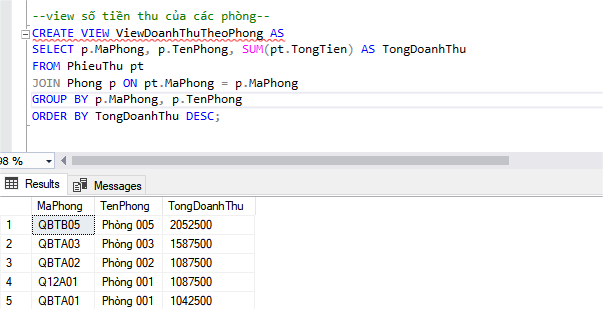










****

**3.5. : Xây dựng các procedure và chạy EXEC GetAl\_**

Các Procedure có trong code:

-- Đảm bảo bảng có đầy đủ các cột cần thiết

ALTER TABLE DangNhap ADD Username NVARCHAR(50);

ALTER TABLE GhiNho ADD NoiDung NVARCHAR(255);

ALTER TABLE DangNhap ADD Username NVARCHAR(50), Password NVARCHAR(50);

ALTER TABLE PhieuThu ADD MaPhieu INT, SoTien MONEY;

GO

Thêm cột Username vào bảng DangNhap (kiểu NVARCHAR(50)).

· Thêm cột NoiDung vào bảng GhiNho (kiểu NVARCHAR(255)).

· Thêm cột Username và Password vào bảng DangNhap (cả hai đều kiểu NVARCHAR(50)).

· Thêm cột MaPhieu (kiểu INT) và SoTien (kiểu MONEY) vào bảng PhieuThu.

-- PROCEDURE thêm dữ liệu vào bảng GhiNho

CREATE PROCEDURE Insert\_GhiNho

@ID INT,

@NoiDung NVARCHAR(255)

AS

BEGIN

INSERT INTO GhiNho (ID, NoiDung) VALUES (@ID, @NoiDung);

END;

GO

Insert\_GhiNho thêm một bản ghi vào bảng GhiNho với giá trị ID và NoiDung được truyền vào khi gọi thủ tục.

-- PROCEDURE cập nhật dữ liệu trong bảng GhiNho

CREATE PROCEDURE Update\_GhiNho

@ID INT,

@NoiDung NVARCHAR(255)

AS

BEGIN

UPDATE GhiNho SET NoiDung = @NoiDung WHERE ID = @ID;

END;

GO

Update\_GhiNho cập nhật nội dung (NoiDung) của một bản ghi trong bảng GhiNho theo ID được truyền vào khi gọi thủ tục.

-- PROCEDURE xóa dữ liệu trong bảng GhiNho

CREATE PROCEDURE Delete\_GhiNho

@ID INT

AS

BEGIN

DELETE FROM GhiNho WHERE ID = @ID;

END;

GO

Delete\_GhiNho xóa một bản ghi trong bảng GhiNho theo ID được truyền vào khi gọi thủ tục.

-- PROCEDURE thêm dữ liệu vào bảng DangNhap

CREATE PROCEDURE Insert\_DangNhap

@Username NVARCHAR(50),

@Password NVARCHAR(50)

AS

BEGIN

INSERT INTO DangNhap (Username, Password) VALUES (@Username, @Password);

END;

GO

nsert\_DangNhap thêm một bản ghi mới vào bảng DangNhap với giá trị Username và Password được truyền vào khi gọi thủ tục.

-- PROCEDURE kiểm tra đăng nhập

CREATE PROCEDURE Check\_DangNhap

@Username NVARCHAR(50),

@Password NVARCHAR(50)

AS

BEGIN

SELECT \* FROM DangNhap WHERE Username = @Username AND Password = @Password;

END;

GO

Check\_DangNhap kiểm tra thông tin đăng nhập bằng cách truy vấn bảng DangNhap và tìm bản ghi có Username và Password khớp với giá trị được truyền vào khi gọi thủ tục.

-- PROCEDURE thêm dữ liệu vào bảng PhieuThu

CREATE PROCEDURE Insert\_PhieuThu

@MaPhieu INT,

@SoTien MONEY

AS

BEGIN

INSERT INTO PhieuThu (MaPhieu, SoTien) VALUES (@MaPhieu, @SoTien);

END;

GO

Insert\_PhieuThu thêm một bản ghi mới vào bảng PhieuThu với giá trị MaPhieu và SoTien được truyền vào khi gọi thủ tục.

-- PROCEDURE cập nhật số tiền trong bảng PhieuThu

CREATE PROCEDURE Update\_PhieuThu

@MaPhieu INT,

@SoTien MONEY

AS

BEGIN

UPDATE PhieuThu SET SoTien = @SoTien WHERE MaPhieu = @MaPhieu;

END;

GO

Update\_PhieuThu cập nhật giá trị SoTien trong bảng PhieuThu theo MaPhieu được truyền vào khi gọi thủ tục.

-- PROCEDURE xóa dữ liệu trong bảng PhieuThu

CREATE PROCEDURE Delete\_PhieuThu

@MaPhieu INT

AS

BEGIN

DELETE FROM PhieuThu WHERE MaPhieu = @MaPhieu;

END;

GO

Delete\_PhieuThu xóa một bản ghi trong bảng PhieuThu theo MaPhieu được truyền vào khi gọi thủ tục.

-- PROCEDURE lấy tất cả dữ liệu từ bảng GhiNho

CREATE PROCEDURE GetAll\_GhiNho

AS

BEGIN

SELECT \* FROM GhiNho;

END;

GO

GetAll\_GhiNho truy vấn và trả về tất cả các bản ghi từ bảng GhiNho.

-- PROCEDURE lấy tất cả dữ liệu từ bảng DangNhap

CREATE PROCEDURE GetAll\_DangNhap

AS

BEGIN

SELECT \* FROM DangNhap;

END;

GO

GetAll\_DangNhap truy vấn và trả về tất cả các bản ghi từ bảng DangNhap.

-- PROCEDURE lấy tất cả dữ liệu từ bảng PhieuThu

CREATE PROCEDURE GetAll\_PhieuThu

AS

BEGIN

SELECT \* FROM PhieuThu;

END;

GO

GetAll\_PhieuThu truy vấn và trả về tất cả các bản ghi từ bảng PhieuThu.

-- PROCEDURE lấy thông tin GhiNho theo ID

CREATE PROCEDURE Get\_GhiNho\_ByID

@ID INT

AS

BEGIN

SELECT \* FROM GhiNho WHERE ID = @ID;

END;

GO

Get\_GhiNho\_ByID truy vấn và trả về bản ghi từ bảng GhiNho có ID khớp với giá trị được truyền vào khi gọi thủ tục.

-- PROCEDURE lấy thông tin PhieuThu theo MaPhieu

CREATE PROCEDURE Get\_PhieuThu\_ByMaPhieu

@MaPhieu INT

AS

BEGIN

SELECT \* FROM PhieuThu WHERE MaPhieu = @MaPhieu;

END;

GO

Get\_PhieuThu\_ByMaPhieu truy vấn và trả về bản ghi từ bảng PhieuThu có MaPhieu khớp với giá trị được truyền vào khi gọi thủ tục.

-- PROCEDURE lấy thông tin DangNhap theo Username

CREATE PROCEDURE Get\_DangNhap\_ByUsername

@Username NVARCHAR(50)

AS

BEGIN

SELECT \* FROM DangNhap WHERE Username = @Username;

END;

GO

Get\_DangNhap\_ByUsername truy vấn và trả về bản ghi từ bảng DangNhap có Username khớp với giá trị được truyền vào khi gọi thủ tục.

**=> Khi muốn in ra kết quả chỉ cần gõ câu lệnh EXEC GetAll\_(procedure cần hiển thị) thì sql sẽ tự động in ra màn hình**

**3.6. Xây dựng các trigger**

-- TRIGGER kiểm tra trước khi xóa dữ liệu trong bảng PhieuThu

CREATE TRIGGER trg\_PreventDelete\_PhieuThu

ON PhieuThu

INSTEAD OF DELETE

AS

BEGIN

PRINT 'Không thể xóa dữ liệu trong bảng PhieuThu!';

END;

GO

**trg\_PreventDelete\_PhieuThu**

* **Mục đích**: Ngăn không cho xóa dữ liệu trong bảng PhieuThu.
* **Cách hoạt động**: Khi có lệnh DELETE trên bảng PhieuThu, trigger này sẽ chặn thao tác và hiển thị thông báo 'Không thể xóa dữ liệu trong bảng PhieuThu!'.

-- TRIGGER tự động cập nhật số tiền khi thêm vào bảng PhieuThu

CREATE TRIGGER trg\_AfterInsert\_PhieuThu

ON PhieuThu

AFTER INSERT

AS

BEGIN

UPDATE PhieuThu

SET SoTien = SoTien + 1000 -- Ví dụ: tự động cộng thêm 1000 vào mỗi giao dịch

FROM PhieuThu p

INNER JOIN inserted i ON p.MaPhieu = i.MaPhieu;

END;

GO

**trg\_AfterInsert\_PhieuThu**

* **Mục đích**: Tự động cập nhật số tiền khi thêm mới một phiếu thu.
* **Cách hoạt động**: Khi một bản ghi mới được thêm vào bảng PhieuThu, trigger sẽ cộng thêm 1000 vào cột SoTien của phiếu thu vừa được chèn.

-- Trigger cho bảng Phong

CREATE TRIGGER trg\_AfterInsert\_Phong

ON dbo.Phong

FOR INSERT

AS

BEGIN

DECLARE @MaPhong nchar(10), @TrangThai nvarchar(10)

DECLARE cursor\_phong CURSOR FOR SELECT MaPhong, TrangThai FROM inserted

OPEN cursor\_phong

FETCH NEXT FROM cursor\_phong INTO @MaPhong, @TrangThai

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'Phòng mới: ' + @MaPhong + CASE WHEN @TrangThai = N'Đã thuê' THEN ' - Đã thuê' ELSE ' - Phòng trống' END

FETCH NEXT FROM cursor\_phong INTO @MaPhong, @TrangThai

END

CLOSE cursor\_phong

DEALLOCATE cursor\_phong

END

GO

**trg\_AfterInsert\_Phong**

* **Mục đích**: Hiển thị thông báo khi thêm một phòng mới vào hệ thống.
* **Cách hoạt động**: Khi một phòng mới được thêm vào bảng Phong, trigger này sẽ in ra thông báo về trạng thái của phòng (đã thuê hoặc trống).

-- Trigger cho bảng ThongTinThuePhong

CREATE TRIGGER trg\_AfterUpdate\_ThongTinThuePhong

ON dbo.ThongTinThuePhong

FOR UPDATE

AS

BEGIN

DECLARE @MaHD nchar(10), @MaKhachTro nchar(10), @NgayThue date

DECLARE cursor\_thue CURSOR FOR SELECT MaHD, MaKhachTro, NgayThue FROM inserted

OPEN cursor\_thue

FETCH NEXT FROM cursor\_thue INTO @MaHD, @MaKhachTro, @NgayThue

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT '🔄 Hợp đồng: ' + @MaHD + ' cập nhật. Khách: ' + @MaKhachTro + ', Ngày thuê: ' + CONVERT(nvarchar(10), @NgayThue, 103)

FETCH NEXT FROM cursor\_thue INTO @MaHD, @MaKhachTro, @NgayThue

END

CLOSE cursor\_thue

DEALLOCATE cursor\_thue

END

GO

**trg\_AfterUpdate\_ThongTinThuePhong**

* **Mục đích**: Ghi nhận thay đổi thông tin thuê phòng.
* **Cách hoạt động**: Khi hợp đồng thuê phòng (ThongTinThuePhong) bị cập nhật, trigger này sẽ hiển thị thông tin về hợp đồng, khách trọ và ngày thuê.

-- Trigger cho bảng ThongTinKhach

CREATE TRIGGER trg\_AfterInsert\_ThongTinKhach

ON dbo.ThongTinKhach

FOR INSERT

AS

BEGIN

DECLARE @MaKhachTro nchar(10), @MaPhong nchar(10)

DECLARE cursor\_khach CURSOR FOR SELECT MaKhachTro, MaPhong FROM inserted

OPEN cursor\_khach

FETCH NEXT FROM cursor\_khach INTO @MaKhachTro, @MaPhong

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT '🆕 Khách mới: ' + @MaKhachTro + ' thuê phòng: ' + @MaPhong

FETCH NEXT FROM cursor\_khach INTO @MaKhachTro, @MaPhong

END

CLOSE cursor\_khach

DEALLOCATE cursor\_khach

END

GO

**trg\_AfterInsert\_ThongTinKhach**

* **Mục đích**: Thông báo khi có khách mới thuê phòng.
* **Cách hoạt động**: Khi một khách trọ mới được thêm vào bảng ThongTinKhach, trigger sẽ in ra thông báo với mã khách và mã phòng mà khách đó thuê.

**3.7. Phân quyền và bảo vệ cơ sở dữ liệu**

CREATE LOGIN TenLogin WITH PASSWORD = 'MatKhauAnToan'

GO

USE [QLPhongTro]

GO

CREATE USER TenNguoiDung FOR LOGIN TenLogin

GO

GRANT SELECT ON dbo.Phong TO TenNguoiDung

GO

GRANT INSERT ON dbo.ThongTinKhach TO TenNguoiDung

GO

GRANT EXECUTE ON dbo.SaveP TO TenNguoiDung

GO

GRANT EXECUTE ON dbo.ChangePass TO TenNguoiDung

GO

ALTER LOGIN TenLogin WITH CHECK\_POLICY = ON

GO

BACKUP DATABASE [QLPhongTro] TO DISK = 'C:\Backup\QLPhongTro.bak'

GO

### 1. ****Tạo Login và User****

* **Login** (CREATE LOGIN) là đối tượng xác thực trong SQL Server, cho phép người dùng đăng nhập vào SQL Server. Sau khi tạo login, ta có thể tạo **user** trong một cơ sở dữ liệu cụ thể để liên kết với login này.
* **User** (CREATE USER) là đối tượng trong cơ sở dữ liệu, cho phép người dùng thực hiện các thao tác trên cơ sở dữ liệu đó. Việc tạo user trong cơ sở dữ liệu QLPhongTro và liên kết với login TenLogin giúp quản lý quyền truy cập vào cơ sở dữ liệu này.

### 2. ****Cấp Quyền Truy Cập****

* Cấp quyền (GRANT) cho phép người dùng thực hiện các thao tác nhất định trên các đối tượng trong cơ sở dữ liệu như bảng hoặc thủ tục lưu trữ.
  + GRANT SELECT: Cho phép người dùng truy vấn dữ liệu từ bảng Phong.
  + GRANT INSERT: Cho phép người dùng chèn dữ liệu vào bảng ThongTinKhach.
  + GRANT EXECUTE: Cho phép người dùng thực thi thủ tục lưu trữ SaveP và ChangePass. Quyền này rất quan trọng để người dùng có thể thực thi các quy trình logic đã được định nghĩa trong các thủ tục.

### 3. ****Kiểm Tra Chính Sách Mật Khẩu****

* WITH CHECK\_POLICY = ON: Điều này kích hoạt chính sách kiểm tra mật khẩu cho login. Chính sách này đảm bảo rằng mật khẩu của người dùng đáp ứng các yêu cầu bảo mật như độ dài tối thiểu, phức tạp (bao gồm chữ hoa, chữ thường, số, ký tự đặc biệt). Điều này giúp bảo vệ hệ thống khỏi các mối đe dọa từ việc sử dụng mật khẩu yếu hoặc dễ đoán.

### 4. ****Sao Lưu Dữ Liệu****

* **Sao lưu cơ sở dữ liệu** (BACKUP DATABASE) là một hành động quan trọng để bảo vệ dữ liệu. Việc sao lưu cơ sở dữ liệu QLPhongTro giúp đảm bảo dữ liệu không bị mất trong trường hợp có sự cố hoặc lỗi hệ thống. Sao lưu dữ liệu vào file QLPhongTro.bak tại C:\Backup\ giúp dễ dàng phục hồi cơ sở dữ liệu nếu cần.

### Phân tích chung:

**Quản lý người dùng và bảo mật**: Các thao tác như tạo login, user, và bật chính sách mật khẩu mạnh giúp bảo vệ hệ thống và cơ sở dữ liệu khỏi các mối nguy cơ bảo mật. Việc kiểm soát quyền truy cập giúp đảm bảo người dùng chỉ có thể thực hiện các thao tác cần thiết, giảm thiểu rủi ro sai sót hoặc tấn công.

**Cấp quyền chi tiết**: Việc cấp quyền truy cập theo từng thao tác cụ thể (SELECT, INSERT, EXECUTE) cho người dùng giúp kiểm soát rõ ràng và chặt chẽ quyền hạn của họ đối với các đối tượng trong cơ sở dữ liệu, từ đó bảo vệ dữ liệu và các quy trình quan trọng trong hệ thống.

**Bảo vệ dữ liệu**: Sao lưu cơ sở dữ liệu là một phần quan trọng trong chiến lược quản lý dữ liệu, đảm bảo rằng dữ liệu có thể được phục hồi khi cần thiết và giảm thiểu nguy cơ mất mát dữ liệu.

Phân quyền cho người dùng:

-- Tạo tài khoản người dùng

CREATE LOGIN UserQuanLy WITH PASSWORD = 'QuanLy@123';

CREATE LOGIN UserNhanVien WITH PASSWORD = 'NhanVien@123';

CREATE LOGIN UserKeToan WITH PASSWORD = 'KeToan@123';

GO

-- Tạo người dùng trong cơ sở dữ liệu

CREATE USER QuanLy FOR LOGIN UserQuanLy;

CREATE USER NhanVien FOR LOGIN UserNhanVien;

CREATE USER KeToan FOR LOGIN UserKeToan;

GO

-- Cấp quyền cho UserQuanLy (Quản lý có toàn quyền trên các bảng)

GRANT CONTROL ON DATABASE::YourDatabase TO QuanLy;

GRANT CONTROL ON SCHEMA::dbo TO QuanLy;

GO

-- Cấp quyền cho UserNhanVien (Nhân viên chỉ có quyền thao tác dữ liệu nhưng không thể xóa)

GRANT SELECT, INSERT, UPDATE ON dbo.Phong TO NhanVien;

GRANT SELECT, INSERT, UPDATE ON dbo.ThongTinThuePhong TO NhanVien;

GRANT SELECT, INSERT, UPDATE ON dbo.ThongTinKhach TO NhanVien;

GO

-- Cấp quyền cho UserKeToan (Kế toán chỉ có quyền đọc dữ liệu và thêm dữ liệu)

GRANT SELECT, INSERT ON dbo.PhieuThu TO KeToan;

GRANT SELECT, INSERT ON dbo.HoaDon TO KeToan;

GO

-- Ngăn UserNhanVien xóa dữ liệu trong bảng PhieuThu

DENY DELETE ON dbo.PhieuThu TO NhanVien;

GO

-- Ngăn UserKeToan xóa hoặc sửa dữ liệu trong bảng HoaDon

DENY DELETE, UPDATE ON dbo.HoaDon TO KeToan;

GO

**CHƯƠNG IV: ỨNG DỤNG, ĐÁNH GIÁ VÀ ĐỊNH HƯỚNG TƯƠNG LAI**

### ****Đánh giá Hệ Thống Quản Lý Phòng Trọ****

Hệ thống hiện tại đã được thiết kế để quản lý thông tin phòng trọ, khách hàng, hợp đồng thuê và các giao dịch tài chính. Nó sử dụng **Trigger** để tự động cập nhật dữ liệu và **phân quyền** để bảo vệ thông tin. Tuy nhiên, để đánh giá tổng quan về mức độ phù hợp với thực tế, cần phân tích chi tiết hơn về **ưu điểm, nhược điểm, và hướng phát triển** của hệ thống.

## **1. Nhận xét tổng quan về hệ thống**

Hệ thống **cơ bản đáp ứng được nhu cầu quản lý nhà trọ** với các tính năng như:  
✅ **Quản lý phòng** (trạng thái phòng: trống, đã thuê)  
✅ **Quản lý hợp đồng thuê** (thời gian thuê, thông tin khách)  
✅ **Quản lý giao dịch tài chính** (phiếu thu tiền phòng)  
✅ **Bảo mật dữ liệu** thông qua phân quyền người dùng  
✅ **Trigger tự động cập nhật** dữ liệu khi có thay đổi

Tuy nhiên, để vận hành trơn tru trong thực tế, hệ thống cần **cải thiện hiệu suất, bảo mật và giao diện người dùng**.

## **2. Ưu điểm của hệ thống**

### ****2.1. Hoạt động tự động hóa tốt****

* Sử dụng **Trigger** để cập nhật trạng thái phòng, hợp đồng thuê mà không cần nhập thủ công.
* Trigger ngăn **xóa dữ liệu quan trọng** như phiếu thu, đảm bảo tính toàn vẹn dữ liệu.

**🔹 Ví dụ:** Khi thêm một khách thuê phòng, Trigger sẽ tự động cập nhật trạng thái phòng đó thành "Đã thuê".

### ****2.2. Phân quyền chặt chẽ****

Hệ thống có **các nhóm tài khoản với quyền hạn khác nhau**:

* **Quản lý**: Toàn quyền truy cập, chỉnh sửa dữ liệu.
* **Nhân viên**: Chỉ có quyền thêm/sửa dữ liệu nhưng không thể xóa.
* **Kế toán**: Chỉ có quyền xem và nhập dữ liệu tài chính.

**🔹 Ví dụ:** Nhân viên không thể xóa hợp đồng thuê để tránh gian lận hoặc sai sót dữ liệu.

### ****2.3. Cấu trúc dữ liệu linh hoạt****

* Hệ thống sử dụng **cơ sở dữ liệu quan hệ** (SQL), giúp dễ mở rộng thêm bảng mới nếu cần.
* Có thể dễ dàng **tích hợp thêm các tính năng** như quản lý điện nước, dịch vụ kèm theo.

## **3. Nhược điểm và hạn chế**

### ****3.1. Hiệu suất chưa tối ưu****

❌ **Sử dụng CURSOR trong Trigger** có thể làm giảm hiệu suất khi dữ liệu lớn.

* **Cách khắc phục**: Thay thế CURSOR bằng **câu lệnh SQL truy vấn trực tiếp**

### ****3.2. Chưa có hệ thống Log thay đổi dữ liệu****

❌ **Không theo dõi được ai đã thay đổi dữ liệu nào và khi nào.**

* Nếu một nhân viên **sửa hợp đồng thuê**, hệ thống **không lưu lại lịch sử sửa đổi**, dễ gây thất thoát dữ liệu.

🔹 **Cách khắc phục**:

* Tạo **bảng Log thay đổi dữ liệu** để lưu thông tin các thao tác chỉnh sửa.
* Dùng **Trigger ghi Log** mỗi khi có UPDATE hoặc DELETE.

### ****3.3. Bảo mật chưa cao****

❌ **Dữ liệu quan trọng như số tiền chưa được mã hóa.**

* Hiện tại, số tiền trong bảng **PhieuThu** có thể bị xem bởi tất cả nhân viên kế toán, dễ dẫn đến rủi ro rò rỉ dữ liệu.
* **Cách khắc phục**: Mã hóa dữ liệu tài chính bằng **AES hoặc SHA-256**.

### ****3.4. Thiếu giao diện người dùng****

❌ **Hệ thống chỉ chạy bằng SQL, không có giao diện thân thiện với nhân viên.**

* Nhân viên phải nhập lệnh SQL trực tiếp, gây khó khăn trong vận hành thực tế.

🔹 **Cách khắc phục:**

* Xây dựng **giao diện Web hoặc ứng dụng** để nhân viên thao tác dễ dàng.
* Dùng **Power BI hoặc Excel kết nối SQL** để nhân viên kế toán dễ theo dõi báo cáo doanh thu.

## **4. Đề xuất phát triển trong tương lai**

### 🔹 ****Cải thiện hiệu suất****

✅ **Thay CURSOR bằng truy vấn SQL tối ưu**.  
✅ **Tạo Index cho các bảng có dữ liệu lớn** (ví dụ: Hợp đồng thuê, Phiếu thu).

### 🔹 ****Nâng cấp bảo mật****

✅ **Mã hóa dữ liệu quan trọng** (số tiền, thông tin khách thuê).  
✅ **Thêm hệ thống Log thay đổi dữ liệu**.  
✅ **Hạn chế quyền truy cập trực tiếp vào cơ sở dữ liệu**.

### 🔹 ****Phát triển thêm tính năng****

✅ **Xây dựng giao diện Web/App** cho nhân viên và khách thuê.  
✅ **Tích hợp thanh toán online** (Momo, VietQR) để khách hàng thanh toán tiền phòng trực tuyến.  
✅ **Hỗ trợ quản lý hóa đơn điện, nước, internet**, giúp quản lý dễ dàng hơn.  
✅ **Tích hợp hệ thống gửi thông báo SMS/Zalo** nhắc nhở khách hàng khi sắp đến hạn thanh toán tiền phòng.

## **5. Đánh giá mức độ phù hợp với thực tế**

🔹 **Ưu điểm**: Hệ thống phù hợp với các nhà trọ **quy mô nhỏ và vừa**, giúp tự động hóa quản lý hợp đồng và tài chính.  
🔹 **Hạn chế**: Chưa hỗ trợ tốt **quy mô lớn** do hạn chế về hiệu suất và bảo mật.  
🔹 **Hướng phát triển**: Nếu cải thiện các vấn đề trên, hệ thống có thể **triển khai cho nhà trọ lớn hoặc khách sạn nhỏ**.

**KẾT LUẬN**

Trong bối cảnh công nghệ thông tin ngày càng phát triển và nhu cầu số hóa quy trình quản lý ngày càng gia tăng, việc xây dựng một hệ thống quản lý phòng trọ dựa trên cơ sở dữ liệu SQL là một giải pháp thiết thực và hiệu quả. Bài báo cáo này đã đi sâu vào phân tích, thiết kế và triển khai một hệ thống quản lý phòng trọ, giúp tối ưu hóa quá trình lưu trữ, truy xuất và xử lý dữ liệu liên quan đến khách thuê, hợp đồng, thanh toán và các yếu tố vận hành khác.

Thông qua quá trình nghiên cứu và thực hiện, nhóm chúng tôi đã xây dựng được một mô hình cơ sở dữ liệu quan hệ chặt chẽ, đảm bảo tính toàn vẹn dữ liệu, giảm thiểu trùng lặp và sai sót trong quá trình quản lý. Việc áp dụng các công cụ của SQL như **Stored Procedures, Triggers, Views** và **các cơ chế phân quyền** không chỉ giúp tự động hóa quy trình mà còn nâng cao tính bảo mật của hệ thống, hạn chế tối đa các thao tác thủ công có thể dẫn đến nhầm lẫn hoặc thất thoát dữ liệu.

Hệ thống quản lý phòng trọ này đã chứng minh được những lợi ích đáng kể so với các phương pháp quản lý truyền thống như sử dụng sổ sách hoặc bảng tính Excel. Những ưu điểm nổi bật có thể kể đến bao gồm:

* **Tính chính xác và toàn vẹn dữ liệu**: Các ràng buộc toàn vẹn (Primary Key, Foreign Key, Unique, Check) giúp đảm bảo dữ liệu không bị trùng lặp hoặc nhập sai.
* **Tính tự động hóa cao**: Các Trigger giúp cập nhật trạng thái phòng, hợp đồng thuê và hóa đơn mà không cần thao tác thủ công, giảm thiểu sai sót.
* **Tối ưu hóa quy trình quản lý**: Hệ thống giúp chủ trọ dễ dàng theo dõi danh sách phòng, tình trạng thuê, hợp đồng và công nợ một cách trực quan, nhanh chóng.
* **Bảo mật dữ liệu**: Phân quyền người dùng giúp kiểm soát chặt chẽ quyền truy cập, ngăn chặn rủi ro rò rỉ hoặc chỉnh sửa dữ liệu trái phép.
* **Khả năng mở rộng linh hoạt**: Hệ thống được thiết kế để có thể tích hợp với các công nghệ khác như thanh toán trực tuyến, ứng dụng di động hoặc phân tích dữ liệu nâng cao trong tương lai.

Tuy nhiên, trong quá trình triển khai, chúng tôi cũng nhận thấy một số hạn chế cần được cải thiện để hệ thống trở nên hoàn thiện hơn:

1. **Hiệu suất hệ thống**: Khi lượng dữ liệu tăng lên đáng kể, hiệu suất truy vấn có thể bị ảnh hưởng. Việc tối ưu hóa truy vấn SQL, sử dụng Indexing và các kỹ thuật phân mảnh dữ liệu có thể giúp cải thiện tốc độ xử lý.
2. **Thiếu giao diện trực quan**: Hiện tại, hệ thống chỉ có thể sử dụng qua truy vấn SQL, gây khó khăn cho những người không chuyên về kỹ thuật. Việc xây dựng một giao diện web hoặc ứng dụng di động sẽ giúp hệ thống trở nên thân thiện và dễ sử dụng hơn.
3. **Chưa có hệ thống log theo dõi thay đổi dữ liệu**: Hệ thống chưa có cơ chế ghi nhận lịch sử thay đổi, khiến việc kiểm soát các thao tác chỉnh sửa, xóa dữ liệu trở nên khó khăn. Việc bổ sung một **bảng Log dữ liệu** kết hợp với Trigger ghi nhận các thay đổi sẽ giúp hệ thống minh bạch và an toàn hơn.
4. **Chưa tích hợp thanh toán trực tuyến**: Trong thực tế, nhiều khách thuê mong muốn có phương thức thanh toán linh hoạt hơn như **Momo, VietQR, Internet Banking**. Việc bổ sung cổng thanh toán sẽ giúp tăng trải nghiệm người dùng và giảm thiểu rủi ro thanh toán chậm.

### ****Hướng phát triển trong tương lai****

Dựa trên những ưu điểm và hạn chế trên, hệ thống có thể được cải thiện và mở rộng theo các hướng sau:

* **Cải thiện hiệu suất và tối ưu hóa truy vấn**: Sử dụng Indexing, phân vùng dữ liệu, tối ưu hóa Execution Plan để đảm bảo hệ thống có thể xử lý dữ liệu lớn mà vẫn hoạt động nhanh chóng.
* **Xây dựng giao diện người dùng trực quan**: Phát triển **ứng dụng web hoặc mobile** giúp chủ trọ và khách thuê dễ dàng thao tác, tra cứu thông tin mà không cần truy vấn SQL trực tiếp.
* **Nâng cao bảo mật hệ thống**: Áp dụng các thuật toán mã hóa dữ liệu nhạy cảm (AES, SHA-256), bổ sung tính năng xác thực hai lớp (2FA) để bảo vệ tài khoản người dùng.
* **Tích hợp thanh toán điện tử**: Kết nối với các cổng thanh toán online giúp khách thuê có thể **thanh toán tiền phòng, điện, nước trực tiếp** qua ứng dụng, tăng tính thuận tiện.
* **Áp dụng trí tuệ nhân tạo (AI) và dữ liệu lớn (Big Data)**: AI có thể được sử dụng để **dự đoán nhu cầu thuê phòng**, phân tích mô hình thanh toán của khách thuê, tối ưu hóa giá thuê để đảm bảo doanh thu ổn định.
* **Tích hợp hệ thống thông báo tự động**: Hệ thống có thể gửi **email, SMS hoặc thông báo qua Zalo** để nhắc nhở khách thuê về thời hạn thanh toán hoặc hợp đồng sắp hết hạn.

### ****Kết luận tổng thể****

Hệ thống quản lý phòng trọ bằng SQL đã đạt được những kết quả quan trọng trong việc hiện đại hóa quy trình quản lý, giúp chủ trọ tiết kiệm thời gian, tăng hiệu quả vận hành và giảm thiểu sai sót. Mặc dù còn tồn tại một số hạn chế về giao diện, hiệu suất và tính bảo mật, nhưng với những định hướng phát triển nêu trên, hệ thống hoàn toàn có khả năng trở thành một nền tảng quản lý chuyên nghiệp, không chỉ áp dụng cho nhà trọ mà còn có thể mở rộng sang các mô hình như **quản lý khách sạn, chung cư mini, ký túc xá hoặc dịch vụ lưu trú ngắn hạn**.

Với sự phát triển không ngừng của công nghệ, đặc biệt là các xu hướng **chuyển đổi số, trí tuệ nhân tạo và dữ liệu lớn**, hệ thống có tiềm năng nâng cấp thành một nền tảng thông minh, hỗ trợ ra quyết định và tối ưu hóa doanh thu. Để đạt được điều này, cần tiếp tục nghiên cứu, phát triển và cải tiến hệ thống để đáp ứng nhu cầu thực tế một cách tốt nhất.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. **Nguyễn Xuân Phương (2019)** - Hệ quản trị cơ sở dữ liệu - Lý thuyết và thực hành, Nhà xuất bản Bách Khoa Hà Nội.
2. **Phạm Hồng Thái (2021)** - SQL Server - Lập trình và quản trị, Nhà xuất bản Thông Tin & Truyền Thông.
3. **Elmasri & Navathe (2016)** - Fundamentals of Database Systems, Pearson Education.
4. **Microsoft Docs (2024)** - SQL Server Documentation, Microsoft.
5. **Date, C. J. (2019)** - An Introduction to Database Systems, Addison-Wesley.