**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 CSDL**

**ĐỀ TÀI: QUẢN LÝ DỰ ÁN**

**Giáo viên hướng dẫn: Ths. NGUYỄN NGỌC ÂN**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sv** | **Họ và tên** | **Lớp** |
| **1** | **1771020499** | **Nguyễn Phương Nam** | **CNTT-1702** |
| **2** | **1771020537** | **Tạ Văn Phong** | **CNTT-1702** |
| **3** | **1771020227** | **NguyễnTrường Giang** | **CNTT-1702** |

**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 CSDL**

**ĐỀ TÀI: QUẢN LÝ DỰ ÁN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020499 | Nguyễn Phương Nam | 22/09/2005 |  |  |
| 2 | 1771020537 | Tạ Văn Phong | 15/03/2005 |  |  |
| 3 | 1771020227 | Nguyễn Trường Giang | 6/03/2005 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

[LỜI NÓI ĐẦU 6](#_Toc193288327)

[Chương 1: Giới Thiệu Tổng Quan về Đề Tài 8](#_Toc193288328)

[1.1. Giới Thiệu Tổng Quan về Quản Lý Dự Án 8](#_Toc193288329)

[1.1.1. Khái Niệm Quản Lý Dự Án 8](#_Toc193288330)

[1.1.2. Các Yếu Tố Chính Của Quản Lý Dự Án 8](#_Toc193288331)

[1.1.3. Các Giai Đoạn Của Quản Lý Dự Án 8](#_Toc193288332)

[1.1.4. Vai Trò Của Nhà Quản Lý Dự Án 8](#_Toc193288333)

[1.2. Mục Đích của Đề Tài 9](#_Toc193288334)

[1.3. Lý Do Lựa Chọn Đề Tài 9](#_Toc193288335)

[1. 3.1. Tính thực tiễn và ứng dụng cao 9](#_Toc193288336)

[1.3.2. Giải quyết các vấn đề hiện tại 9](#_Toc193288337)

[1.3.3. Ứng dụng công nghệ để nâng cao hiệu quả quản lý 10](#_Toc193288338)

[1.3.4. Hỗ trợ công tác nghiên cứu và phát triển kỹ năng 10](#_Toc193288339)

[1.3.5. Tiềm năng mở rộng và phát triển 10](#_Toc193288340)

[1.4. Phạm Vi và Giới Hạn của Đề Tài 10](#_Toc193288341)

[1.5.Phương Pháp Nghiên Cứu 11](#_Toc193288342)

[1.5.1 Nghiên cứu tài liệu 11](#_Toc193288343)

[1.5.2. Phương pháp phân tích và thiết kế hệ thống 11](#_Toc193288344)

[1.5.3. Phương pháp xây dựng và triển khai 11](#_Toc193288345)

[1.5.4. Phương pháp kiểm thử và đánh giá 11](#_Toc193288346)

[Chương 2: Phân Tích và Thiết Kế Cơ Sở Dữ Liệu 13](#_Toc193288347)

[2.1. Xác Định Các Yêu Cầu Của Hệ Thống 13](#_Toc193288348)

[2.1.1. Yêu cầu chức năng 13](#_Toc193288349)

[2.1.2. Yêu cầu phi chức năng 13](#_Toc193288350)

[2.2. Thiết Kế Mô Hình Dữ Liệu (ERD) 14](#_Toc193288351)

[2.3. Chuyển Đổi Mô Hình Dữ Liệu Sang Mô Hình Vật Lý 14](#_Toc193288352)

[Chương 3: Triển Khai Cơ Sở Dữ Liệu 15](#_Toc193288353)

[3.1. Tạo Cơ Sở Dữ Liệu và Các Bảng 15](#_Toc193288354)

[3.2. Nhập Dữ Liệu Mẫu 16](#_Toc193288356)

[3.3Các truy vấn cơ bản và nâng cao 17](#_Toc193288357)

[Chương4: Lập trình T-SQL 21](#_Toc193288358)

[4.1.Tạo view từ cơ bản đến nâng cao 21](#_Toc193288359)

[4.2.Tạo index cần thiết cho các bảng 23](#_Toc193288360)

[4.3. Tạo function 23](#_Toc193288361)

[4.4.Tạo trigger để kiểm soát dữ liệu 25](#_Toc193288370)

[4.5. Xây dựng Stored Procedure 28](#_Toc193288371)

[Chương 5: Bảo mật và quản trị 30](#_Toc193288372)

[5.1. Tạo người dùng 30](#_Toc193288373)

[5.2. Quản lý sao lưu và phục hồi dữ liệu 30](#_Toc193288374)

[Chương 6: Kết luận 32](#_Toc193288375)

[1. Đánh giá ưu và nhược điểm của hệ thống: 32](#_Toc193288376)

[2. Đề xuất các hướng phát triển tiếp theo: 32](#_Toc193288377)

# LỜI NÓI ĐẦU

Trong thời đại công nghệ số và toàn cầu hóa, việc quản lý nhân sự, phòng ban, hoạt động và dự án đóng vai trò quan trọng trong sự phát triển bền vững của các doanh nghiệp và tổ chức. Một hệ thống quản lý hiệu quả không chỉ giúp tối ưu hóa việc phân bổ nguồn lực mà còn nâng cao năng suất lao động, cải thiện quy trình làm việc và hỗ trợ ra quyết định chính xác.

Tuy nhiên, trên thực tế, nhiều doanh nghiệp vẫn gặp khó khăn trong việc tổ chức nhân sự, phân công nhiệm vụ, theo dõi tiến độ dự án cũng như đánh giá hiệu quả làm việc. Các phương pháp quản lý thủ công hoặc sử dụng các công cụ rời rạc thường dẫn đến tình trạng dữ liệu phân tán, thiếu minh bạch và không tối ưu hóa được nguồn lực. Điều này có thể gây ra tình trạng chồng chéo công việc, thiếu trách nhiệm trong quản lý và giảm hiệu suất làm việc. Chính vì vậy, nhu cầu về một hệ thống quản lý nhân sự và dự án hiện đại, khoa học và hiệu quả trở nên cấp thiết hơn bao giờ hết.

Xuất phát từ thực tế đó, đề tài "Xây dựng hệ thống quản lý nhân sự, phòng ban, hoạt động và dự án" được thực hiện với mục tiêu cung cấp một giải pháp công nghệ nhằm hỗ trợ các doanh nghiệp và tổ chức quản lý nguồn nhân lực một cách chặt chẽ và hiệu quả hơn. Hệ thống này giúp lưu trữ và quản lý thông tin nhân sự, phân chia phòng ban hợp lý, theo dõi hoạt động, phân công nhiệm vụ và kiểm soát tiến độ dự án một cách khoa học. Đồng thời, hệ thống cũng cung cấp các công cụ báo cáo, thống kê trực quan giúp nhà quản lý dễ dàng đánh giá hiệu quả làm việc và ra quyết định nhanh chóng.

Để thực hiện đề tài này, nhóm nghiên cứu đã áp dụng nhiều phương pháp khác nhau, bao gồm nghiên cứu tài liệu, khảo sát thực tế, phân tích hệ thống, thiết kế và lập trình phần mềm, kiểm thử và đánh giá hiệu quả. Đề tài không chỉ giúp nâng cao kỹ năng chuyên môn về quản lý dữ liệu, thiết kế hệ thống, lập trình và kiểm thử phần mềm mà còn giúp nhóm nghiên cứu hiểu rõ hơn về cách thức vận hành một tổ chức, cách tổ chức công việc và quản lý nguồn nhân lực hiệu quả.

Hệ thống được xây dựng dựa trên các công nghệ hiện đại, có thể mở rộng và tích hợp với nhiều nền tảng khác nhau trong tương lai. Tuy nhiên, do thời gian và nguồn lực có hạn, hệ thống vẫn còn một số hạn chế cần được cải thiện. Chúng tôi mong nhận được những góp ý và phản hồi từ giảng viên, hội đồng đánh giá và các chuyên gia để tiếp tục hoàn thiện và nâng cao chất lượng sản phẩm.

# Chương 1: Giới Thiệu Tổng Quan về Đề Tài

## Giới Thiệu Tổng Quan về Quản Lý Dự Án

### 1.1.1. Khái Niệm Quản Lý Dự Án

Quản lý dự án là quá trình lập kế hoạch, tổ chức, thực hiện và giám sát các hoạt động nhằm đạt được mục tiêu cụ thể trong phạm vi, thời gian và ngân sách cho phép. Mục tiêu của quản lý dự án là đảm bảo dự án hoàn thành đúng tiến độ, trong phạm vi ngân sách và đạt chất lượng yêu cầu.

### 1.1.2. Các Yếu Tố Chính Của Quản Lý Dự Án

Một dự án thường bao gồm các yếu tố chính sau:

* Mục tiêu dự án: Xác định kết quả mong muốn đạt được.
* Phạm vi dự án: Xác định công việc cần thực hiện để hoàn thành mục tiêu.
* Nguồn lực: Bao gồm nhân lực, tài chính, công nghệ và vật tư cần thiết.
* Tiến độ và thời gian: Xác định các mốc thời gian và thời hạn hoàn thành.
* Ngân sách: Dự toán chi phí và quản lý ngân sách dự án.
* Quản lý rủi ro: Dự báo các vấn đề có thể phát sinh và đề ra phương án xử lý.

### 1.1.3. Các Giai Đoạn Của Quản Lý Dự Án

* Quản lý dự án thường được chia thành 5 giai đoạn chính:
* Khởi tạo: Xác định mục tiêu, phạm vi và các bên liên quan.
* Lập kế hoạch: Xây dựng kế hoạch chi tiết về nguồn lực, tiến độ và ngân sách.
* Thực hiện: Phân công công việc, triển khai kế hoạch và giám sát tiến độ.
* Kiểm soát: Theo dõi tiến độ, đánh giá hiệu quả và điều chỉnh nếu cần.
* Kết thúc: Hoàn thành dự án, tổng kết và đánh giá kết quả.

### 1.1.4. Vai Trò Của Nhà Quản Lý Dự Án

Nhà quản lý dự án có trách nhiệm:

* Xây dựng kế hoạch và phân bổ nguồn lực hợp lý.
* Giám sát tiến độ và chất lượng công việc.
* Quản lý rủi ro và giải quyết vấn đề phát sinh.
* Điều phối và giao tiếp với các bên liên quan.

## Mục Đích của Đề Tài

Đề tài này nhằm mục đích xây dựng một hệ thống quản lý nhân sự, phòng ban, hoạt động và dự án trong doanh nghiệp hoặc tổ chức. Cụ thể, hệ thống giúp:

✅ Tối ưu hóa quản lý nhân sự:

* Phân chia nhân viên vào các phòng ban một cách khoa học.
* Quản lý danh sách nhân viên, vai trò và phòng ban một cách hiệu quả.
* Xác định rõ vai trò của nhân viên phụ trách và người quản lý phòng ban.

✅ Nâng cao hiệu quả phân công công việc:

* Mỗi phòng ban chỉ tham gia vào một hoạt động duy nhất, giúp đảm bảo tính chuyên môn hóa.
* Một hoạt động có thể được phân bổ cho nhiều phòng ban để tăng tính hợp tác liên phòng.

✅ Quản lý và theo dõi dự án:

* Hỗ trợ việc phân công nhân viên vào các dự án một cách linh hoạt.
* Đảm bảo mỗi dự án có đủ nhân lực cần thiết.
* Cho phép nhân viên tham gia nhiều dự án để tận dụng tối đa kỹ năng và năng lực.

✅ Hỗ trợ ra quyết định cho nhà quản lý:

* Cung cấp dữ liệu về nhân sự, phòng ban, hoạt động và dự án một cách trực quan.
* Giúp nhà quản lý theo dõi tình trạng hoạt động của phòng ban và tiến độ dự án.
* Hỗ trợ lập kế hoạch nhân sự và điều phối nguồn lực hiệu quả.

✅ Tăng cường tính minh bạch và chuyên nghiệp trong quản lý:

* Hạn chế tình trạng chồng chéo nhiệm vụ giữa các phòng ban.
* Giúp nhân viên nắm rõ vai trò, trách nhiệm của mình trong tổ chức.
* Đảm bảo mọi hoạt động và dự án đều được theo dõi, đánh giá một cách minh bạch.

## Lý Do Lựa Chọn Đề Tài

### 1. 3.1. Tính thực tiễn và ứng dụng cao

* Hệ thống quản lý nhân sự và dự án là nhu cầu thiết yếu của các doanh nghiệp, tổ chức.
* Giúp quản lý chặt chẽ nguồn nhân lực, tối ưu hóa công việc và nâng cao hiệu suất làm việc.
* Ứng dụng linh hoạt trong nhiều mô hình tổ chức khác nhau như công ty, trường học, bệnh viện, cơ quan nhà nước,...

### 1.3.2. Giải quyết các vấn đề hiện tại

* Trong nhiều tổ chức, việc quản lý nhân sự, hoạt động và dự án còn thủ công, thiếu hệ thống khoa học.
* Dữ liệu phân tán, gây khó khăn trong việc theo dõi nhân sự, phân công công việc và quản lý tiến độ dự án.
* Thiếu minh bạch trong việc phân công nhiệm vụ và quản lý nhân viên.

### 1.3.3. Ứng dụng công nghệ để nâng cao hiệu quả quản lý

* Sử dụng hệ thống phần mềm giúp giảm thiểu sai sót trong quản lý nhân sự, phòng ban và dự án.
* Dữ liệu được số hóa, truy xuất dễ dàng, hỗ trợ nhà quản lý ra quyết định nhanh chóng.
* Có thể tích hợp với các hệ thống khác như chấm công, tính lương, đánh giá nhân sự để tạo thành một hệ sinh thái quản lý nhân sự toàn diện.

### 1.3.4. Hỗ trợ công tác nghiên cứu và phát triển kỹ năng

* Giúp nhóm nghiên cứu áp dụng các kiến thức về cơ sở dữ liệu, lập trình, quản lý dự án vào thực tế.
* Cải thiện kỹ năng phân tích, thiết kế hệ thống, lập trình và tối ưu thuật toán.
* Là tiền đề để phát triển các hệ thống quản lý nhân sự thông minh hơn trong tương lai.

### 1.3.5. Tiềm năng mở rộng và phát triển

* Hệ thống có thể mở rộng thêm các tính năng như AI hỗ trợ phân công nhân sự, báo cáo tự động, dự báo tiến độ dự án.
* Có thể triển khai trên nền tảng web, mobile để tiện lợi hơn cho người dùng.
* Mở rộng sang các lĩnh vực khác như quản lý tài nguyên doanh nghiệp (ERP), quản lý đào tạo nhân sự,...

## 1.4. Phạm Vi và Giới Hạn của Đề Tài

Phạm vi của đề tài này bao gồm việc nghiên cứu các phương pháp quản lý dự án phổ biến và phân tích các yếu tố ảnh hưởng đến thành công của dự án. Tuy nhiên, đề tài sẽ không đi sâu vào các chi tiết kỹ thuật cụ thể của từng phương pháp mà tập trung vào việc đánh giá hiệu quả và ứng dụng của chúng trong thực tế.

Giới hạn của đề tài là chỉ tập trung vào các phương pháp quản lý dự án hiện đại và không bao gồm các phương pháp truyền thống. Ngoài ra, đề tài sẽ không nghiên cứu về các công cụ và phần mềm hỗ trợ quản lý dự án một cách chi tiết.

## 1.5.Phương Pháp Nghiên Cứu

### 1.5.1 Nghiên cứu tài liệu

* Tìm hiểu về quản lý nhân sự và dự án
* Nghiên cứu các mô hình quản lý nhân sự, phòng ban trong doanh nghiệp.
* Khám phá các phương pháp quản lý dự án phổ biến (Agile, Scrum, Waterfall, Kanban,...).
* Xác định các yêu cầu cơ bản của một hệ thống quản lý nhân sự và dự án.
* Tham khảo các hệ thống tương tự
* Nghiên cứu các phần mềm quản lý dự án hiện có như Trello, Asana, Microsoft Project.
* Phân tích điểm mạnh, điểm yếu để rút kinh nghiệm trong quá trình thiết kế hệ thống.

### 1.5.2. Phương pháp phân tích và thiết kế hệ thống

* Thu thập yêu cầu
* Khảo sát nhu cầu thực tế của doanh nghiệp về quản lý nhân sự và dự án.
* Xác định các chức năng cần thiết và xây dựng đặc tả yêu cầu hệ thống.
* Phân tích hệ thống
* Sử dụng mô hình Use Case Diagram để xác định các tác nhân và chức năng chính.
* Áp dụng mô hình ERD (Entity-Relationship Diagram) để thiết kế cơ sở dữ liệu.
* Thiết kế hệ thống
* Thiết kế kiến trúc hệ thống theo mô hình MVC (Model-View-Controller).
* Xây dựng giao diện người dùng thân thiện, dễ sử dụng.

### 1.5.3. Phương pháp xây dựng và triển khai

* Công nghệ sử dụng
* Ngôn ngữ lập trình: Python, Java, hoặc JavaScript.
* Cơ sở dữ liệu: MySQL, PostgreSQL hoặc MongoDB.
* Giao diện: ReactJS, Angular hoặc VueJS.
* Quy trình phát triển
* Áp dụng mô hình phát triển phần mềm Agile để triển khai linh hoạt.
* Chia hệ thống thành các giai đoạn: Phân tích → Thiết kế → Phát triển → Kiểm thử → Triển khai.

### 1.5.4. Phương pháp kiểm thử và đánh giá

* Kiểm thử phần mềm
* Áp dụng kiểm thử đơn vị (Unit Test) để đảm bảo từng chức năng hoạt động đúng.
* Kiểm thử tích hợp (Integration Test) để kiểm tra sự liên kết giữa các thành phần.
* Kiểm thử giao diện (UI/UX Test) để đánh giá trải nghiệm người dùng.
* Đánh giá hiệu quả hệ thống
* So sánh hiệu suất của hệ thống với các phần mềm hiện có.
* Thực hiện khảo sát người dùng để đánh giá mức độ đáp ứng yêu cầu thực tế.

# Chương 2: Phân Tích và Thiết Kế Cơ Sở Dữ Liệu

## 2.1. Xác Định Các Yêu Cầu Của Hệ Thống

### 2.1.1. Yêu cầu chức năng

Hệ thống cần cung cấp các chức năng sau:

Quản lý phòng ban

* Thêm, sửa, xóa, và xem danh sách phòng ban.
* Mỗi phòng ban có một người quản lý duy nhất.

Quản lý nhân viên

* Thêm, sửa, xóa, và xem danh sách nhân viên.
* Phân loại nhân viên: nhân viên thường hoặc nhân viên phụ trách.
* Mỗi nhân viên chỉ thuộc một phòng ban (trừ nhân viên phụ trách).
* Phân công nhân viên làm quản lý phòng ban (mỗi phòng chỉ có một người quản lý).

Quản lý hoạt động

* Thêm, sửa, xóa, và xem danh sách hoạt động.
* Mỗi hoạt động có thể được phân chia cho nhiều phòng ban.
* Mỗi phòng ban chỉ được phân chia một hoạt động.

Quản lý dự án

* Thêm, sửa, xóa, và xem danh sách dự án.
* Mỗi dự án có thể có nhiều nhân viên tham gia.
* Một nhân viên có thể tham gia nhiều dự án.

Báo cáo và truy vấn

* Xem danh sách nhân viên theo phòng ban.
* Xem danh sách dự án và nhân viên tham gia.
* Xem hoạt động đang diễn ra tại các phòng ban.
* Thống kê số lượng nhân viên tham gia mỗi dự án.

### 2.1.2. Yêu cầu phi chức năng

Hiệu suất

* Hệ thống cần hỗ trợ tìm kiếm và truy vấn dữ liệu nhanh chóng.
* Hỗ trợ số lượng lớn nhân viên, phòng ban, hoạt động, dự án.

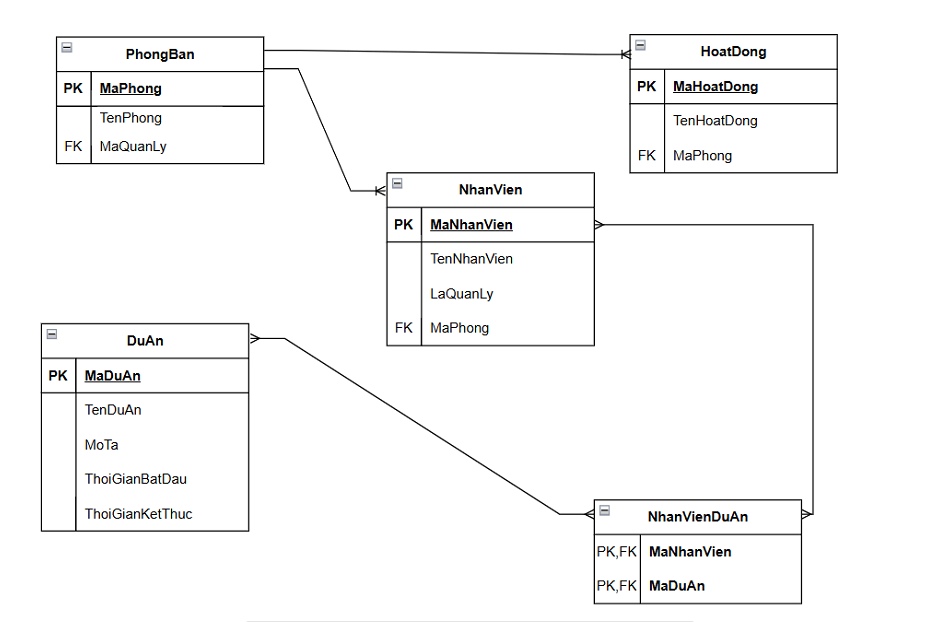
Bảo mật

* Phân quyền truy cập: quản trị viên, nhân viên, người quản lý.
* Nhân viên chỉ có thể xem thông tin cá nhân và dự án của mình.
* Người quản lý có thể cập nhật thông tin phòng ban và nhân viên thuộc phòng.
* Quản trị viên có toàn quyền thao tác dữ liệu.

Khả năng mở rộng

* Có thể mở rộng để quản lý nhiều loại hoạt động hoặc dự án khác nhau.
* Hỗ trợ tích hợp với hệ thống chấm công hoặc quản lý tài nguyên doanh nghiệp (ERP).

## 2.2. Thiết Kế Mô Hình Dữ Liệu (ERD)



## 2.3. Chuyển Đổi Mô Hình Dữ Liệu Sang Mô Hình Vật Lý

Để chuyển đổi mô hình dữ liệu sang mô hình vật lý, cần thiết kế các bảng, khóa và ràng buộc trong cơ sở dữ liệu.

**PhongBan(**MaPhong, MaQuanLy,TenPhong)

NhanVien (MaNhanVien, MaPhong TenNhanVien, LaQuanLy,)

HoatDong (MaHoatDong, MaPhong TenHoatDong,)

DuAn (MaDuAn, TenDuAn, MoTa, ThoiGianBatDau, ThoiGianKetThuc)

NhanVienDuAn (MaNhanVien, MaDuAn)

# Chương 3: Triển Khai Cơ Sở Dữ Liệu

## 3.1. Tạo Cơ Sở Dữ Liệu và Các Bảng

Để triển khai cơ sở dữ liệu cho hệ thống quản lý dự án, chúng ta cần tạo các bảng với các định nghĩa khóa, ràng buộc và các mối quan hệ.

CREATE DATABASE QLNhanSu;

USE QLNhanSu;

-- Tạo bảng PhongBan

CREATE TABLE PhongBan (

MaPhong INT PRIMARY KEY IDENTITY(1,1),

TenPhong NVARCHAR(255) NOT NULL,

MaQuanLy INT,

FOREIGN KEY (MaQuanLy) REFERENCES NhanVien(MaNhanVien)

);

-- Tạo bảng NhanVien

CREATE TABLE NhanVien (

MaNhanVien INT PRIMARY KEY IDENTITY(1,1),

TenNhanVien NVARCHAR(255) NOT NULL,

LaQuanLy BIT NOT NULL,

MaPhong INT

);

-- Tạo bảng HoatDong

CREATE TABLE HoatDong (

MaHoatDong INT PRIMARY KEY IDENTITY(1,1),

TenHoatDong NVARCHAR(255) NOT NULL,

MaPhong INT,

FOREIGN KEY (MaPhong) REFERENCES PhongBan(MaPhong)

);

-- Tạo bảng DuAn

CREATE TABLE DuAn (

MaDuAn INT PRIMARY KEY IDENTITY(1,1),

TenDuAn NVARCHAR(255) NOT NULL,

MoTa NVARCHAR(MAX),

ThoiGianBatDau DATE,

ThoiGianKetThuc DATE

);

-- Tạo bảng NhanVienDuAn

CREATE TABLE NhanVienDuAn (

MaNhanVien INT,

MaDuAn INT,

PRIMARY KEY (MaNhanVien, MaDuAn),

FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien),

FOREIGN KEY (MaDuAn) REFERENCES DuAn(MaDuAn)

## );

## 3.2. Nhập Dữ Liệu Mẫu

Để kiểm tra hoạt động của cơ sở dữ liệu, chúng ta cần nhập một số dữ liệu mẫu vào các bảng.

INSERT INTO PhongBan (TenPhong, MaQuanLy) VALUES (N'Phòng Kinh Doanh', NULL);

INSERT INTO PhongBan (TenPhong, MaQuanLy) VALUES (N'Phòng Kỹ Thuật', 1); -- Giả sử nhân viên có MaNhanVien = 1 là quản lý

INSERT INTO PhongBan (TenPhong, MaQuanLy) VALUES (N'Phòng Hành Chính', 1);

INSERT INTO PhongBan (TenPhong, MaQuanLy) VALUES (N'Phòng Marketing', 2); -- Giả sử nhân viên có MaNhanVien = 2 là quản lý

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Nguyễn Văn A', 1, 1); -- Quản lý Phòng Kinh Doanh

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Trần Thị B', 0, 1); -- Nhân viên Phòng Kinh Doanh

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Lê Văn C', 1, 2); -- Quản lý Phòng Kỹ Thuật

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Phạm Văn D', 0, 2); -- Nhân viên Phòng Kỹ Thuật

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Nguyễn Thị E', 1, 3); -- Quản lý Phòng Hành Chính

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Trần Văn F', 0, 3); -- Nhân viên Phòng Hành Chính

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Lê Thị G', 1, 4); -- Quản lý Phòng Marketing

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Nguyễn Văn H', 0, 4); -- Nhân viên Phòng Marketing

INSERT INTO DuAn (TenDuAn, MoTa, ThoiGianBatDau, ThoiGianKetThuc) VALUES (N'Dự án A', N'Mô tả dự án A', '2023-01-01', '2023-12-31');

INSERT INTO DuAn (TenDuAn, MoTa, ThoiGianBatDau, ThoiGianKetThuc) VALUES (N'Dự án B', N'Mô tả dự án B', '2023-02-01', '2023-11-30');

INSERT INTO DuAn (TenDuAn, MoTa, ThoiGianBatDau, ThoiGianKetThuc) VALUES (N'Dự án C', N'Mô tả dự án C', '2023-03-01', '2023-10-31');

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (1, 1); -- Nguyễn Văn A tham gia Dự án A

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (2, 1); -- Trần Thị B tham gia Dự án A

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (3, 2); -- Lê Văn C tham gia Dự án B

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (4, 2); -- Phạm Văn D tham gia Dự án B

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (5, 3); -- Nguyễn Thị E tham gia Dự án C

INSERT INTO NhanVienDuAn (MaNhanVien, MaDuAn) VALUES (6, 3); -- Trần Văn F tham gia Dự án C

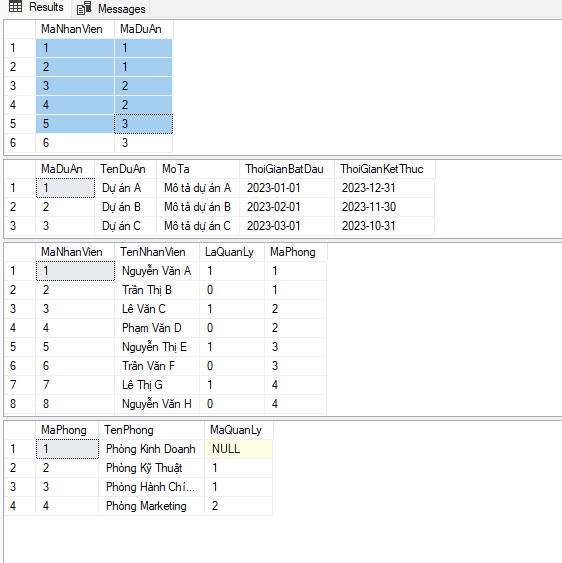
## 3.3Các truy vấn cơ bản và nâng cao

select \* from NhanVienDuAn

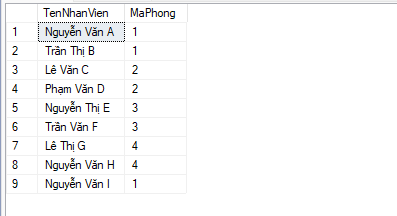
select \* from DuAn

select \* from NhanVien

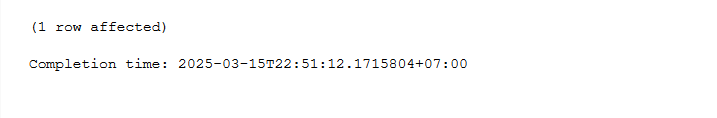
**select \* from PhongBan**



SELECT TenNhanVien, MaPhong FROM NhanVien; --lấy tên và phòng của tất cả nhân viên



INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong) VALUES (N'Nguyễn Văn I', 0, 1);-- thêm một nhân viên mới



UPDATE NhanVien SET TenNhanVien = N'Nguyễn Văn A1' WHERE MaNhanVien = 1;--cập nhật tên của nhân viên

--lấy tên nhân viên cùng với phòng

SELECT N.TenNhanVien, P.TenPhong

FROM NhanVien N

INNER JOIN PhongBan P ON N.MaPhong = P.MaPhong;

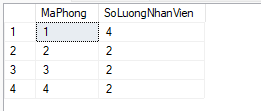


--đếm số lượng nhân viên trong mỗi phòng

SELECT MaPhong, COUNT(\*) AS SoLuongNhanVien

FROM NhanVien

GROUP BY MaPhong;



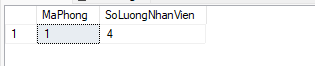
--lấy các phòng có hơn 2 nhân viên

SELECT MaPhong, COUNT(\*) AS SoLuongNhanVien

FROM NhanVien

GROUP BY MaPhong

HAVING COUNT(\*) > 2;



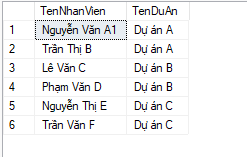
--lấy danh sách nhân viên cùng với dự án mà họ tham gia

SELECT N.TenNhanVien, D.TenDuAn

FROM NhanVien N

INNER JOIN NhanVienDuAn NDA ON N.MaNhanVien = NDA.MaNhanVien

INNER JOIN DuAn D ON NDA.MaDuAn = D.MaDuAn;



# Chương4: Lập trình T-SQL

## 4.1.Tạo view từ cơ bản đến nâng cao

Trong SQL, "view" (hay còn gọi là khung nhìn) là một bảng ảo được tạo ra từ kết quả của một câu lệnh SELECT. Nó không lưu trữ dữ liệu thực tế, mà chỉ lưu trữ định nghĩa của câu lệnh SELECT. Khi bạn truy vấn một view, hệ thống sẽ thực thi câu lệnh SELECT cơ bản và trả về kết quả.

Mục đích của việc tạo view:

Đơn giản hóa truy vấn phức tạp: View cho phép bạn đóng gói các truy vấn phức tạp thành một đối tượng duy nhất, giúp cho việc truy vấn dữ liệu trở nên dễ dàng hơn.

Tăng cường bảo mật: View có thể được sử dụng để giới hạn quyền truy cập vào dữ liệu nhạy cảm bằng cách chỉ hiển thị những thông tin cần thiết cho người dùng.

Tùy chỉnh dữ liệu: View cho phép bạn thay đổi cách dữ liệu được hiển thị mà không cần thay đổi cấu trúc của bảng gốc.

Tính nhất quán: View giúp đảm bảo rằng các truy vấn được thực hiện một cách nhất quán trên toàn bộ ứng dụng.

Dưới đây là một số view cơ bản đến nâng cao.

--VIEW danh sách tất cả nhân viên

CREATE VIEW View\_DanhSachNhanVien AS

SELECT \* FROM NhanVien;

--view danh sách phòng ban

CREATE VIEW View\_DanhSachPhongBan AS

SELECT \* FROM PhongBan;

--view danh sách nhân viên là quản lý

CREATE VIEW View\_QuanLy AS

SELECT \* FROM NhanVien

WHERE LaQuanLy = 1;

--view danh sách nhân viên theo phòng

CREATE VIEW View\_NhanVienTheoPhong AS

SELECT N.TenNhanVien, P.TenPhong

FROM NhanVien N

INNER JOIN PhongBan P ON N.MaPhong = P.MaPhong;

--view số lượng nhân viên trong mỗi phòng

CREATE VIEW View\_SoLuongNhanVienTheoPhong AS

SELECT P.TenPhong, COUNT(N.MaNhanVien) AS SoLuongNhanVien

FROM PhongBan P

LEFT JOIN NhanVien N ON P.MaPhong = N.MaPhong

GROUP BY P.TenPhong;

--view danh sách nhân viên với dự án mà họ tham gia

CREATE VIEW View\_NhanVienDuAn AS

SELECT N.TenNhanVien, D.TenDuAn

FROM NhanVien N

INNER JOIN NhanVienDuAn NDA ON N.MaNhanVien = NDA.MaNhanVien

INNER JOIN DuAn D ON NDA.MaDuAn = D.MaDuAn;

--view danh sách phòng có nhiều hơn 2 nhân viên

CREATE VIEW View\_PhongCoNhieuNhanVien AS

SELECT P.TenPhong, COUNT(N.MaNhanVien) AS SoLuongNhanVien

FROM PhongBan P

LEFT JOIN NhanVien N ON P.MaPhong = N.MaPhong

GROUP BY P.TenPhong

HAVING COUNT(N.MaNhanVien) > 2;

## 4.2.Tạo index cần thiết cho các bảng

Trong SQL, "index" (chỉ mục) là một cấu trúc dữ liệu đặc biệt giúp tăng tốc độ truy vấn dữ liệu từ bảng. Nó hoạt động tương tự như mục lục của một cuốn sách, cho phép hệ thống cơ sở dữ liệu nhanh chóng tìm thấy các hàng dữ liệu cần thiết mà không cần phải quét toàn bộ bảng.

Mục đích của việc tạo index:

Tăng tốc độ truy vấn: Index giúp cải thiện đáng kể hiệu suất của các truy vấn SELECT, đặc biệt là trên các bảng lớn.

Tối ưu hóa các phép toán JOIN: Index có thể giúp tối ưu hóa hiệu suất của các phép toán JOIN, giúp kết hợp dữ liệu từ nhiều bảng nhanh hơn.

Đảm bảo tính duy nhất của dữ liệu: Index UNIQUE có thể được sử dụng để đảm bảo rằng các giá trị trong một cột là duy nhất

--tạo index cho bảng nhân viên( cột mã phòng)

CREATE INDEX IDX\_NhanVien\_MaPhong ON NhanVien(MaPhong);

--tạo index cho bảng phòng ban( cột mã quản lý)

CREATE INDEX IDX\_PhongBan\_MaQuanLy ON PhongBan(MaQuanLy);

--tạo index cho bảng hoạt động( cột mã phòng)

CREATE INDEX IDX\_HoatDong\_MaPhong ON HoatDong(MaPhong);

--tạo index cho bảng dự án( cột tên dự án)

CREATE INDEX IDX\_DuAn\_TenDuAn ON DuAn(TenDuAn);

--tạo index cho bảng nhân viên dự án

--( cột mã dự án)

CREATE INDEX IDX\_NhanVienDuAn\_MaDuAn ON NhanVienDuAn(MaDuAn);

--(cột mã nhân viên)

CREATE INDEX IDX\_NhanVienDuAn\_MaNhanVien ON NhanVienDuAn(MaNhanVien);

## 4.3. Tạo function

Trong SQL, "function" (hàm) là một khối mã được đặt tên, thực hiện một nhiệm vụ cụ thể và trả về một giá trị. Hàm giúp bạn đóng gói các logic phức tạp, tái sử dụng mã và cải thiện tính dễ đọc của truy vấn SQL.

Mục đích của việc tạo function:

Tái sử dụng mã: Hàm cho phép bạn viết mã một lần và sử dụng lại nhiều lần trong các truy vấn khác nhau.

Đơn giản hóa truy vấn phức tạp: Hàm giúp bạn đóng gói các logic phức tạp thành một đơn vị duy nhất, giúp cho truy vấn SQL trở nên dễ đọc và dễ bảo trì hơn.

Tùy chỉnh logic: Hàm cho phép bạn tạo các logic tùy chỉnh để xử lý dữ liệu theo yêu cầu cụ thể của ứng dụng.

Các loại function:

Scalar function: Hàm trả về một giá trị đơn (ví dụ: số, chuỗi, ngày tháng).

Table-valued function: Hàm trả về một bảng.

Dưới đây là một số function mà em đã tạo.

## Hàm tính số lượng nhân viên trong một phòng

CREATE FUNCTION dbo.CountEmployeesInRoom(@MaPhong INT)

RETURNS INT

AS

BEGIN

DECLARE @Count INT;

SELECT @Count = COUNT(\*) FROM NhanVien WHERE MaPhong = @MaPhong;

RETURN @Count;

END;

## Hàm kiểm tra xem một nhân viên có phải là quản lý không

CREATE FUNCTION dbo.IsManager(@MaNhanVien INT)

RETURNS BIT

AS

BEGIN

DECLARE @Result BIT;

SELECT @Result = CASE WHEN LaQuanLy = 1 THEN 1 ELSE 0 END

FROM NhanVien WHERE MaNhanVien = @MaNhanVien;

RETURN @Result;

END;

## Hàm lấy tên phòng theo mã phòng

CREATE FUNCTION dbo.GetRoomName(@MaPhong INT)

RETURNS NVARCHAR(255)

AS

BEGIN

DECLARE @TenPhong NVARCHAR(255);

SELECT @TenPhong = TenPhong FROM PhongBan WHERE MaPhong = @MaPhong;

RETURN @TenPhong;

END;

## Hàm tính tổng số dự án

CREATE FUNCTION dbo.CountTotalProjects()

RETURNS INT

AS

BEGIN

DECLARE @Total INT;

SELECT @Total = COUNT(\*) FROM DuAn;

RETURN @Total;

# END;

## Hàm lấy tên nhân viên theo mã nhân viên

CREATE FUNCTION dbo.GetEmployeeName(@MaNhanVien INT)

RETURNS NVARCHAR(255)

AS

BEGIN

DECLARE @TenNhanVien NVARCHAR(255);

SELECT @TenNhanVien = TenNhanVien FROM NhanVien WHERE MaNhanVien = @MaNhanVien;

RETURN @TenNhanVien;

END;

## Hàm lấy danh sách nhân viên theo phòng

CREATE FUNCTION dbo.GetEmployeesByRoom(@MaPhong INT)

RETURNS TABLE

AS

RETURN (

SELECT \* FROM NhanVien WHERE MaPhong = @MaPhong

);

## Hàm lấy danh sách nhân viên là quản lý

CREATE FUNCTION dbo.GetManagers()

RETURNS TABLE

AS

RETURN (

SELECT \* FROM NhanVien WHERE LaQuanLy = 1);

## 4.4.Tạo trigger để kiểm soát dữ liệu

Trong SQL, "trigger" (bộ kích hoạt) là một đoạn mã SQL được thực thi tự động khi một sự kiện cụ thể xảy ra trong cơ sở dữ liệu. Các sự kiện này có thể bao gồm chèn (INSERT), cập nhật (UPDATE) hoặc xóa (DELETE) dữ liệu từ một bảng.

Mục đích của việc tạo trigger:

Đảm bảo tính toàn vẹn dữ liệu: Trigger có thể được sử dụng để kiểm tra và duy trì tính nhất quán của dữ liệu trong cơ sở dữ liệu.

Tự động hóa các tác vụ: Trigger có thể được sử dụng để tự động hóa các tác vụ như ghi nhật ký thay đổi dữ liệu, cập nhật các bảng liên quan hoặc gửi thông báo.

Thực thi các quy tắc nghiệp vụ: Trigger có thể được sử dụng để thực thi các quy tắc nghiệp vụ phức tạp mà không thể thực hiện bằng các ràng buộc thông thường.

Kiểm tra và xác thực dữ liệu: Trigger có thể được sử dụng để kiểm tra và xác thực dữ liệu trước khi nó được chèn, cập nhật hoặc xóa.

Các loại trigger:

Trigger AFTER: Được thực thi sau khi sự kiện xảy ra.

Trigger BEFORE: Được thực thi trước khi sự kiện xảy ra.

Trigger INSTEAD OF: Thay thế sự kiện kích hoạt.

Dưới đây là một số trigger mà em đã tạo.

--Trigger kiểm tra dữ liệu trước khi chèn

CREATE TRIGGER trg\_CheckTenNhanVien

ON NhanVien

INSTEAD OF INSERT

AS

BEGIN

IF EXISTS (SELECT \* FROM inserted WHERE TenNhanVien IS NULL OR TenNhanVien = '')

BEGIN

RAISERROR('Tên nhân viên không được để trống.', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong)

SELECT TenNhanVien, LaQuanLy, MaPhong FROM inserted;

END

END;

--Trigger ghi lại lịch sử thay đổi

CREATE TABLE LichSuThayDoi (

MaLichSu INT PRIMARY KEY IDENTITY(1,1),

MaNhanVien INT,

ThayDoi NVARCHAR(255),

ThoiGian DATETIME DEFAULT GETDATE()

);

CREATE TRIGGER trg\_GhiLichSuThayDoi

ON NhanVien

AFTER UPDATE

AS

BEGIN

INSERT INTO LichSuThayDoi (MaNhanVien, ThayDoi)

SELECT MaNhanVien, 'Cập nhật thông tin nhân viên' FROM inserted;

END;

--Trigger ngăn chặn xóa nhân viên quản lý

CREATE TRIGGER trg\_NganChanXoaQuanLy

ON NhanVien

INSTEAD OF DELETE

AS

BEGIN

IF EXISTS (SELECT \* FROM deleted WHERE LaQuanLy = 1)

BEGIN

RAISERROR('Không thể xóa nhân viên là quản lý.', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

DELETE FROM NhanVien

WHERE MaNhanVien IN (SELECT MaNhanVien FROM deleted);

END

END;

--Trigger tự động cập nhật ngày sửa đổi

ALTER TABLE NhanVien ADD NgayCapNhat DATETIME;

CREATE TRIGGER trg\_CapNhatNgayCapNhat

ON NhanVien

AFTER UPDATE

AS

BEGIN

UPDATE NhanVien

SET NgayCapNhat = GETDATE()

FROM NhanVien n

INNER JOIN inserted i ON n.MaNhanVien = i.MaNhanVien;

END;

## 4.5. Xây dựng Stored Procedure

Stored procedure (thủ tục lưu trữ) là một tập hợp các câu lệnh SQL được lưu trữ trong cơ sở dữ liệu và có thể được thực thi nhiều lần. Nó giống như một chương trình nhỏ được viết bằng SQL, cho phép bạn đóng gói các logic phức tạp và thực hiện các tác vụ lặp đi lặp lại một cách hiệu quả.

Mục đích của việc sử dụng stored procedure:

Tăng hiệu suất: Stored procedure được biên dịch và lưu trữ trong cơ sở dữ liệu, giúp giảm thời gian thực thi so với việc gửi từng câu lệnh SQL riêng lẻ từ ứng dụng.

Tái sử dụng mã: Bạn có thể viết stored procedure một lần và sử dụng lại nhiều lần trong các ứng dụng khác nhau.

Tăng cường bảo mật: Stored procedure cho phép bạn kiểm soát quyền truy cập vào dữ liệu, chỉ cho phép người dùng thực thi các thủ tục được ủy quyền.

Giảm lưu lượng mạng: Thay vì gửi nhiều câu lệnh SQL từ ứng dụng đến cơ sở dữ liệu, bạn chỉ cần gửi một lệnh gọi stored procedure.

Đơn giản hóa logic ứng dụng: Stored procedure giúp bạn di chuyển logic phức tạp từ ứng dụng sang cơ sở dữ liệu, giúp ứng dụng trở nên đơn giản và dễ bảo trì hơn.

--Lấy danh sách tất cả nhân viên.

CREATE PROCEDURE sp\_GetAllEmployees

AS

BEGIN

SELECT \* FROM NhanVien;

END;

--Lấy thông tin nhân viên theo mã nhân viên

CREATE PROCEDURE sp\_GetEmployeeById

@MaNhanVien INT

AS

BEGIN

SELECT \* FROM NhanVien WHERE MaNhanVien = @MaNhanVien;

END;

-- Lấy danh sách các phòng ban.

CREATE PROCEDURE sp\_GetAllDepartments

AS

BEGIN

SELECT \* FROM PhongBan;

END;

-- Thêm một nhân viên mới.

CREATE PROCEDURE sp\_AddEmployee

@TenNhanVien NVARCHAR(255),

@LaQuanLy BIT,

@MaPhong INT

AS

BEGIN

INSERT INTO NhanVien (TenNhanVien, LaQuanLy, MaPhong)

VALUES (@TenNhanVien, @LaQuanLy, @MaPhong);

END;

-- Lấy danh sách các dự án.

CREATE PROCEDURE sp\_GetAllProjects

AS

BEGIN

SELECT \* FROM DuAn;

END;

-- Xóa một nhân viên theo mã nhân viên.

CREATE PROCEDURE sp\_DeleteEmployee

@MaNhanVien INT

AS

BEGIN

DELETE FROM NhanVien WHERE MaNhanVien = @MaNhanVien;

END;

-- Lấy danh sách nhân viên theo phòng

CREATE PROCEDURE sp\_GetEmployeesByDepartment

@MaPhong INT

AS

BEGIN

SELECT \* FROM NhanVien WHERE MaPhong = @MaPhong;

END;

# Chương 5: Bảo mật và quản trị

Phần bảo mật và quản trị đóng vai trò then chốt trong việc bảo vệ dữ liệu và duy trì hoạt động ổn định của bất kỳ hệ thống thông tin nào. Bảo mật bao gồm các biện pháp phòng ngừa, phát hiện và ứng phó với các mối đe dọa an ninh mạng, đảm bảo tính bảo mật, toàn vẹn và khả dụng của dữ liệu. Quản trị tập trung vào việc quản lý và vận hành hệ thống, bao gồm cấu hình, giám sát, bảo trì và khắc phục sự cố.

Trong lĩnh vực bảo mật, các biện pháp thường được áp dụng bao gồm:

* Kiểm soát truy cập: Xác thực người dùng và phân quyền truy cập dựa trên vai trò.
* Mã hóa dữ liệu: Bảo vệ dữ liệu nhạy cảm bằng cách chuyển đổi chúng thành dạng không thể đọc được.
* Tường lửa và hệ thống phát hiện xâm nhập: Ngăn chặn các truy cập trái phép và phát hiện các hoạt động đáng ngờ.
* Sao lưu và phục hồi dữ liệu: Đảm bảo dữ liệu có thể được khôi phục trong trường hợp xảy ra sự cố.
* Cập nhật phần mềm và vá lỗi: Khắc phục các lỗ hổng bảo mật để ngăn chặn khai thác.

Trong lĩnh vực quản trị, các hoạt động chính bao gồm:

* Quản lý người dùng: Tạo, xóa và quản lý tài khoản người dùng.
* Giám sát hiệu suất hệ thống: Theo dõi các chỉ số hiệu suất để đảm bảo hệ thống hoạt động ổn định.
* Bảo trì hệ thống: Thực hiện các công việc bảo trì định kỳ để đảm bảo hệ thống hoạt động trơn tru.
* Xử lý sự cố: Khắc phục các sự cố phần cứng và phần mềm để giảm thiểu thời gian ngừng hoạt động.
* Quản lý tài nguyên: Phân bổ và quản lý tài nguyên hệ thống một cách hiệu quả.

Sự kết hợp giữa bảo mật và quản trị giúp đảm bảo rằng hệ thống thông tin được bảo vệ an toàn và hoạt động hiệu quả.

## 5.1. Tạo người dùng

--tạo login

CREATE LOGIN Phong1 WITH PASSWORD = 'Phong2005';

--tạo user

USE QLNhanSu; -- Chọn cơ sở dữ liệu

CREATE USER Phong2 FOR LOGIN Phong1;

## 5.2. Quản lý sao lưu và phục hồi dữ liệu

-- Gán người dùng vào vai trò

EXEC sp\_addrolemember 'Role\_QuanLy', 'Phong2';

--Sao lưu

BACKUP DATABASE QLNhanSu

TO DISK = 'C:\saoluu\QLNS.bak'

WITH FORMAT, INIT;

--Phục hồi

RESTORE DATABASE QLNhanSu

FROM DISK = 'C:\saoluu\QLNS.bak'

WITH

REPLACE;

# Chương 6: Kết luận

## 1. Đánh giá ưu và nhược điểm của hệ thống:

*Ưu điểm:*

* Khả năng xử lý dữ liệu đa dạng: Hệ thống có thể thu thập và xử lý dữ liệu từ nhiều nguồn khác nhau, bao gồm cơ sở dữ liệu, tệp tin, API, v.v.
* Quy trình phân tích dữ liệu toàn diện: Hệ thống tuân thủ quy trình phân tích dữ liệu chuẩn, bao gồm thu thập, làm sạch, khám phá, xây dựng mô hình, trực quan hóa và đưa ra kết luận.
* Khả năng trực quan hóa dữ liệu hiệu quả: Hệ thống cung cấp các công cụ trực quan hóa dữ liệu mạnh mẽ, giúp người dùng dễ dàng hiểu và diễn giải kết quả phân tích.
* Tính linh hoạt và khả năng mở rộng: Hệ thống có thể được tùy chỉnh và mở rộng để đáp ứng các yêu cầu phân tích dữ liệu khác nhau.
* Tăng cường khả năng ra quyết định: Hệ thống cung cấp thông tin hữu ích và khuyến nghị dựa trên dữ liệu, giúp người dùng đưa ra các quyết định sáng suốt.

*Nhược điểm:*

* Yêu cầu kiến thức chuyên môn: Để sử dụng hiệu quả hệ thống, người dùng cần có kiến thức về phân tích dữ liệu và các công cụ liên quan.
* Khả năng xử lý dữ liệu thời gian thực còn hạn chế: Hệ thống có thể chưa tối ưu cho việc xử lý dữ liệu thời gian thực hoặc dữ liệu luồng.
* Chi phí triển khai và bảo trì: Việc triển khai và bảo trì hệ thống có thể đòi hỏi chi phí đáng kể, đặc biệt là đối với các hệ thống phức tạp.
* Phụ thuộc vào chất lượng dữ liệu đầu vào: Kết quả phân tích phụ thuộc lớn vào chất lượng dữ liệu đầu vào. Dữ liệu không chính xác hoặc không đầy đủ có thể dẫn đến kết quả sai lệch.
* Vấn đề bảo mật dữ liệu: Cần đảm bảo an toàn và bảo mật cho dữ liệu, đặc biệt là dữ liệu nhạy cảm.

## 2. Đề xuất các hướng phát triển tiếp theo:

* Tích hợp khả năng xử lý dữ liệu thời gian thực: Nâng cấp hệ thống để có thể xử lý dữ liệu thời gian thực và dữ liệu luồng, phục vụ các ứng dụng yêu cầu cập nhật dữ liệu liên tục.
* Tăng cường khả năng tự động hóa: Tự động hóa các bước trong quy trình phân tích dữ liệu, chẳng hạn như làm sạch dữ liệu và xây dựng mô hình, để giảm thời gian và công sức của người dùng.
* Phát triển các mô hình dự đoán và phân tích nâng cao: Tích hợp các thuật toán học máy và thống kê tiên tiến để nâng cao khả năng dự đoán và phân tích dữ liệu.
* Tích hợp các công cụ trực quan hóa tương tác: Cung cấp các công cụ trực quan hóa dữ liệu tương tác, cho phép người dùng khám phá dữ liệu một cách linh hoạt.
* Tăng cường bảo mật dữ liệu: Áp dụng các biện pháp bảo mật mạnh mẽ để bảo vệ dữ liệu khỏi các truy cập trái phép.
* Phát triển giao diện người dùng thân thiện hơn: Cải thiện giao diện người dùng để người dùng dễ dàng sử dụng hệ thống hơn, ngay cả khi không có kiến thức chuyên môn sâu về phân tích dữ liệu.
* Tích hợp khả năng phân tích dữ liệu trên thiết bị di động: Cho phép người dùng truy cập và phân tích dữ liệu trên các thiết bị di động, tăng tính linh hoạt và tiện lợi.
* Tích hợp các công cụ hợp tác: Cho phép nhiều người dùng cùng làm việc trên một dự án phân tích dữ liệu, tăng cường hiệu quả làm việc nhóm.
* Xây dựng cộng đồng người dùng: Tạo ra một cộng đồng người dùng để chia sẻ kiến thức, kinh nghiệm và hỗ trợ lẫn nhau trong việc sử dụng hệ thống.