**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN NỘI DUNG BÁO CÁO 6](#_Toc68693299)

[1.1. GIỚI THIỆU VỀ CÔNG CỤ THỰC HIỆN ĐỀ TÀI 6](#_Toc68693300)

[1.1.1. Tổng quan về Visual Studio 6](#_Toc68693301)

[1.1.3. Ngôn ngữ lập trình 9](#_Toc68693302)

[1.2. GIỚI THIỆU VỀ MÃ HÓA THÔNG TIN 9](#_Toc68693303)

[CHƯƠNG 2. CƠ CHẾ HOẠT ĐỘNG 11](#_Toc68693304)

[2.1. Đôi nét về mật mã Vigenere 11](#_Toc68693305)

[2.2. Phân quyền là gì? 17](#_Toc68693306)

[2.2.1. Bạn thường dùng loại phân quyền như thế nào? 17](#_Toc68693307)

[2.2.2. Phân quyền theo group là gì và làm như thế nào? 18](#_Toc68693308)

[2.2.2.1. Phân quyền theo cấp bậc 18](#_Toc68693309)

[2.2.2.2 Phân quyền theo chức năng 20](#_Toc68693310)

[2.2.3. Kiểm tra lại dữ liệu trong bảng 22](#_Toc68693311)

[2.2.4. Làm một số ví dụ 24](#_Toc68693312)

[2.2.5. Kết luận 26](#_Toc68693313)

[2.2.6. Phân quyền theo Hành động của các nhóm Group theo những cấp bậc khác nhau 26](#_Toc68693314)

[CHƯƠNG 3. CHƯƠNG TRÌNH THỰC NGHIỆM 28](#_Toc68693315)

[3.1. Dữ liệu thực nghiệm 28](#_Toc68693316)

[3.2. form Đăng nhập 28](#_Toc68693317)

[3.3. Form Trang chủ 29](#_Toc68693318)

[3.4. Form quản lý tài khoản 30](#_Toc68693319)

[3.5. Form đổi mật khẩu. 31](#_Toc68693320)

[3.6. Form Lịch sử hoạt động 32](#_Toc68693321)

[3.7. TÀI LIỆU THAM KHẢO 32](#_Toc68693322)

# Lời Mở Đầu

Đồ án môn học với những ngôn ngữ lập trình khá phổ biến với sinh viên khoa Công Nghệ Thông Tin như ngôn ngữ Java để tự thống kê kiến thức và lên ý tưởng cho những đồ án của mình. Đó là yêu cầu và cũng như thách thức đối với sinh viên khi làm quen với thực hành từ nhữnh lý thuyết sách vở.

Để cho em có thể nắm chắc kiến thức và tiếp cận thực tế với một sản phẩm phần mềm, em đã nhận được sự trợ giúp tận tình từ thầy ***Tống Thanh Văn*** giảng viên hướng dẫn, cùng với sự góp ý của các thành viên trong lớp để giúp em hoàn thành đồ án này.

Khi nhìn vào định nghĩa mã hóa thông tin, chúng ta có thể thấy một trong những lĩnh vực rất cần thiết để áp dụng mã hóa trong quá khứ: trao đổi liên lạc giữa các bên trong chiến tranh. Hai đồng minh cần phải trao đổi thông tin mà đảm bảo kẻ thù không thể đọc được, vì vậy việc mã hóa dữ liệu khi trao đổi là một trong những yếu tố quyết định trong việc truyền tin mà không thể tạo được một kênh kết nối an toàn và bí mật.

Xuất phát từ những lý do trên, cùng với sự hướng dẫn của Giảng viên Tống Thanh Văn, chúng em thực hiện 1 số chức năng như mã hóa mật khẩu, phân quyền account, theo dõi người dùng … để vận dụng những kiến thức đã được học vào thực tiễn, xây dựng một hệ thống bảo mật cho người dùng.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

**Phan Đình Tuệ**

**Nguyễn Nhất Tuyển**

**Lương Thanh Tuấn**

# CHƯƠNG 1. TỔNG QUAN NỘI DUNG BÁO CÁO

## 1.1. GIỚI THIỆU VỀ CÔNG CỤ THỰC HIỆN ĐỀ TÀI

### 1.1.1. Tổng quan về Visual Studio



**Microsoft Visual Studio** là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

Microsoft cung cấp phiên bản "Express" (đối với phiên bản Visual Studio 2013 trở về trước) và "Community" (đối với bản Visual Studio 2015 trở về sau) là phiên bản miễn phí của Visual Studio.

***1.1.2. Tổng quan về hệ quản trị cơ sở dữ liệu SQL Server***



Có rất nhiều hệ quản trị cơ sở dữ liệu mạnh mẽ như: Oracle, My SQL… nhưng chúng em sẽ sử dụng SQL Server để thực hiện đề tài*“Xây dựng phần mềm quản lý điểm sinh viên”* này.

Hệ quản trị cơ sở dữ liệu Microsoft SQL Server (MSSQL) là một trong những hệ quản trị cơ sở dữ liệu thông dụng nhất hiện nay. Đây là hệ quản trị cơ sở dữ liệu thường được sử dụng trong các hệ thống trung bình, với ưu điểm có các công cụ quản lý mạnh mẽ giúp cho việc quản lý và bảo trì hệ thống dễ dàng, hỗ trợ nhiều phương pháp lưu trữ, phân vùng và đánh chỉ mục phục vụ cho việc tối ưu hóa hiệu năng.

SQL Server luôn được Microsoft cải tiến để nâng cao hiệu năng, tính sẵn sàng của hệ thống, khả năng mở rộng và bảo mật, cung cấp nhiều công cụ cho người phát triển ứng dụng được tích hợp với bộ Visual Studio do Microsoft cung cấp. SQL Server có 4 dịch vụ lớn là Database Engine, Intergration Service, Reporting service và Analysis Services.

DataBase Engine: được phát triển để thực thi tốt hơn với việc hỗ trợ cả dữ liệu có cấu trúc và dữ liệu phi cấu trúc (XML).

Khả năng sẵn sàng của hệ thống được nâng cao, hỗ trợ các chức năng: Database mirroring (cơ sở dữ liệu gương), failover clustering , snapshots và khôi phục dữ liệu nhanh.

Việc quản lý chỉ mục được thực hiện song song với việc hoạt động của hệ thống. Người dùng có thể thêm chỉ mục, xây dựng lại chỉ mục hay xóa một chỉ mục đi trong khi hệ thống vẫn được sử dụng.

Chức năng phân vùng dữ liệu được hỗ trợ: Người dùng có thể phân vùng các bảng và chỉ mục cũng như quản lý phân vùng dữ liệu một cách dễ dàng. Việc hỗ trợ phân vùng dữ liệu giúp nâng cao hiệu năng hoạt động của hệ thống.

Dịch vụ đồng bộ hóa dữ liệu được mở rộng với việc hỗ trợ mô hình đồng bộ hóa ngang hàng. Đây là dịch vụ giúp đồng bộ hóa dữ liệu giữa các máy chủ dữ liệu, giúp mở rộng khả năng của hệ thống.

Dịch vụ tích hợp (Integration Service) thiết kế lại cho phép người dùng tích hợp dữ liệu và phân tích dữ liệu từ nhiều nguồn khác nhau. Hỗ trợ việc quản lý chất lượng dữ liệu và làm sạch dữ liệu, một công việc quan trọng trong tiến trình ETL.

Dịch vụ phân tích dữ liệu (Analysis Service): cung cấp khung nhìn tích hợp và thống nhất về dữ liệu cho người dùng, hỗ trợ việc phân tích dữ liệu.

Công cụ khai phá dữ liệu (Data mining) được tích hợp hỗ trợ nhiều thuật toán khai phá dữ liệu, hỗ trợ cho việc phân tích, khai phá dữ liệu và xây dựng các hệ thống hỗ trợ ra quyết định cho người quản lý.

Dịch vụ xây dựng quản lý báo cáo (Reporting Service) dựa trên nền tảng quản trị doanh nghiệp thông minh và được quản lý qua dịch vụ web. Báo cáo có thể được xây dựng dễ dàng với ngôn ngữ truy vấn MDX. Thông qua các công cụ trên Business Intelligent, người dùng dễ dàng truy cập báo cáo và trích xuất ra nhiều định dạng khác nhau thông qua trình duyệt web.

### 1.1.3. Ngôn ngữ lập trình

- C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

- C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

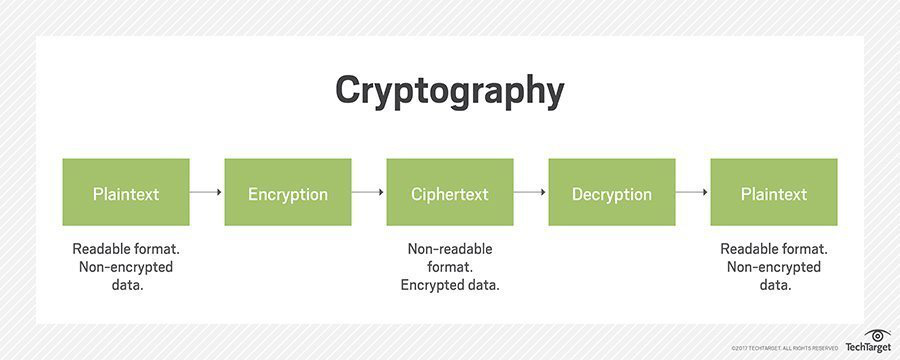
- C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

- C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

## 1.2. GIỚI THIỆU VỀ MÃ HÓA THÔNG TIN

Mã hóa thông tin là một hình thức biến đổi dữ liệu thành một dạng dữ liệu khác có ý nghĩa khác với dữ liệu trước khi bị biến đổi ban đầu, với mục đích chỉ cho phép một số người nhất định có thể đọc được dữ liệu ban đầu, thông qua việc giải mã dữ liệu sau khi biến đổi.

Hay nói cách khác, mã hóa là biến dữ liệu ban đầu A thành dữ liệu B, và việc đọc dữ liệu A sẽ thông qua việc giải mã dữ liệu B về A.

[](https://techvccloud.mediacdn.vn/2020/10/10/ma-hoa-thong-tin-1-16022970290862126517679.jpg)

*Quá trình mã hóa và giải mã*

Định nghĩa khá phức tạp, nên chúng ta sẽ trả lời một số câu hỏi để làm rõ khái niệm mã hóa ở trên:

**Biến đổi dữ liệu là gì?**

Biến đổi dữ liệu là một quy tắc nào đó biến đổi một lượng dữ liệu này thành một lượng dữ liệu khác. Nếu nhìn theo khía cạnh toán học, thì biến đổi dữ liệu chính là một dạng hàm số y = f(x) với x là dữ liệu ban đầu, y là dữ liệu sau khi biến đổi từ dữ liệu x và f là hàm biến đổi.

Trong mật mã học, khi nghiên cứu về mã hóa thông tin, dữ liệu ban đầu được gọi là Plaintext (kí hiệu là P), dữ liệu sau khi mã hóa được gọi là Ciphertext (kí hiệu là C), hàm biến đổi/mã hóa được gọi là phương pháp mã hóa và được kí hiệu là E (Encryption).

# CHƯƠNG 2. CƠ CHẾ HOẠT ĐỘNG

## 2.1. Đôi nét về mật mã Vigenere

Mật mã Vigenère là một phương pháp mã hóa chữ văn bản tiếng Anh, lần đầu tiên được mô tả bởi Giovan Battista Bellaso vào năm 1553. Phương pháp mã hóa mật mã Vigenère dễ hiểu và dễ thực hiện, nhưng chỉ đến năm 1863 với nhiều nỗ lực suốt ba thế kỷ, Friedrich Kasiski mới xuất bản một phương pháp chung để giải mã mật mã Vigenère.

Mật mã Vigenère là tập hợp các quy tắc thay thế chữ cái đơn trong bảng chữ cái tiếng Anh qua việc sử dụng 26 mật mã Caesar với các bước dịch chuyển từ 0 đến 25 tương ứng từ chữ ‘a’ đến chữ ‘z’. Cụ thể, bản mã Vigenère được lập theo công thức sau:

ci = (pi + ki) mod 26, i=1,…,l

trong đó, C = {c1…cl} là bản mã, P={p1…pl} là bản gốc, K = {k1…kl} là dãy khóa và l là độ dài bản rõ. Tương tự, bản gốc P có thể được tính nếu biết khóa và bản mã theo công thức:

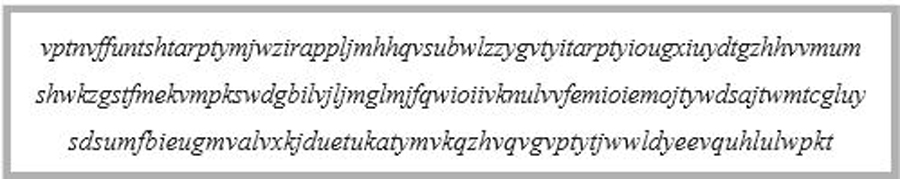
pj = (cj – kj) mod 26, j = 1,…,l

Mật mã Vigenère không thể phá vỡ trong trường hợp sử dụng các khóa đủ dài. Nhưng với các khóa ngắn hoặc nếu nhà thám mã có đủ nhiều bản mã so với độ dài khóa thì khá dễ để phá vỡ. Việc thám mật mã Vigenère thường tiến hành theo hai bước là: xác định độ dài chu kỳ của khóa trước, sau đó tìm khóa cụ thể.

***Tìm chu kỳ khóa của mật mã Vigenère***

Đầu tiên cần lưu ý là chu kỳ của khóa tìm được có thể không đúng với thực tế được sử dụng. Nếu bản mã đủ dài thì có thể là chính xác, các phương pháp được cung cấp ở đây là gần đúng.

Mật mã Vigenère áp dụng các mật mã Caesar khác nhau cho các chữ cái liên tiếp. Ví dụ một bản mã Vigenère như sau:



Hình 1. Bản mã hóa sử dụng mật mã Vigenère

Mật mã Caesar là một dạng của mật mã thay thế, theo đó mỗi ký tự trong bản rõ được thay thế bằng một ký tự cách nó một đoạn trong bảng chữ cái để tạo thành bản mã. Giả sử với khóa là 3 (dịch 3 vị trí trong bảng chữ cái), thì chữ ‘a’ sẽ được thay bằng chữ ‘d’, chữ ‘b’ sẽ được thay bằng ‘e’ và cứ thế đến hết bản rõ. Phương pháp này được đặt tên là Caesar, vị Hoàng đế đã sử dụng loại mật mã này thường xuyên trong công việc.

Nếu mật mã Vigenère sử dụng khóa có chu kỳ 3 là 'PUB', thì chữ cái rõ đầu tiên được mã hóa bằng mật mã Caesar với khóa là 16 (P là chữ cái thứ 16 của bảng chữ cái), chữ cái thứ hai được mã với khóa là 21 (chữ cái U) và chữ cái thứ ba được mã với khóa là 2 (chữ cái B). Chữ cái rõ thứ 4 được mã hóa quay lại bằng chữ khóa thứ nhất (khóa 16). Kết quả là, các chữ cái ở các vị trí 1,4,7,10,... đều được mã hóa bằng cùng một mật mã Caesar với chữ khóa là P. Các chữ cái ở các vị trí 2,5,8,11,... và 3,6,9,12,... được mã hóa bằng mật mã Caesar với khóa tương ứng là chữ U và B.

Như vậy, trình tự chính xác sẽ phụ thuộc vào chu kỳ của khóa mật mã, tức là độ dài khóa, như với ví dụ trên thì độ dài chu kỳ khóa là 3.

***Phương pháp tìm chu kỳ khóa theo sự lặp lại của nhóm chữ cái***

Để xác định chu kỳ của khóa mật mã Vigenère, phương pháp Kasiski xem xét sự lặp lại của các nhóm chữ cái như Hình 2.

*Hình 2. Sự lặp lại của nhóm chữ cái*

Đoạn lặp lại loạt **VHVS** gồm 18 ký tự, gợi ý rằng độ dài khóa có thể là 18, 9, 6, 3, 2. Còn đoạn lặp lại loạt **QUCE** là 30 ký tự, gợi ý độ dài khóa là 30, 15, 10, 6, 5, 3, 2. Kết hợp lại, độ dài khóa có thể là 6, 3 hoặc 2.

***Chỉ số trùng hợp (Index of coincidence - I.C. còn được ký hiệu là***Ic()***)***

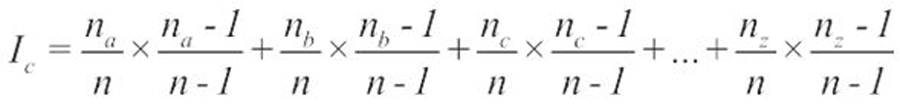
Nếu trong bản mã không có sự lặp lại của một loạt chữ cái nào, người thám mã sẽ sử dụng đến chỉ số trùng hợp Ic.

Giả sử X là một chuỗi ký tự trong tiếng Anh, ký hiệu xác suất xuất hiện của các chữ a, b,…, z lần lượt là p0, p1,…, p25. Khi đó:



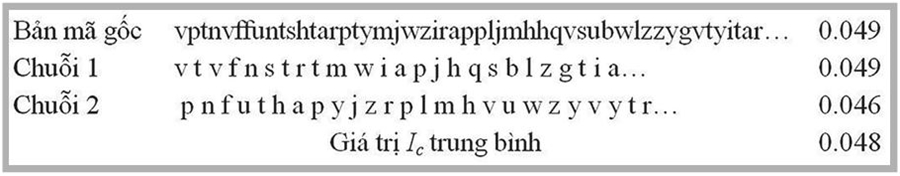
• Khi đó:

Ic (x) = ∑  = 0.0822+0.0152+…+0.0012 = 0.065

Chỉ số trùng hợp đôi khi được gọi là tỷ lệ lặp lại. Nếu bản mã cụ thể có độ dài n, na là tần số xuất hiện của chữ ‘a’, nb là tần số xuất hiện của chữ ‘b’…, thì chỉ số trùng hợp gần đúng được tính theo công thức sau: 

Chỉ số trùng hợp (Ic) là một kỹ thuật thống kê giúp xác định một đoạn văn bản có đáp ứng quy luật ngôn ngữ của tiếng Anh. Một tính chất quan trọng của kỹ thuật là giá trị Ic không thay đổi nếu áp dụng mật mã thay thế đơn cho văn bản. Điều này là do Ic dựa trên tần số xuất hiện của chữ cái và mật mã thay thế đơn không làm thay đổi tần số của bộ chữ cái riêng lẻ. Với văn bản tiếng Anh sẽ có giá trị Ic làm tròn là 0.06, nếu các ký tự có phân phối đồng đều thì Ic gần hơn với 0,03 - 0,04.

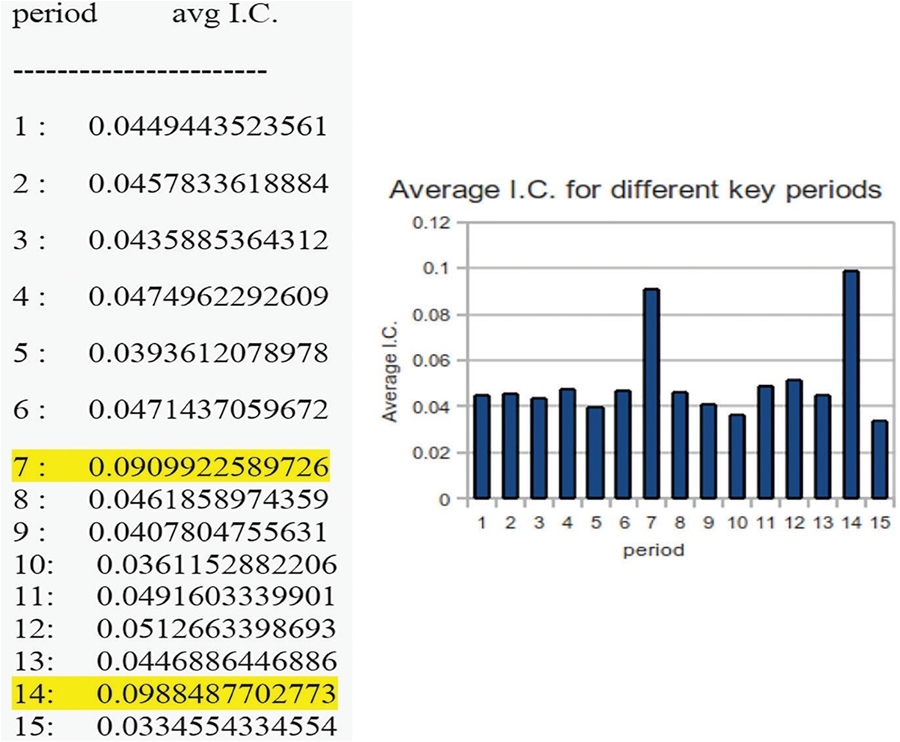
Phương pháp dùng chỉ số trùng hợp để xác định chu kỳ của khóa mật mã Vigenère thực hiện như sau. Trước tiên, giả sử độ dài khóa là 2, thực hành trích xuất hai chuỗi tại các vị trí 1, 3, 5, 7,... và 2, 4, 6, 8,... từ bản mã như Hình 1 (lưu ý rằng Ic được tính bằng cách sử dụng toàn bộ chuỗi mã, không chỉ là phần được hiển thị).



Tương tự với trường hợp độ dài khóa là 3 sẽ có 3 chuỗi, tương ứng với các giá trị Ic như sau:

Như vậy, Ic trung bình đối với trường hợp chu kỳ 2 là khoảng 0,048 và đối với trường hợp chu kỳ 3 là khoảng 0,047.

Quy trình này sẽ được lặp lại cho tất cả các độ dài khóa muốn kiểm tra. Ví dụ tiếp tục tính với chu kỳ khóa lên đến 15 sẽ có các giá trị trung bình I.C (avg I.C.) tương ứng như Hình 3.



*Hình 3. Các giá trị trung bình I.C.*

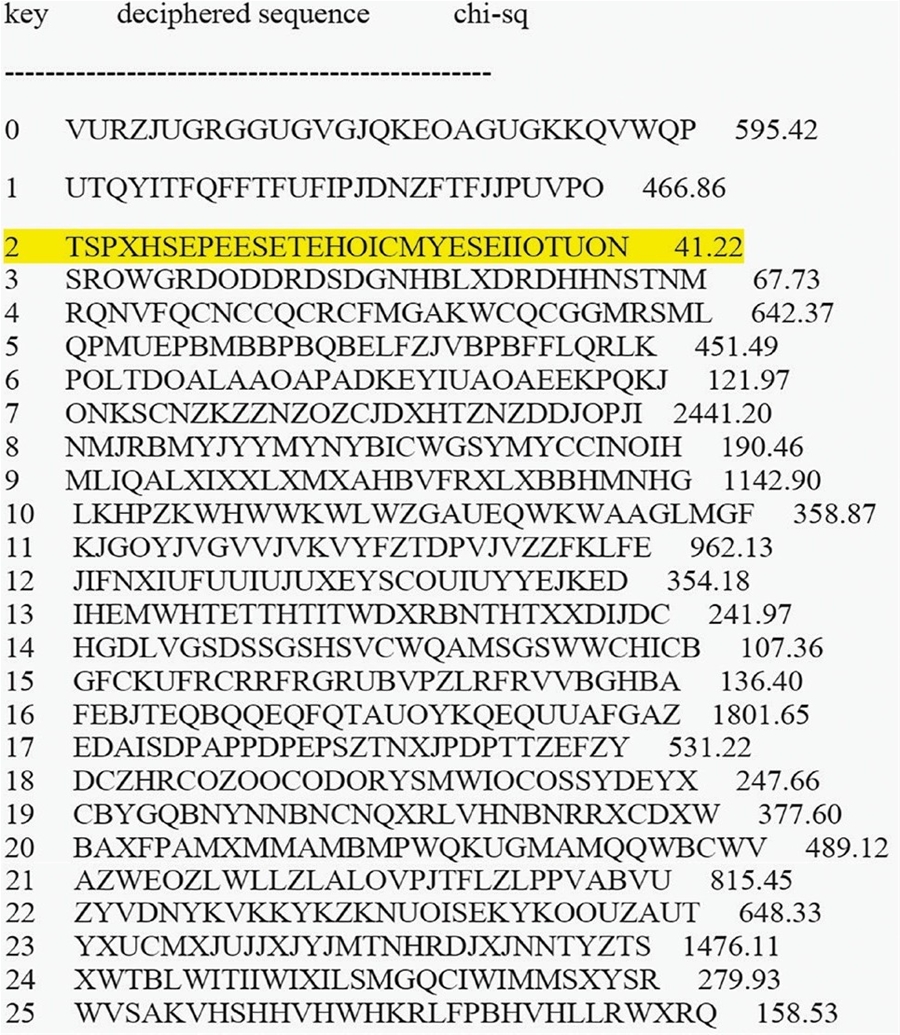
Theo cột giá trị, hoặc theo biểu đồ, có 2 giá trị trung bình I.C cao đột biến đã gợi ý rằng khóa mật mã có thể có độ dài 7 hoặc 14. Cả hai xác suất này phải được tiếp tục kiểm tra.

***Tìm khóa mật mã theo kỹ thuật thống kê khi bình phương***

Ví dụ thám mã với khóa chu kỳ 7 (sử dụng 7 mật mã Caesar) cho bản mã ở Hình 1, việc tìm khóa khá dễ dàng. Thám mã sẽ so sánh giá trị thống kê Khi bình phương của dãy phá mã với giá trị phân phối tần số xuất hiện chữ cái tiếng Anh.

Lập   chuỗi   chữ   cái   lấy   từ   các   vị   trí   1, 8, 15, 22,… của bản mã ở Hình 1 (vurzjugrggugvgjqkeoagugkkqvwqp…). Đây là chuỗi được mã hóa với cùng một mật mã Caesar.

Giải mã chuỗi này với cả 26 mật mã Caesar có thể, lập bảng so sánh phân phối tần số của văn bản được giải mã với phân phối tần số tiếng Anh cho mỗi khóa. Tương ứng, sẽ thu được 26 giá trị thống kê Khi bình phương. Khóa chính xác sẽ tương ứng với văn bản được giải mã với thống kê Khi bình phương thấp nhất. Kết quả cụ thể như Hình 4 đã tìm được chữ cái khóa đầu tiên, theo đó giá trị Khi bình phương nhỏ nhất là 41.22, tương ứng với khóa là 2 (chữ cái ‘c’).



*Hình 4. Giá trị thống kê Khi bình phương của chuỗi giải mã*

Tiếp tục tìm 6 chữ cái khóa còn lại theo cách cực tiểu Khi bình phương tương tự để tìm các khóa tương ứng sẽ thu được chuỗi khóa 2,8,0,7,4,17,18. Chuyển về dạng chữ cái là chuỗi 'CIAHERS', chuỗi khóa này bị sai một vị trí. Điều này cho thấy không thể hoàn toàn dựa vào kỹ thuật thám mã này trừ khi thu được bản mã đủ dài. Khóa chính xác trong trường hợp này là 'CIPHERS' và thực tế kiểm tra Khi bình phương có hai giá trị  rất thấp cho dãy con thứ 3. Thật không may, giá trị nhỏ nhất lại không đúng, giá trị khóa đúng có giá trị Khi bình phương lớn hơn giá trị nhỏ nhất một chút.

Thực tế trong kiểm tra Khi bình phương cũng như I.C, xác suất xuất hiện của các chữ cái không phải luôn luôn đúng, chúng gần đúng. Hơn nữa, tần số của các chữ cái trong bản mã không phản ánh chính xác phân phối xác suất các chữ cái trên văn bản mã. Đó là lý do kết quả trên cho ra chữ khóa ‘A’ mà lẽ ra phải là ‘P’. Do đó, việc xem xét thêm các khía cạnh khác như là dựa vào quy luật ngôn ngữ để chỉnh sửa kết quả là rất cần thiết.

## 2.2. Phân quyền là gì?

Tưởng tượng đơn giản như này nhé.

- Bạn là sếp bạn có toàn quyền với nhân viên của mình

- Bạn là trưởng phòng bạn chỉ có quyền với nhân viên trong phòng của mình

- Bạn là trưởng nhóm dự án bạn chỉ có quyền với các thành viên trong nhóm

- Và Nhóm < Phòng < công ty.

- Như vậy bạn có thể thấy trong công ty này có 3 loại quyền hạn và chúng ta cần phân quyền theo nó

### 2.2.1. Bạn thường dùng loại phân quyền như thế nào?

Câu hỏi này mình đặt ra ban đầu là "Có các loại phân quyền nào?" nhưng thật sự việc phân quyền này rất ít được public do nó mang tính chất riêng tư của những dự án. Nên nếu bạn search thì chủ yếu sẽ chỉ tìm được cách tạo tài khoản sử dụng cho user trong các HQTCSDL.

Ở bài viết này mình sẽ nói về việc phân quyền bằng nhóm quyền (Group) Và đưa ra 1 số cách dựng CSDL (db).

## 2.2.2 Phân quyền truy cập và nhận dạng người dùng

- Các hệ QTCSDL đều có cơ chế cho phép nhiều người cùng khai thác CSDL, phục vụ nhiều mục đích rất đa dạng. Tuỳ theo vai trò khác nhau của người dùng mà họ được cấp quyền khác nhau để khai thác CSDL.

- Bảng phân quyền truy cập cũng là dữ liệu của CSDL, được tổ chức và xây dựng như những dữ liệu khác. Điểm khác biệt duy nhất là nó được quản lí chặt chẽ, không giới thiệu công khai và chỉ có những người quản trị hệ thống mới có quyền truy cập, bổ sung, sửa.

- Ví dụ: một số hệ quản lí học tập và giảng dạy của nhà trường cho phép mọi PHHS truy cập để biết kết quả học tập của con em mình. Mỗi PHHS chỉ có quyền xem điểm của con em mình. Đây là quyền truy cập hạn chế nhất. Các thầy cô giáo trong trường có quyền truy cập cao hơn: xem kết quả và mọi thông tin khác của bất kì học sinh nào trong trường. Người quản lí học tập có quyền nhập điểm, cập nhật các thông tin khác trong CSDL. Bảng phân quyền truy cập:

Đ: Đọc;                K: Không được truy cập;

S: Sửa;                 X: Xoá.              B: Bổ sung.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MaHS** | **Các điểm số** | **Các thông tin khác** |
| K10 | Đ | Đ | K |
| K11 | Đ | Đ | K |
| K11 | Đ | Đ | K |
| Giáo viên | Đ | Đ | K |
| Người quản lí | ĐSBX | ĐSBX | ĐSBX |

 - Người quản trị hệ CSDL cần cung cấp:

* Bảng phân quyền truy cập cho hệ QTCSDL
* Phương tiện cho người dùng để hệ QTCSDL nhận biết đúng được họ.

- Người dùng muốn truy cập vào hệ thống cần khai báo:

* Tên người dùng;
* Mật khẩu.

=> Dựa vào các thông tin này, hệ QTCSDl xác minh để cho phép hoặc từ chối quyền truy cập CSDL.

### 2.2.3. Phân quyền theo group là gì và làm như thế nào?

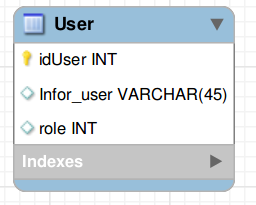
Phân quyền theo group là cách gọi chung của mình cho việc bạn nhóm nhiều thành viên trong 1 tổ chức có cùng một quyền hạn thực thi công việc. Lúc đó ta có thể nhóm họ vào 1 group để dễ dàng trao quyền hạn.

Như vậy từ cách phân quyền theo group này ta có thể sử dụng với "3 nấc" khác nhau.

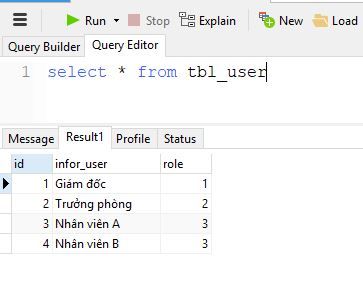
Ví dụ đơn giản về 3 cách này nhé

#### 2.2.3.1. Phân quyền theo cấp bậc

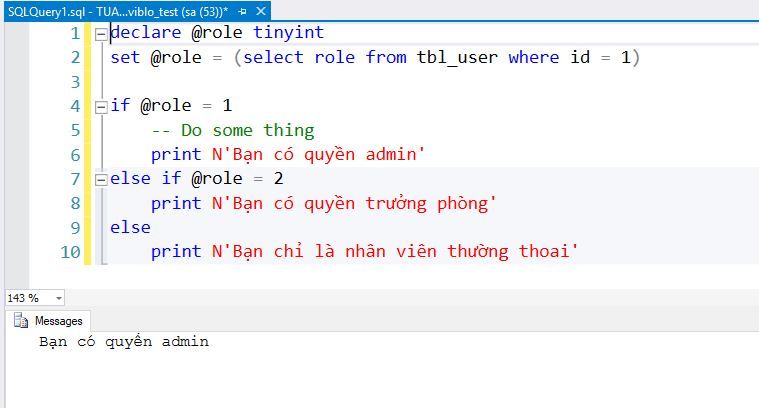
Loại hình này chúng ta thường thiết kế db đơn giản như sau



Khi đó dữ liệu bạn dùng sẽ có dạng như thế này.



role ở đây là 1, 2, 3 tức là có 3 mức quyền hạn và lớn nhất hay bé nhất còn tùy thuộc vào quy định của mỗi công ty. Ví dụ



- Ưu điểm

Việc sử dụng kiểu phân quyền này dễ dàng cho những người mới bắt đầu. Những nhóm quyền được lập lên nhanh chóng có thể sử dụng luôn, Và việc phải check cũng tương đối là đơn giản, bạn chỉ cần 1 cần

select count(\*) from tbl\_... where id = ? and role = ?

hoặc

select role from tbl\_... where id = ?

Rất dễ dàng để sử dụng đúng không.

- Nhược điểm

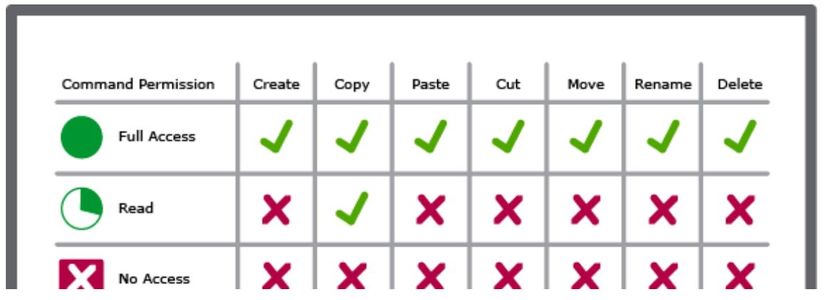
Bạn biết đó việc sử dụng dữ liệu như này tồn tại 1 số nhược điểm rất lớn sau

Rất khó có thể mở rộng dự án

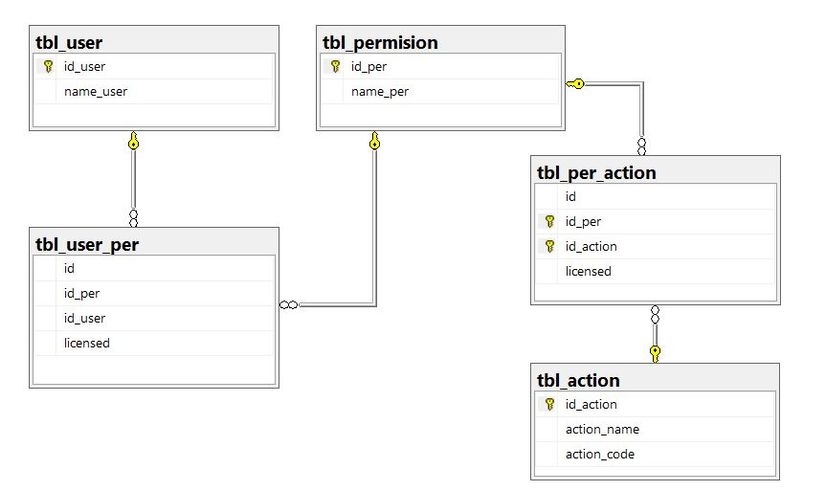
Trong thực tế không phải lúc nào cũng có 3 role. Nó có thể phát sinh nhiều role kì dị. Ví dụ: **Thư ký giám đốc ngoài quyền đuổi việc ra còn lại nó sẽ có quyền của giám đốc** vậy trường hợp này thuộc role 1 hay 2?.

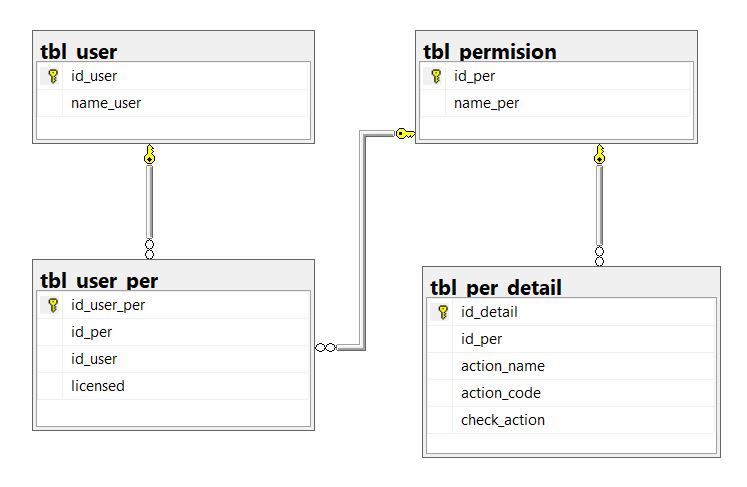
Rất khó để phân quyền chi tiết.

#### 2.2.2.2 Phân quyền theo chức năng



Loại phân quyền này được sử dụng rất nhiều trong thực tế. Nó rất hiệu quả và dễ thao tác đối với người cấp quyền.

Ta thiết kế db đơn giản trong ví dụ này như sau:  Nhưng để Demo mình sẽ tóm gọn 2 bảng tbl\_action và tbl\_per\_action thành bảng tbl\_per\_detail để dễ thao tác. Và ta có một Database như sau.



Chi tiết của việc thiết kế DB như sau:

**tbl\_user**: bảng lưu người dùng bao gồm các thuộc tính như ID, Name,.... Bảng không có khóa ngoại.

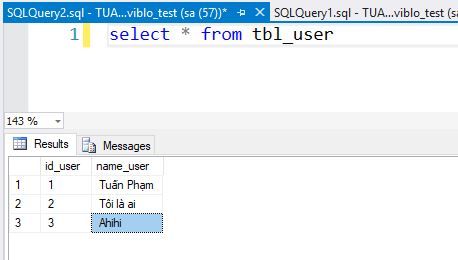
**tbl\_permision**: bảng chứa nhóm quyền hạn. bao gồm các thuộc tính, ID nhóm quyền hạn, tên nhóm quyền hạn.

**tbl\_permision\_detail**: là bảng sẽ chứa những quyền hạn cụ thể dành cho nhóm quyền hạn. Trường **action\_name** không cần thiết bạn có thể bỏ. Trường **action\_code** là để khi lập trình mình định nghĩa một thao tác nhất định trong bằng code này ví dụ quyền sửa thì code nó là **EDIT** chẳng hạn.

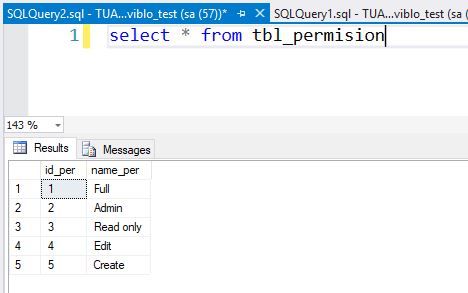
tbl\_per\_relationship: là bảng lưu mối liên hệ giữa người dùng và nhóm quyền hạn. Mục đích của bảng này không phải là để một người dùng có nhiều nhóm quyền mà để không phải truy vấn lại bảng user chứa thông tin nhạy cảm như username và password. Bạn cũng có thể bỏ qua bảng này và liên hệ trực tiếp giữa bảng user và permision luôn, nhưng mình khuyên bạn nên sử dụng thêm bảng này vì có nhiều trường hợp user có nhiều quyền hạn.

### 2.2.3. Kiểm tra lại dữ liệu trong bảng

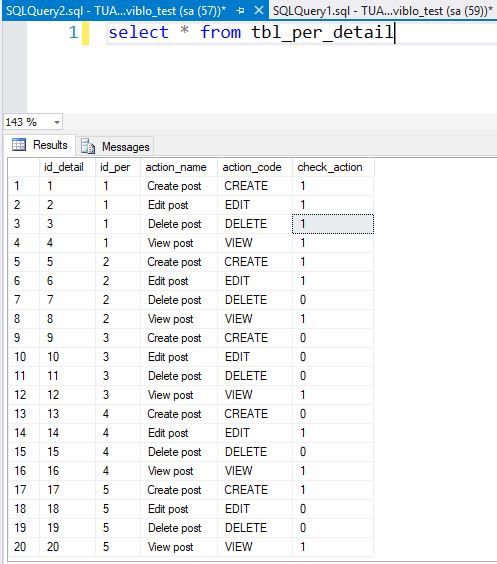
**tbl\_user**



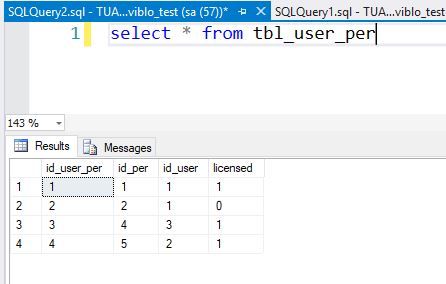
**tbl\_permision**



**tbl\_per\_detail**



**tbl\_user\_per**



### 2.2.4. Làm một số ví dụ

Kiểm tra quyền của người dùng ví dụ: Hãy kiểm tra quyền của user có id là 1:

DECLARE @result NVARCHAR(1000)

SET @result = N'Những quyền hiện tại của user ('

select @result = @result + name\_user + ') là: ' from tbl\_user where id\_user = 1

select @result = @result + action\_name + ', ' from tbl\_user as u

join tbl\_user\_per as up on u.id\_user = up.id\_user

join tbl\_permision as p on up.id\_per = p.id\_per

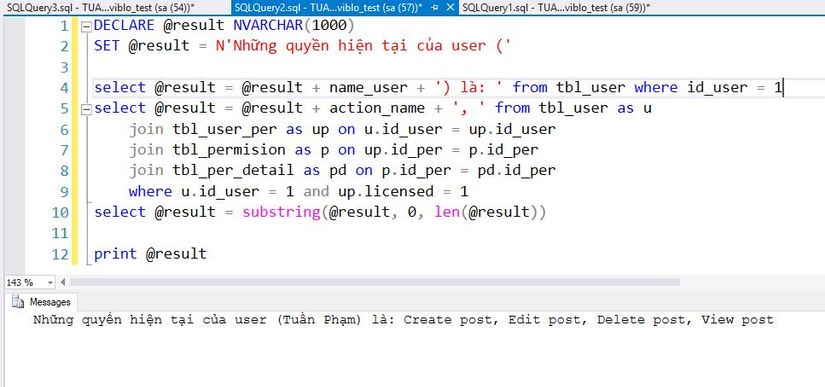
join tbl\_per\_detail as pd on p.id\_per = pd.id\_per

where u.id\_user = 1 and up.licensed = 1 and pd.check\_action = 1

select @result = substring(@result, 0, len(@result))

print @result

kết quả



Kiểm tra xem user 2 có quyền xóa bài viết không?

DECLARE @result bit

select @result = check\_action from tbl\_user as u

join tbl\_user\_per as up on u.id\_user = up.id\_user

join tbl\_permision as p on up.id\_per = p.id\_per

join tbl\_per\_detail as pd on p.id\_per = pd.id\_per

where u.id\_user = 2 and up.licensed = 1 and action\_code = 'DELETE'

begin

if @result = 1

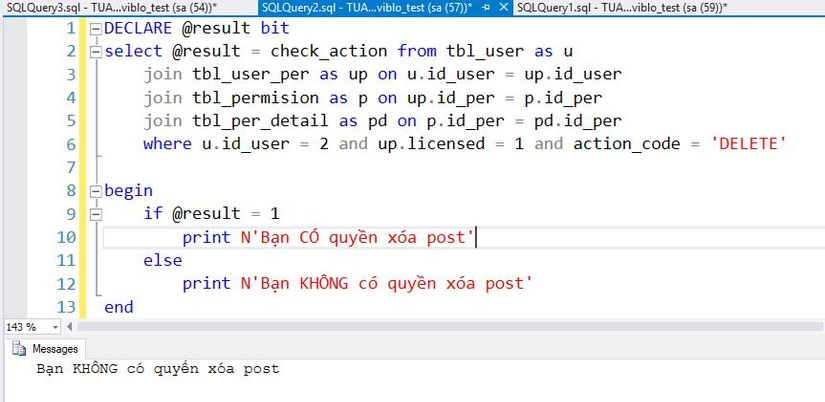
print N'Bạn CÓ quyền xóa post'

else

print N'Bạn KHÔNG có quyền xóa post'

end

kết quả



Đó là một số ví dụ đơn giản tron tình huống này khi sử dụng phân quyền theo nhóm (group)

### 2.2.5. Kết luận

- Ưu điểm

Việc phân quyền này như mình có nêu ngay từ ban đầu rất dễ thao tác đối với những admin khi họ muốn chuyển nhóm quyền hoặc thỏa sức sáng tạo trong việc tạo ra những quyền mới từ những quyền ban đầu. Ví dụ như người dùng vừa có thể **EDIT** và **DELETE**,....

Ngoài ra việc thực hiện những câu querry cũng rất dễ dàng cho những lập trình viên.

- Nhược điểm

Vấn đề sử dụng quyền hành động rất dễ khi chúng ta làm việc trên 1 group, nhưng nếu trong chương trình của bạn có nhiều group và phân cấp nhiều tầng thì nó lại là một vấn đề nan giải khác, khi bạn không chỉ phải check quyền hành động mà bạn còn phải check xem quyền hành động này của người dùng có thể áp dùng được trong group khác hay không?..

### 2.2.6. Phân quyền theo Hành động của các nhóm Group theo những cấp bậc khác nhau

Đây là loại phân quyền lằng ngằng nhất nhưng lại là quan trọng nhất, bởi các lý do sau đây:

Các tổ chức sử dụng phần mềm để thực hiện thao tác của họ đều có phân cấp rõ ràng

Trong những tổ chức có những người nắm full quyền của nhiều nhóm

Có những thành viên thuộc nhiều nhóm

Có những thành viên tuy chỉ là nhân viên nhưng lại có quyền của các sếp (thư ký)

Chính vì có nhiều trường hợp như vậy những lập trình viên sinh ra được rất nhiều case trong code.

Giải quyết vấn đề này bạn có thể tìm hiểu cách thự hiện phân quyền trong odoo.

Phân quyền theo model: Có nghĩ là người dùng được thao tác thực hiện với những bảng dữ liệu nào

Ví dụ: Admin có thể thực hiện với thao tác với bảng user của họ

phân quyền theo raw: Người dùng được thực hiện việc thao tác với các raw được chỉ định

Ví dụ: Leader A chỉ có thể thực hiện thao tác với những thành viên của mình trong bảng user

Phân quyền theo column: Người dùng sẽ được quyền thao tac với nhưng column đó

Ví dụ: Chỉ giám đốc mới có thể đuổi việc nhân viên, ở đây ta sẽ có 1 column tên là is\_working để biết việc nhân viên đó còn đi làm hay không

***2. Cấu trúc phần quyền của 1 Folder:***

Đối với 1 folder hay 1 directory bất kì thì đều bị tác động bởi 3 loại phần quyền người dùng ở phần 1 mình đã giới thiệu đó là: Own, Group, Other.

*Ví dụ :*

[root@task ~]# ls -l

total 32

-rw-------. 1 root root 1412 Aug 10 16:06 con-ks.cfg

-rw-r--r--. 1 root root 14522 Aug 10 18:06 install.ext

-rw-r--r--. 1 root root 5037 Jan 10 13:06 hello.pdf

drwxr-xr-x 6 root root 4196 Feb 9 10:02 text.txt

Các quyền hạn :

Read – r – 4 : cho phép đọc nội dung.

Write – w – 2 : dùng để tạo, thay đổi hay xóa.

Execute – x – 1 : thực thi chương trình.

Ví dụ : quyền r, w, x : 4+2+1 = 7

Tổ hợp 3 quyền trên có giá trị từ 0 đến 7.

***3. Một số lệnh phân quyền thường dùng:***

Lệnh Chmod : dùng để cấp quyền hạn.

Cú pháp : #chmod <specification> <file>

Ví dụ: #chmod 644 hello.txt

-> cấp quyền cho owner có thể ghi các nhóm các chỉ có quyền đọc với file hello.txt

Lệnh Chown : dùng thay đổi người sở hữu.

Cú pháp : #chown <owner> <filename>

Lệnh Chgrp : dùng thay đổi nhóm sở hữu.

Cú pháp : #chgrp <group> <filename>

# CHƯƠNG 3. CHƯƠNG TRÌNH THỰC NGHIỆM

## 3.1. Dữ liệu thực nghiệm :

Table tài khoản gồm : Tài khoản , Mật khẩu, quyền , kiểm soát

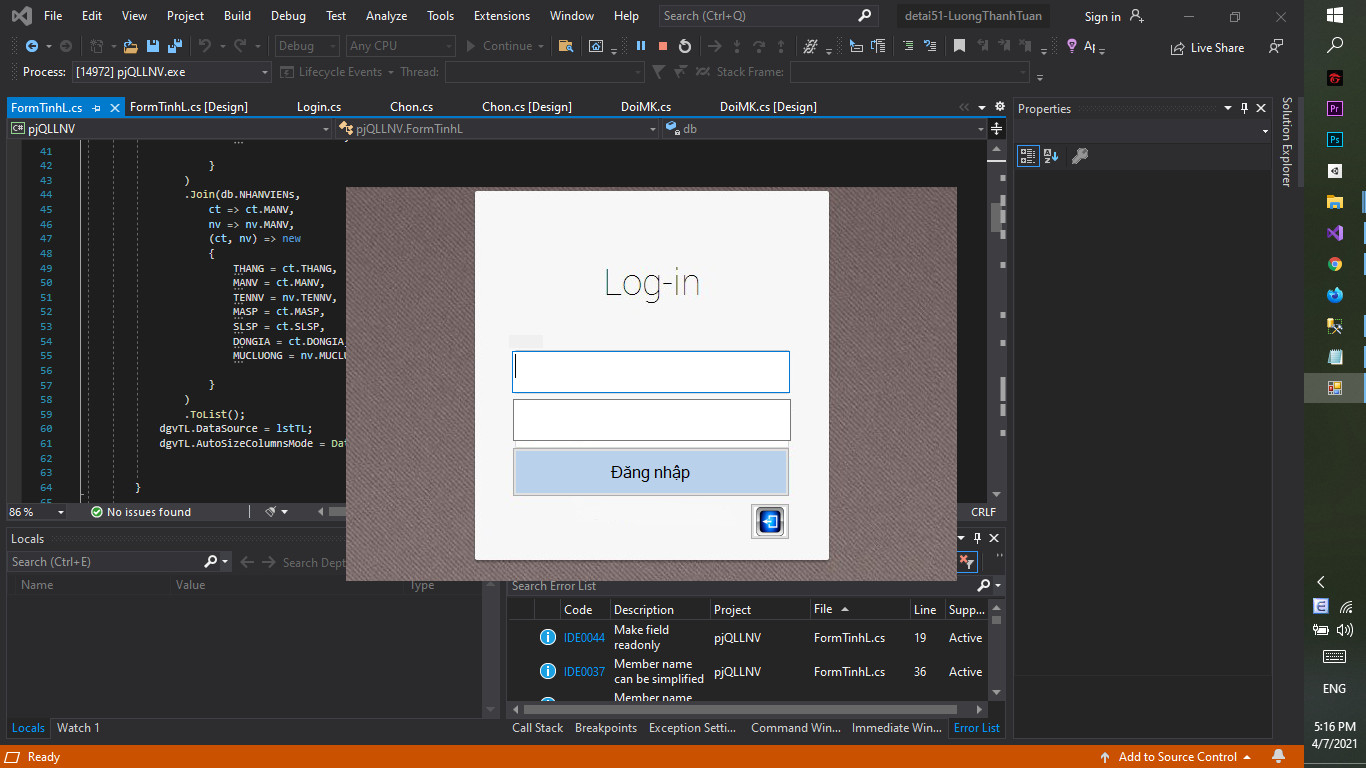
Tài khoản Admin : tuan , mật khẩu: admin , quyền : Admin , Kiểm soát : Tắt

Tài khoản User: User1, mật khẩu : user , quyền : user, kiểm soát : bật

User2, mật khẩu : user , quyền : user, kiểm soát : bật

User3, mật khẩu : user , quyền : user, kiểm soát : bật

## 3.2. Form Đăng nhập :



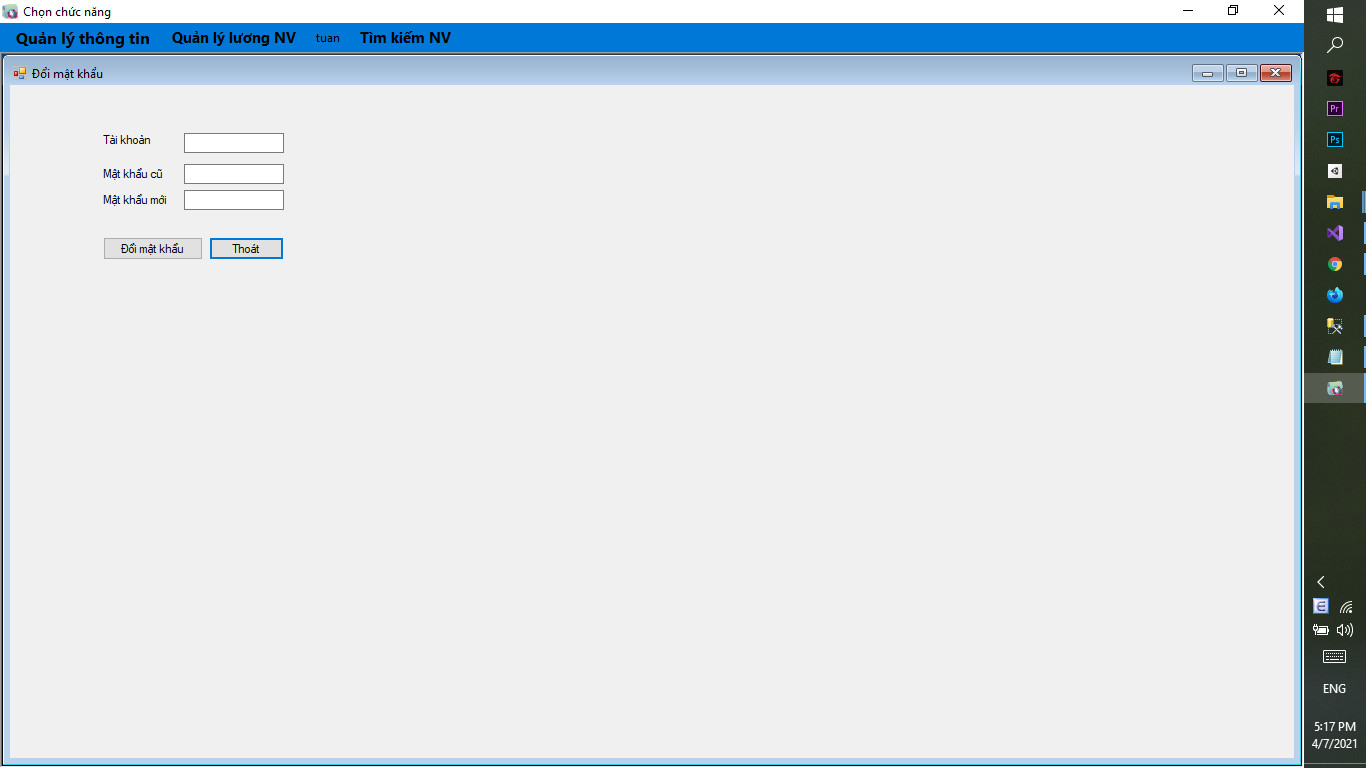
Sau khi ấn nút đăng nhập mật khẩu sẽ được mã hóa và so sánh với mật khẩu trong CSDL, thao tác này sẽ lưu lịch sử đăng nhập và chuyển đến form trang chủ.

***3.3. Form Trang chủ :***

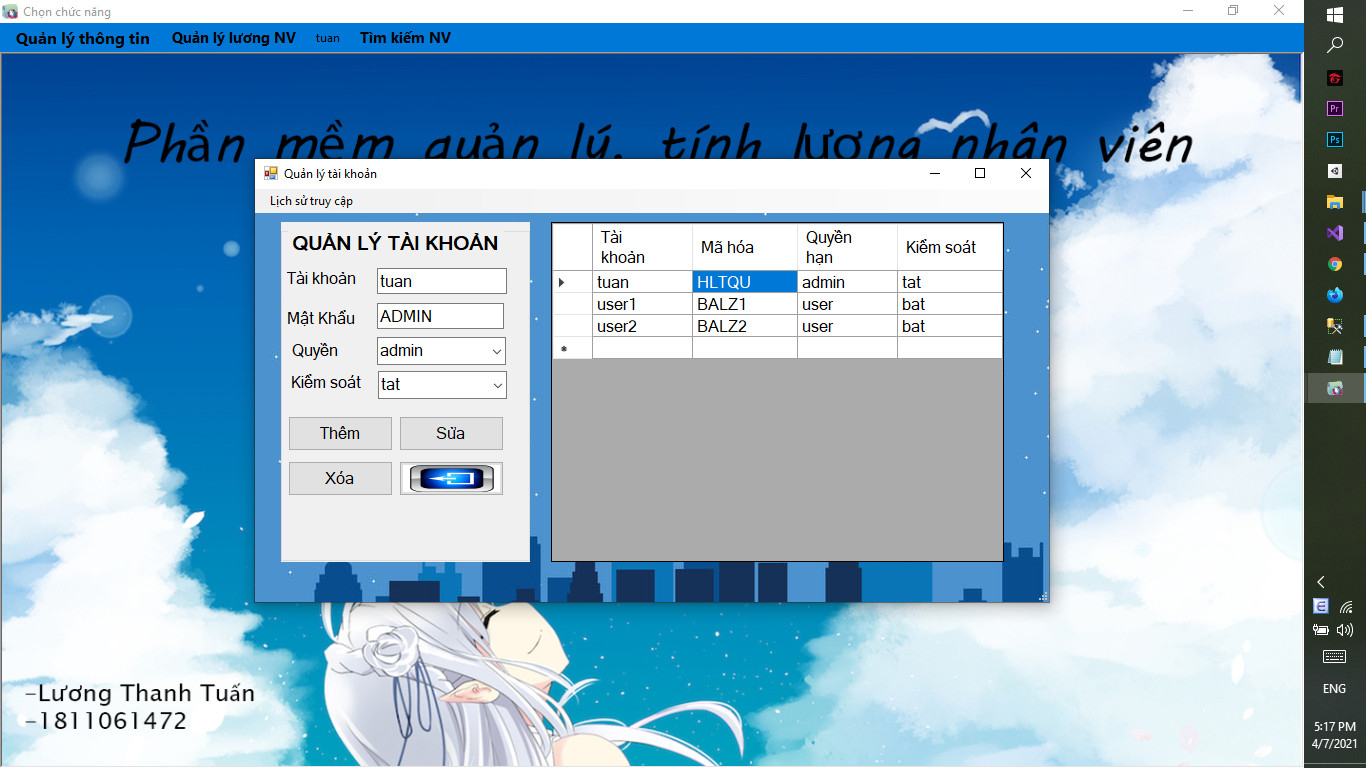


Trong trang chủ, sẽ hiển thị tên tài khoản vừa đăng nhập để tiến hành các thao tác như là : Quản lý tài khoản, Đổi mật khẩu, Đăng xuất

* + Chức năng quản lí tài khoản : Chỉ những tài khoản có quyền ( Admin) mới truy cập được
  + Tất cả các tài khoản đều có thể dùng chức năng đổi mật khẩu
  + Khi ấn vào nút đăng xuất , sẽ lưu vào lịch sử thời gian đăng xuất của tài khoản đó và quay về trang đăng nhập.

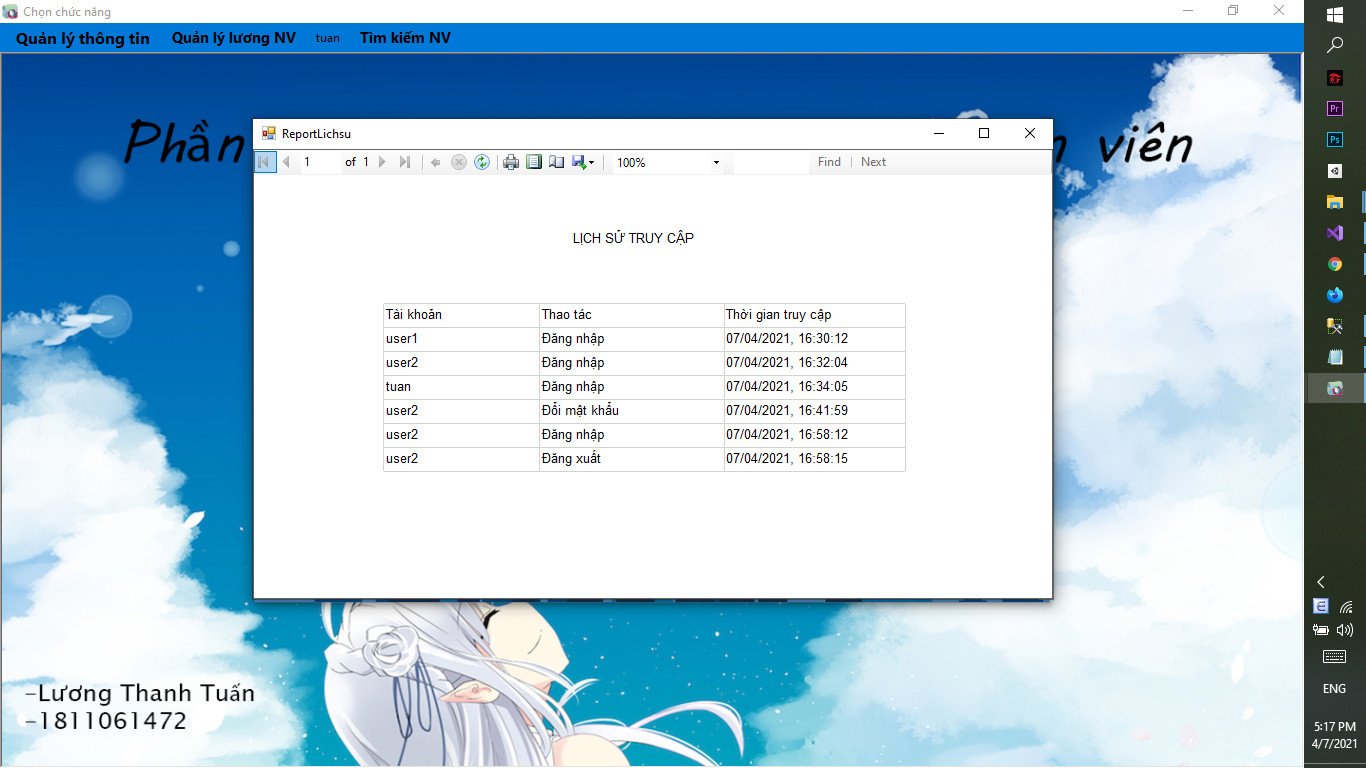
***3.4. Form đổi mật khẩu :***

* + Khi đổi mật khẩu , mật khẩu sẽ được lưu lại dưới dạng mã hóa ,và lưu hành động này ở phần lịch sử.

***3.5. Form Quản Lí Tài Khoản :***

- Trong đây các admin có thể : thêm , xóa , sửa các tài khoản và xem lại lịch sử hoạt động của các tài khoản bật quyền kiểm soát.

***3.6. Form lịch sử hoạt động :***



* + Hiển thị tên tài khoản, thao tác và thời gian truy cập của các tài khoản bật quyền kiểm soát

## 3.7. TÀI LIỆU THAM KHẢO

[1] Dr. S.B. Sadkhan, Cryptanalysis of a Vigenère, Security of Networks, 2011-2012

[2] Chris Christensen, Cryptanalysis of the Vigenère Cipher: The Friedman Test, Spring 2015, MAT/ CSC 483

[3] Jonathan Taylor, Lecture # 4 – Vigenère Cipher –Kasiski Attack, Statistics 116-Fall 2002

[4] Author: Jeremy Druin, Learning Cryptography by Doing It Wrong: Cryptanalysis of the Vigenère Cipher, jdruin@gmail.com, Advisor: Christopher Walker, CISSP, CCISO, GCED, GWEB, Accepted: 2/1/2018

[5] https://shodhganga.inflibnet.ac.in/ bitstream/10603/26543/10/10\_chapter5.pdf, CRYPTANALYSIS OF VIGENÈRE CIPHER AND SUBSTITUTION CIPHER

[6] S. S. Omran A. S. Al-Khalid D. M. Al-Saady, A Cryptanalytic Attack on Vigenère Cipher Using Genetic Algorithm, 2011 IEEE Conference on Open Systems (ICOS2011), September 25 - 28, 2011, Langkawi, Malaysia

[7] Mehmet E. Dalkilic and Cengiz Gungor, An Interactive Cryptanalysis Algorithm for the Vigenère Cipher, Ege University, International Computer Institute Bornova 35100 Izmir, TURKEY, fdalkilic,cgungorg@bornova.ege.edu.trc