**H**

**Ọ**

**C VI**

**Ệ**

**N CÔNG NGH**

**Ệ**

**BƯU CHÍNH VI**

**Ễ**

**N THÔNG**

**KHOA CÔNG NGH**

**Ệ**

**THÔNG TIN**

꧁

༺

**\***

༻

꧂

**BÁO CÁO**

**Đ**

**Ề**

**TÀI MÔN**

**H**

**Ọ**

**C**

**TH**

**Ự**

**C T**

**Ậ**

**P CƠ S**

**Ở**

**XÂY D**

**Ự**

**NG PH**

**Ầ**

**N M**

**Ề**

**M MÔ PH**

**Ỏ**

**NG TÍNH NĂNG**

**SQL SERVER**

**QU**

**Ả**

**N LÍ THƯ VI**

**Ệ**

**N**

**TH**

**Ự**

**C HI**

**Ệ**

**N Đ**

**Ề**

**TÀI:**

**N19DCCN120**

**-**

**D19CQCN01**

**-**

**N**

**-**

**NGUY**

**Ễ**

**N THANH NGH**

**Ị**

**N19DCCN175**

**-**

**D19CQCN01**

**-**

**N**

**-**

**NGUY**

**Ễ**

**N ANH TÚ**

**N19DCCN081**

**-**

**D19CQCN01**

**-**

**N**

**–**

**PHAN ANH KI**

**Ệ**

**T**



# Chương 1: Giới thiệu đề tài

## 1. Giới thiệu chung

SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ.

SQL cung cấp các tập lệnh phong phú cho các công việc hỏi đáp dữ liệu như:

* Chèn, xóa và cập nhật các hàng trong 1 quan hệ
* Tạp, thêm, xóa và sửa đổi các đối tượng trong của cơ sở dữ liệu.
* Điều khiển việc truy cấp tới cơ sở dữ liệu và các đối tượng của cơ sở dữ liệu để đảm bảo tính bảo mật, tính nhất quán và sự ràng buộc của cơ sở dữ liệu.

Đối tượng của SQL Server là các bảng dữ liệu với các cột và các hàng. Cột được gọi là trường dữ liệu và hàng là bản ghi của bảng. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu.

Một số lý do cho phép người sử dụng ưa chuộng sử dụng SQL để thực hiện đa dạng các chức năng như sau: o Nó cho phép người sử dụng có thể dùng để truy cập dữ liệu bên trong hệ thống quản lý cơ sở dữ liệu quan hệ. o Người ta cho phép người dùng mô tả dữ liệu. o Cho phép người dùng thực hiện xác định dữ liệu bên trong cơ sở dữ liệu và thực hiện các thao tác dữ liệu. o Nó cho phép nhúng trong các ngôn ngữ khác có thể sử dụng mô-đun SQL, thư viện và thực hiện trình biên dịch trước.

o Nó sẽ cho phép người dùng tạo và thả cho các cơ sở dữ liệu cũng như bảng. o SQL cho phép người sử dụng để thực hiện tạo ra các chế độ view, các thủ tục lưu trữ và chức năng trong cơ sở dữ liệu. o Nó sẽ cho phép người dùng để thực hiện thiết lập quyền trên các bảng, view và thủ tục.

## 2. Mô tả đề tài

### a) Sơ lược về nghiệp vụ quản lí thư viện

Thư viện có nhiều thủ thư. Thông tin của một thủ thư có thông tin như sau: mã thủ thư, họ, tên, phái, địa chỉ, ngày sinh...

Thư viện có nhiều độc giả. Mỗi độc giả có thông tin sau đây: Mã độc giả, họ, tên, phái, địa chỉ, ngày sinh, số điện thoại.

Thư viện có rất nhiều đầu sách. Một đầu sách có những thông tin sau đây : mã đầu sách, tên sách, số trang, đơn giá, năm xuất bản…

Một đầu sách thuộc một trong các thể loại sách. Thể Loại sách có các thông tin sau đây: Mã thể loại, tên thể loại.

Đầu Sách được sáng tác bởi một hoặc nhiều tác giả, và một tác giả sáng tác một hoặc nhiều đầu sách. Thông tin tác giả có những nội dung sau đây: Mã tác giả, họ, tên, địa chỉ, phái, email, ngày sinh.

Một đầu sách được sản xuất bởi một nhà xuất bản. Một nhà xuất bản thì xuất bản nhiều đầu sách. Thông tin của nhà xuất bản bao gồm: mã nhà xuất bản, tên , địa chỉ, số điện thoại, email.

Một cuốn sách được đặt trong một ngăn. Một ngăn có thể chứa nhiều cuốn sách và mỗi ngăn được đánh số thứ tự từ trái qua phải. và thuộc 1 kệ sách và cũng được đánh số thứ tự. Sách có thể cho mượn về hoặc chỉ cho mượn đọc tại chỗ hoặc không cho mượn.

Một đầu sách có nhiều cuốn sách giống nhau được đánh mã . Được tự sinh từ chi tiết phiếu nhập hàng khi nhập hàng. Lưu ý số lượng trong chi tiết phiếu nhập chỉ có thể thay đổi khi phiếu nhập do chưa được xếp lên kệ

Một độc giả được quyền mượn tối đa về nhà 3 cuốn và độc giả được quyền gia hạn những sách chưa đọc thêm 1 tuần sau và chỉ gia hạn được tối đa 1 lần cho mỗi cuốn sách, nếu có 1 sách quá hạn thì sẽ không được mượn thêm. Mỗi lần mượn , độc giả phải lập phiếu mượn sách, nhưng có thể trả vào các thời điểm khác nhau. Mỗi lần mượn hay trả sách ta đều phải ghi nhận thủ thư .

Độc giả phải đóng lệ phí theo năm, hết năm thì phải đóng thêm năm tiếp theo, nếu không đóng sẽ không được mượn sách nữa.

Thủ thư có nhiệm vụ sau đây: nhập sách, thống kê sách, quản lý độc giả, thông báo và áp dụng mức hình phạt cho độc giả nếu quá trễ hạn (15 ngày) , quản lý cho mượn/trả sách, thanh lý sách, tạo độc giả mới, hủy độc giả, …

Những độc giả trả sách muộn, làm mất, rách sách thư viện hay cố tình vi phạm nội quy của Thư viện thì sẽ bị xử phạt theo quy định của Thư viện. **b) Phân tích, thiết kế CSDL**

**1/ Xác định các thực thể:**

* **THUTHU**(MATT, HOTEN,SDT, GIOITINH, DIACHI, NGAYSINH, NGHIVIEC)
* **DOCGIA**(MADG, HO, TEN, PHAIDG, DIACHI, NGAYSINH, EMAIL, SDT)
* **TACGIA**(MATG, HOTEN, DIACHI, GIOITINH, NGAYSINH)
* **NHAXUATBAN**(MANXB, TENNXB, DIACHI, SDT, EMAIL)
* **THELOAI** (MATL, TENTL)
* **KE** (MAKE, TENKE)
* **NGAN** (MANGAN)
* **DAUSACH** (MADS, TENSACH, KHOSACH, SOTRANG, NAMXB)
* **CUONSACH** (MAS)
* **PHIEUMUONTRA**(MAPMT, NGAYMUON, NGAYHENTRA)
* **PHIEUNHAP**(MAPN, THOIGIANLAP)

**2/ Mô hình quan hệ từ ERD (đạt dạng chuẩn 3):**

* Khoá chính: gạch chân.
* Khoá ngoại: màu đỏ. **-** Khoá chính và ngoại: màu đỏ, gạch chân.

* **THUTHU**(MATT, HOTEN, SDT, GIOITINH, DIACHI, NGAYSINH, EMAIL, NGHIVIEC)
* **DOCGIA**(MADG, HOTEN, GIOITINH, DIACHI, NGAYSINH, SDT)
* **TACGIA**(MATG, HOTEN, DIACHI, GIOITINH, NGAYSINH)
* **NHAXUATBAN**(MANXB, TENNXB, DIACHI)
* **THELOAI** (MATL, TENTL)
* **KE** (MAKE, TENKE)
* **NGAN** (MANGAN, MAKE)
* **DAUSACH** (MADS, TENSACH, KHOSACH, SOTRANG, NAMXB, MATL, MANXB)
* **CUONSACH** (MAS, MANGAN, MACTPN)
* **PHIEUMUONTRA**(SOPHIEU, NGAYMUON, NGAYHENTRA,

TONGSACH, MADG, MATT)

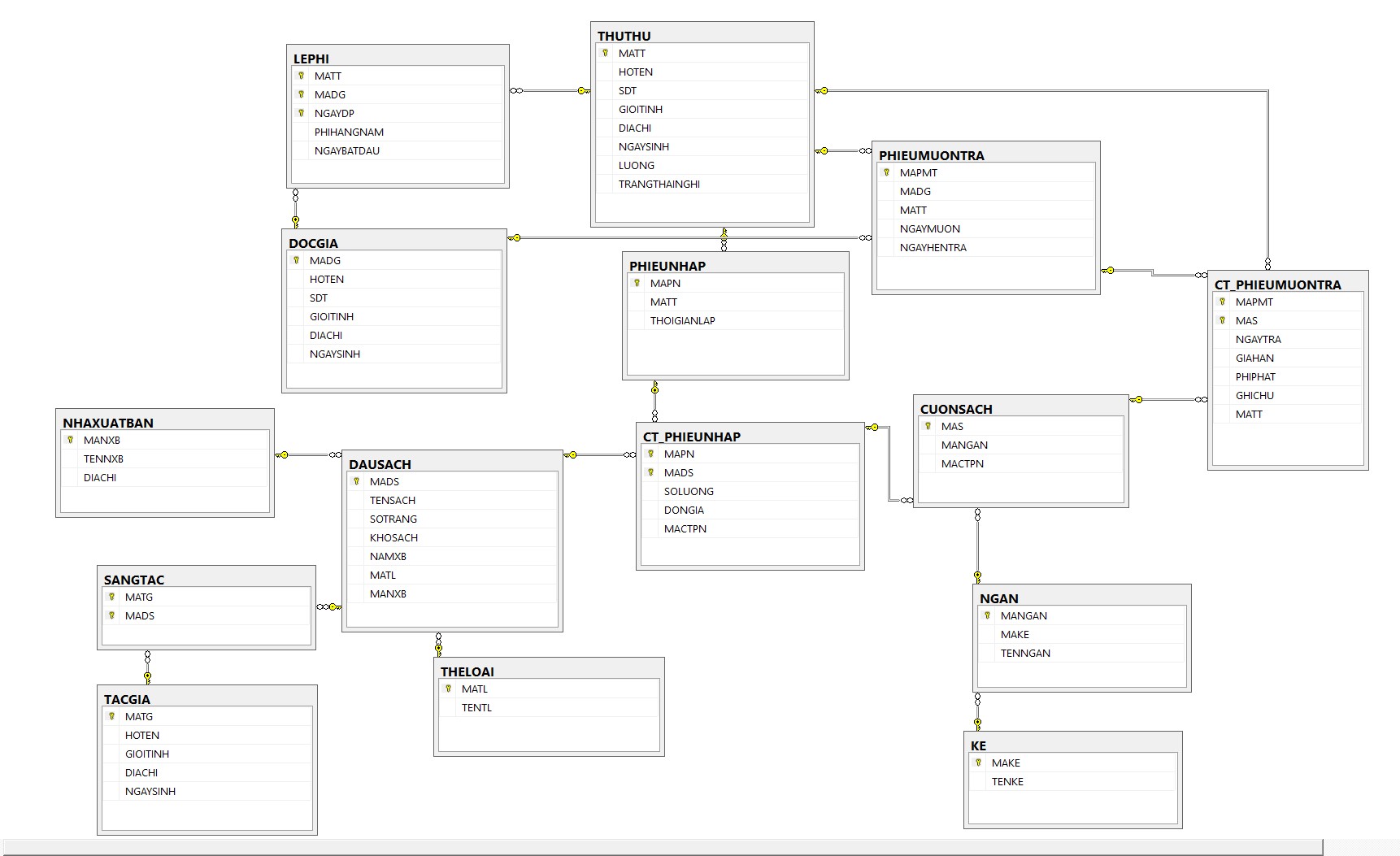
* **PHIEUNHAP**(MAPN, NGAY, MATT)
* **LEPHI** (MADG, MATT, NGAYDP, PHIHANGNAM, NGAYBD)

-**SANGTAC** (MATG, MADS)

-**CT\_PHIEUNHAP**(MAPN, MADS, SOLUONG, DONGIA, MACTPN)

-**CT\_MUONTRA**(MAPMT, MAS, NGAYTRA, PHIPHAT, GIAHAN, GHICHU, MATT)

**3/ Mô hình Diagram:**

**4/ Từ điển dữ liệu:**

**THỦ THƯ** (MATT, HOTEN, SDT, GIOITIH , NGAYSINH, DIACHI, LUONG, NGHIVIEC)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MATT | nchar | 10 | PK | Mã thủ thư |
| 2 | HOTEN | nvarchar | 40 |  |  |
| 3 | SDT | nvarchar | 10 |  |  |
| 4 | GIOITINH | nvarchar | 3 | NULL, Default : ‘Nam’  Chỉ nhân ‘Nam’ hoặc  ‘Nữ’ | Phái thủ thư |
| 5 | NGAYSINH | Date |  | NULL |  |
| 6 | DIACHI | nvarchar | 50 | NULL, Default : ‘ ‘ |  |
| 7 | LUONG | MONEY |  | Default : 0 |  |
| 8 | NGHIVIEC | Bit |  | Default : 0 | Nghỉ việc |

**ĐỘC GIẢ** (MADG, HOTEN, SDT, GIOITINH , NGAYSINH, DIACHI)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MADG | nchar | 10 | PK | Mã độc giả |
| 2 | HOTEN | nvarchar | 40 |  |  |
| 3 | SDT | nvarchar | 10 |  |  |
| 4 | GIOITINH | nvarchar | 3 | NULL, Default : ‘Nam’ chỉ nhân ‘Nam’ hoặc ‘Nữ’ | Phái độc giả |
| 5 | NGAYSINH | Date |  | NULL |  |
| 6 | DIACHI | nvarchar | 50 | NULL, Default : ‘ ‘ |  |

**TÁC GIẢ** (MATG, HOTEN, GIOTINH, NGAYSINH, DIACHI)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MATG | nchar | 10 | PK | Mã tác giả |
| 2 | HOTEN | nvarchar | 40 |  |  |
| 3 | GIOITINH | nvarchar | 10 | NULL, Default : ‘Nam’, chỉ nhân ‘Nam’ hoặc ‘Nữ’ |  |
| 4 | NGAYSINH | Date |  | NULL |  |
| 5 | DIACHI | nvarchar | 50 | NULL, Default : ‘ ‘ |  |

**NHÀ XUẤT BẢN**(MANXB, TENNXB, DIACHI)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MANXB | nchar | 10 | PK | Mã nhà xuất bản |
| 2 | TENNXB | nvarchar | 40 |  |  |
| 3 | DIACHI | nvarchar | 50 | NULL, Default : ‘ ‘ |  |

**THỂ LOẠI**(MATL, TENTL)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MATL | nchar | 10 | PK | Mã thể loại |
| 2 | TENTL | nvarchar | 30 | UK | Tên thể loại |

**CUỐN SÁCH**(**MAS**, MANGAN, MACTPN)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ**  **dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MAS | nchar | 10 | PK | Mã sách |
| 2 | MANGAN | nvarchar | 10 | FK | Mã ngăn |
| 3 | MACTPN | int |  |  | Giá thanh lý |

**ĐẦU SÁCH**(MADS, TENSACH, KHOSACH,SOTRANG, NAM, MATL, MANXB)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ**  **dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MADS | nchar(10) | 10 | PK | Mã nhà xuất bản |
| 2 | TENSACH | nvarchar | 50 | UK |  |
| 3 | KHOSACH | Nvarchar | 3 | NULL, Default : ‘VỪA’  Chỉ nhận‘NHỎ’,’VỪA’,’LỚN’ | Khổ sách |
| 4 | SOTRANG | Int |  | >0 | Số trang |
| 5 | NAMXB | Int |  |  |  |
| 6 | MATL | nchar | 10 | FK | Mã thể loại |
| 7 | MANXB | nchar | 10 | FK | Mã nhà xuất bản |

**KỆ**(MAKE, TENKE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MAKE | nvarchar | 10 | PK | Mã kệ |
| 2 | TENKE | nvarchar | 10 | UK | Tên kệ |

**NGĂN**(MANGAN, MAKE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MANGAN | nvarchar | 10 | PK | Mã ngăn |
| 2 | MAKE | nvarchar | 10 | FK | Mã kệ |

**PHIẾU MƯỢN TRẢ**(MAPMT, NGAYMUON, TONGSACH, NGAYHENTRA, MADG, MATT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MAPMT | nchar | 10 | PK | Số phiếu mượn trả |
| 2 | NGAYMUON | Date | 50 | >= NGAYDK |  |
| 3 | TONGSACH | Int |  | >=1 và <= 3 | Tổng sách |
| 3 | NGAYHENTRA | Date |  | >= NGAYMUON,  (NGAYHENTRA - NGAYMUON) <=7 | Ngày hẹn trả |
| 4 | MADG | nchar | 10 | FK | Mã độc giả |
| 5 | MATT | nchar | 10 | FK | Mã thủ thư |

**PHIẾU NHẬP**(MAPN, THOIGIANLAP, MATT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | SOPHIEUNHAP | nchar | 10 | PK | Số phiếu nhập |
| 2 | THOIGIANLAP | Date |  |  | Ngày nhập |
| 3 | MATT | nchar | 10 | FK | Mã thủ thư |

**LỆ PHÍ**(MATT, MADG, NGAYDP, PHIHANGNAM, NGAYBD)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MATT | Nchar | 10 | PK,FK | Mã thủ thư |
| 2 | MADG | Nchar | 10 | PK,FK | Mã độc giả |
| 3 | NGAYDP | Date |  | PK | Ngày đóng phí |
| 4 | PHIHANGNAM | Money |  | >=0 | Phí hằng năm |
| 5 | NGAYBD | Date |  | >= NGAYDP | Ngày bắt đầu |

**SÁNG TÁC**(MATG, MADS)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MATG | nchar | 10 | PK,FK | Mã tác giả |
| 2 | MADS | Nchar | 10 | PK,FK | Mã đầu sách |

**CHI TIẾT MƯỢN TRẢ**(MAPMT, MAS,LOAI,NGAYTRA,GIAHAN,

PHIPHAT,GHICHU,MATT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MAPMT | Nchar | 10 | PK,FK | Số phiếu mượn trả |
| 2 | MAS | Nchar | 10 | PK,FK | Mã sách |
| 4 | NGAYTRA | Date |  | NULL  >= NGAYMUON |  |
| 5 | GIAHAN | Bit |  | NULL, Default: 0 |  |
| 6 | PHIPHAT | Money |  | NULL, Default: 0 |  |
| 7 | GHICHU | nText |  | Null |  |
| 8 | MATT | Nchar | 10 | FK, NULL | Mã thủ thư |

**CHI TIẾT NHẬP**(MAPN, MADS, SOLUONG, DONGIA, MACTPN)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **stt** | **Thuộc tính** | **Kiểu dữ**  **liệu** | **Độ dài** | **Ràng buộc** | **Ghi chú** |
| 1 | MAPN | Nchar | 10 | PK,FK | Số phiếu nhập |
| 2 | MADS | Nchar | 10 | PK,FK | Mã đầu sách |
| 3 | SOLUONG | Int |  | >0 | Số lượng sách nhập |
| 4 | DONGIA | Money |  | >=0 | Giá nhập của 1 cuốn sách |
| 5 | MACTPN | int |  | UK | Mã chi tiết phiếu nhập duy nhất |

### c)Phần mềm hỗ trợ thực hiện

-Microsoft SQL Server Management Studio (Enterprise) 2014 -Visual Studio 2019

### d)Mô tả đề tài

Sau khi đã thiết kế xong database: tiến hành viết SP thực thi công việc tạo quản lí login trên server, backup restore để tiến hành viết phần mềm mô phỏng các tính năng trên của SQL server. Bên cạnh đó kết hợp viết trigger để xử lí một số quá trình nghiệp vụ trong đồ án. Cụ thể ở phần nhập hàng trong chi tiết khi nhân viên nhập về 1 số lượng sách trigger sẽ tự sinh ra số lượng mã sách tương ứng với đầu sách đã nhập về.

Các chức năng phần mềm:

+Tạo login server, database role

+ Backup full, backup difference databse

+ Đặt lịch backup tự động

+ Restore databse từ những file đã backup

**e)Link github source**

https://github.com/nghi1421/TTCS\_QuanLiThuVien

# Chương 2: Cơ sở lí thuyết

* Ngôn ngữ sử dụng: C# trên nền tảng .NET.
* Ưu điểm:

C#:

C# có rất nhiều ưu điểm nổi bật như:

* + Gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java, Pascal.
  + Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của chúng.
  + Cải tiến các khuyết điểm của C/C++ như con trỏ, hiệu ứng phụ,...
  + Dễ dàng tiếp cận, dễ phát triển.
  + Được sự chống lưng của .NET Framework.

Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows.

Framework .NET:

* + Sử dụng được đa ngôn ngữ, đa nền tảng
  + Giảm thiểu được nhiều xung đột, an toàn, tính bảo mật cao
  + Tăng hiệu suất, giảm chi phí khi sử dụng
  + Khả năng tích hợp cao với nhiều ngôn ngữ
  + Tận dụng được các dịch vụ của hệ điều hành

## 1.Backup

Backup là một bản sao lưu dữ liệu/cơ sở dữ liệu. Việc sao lưu CSDL trong MS

SQL Server là rất quan trọng để bảo vệ dữ liệu trước việc mất CSDL. Có 3 hình thức sao lưu chính trong MS SQL Server là **Full/Database, Differential/Incremental, Transactional Log/Log.**

## 2.Restore

Khôi phục là quá trình sao chép dữ liệu đã sao lưu và đưa các giao dịch được ghi lại vào dữ liệu của MS [SQL Server.](https://quantrimang.com/sql-server) Hiểu đơn giản, đây là quá trình lấy file sao lưu và đưa nó trở lại CSDL.

Đồ án này bao gồm 2 hình thức back up là full backup (thủ công và tự động) và diferential backup được thực hiện hoàn toàn bằng T-SQL.

## 3.Role

Trong bất kỳ hệ thống cơ sở quản trị dữ liệu nào, thì vấn đề an ninh, bảo mật luôn được đặt lên hàng đầu. Và với SQL Server, nếu chúng ta tận dụng được nhiều ưu điểm của ứng dụng này, những người làm công việc quản trị hệ thống sẽ giảm bớt được rất nhiều gánh nặng cũng như áp lực trong công việc. Với bài viết hướng dẫn dưới đây, chúng tôi sẽ giới thiệu với các bạn một số thông tin cơ bản cũng như cách sử dụng, quản lý role của SQL Server.

Việc tạo cơ sở dữ liệu là của riêng người quản trị, nhưng các bạn cần lưu ý một số điểm sau về quy chuẩn chung khi tạo bảng:

* **db\_owner:** toàn bộ người dùng có quyền full – access
* **db\_accessadmin:** người dùng có quyền quản lý các Windows Group và tài khoản SQL Server đăng nhập
* **db\_datareader:** người dùng có thể đọc được toàn bộ dữ liệu
* **db\_datawriter:** người dùng có quyền thêm, xóa hoặc chỉnh sửa dữ liệu trong bảng
* **db\_ddladmin:** người dùng có thể sử dụng các file dynamic – link library (DLL) - **db\_securityadmin:** người dùng có thể chỉnh sửa vai trò role và quản lý các bậc quản lý, phân quyền khác
* **db\_bckupoperator:** người dùng có thể sao lưu cơ sở dữ liệu
* **db\_denydatareader:** người dùng không thể xem dữ liệu trong bảng
* **db\_denydatawriter:** người dùng không thể xem, thay đổi hoặc xóa dữ liệu trong bảng

## 4.Trigger

Trigger là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như Insert, Delete, hay Update. Trigger là một loại stored procedure đặc biệt (không có tham số) được thực thi (execute) một cách tự động khi có một sự kiện thay đổi dữ liệu (data modification). Trigger được lưu trữ và quản lý trong Server DB, được dùng trong trường hợp ta muốn kiểm tra các ràng buộc toàn vẹn trong DB.

Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ (nhiều bảng/table) hoặc trên nhiều dòng (nhiều record) của bảng.

Ngoài ra việc sử dụng Trigger để chương trình có những hàm chạy ngầm nhằm phục vụ nhưng trường hợp hữu hạn và thường không sử dụng cho mục đích kinh doanh hoặc giao dịch.

# Chương 3: Thực nghiệm

## I. Backup & Restore

**1. Tạo backup thủ công:**

**1.1.Full back up:**

**Yêu cầu:** File full backup có thể được tạo mà hầu như không có điều kiện đặc biệt nào.

**Lệnh T-SQL:**

BACKUP DATABASE [QUANLITHUVIEN] TO DISK= N'" + @backupLocation+

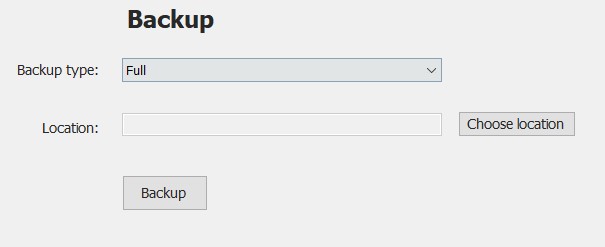
"\\QuanLiThuVien\_FULL\_" + DateTime.Now.ToString("yyyyMMddHHmm") +

".BAK' WITH NOFORMAT, NOINIT, NAME = N'QuanLiThuVien-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

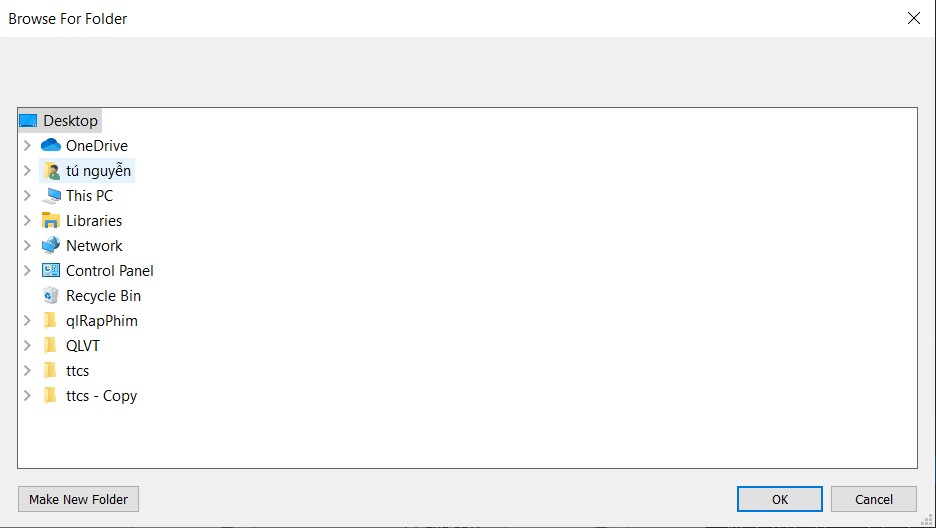
**Các bước tạo full backup:**

Bước 1: Dòng back up type chọn Full

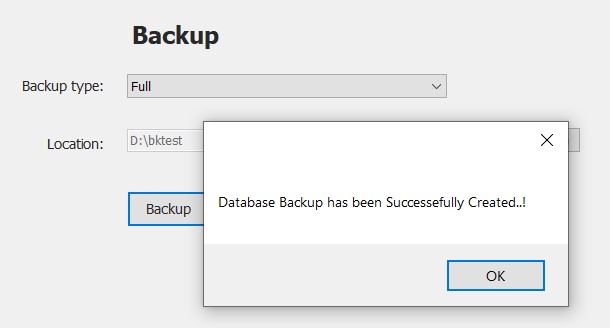
Bước 2: Dòng location ấn choose location



Bước 3: Chọn folder muốn lưu, ấn ok



Bước 4: Ấn backup



**1.2. Difference back up:**

Ưu điểm của difference backup là kích thước file của nó thường nhỏ hơn nhiều so với file full backup, vì vậy , đối với cơ sở dữ liệu rất lớn ta thường dùng differential backup để sao lưu các bản phân biệt. Trong một số trường hợp, các bản sao lưu khác biệt được sử dụng trước khi một thay đổi lớn được thực hiện trong cơ sở dữ liệu. Sau đó, các bản sao lưu có thể được sử dụng để khôi phục những thay đổi đó nếu và khi cần thiết. Nhược điểm khi sử dụng loại sao lưu difference backup này là chúng yêu cầu một bản sao lưu full backup làm điểm bắt đầu cho quá trình khôi phục. Làm cách nào để đảm bảo rằng có một bản sao của các bản sao lưu cơ sở dữ liệu đầy đủ trước khi ta tạo các loại sao lưu difference này?

Cần lưu ý rằng nếu bạn đang tạo các bản difference backup hoặc bất kỳ loại backup nào khác ngoài bản full backup, thì file full backup phải có sẵn. Điều này đặc biệt hữu ích trong những trường hợp bạn cần thực hiện một thay đổi lớn trong cơ sở dữ liệu của mình và muốn có sẵn một bản backup cho mục đích khôi phục trong trường hợp cần thiết.

Sp dưới đây đòi hỏi file full backup gần nhất phải được đặt trong folder muốn tạo file difference backup:

CREATE PROCEDURE [dbo].[sp\_generateDifferentialBackups]

@strBackupPath NVARCHAR (25) ='D:\bktest\' --variable for location of DIFFERENTIAL backups

AS

SET NOCOUNT ON

-- Get the name of all user databases

DECLARE @strTimeStamp NVARCHAR(12) --variable for timestamp value

DECLARE @strSQL NVARCHAR(100) -- used for generating dynamic SQL statements

DECLARE @databaseName NVARCHAR(128) -- used as variable to store database names set @databaseName='quanlithuvien'

BEGIN

PRINT 'Checking for the latest FULL database backup for: [' + @databaseName +']'

DECLARE @strphysical\_device\_name NVARCHAR(400) -- variable for physical\_device\_name

-- Reads the MSDB database to check for the latest FULL database backup

set @strphysical\_device\_name= (SELECT TOP 1 physical\_device\_name

FROM msdb.dbo.backupset a INNER JOIN msdb.dbo.backupmediafamily b

ON a.media\_set\_id = b.media\_set\_id

WHERE database\_name='' + @databaseName + ''

AND type='D'

ORDER BY backup\_finish\_date desc)

--Check if the database does not have any FULL backups at all if (@strphysical\_device\_name is null) BEGIN

return 0;

END

ELSE

--If the database has FULL backups,

BEGIN

DECLARE @result INT

EXEC xp\_fileexist @strphysical\_device\_name, @result output

IF (@result = 1) --@result will return 1 if the specified file exists, if it doesn't, it will return 0

BEGIN

SET @strTimeStamp=(SELECT FORMAT (getdate(), 'yyyyMMddHHmmss'))

SET @strTimeStamp=@strTimeStamp +

REPLACE(CONVERT(CHAR(8), GETDATE(), 108),':','')

EXEC ('BACKUP DATABASE ' + @databaseName + ' TO DISK='''

+ @strBackupPath + @databaseName + '\_DIFF\_' + @strTimeStamp +

'.BAK'' WITH INIT, DIFFERENTIAL, DESCRIPTION=''DIFFERENTIAL Backup for ' + @databasename + ' database''') -- change this value should you decide to change the backup type to something other than differential

PRINT '=================database backup: ' + @strphysical\_device\_name

PRINT 'DIFFERENTIAL database backup: ' + @strBackupPath + @databaseName + '\_DIFF\_' + @strTimeStamp + '.BAK'

RETURN 1;

END

ELSE

BEGIN

PRINT 'WARNING: FULL database backup file is missing. Generate the FULL database backup first. Aborting DIFFERENTIAL backup command.'

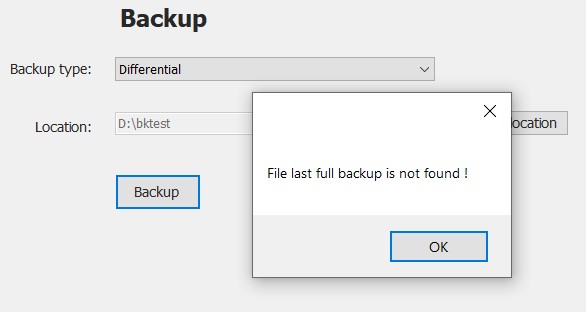
RETURN 0;

END

END

END

Stored procedure trên sẽ cho phép tạo 1 file difference backup mới trong đường dẫn đưa vào nếu như file full backup gần nhất được tìm thấy trong đường dẫn đó. Trường hợp nếu không tìm thấy sẽ báo lỗi:



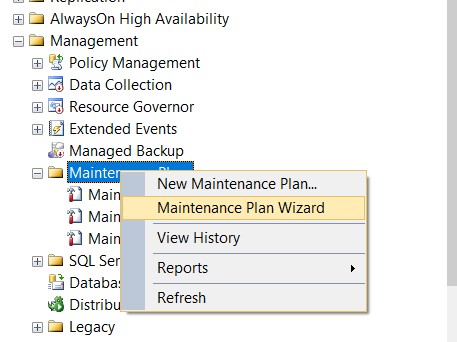
**2.Back up tự động:**

**2.1.Full backup:**

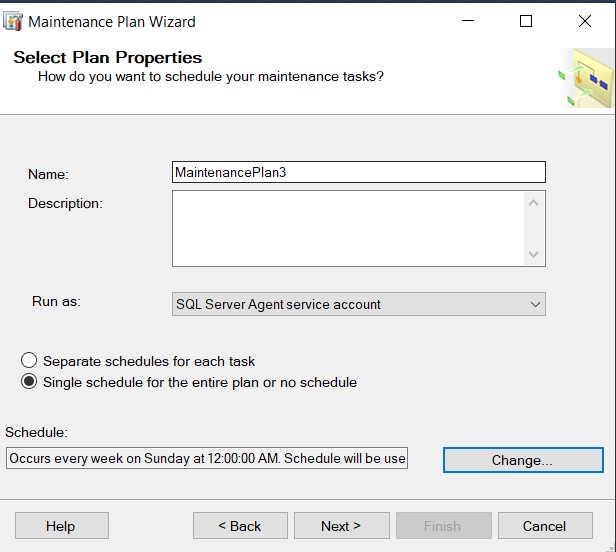
Yêu cầu: tạo 2 maintenance plan 1 cho full database backup và 1 cho difference backup, viết 2 sp back\_up\_full và back\_up\_diff.

Các bước tạo maintainence plan:

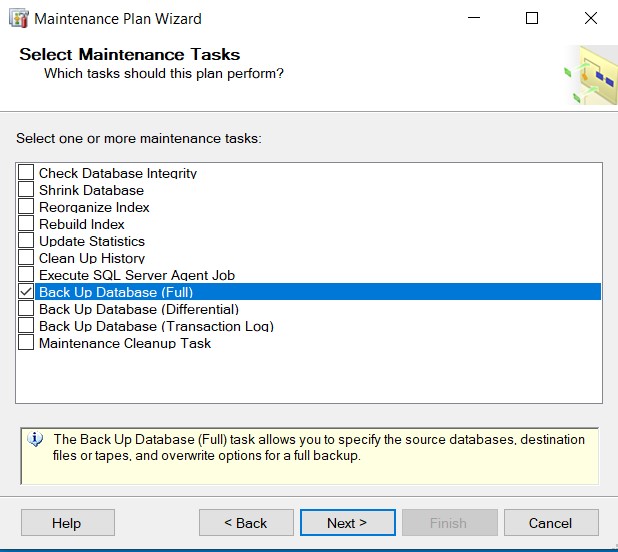
Bước 1: chọn maintenance plan -> maintenance plan wizard



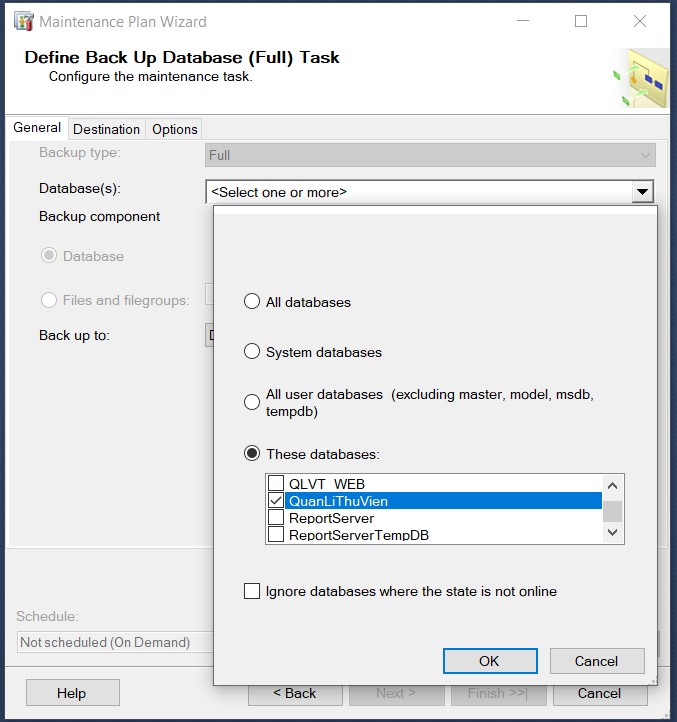
Bước 2:Chọn single chedule for entire plan or no schedule, ấn change đặt 1 lịch back up bất kì (để lấy ra biến @schedule uid , dùng cho việc viết sp backup)



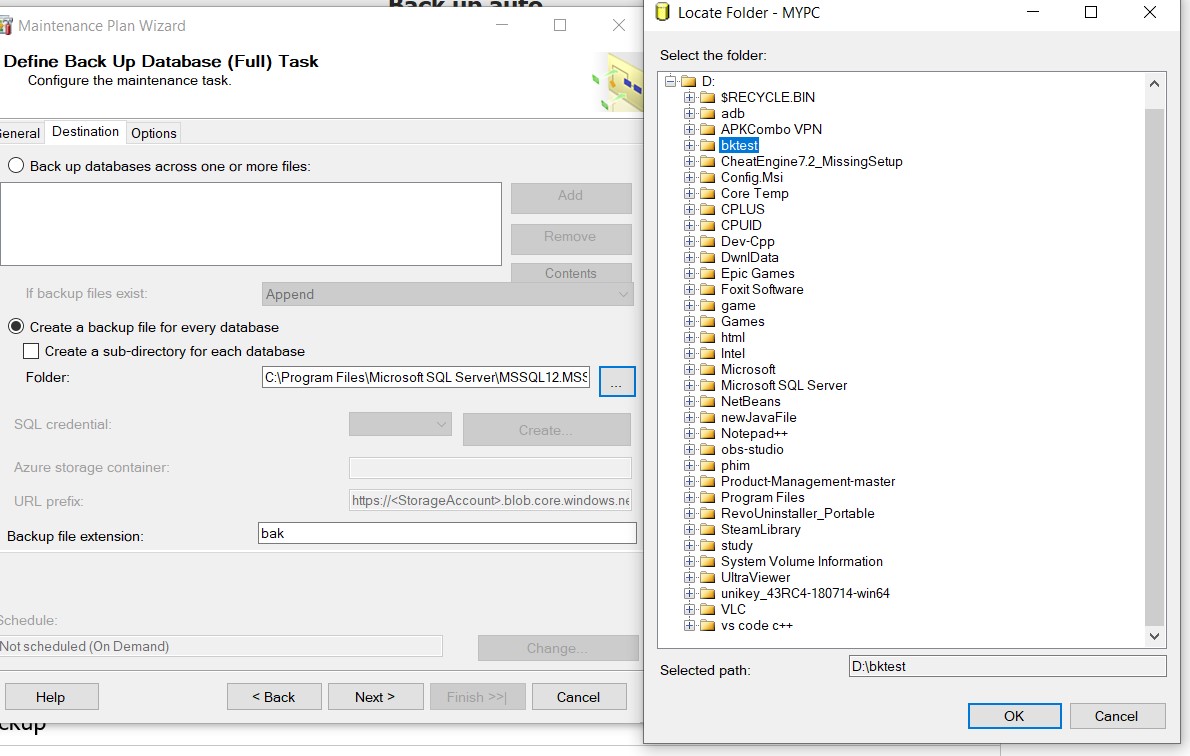
Bước 3:Chọn back up database (Full) /(Diferential)



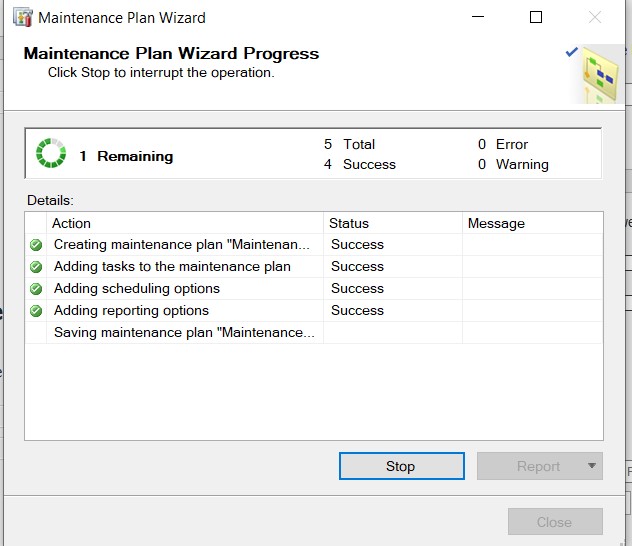
Bước 4: Chọn database quanlithuvien



Bước 5: Chọn nơi lưu file.



Bước 6: Ấn finish



**Tạo sp:**

Sp BACK\_UP\_FULL:

Create PROCEDURE [dbo].[BACK\_UP\_FULL] @subtenjob varchar(3),@f\_type int

,@f\_interval int,@subday\_type int,@subday\_interval int,

@f\_relative\_interval int,@f\_recurrence\_factor int, @ngay\_bat\_dau int,

@ngay\_ket\_thuc int,@gio\_bat\_dau int, @gio\_ket\_thuc int as

BEGIN TRANSACTION

DECLARE @ReturnCode INT

SELECT @ReturnCode = 0

/\*\*\*\*\*\* Object: JobCategory [Database Maintenance] Script Date: 6/3/2022 5:06:35 PM \*\*\*\*\*\*/

IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'Database Maintenance' AND category\_class=1)

BEGIN

EXEC @ReturnCode = msdb.dbo.sp\_add\_category @class=N'JOB',

@type=N'LOCAL', @name=N'Database Maintenance'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)

DECLARE @tenMoi VARCHAR(50)

SET @tenMoi = CONCAT('MaintenancePlan.Subplan\_', @subtenjob)

EXEC @ReturnCode = msdb.dbo.sp\_add\_job @job\_name=@tenMoi,

@enabled=1,

@notify\_level\_eventlog=2,

@notify\_level\_email=0,

@notify\_level\_netsend=0,

@notify\_level\_page=0,

@delete\_level=0,

@description=N'No description available.',

@category\_name=N'Database Maintenance',

@owner\_login\_name=N'sa', @job\_id = @jobId OUTPUT

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback /\*\*\*\*\*\* Object: Step [Subplan\_1] Script Date: 6/3/2022 5:06:35 PM \*\*\*\*\*\*/

DECLARE @sub\_plan\_moi VARCHAR(50)

SET @sub\_plan\_moi = CONCAT('Subplan\_', @subtenjob)

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=@sub\_plan\_moi,

@step\_id=1,

@cmdexec\_success\_code=0,

@on\_success\_action=1,

@on\_success\_step\_id=0,

@on\_fail\_action=2,

@on\_fail\_step\_id=0,

@retry\_attempts=0, @retry\_interval=0,

@os\_run\_priority=0, @subsystem=N'SSIS',

@command=N'/Server "$(ESCAPE\_NONE(SRVR))" /SQL

"Maintenance Plans\MaintenancePlan" /set "\Package\Subplan\_1.Disable;false"',

@flags=0

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_update\_job @job\_id = @jobId, @start\_step\_id = 1

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobschedule @job\_id=@jobId, @name=N'MaintenancePlan',

@enabled=1,

@freq\_type=@f\_type,

@freq\_interval=@f\_interval,

@freq\_subday\_type=@subday\_type,

@freq\_subday\_interval=@subday\_interval,

@freq\_relative\_interval=@f\_relative\_interval,

@freq\_recurrence\_factor=@f\_recurrence\_factor,

@active\_start\_date=@ngay\_bat\_dau,

@active\_end\_date=@ngay\_ket\_thuc,

@active\_start\_time=@gio\_bat\_dau,

@active\_end\_time=@gio\_ket\_thuc,

@schedule\_uid=N'6ceef16d-ac6d-43c1-910d-6c42f50bdc81'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobserver @job\_id = @jobId, @server\_name = N'(local)'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

COMMIT TRANSACTION

GOTO EndSave

QuitWithRollback:

IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION EndSave:

SP BACK\_UP\_DIFF:

CREATE PROCEDURE [dbo].[BACK\_UP\_DIFF] @subtenjob varchar(3),@f\_type int

,@f\_interval int,@subday\_type int,@subday\_interval int, @f\_relative\_interval int,@f\_recurrence\_factor int, @ngay\_bat\_dau int,

@ngay\_ket\_thuc int,@gio\_bat\_dau int, @gio\_ket\_thuc int as

BEGIN TRANSACTION

DECLARE @ReturnCode INT

SELECT @ReturnCode = 0

/\*\*\*\*\*\* Object: JobCategory [Database Maintenance] Script Date: 6/3/2022 5:46:18 PM \*\*\*\*\*\*/

IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'Database Maintenance' AND category\_class=1)

BEGIN

EXEC @ReturnCode = msdb.dbo.sp\_add\_category @class=N'JOB',

@type=N'LOCAL', @name=N'Database Maintenance'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)

DECLARE @tenMoi VARCHAR(50)

SET @tenMoi = CONCAT('MaintenancePlan1.Subplan\_', @subtenjob)

EXEC @ReturnCode = msdb.dbo.sp\_add\_job @job\_name=@tenMoi,

@enabled=1,

@notify\_level\_eventlog=2,

@notify\_level\_email=0,

@notify\_level\_netsend=0,

@notify\_level\_page=0,

@delete\_level=0,

@description=N'No description available.',

@category\_name=N'Database Maintenance',

@owner\_login\_name=N'sa', @job\_id = @jobId OUTPUT

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

/\*\*\*\*\*\* Object: Step [Subplan\_1] Script Date: 6/3/2022 5:46:18 PM \*\*\*\*\*\*/

DECLARE @sub\_plan\_moi VARCHAR(50)

SET @sub\_plan\_moi = CONCAT('Subplan\_', @subtenjob)

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=@sub\_plan\_moi,

@step\_id=1,

@cmdexec\_success\_code=0,

@on\_success\_action=1,

@on\_success\_step\_id=0,

@on\_fail\_action=2,

@on\_fail\_step\_id=0,

@retry\_attempts=0, @retry\_interval=0,

@os\_run\_priority=0, @subsystem=N'SSIS',

@command=N'/Server "$(ESCAPE\_NONE(SRVR))" /SQL

"Maintenance Plans\MaintenancePlan1" /set "\Package\Subplan\_1.Disable;false"',

@flags=0

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_update\_job @job\_id = @jobId, @start\_step\_id = 1

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobschedule @job\_id=@jobId, @name=N'MaintenancePlan1.Subplan\_1',

@enabled=1,

@freq\_type=@f\_type,

@freq\_interval=@f\_interval,

@freq\_subday\_type=@subday\_type,

@freq\_subday\_interval=@subday\_interval,

@freq\_relative\_interval=@f\_relative\_interval,

@freq\_recurrence\_factor=@f\_recurrence\_factor,

@active\_start\_date=@ngay\_bat\_dau,

@active\_end\_date=@ngay\_ket\_thuc,

@active\_start\_time=@gio\_bat\_dau,

@active\_end\_time=@gio\_ket\_thuc,

@schedule\_uid=N'05dfe2fd-697e-4012-9d68-5b5b39fea90f'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobserver @job\_id = @jobId, @server\_name = N'(local)'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

COMMIT TRANSACTION

GOTO EndSave

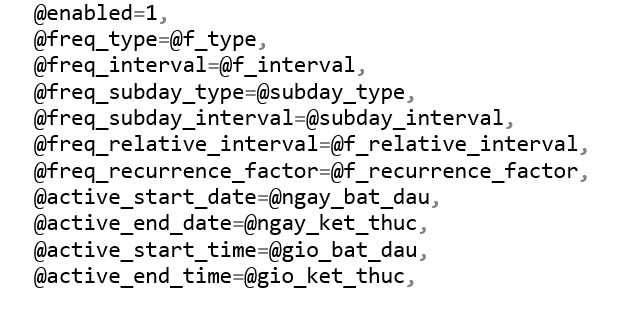
QuitWithRollback:

IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION EndSave:

**Chú ý:** mục @schedule\_uid đặt lại theo schedule\_uid của mainteinence plan.

**Giải thích:** Hai sp trên sẽ chạy dựa theo mainteinence plan tương ứng, đường dẫn lưu file back up là đường dẫn khi tạo mainteinence plan.

Phần quan trọng nhất của 2 sp trên là việc đưa vào các tham số để đặt lịch backup, sau đây là phần giải thích các argument quan trọng:



@enabel: Cho biết trạng thái hiện tại của lịch trình. Mặc định là 1 (bật).

[ @freq\_type = ] Một giá trị cho biết khi nào một công việc sẽ được thực hiện. freq\_type là int, với giá trị mặc định là 0 và có thể là một trong những giá trị này.

**Value Description**

**1** Once

**4** Daily

**8** Weekly

**16** Monthly

**32** Monthly, relative to *freq\_interval*

[ @freg\_interval = ] ngày mà công việc được thực hiện. freq\_interval là int, với giá trị mặc định là 1 và phụ thuộc vào giá trị của freq\_type.

**Value of *freq\_type* Effect on *freq\_interval* 1** (once) *freq\_interval* is unused.

**4** (daily) Every *freq\_interval* days.

**8** (weekly) *freq\_interval* is one or more of the following (combined with an OR logical operator):

1. = Sunday
2. = Monday **4** = Tuesday

**8** = Wednesday

**16** = Thursday

**32** = Friday

**64** = Saturday

**16** (monthly) On the *freq\_interval* day of the month.

**32** (monthly *freq\_interval* is one of the following:

relative)

1. = Sunday
2. = Monday **3** = Tuesday
3. = Wednesday
4. = Thursday
5. = Friday
6. = Saturday
7. = Day
8. = Weekday
9. = Weekend day

[ @freq\_subday\_type = ] Chỉ định các đơn vị cho freq\_subday\_interval.

freq\_subday\_type là int, với giá trị mặc định là 0 và có thể là một trong những giá trị này.

|  |  |
| --- | --- |
| **Value** | **Description (unit)** |
| **0x1** | At the specified time |
| **0x2** | Seconds |
| **0x4** | Minutes |
| **0x8** | Hours |

[ @freq\_subday\_interval = ] Số khoảng thời gian freq\_subday\_type xảy ra giữa mỗi lần thực hiện một công việc. freq\_subday\_interval là int, với giá trị mặc định là 0.

Lưu ý: Khoảng thời gian phải dài hơn 10 giây. freq\_subday\_interval bị bỏ qua trong những trường hợp freq\_subday\_type bằng 1.

[ @freq\_relative\_interval = ] Công việc xuất hiện freq\_interval trong mỗi tháng, nếu freq\_interval là 32 (tương đối hàng tháng). freq\_relative\_interval là int, với giá trị mặc định là 0 và có thể là một trong những giá trị này. freq\_relative\_interval bị bỏ qua trong những trường hợp freq\_type không bằng 32.

**Value Description (unit)**

1. First
2. Second

**4** Third

**8** Fourth

**16** Last

[ @freq\_recurrence\_factor = ] Số tuần hoặc tháng giữa thời gian thực hiện công việc theo lịch trình. freq\_recency\_factor chỉ được sử dụng nếu freq\_type là 8, 16 hoặc 32. freq\_recency\_factor là int, với giá trị mặc định là 0.

[ @active\_start\_date = ] Ngày bắt đầu thực hiện một công việc. active\_start\_date là int, với giá trị mặc định là NULL, cho biết ngày hôm nay. Ngày được định dạng là YYYYMMDD. Nếu active\_start\_date không phải là NULL, thì ngày đó phải lớn hơn hoặc bằng 19900101.

[ @active\_end\_date = ] Ngày mà việc thực hiện một công việc có thể dừng lại.

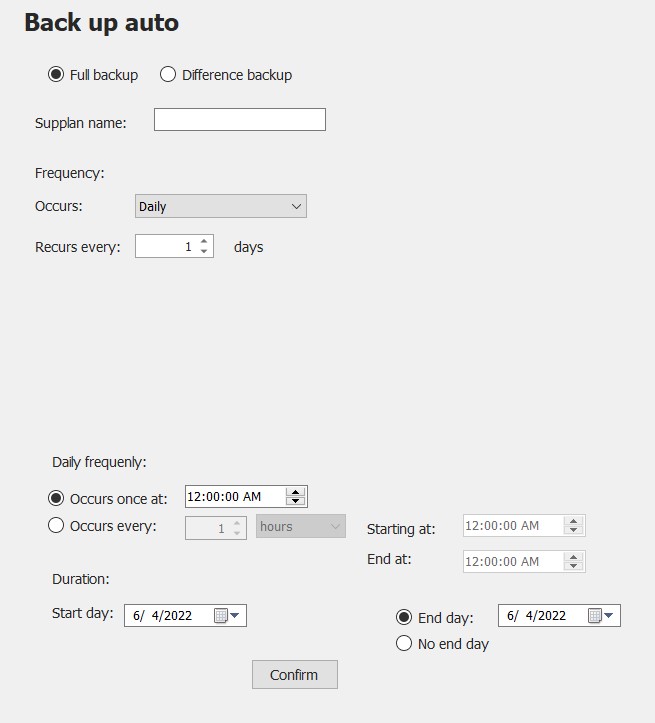
active\_end\_date là int, với giá trị mặc định là 99991231, cho biết ngày 31 tháng 12 năm 9999. Được định dạng là YYYYMMDD.

[ @active\_start\_time = ] Thời gian vào bất kỳ ngày nào từ active\_start\_date đến active\_end\_date để bắt đầu thực hiện một công việc. active\_start\_time là int, với

giá trị mặc định là 000000, cho biết 12:00:00 A.M. trên đồng hồ 24 giờ và phải được nhập bằng biểu mẫu HHMMSS

[ @active\_end\_time = ] Khoảng thời gian vào bất kỳ ngày nào giữa active\_start\_date và active\_end\_date để kết thúc việc thực thi một công việc. active\_end\_time là int, với giá trị mặc định là 235959, cho biết 11:59:59 CH. trên đồng hồ 24 giờ và phải được nhập bằng biểu mẫu HHMMSS.

**Giao diện chính:**



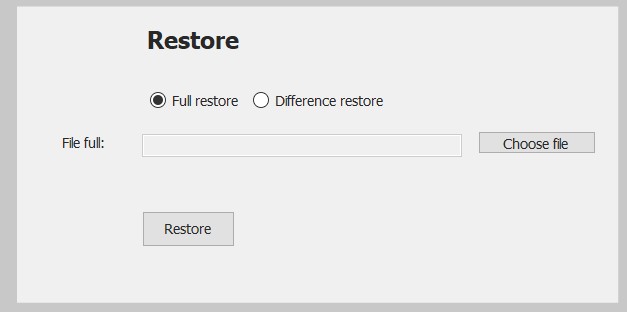
**3.Restore:**

**1.1.Full database restore:**

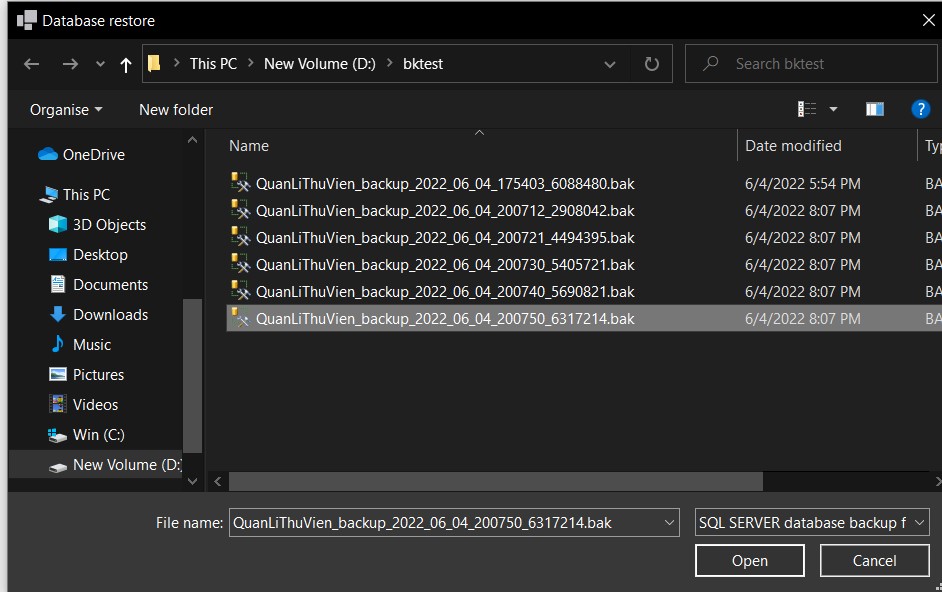
**Yêu cầu:** có file full database backup

Các bước tạo full database restore:

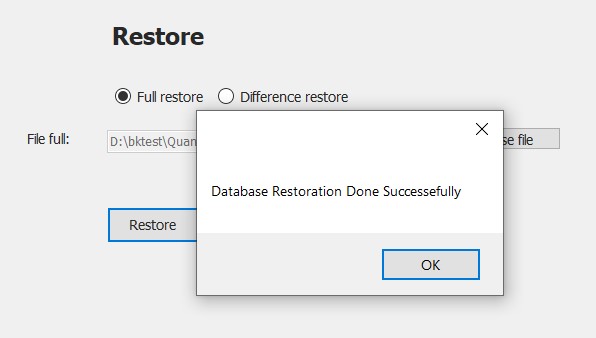
Bước 1: Ấn chọn file



Bước 2:Chọn file full backup



Bước 3:Ấn restore.

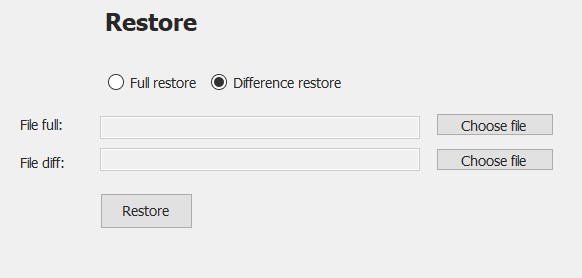


**1.2.Difference database restore:**

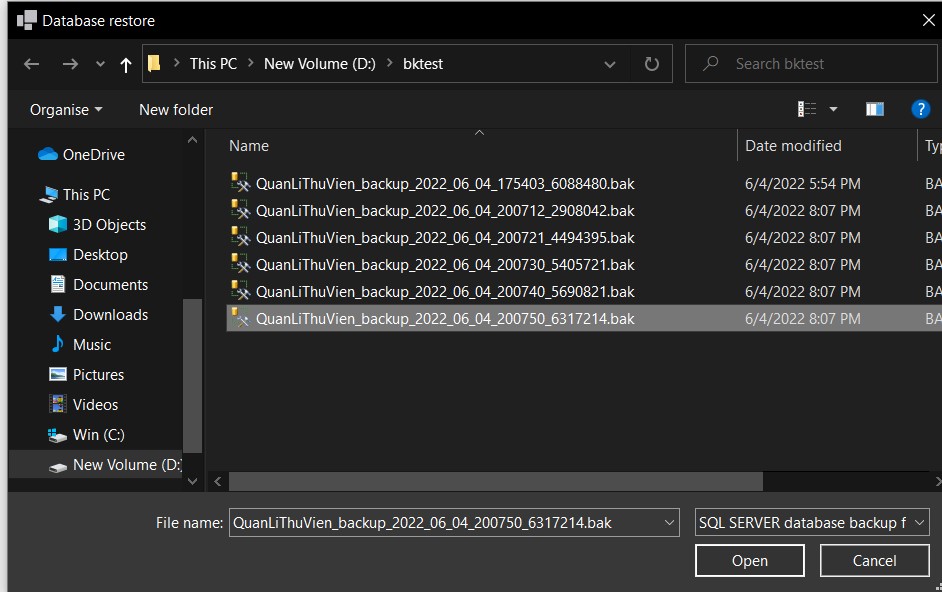
Yêu cầu: Có file difference backup và file full backup tương ứng.

Các bước tạo difference restore :

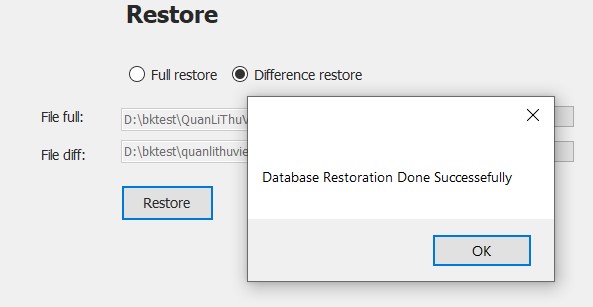
Bước 1: Chọn difference restore.



Bước 2:Chọn file full backup và difference back up cần dùng để restore database.

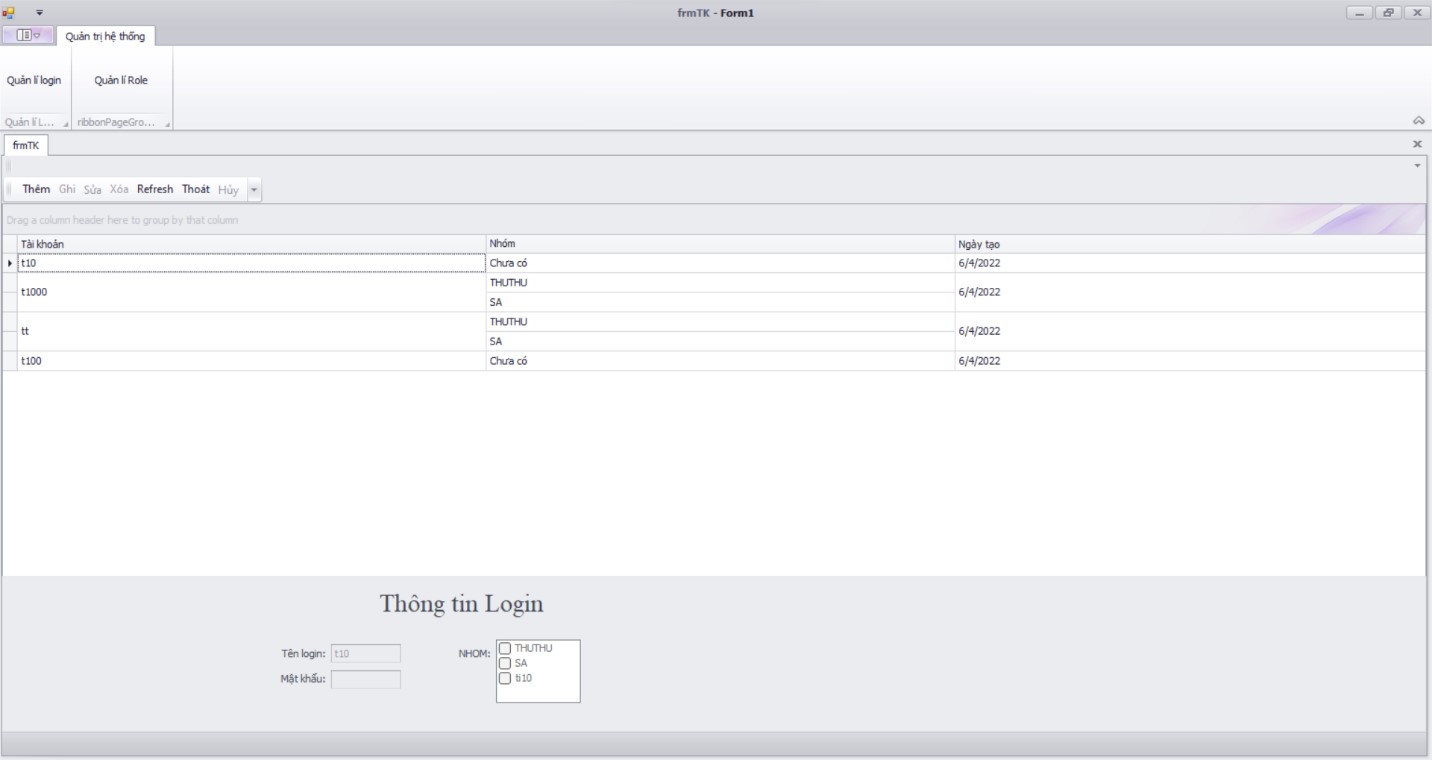


Bước 3:Ấn restore.

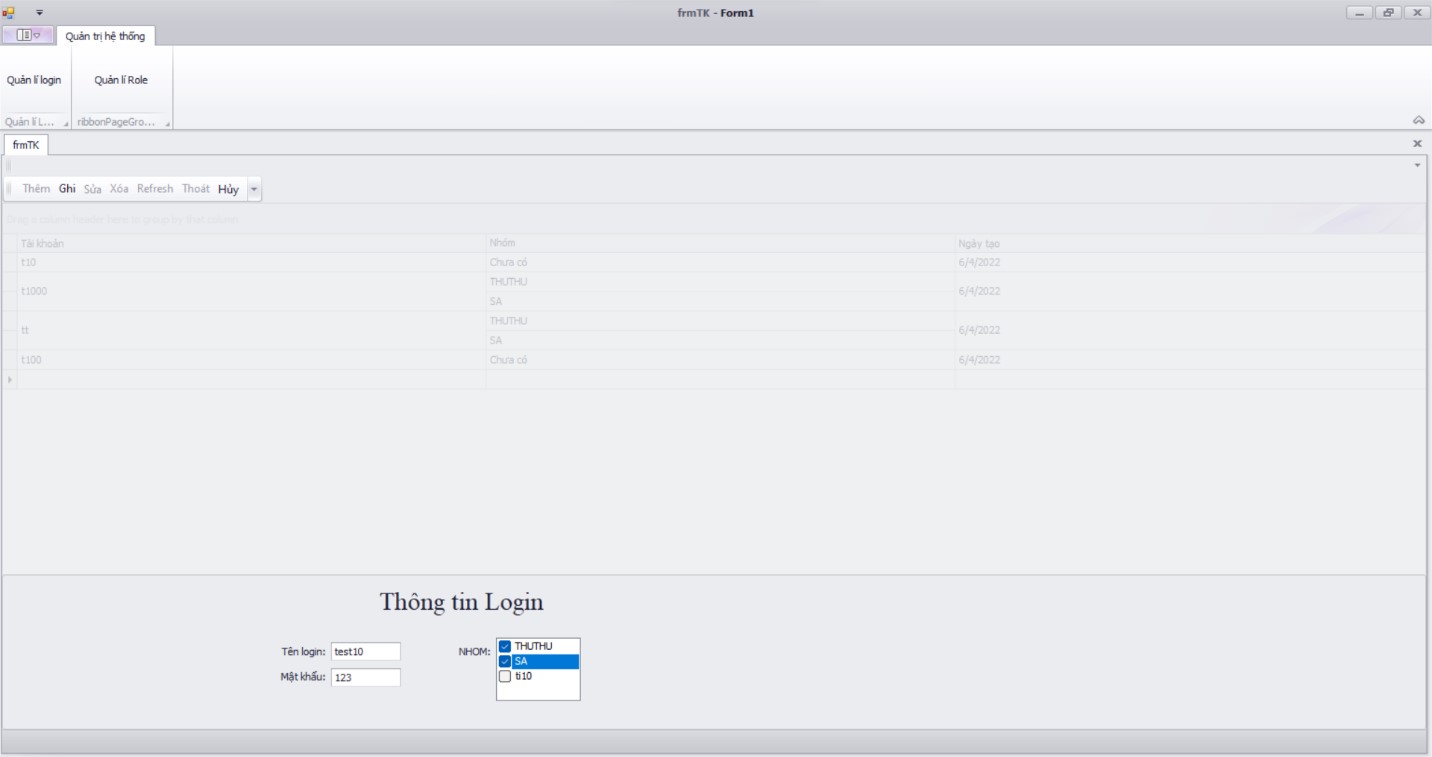


## II. Phân quyền, quản lí login

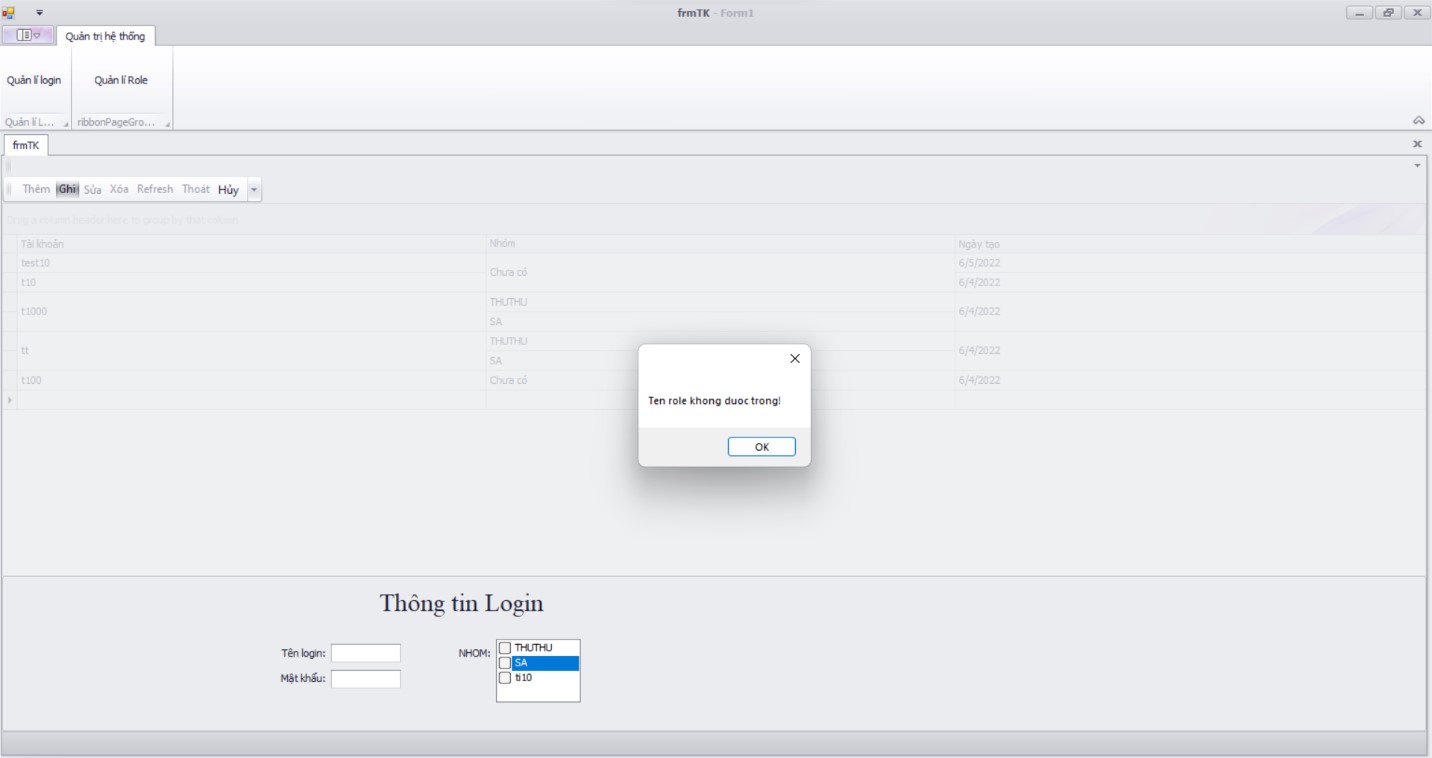
**1.1. Quản lí login**



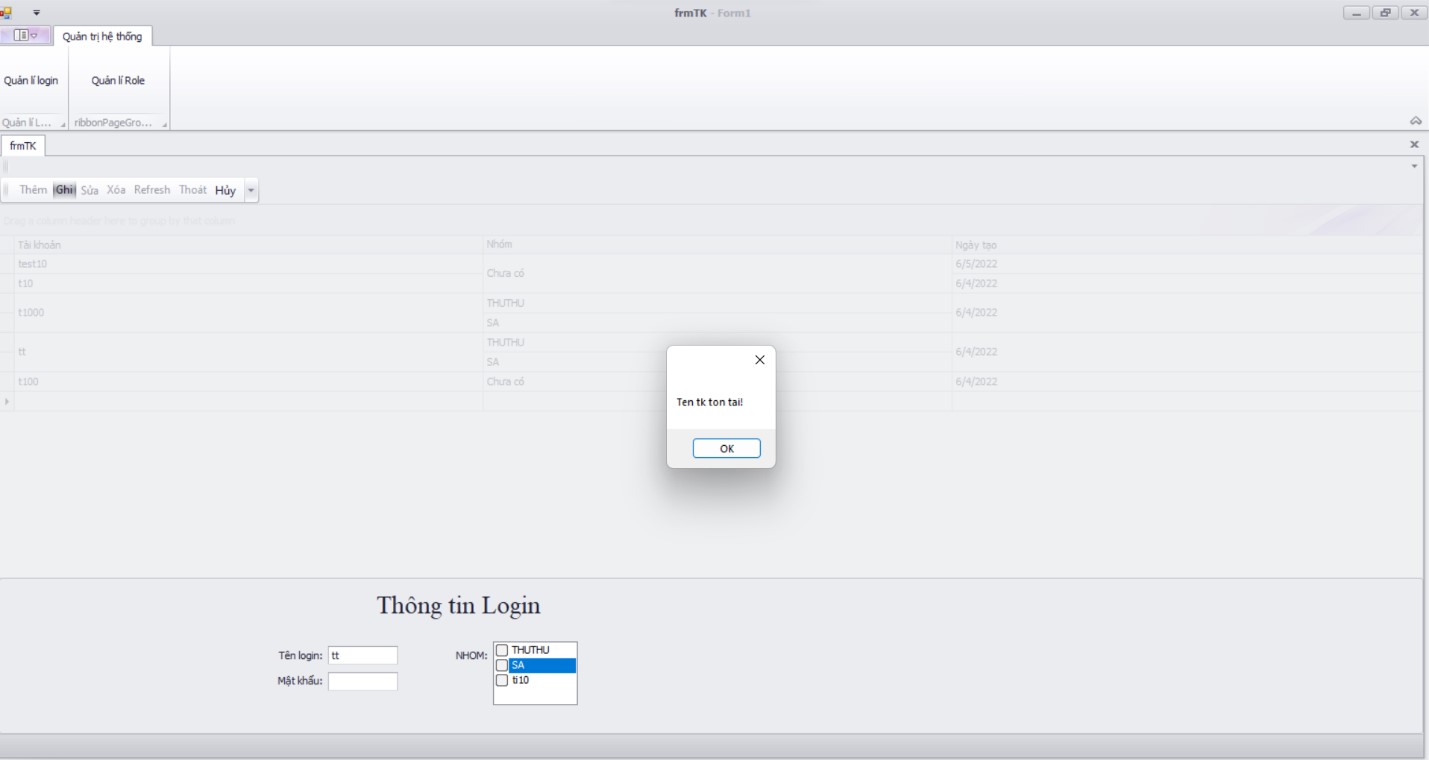
*1.1.a. Màn hình giao diện quản lí login*



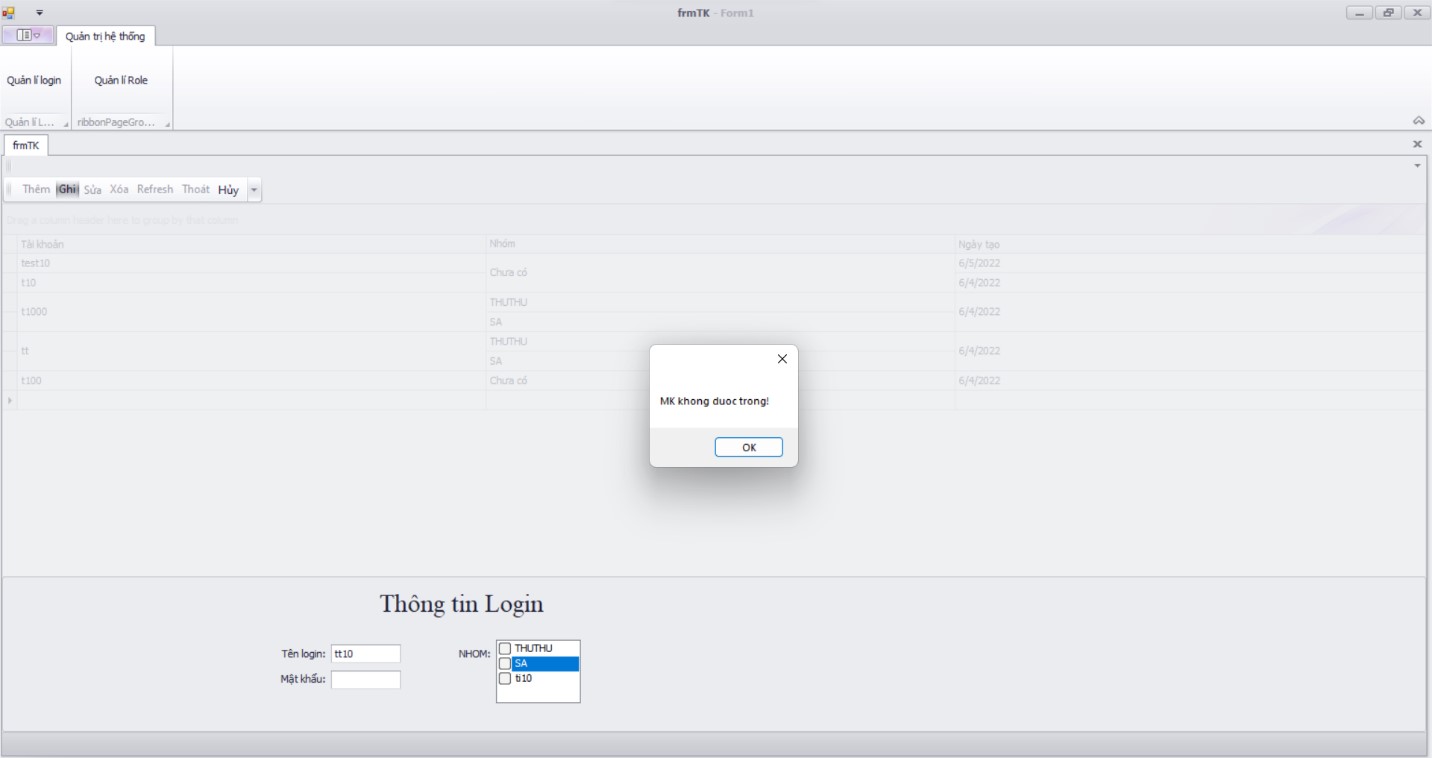
*1.1.b. Màn hình thêm login*



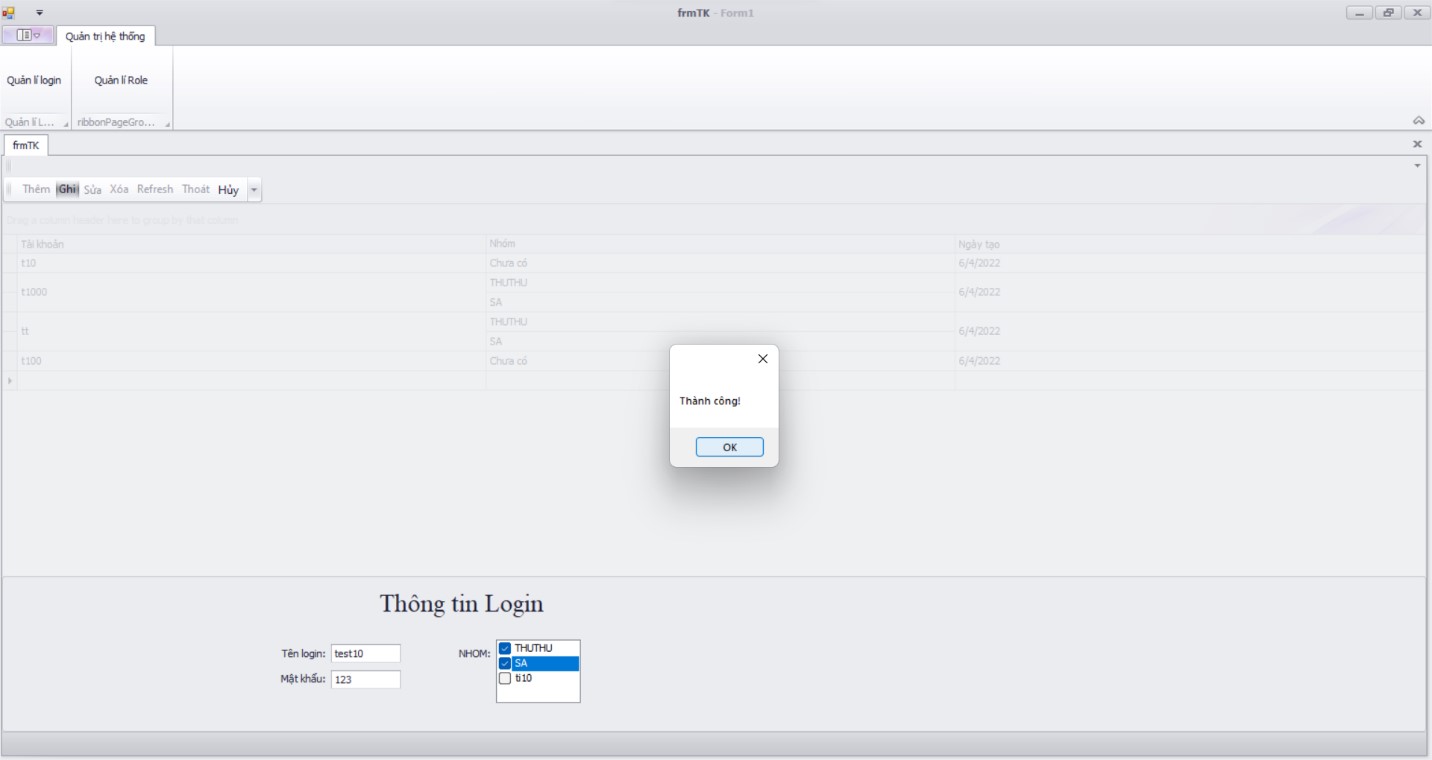
*1.1.c. Vi phạm tên login trống*



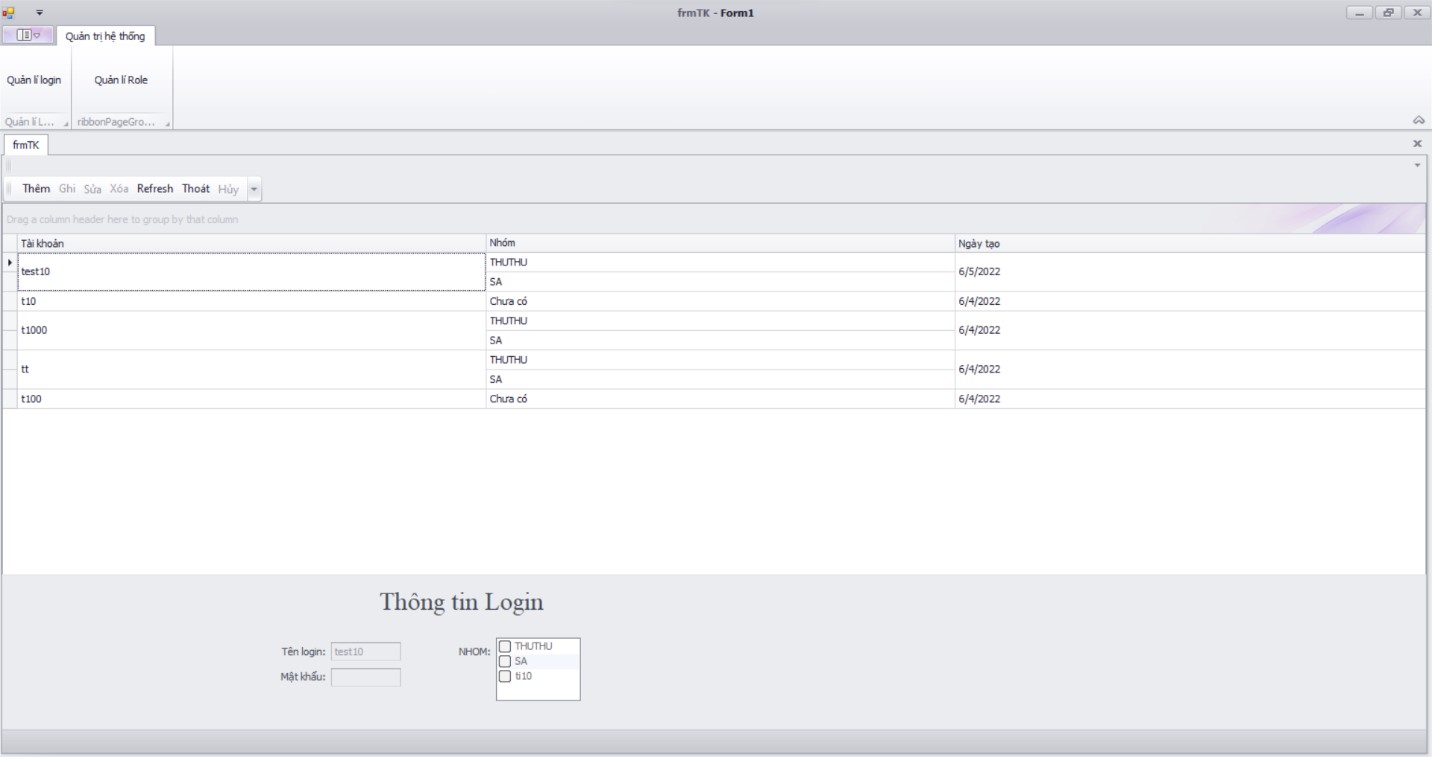
*1.1.d. Vi phạm tên login trùng*



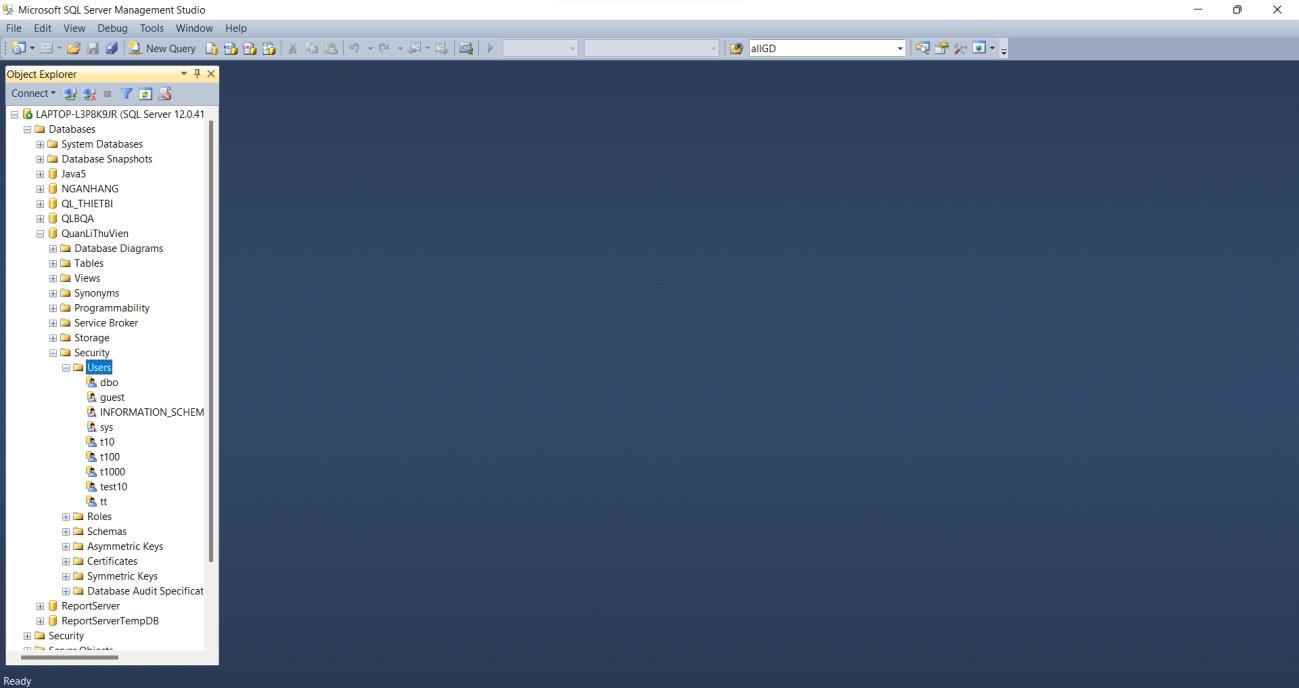
*1.1.e. Vi phạm mật khẩu trống*



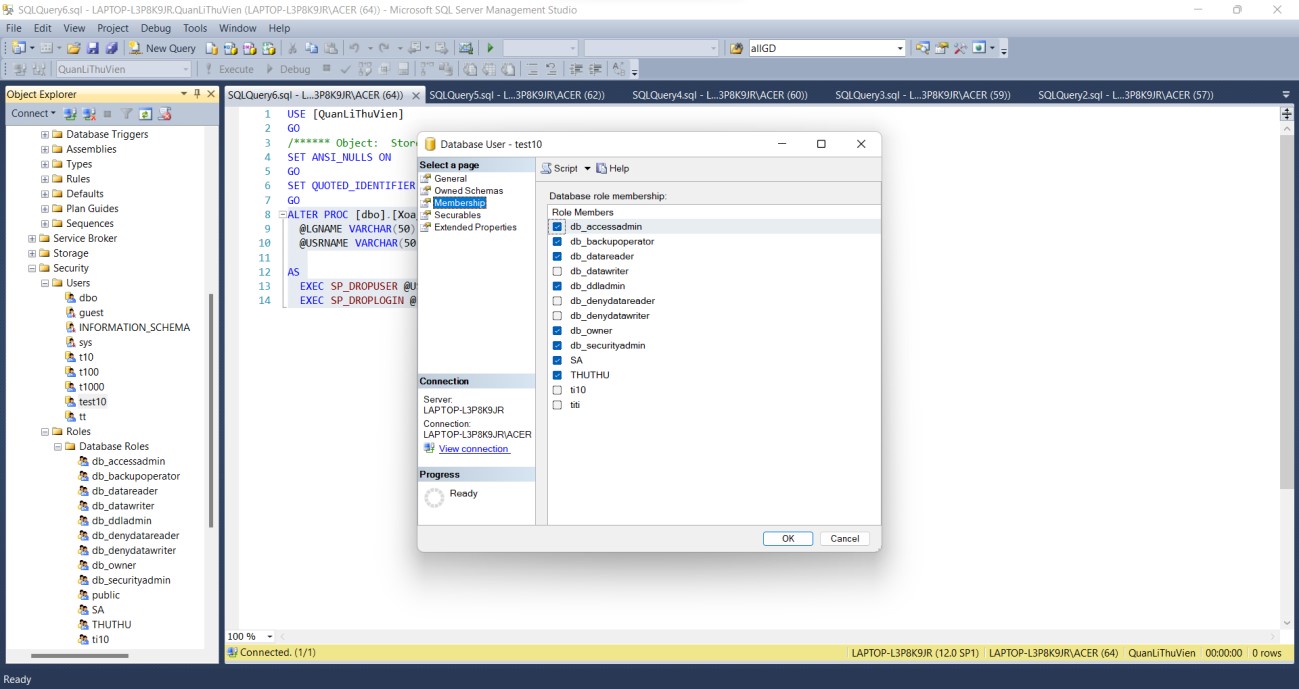
*1.1.f. Màn hình thêm login thành công*



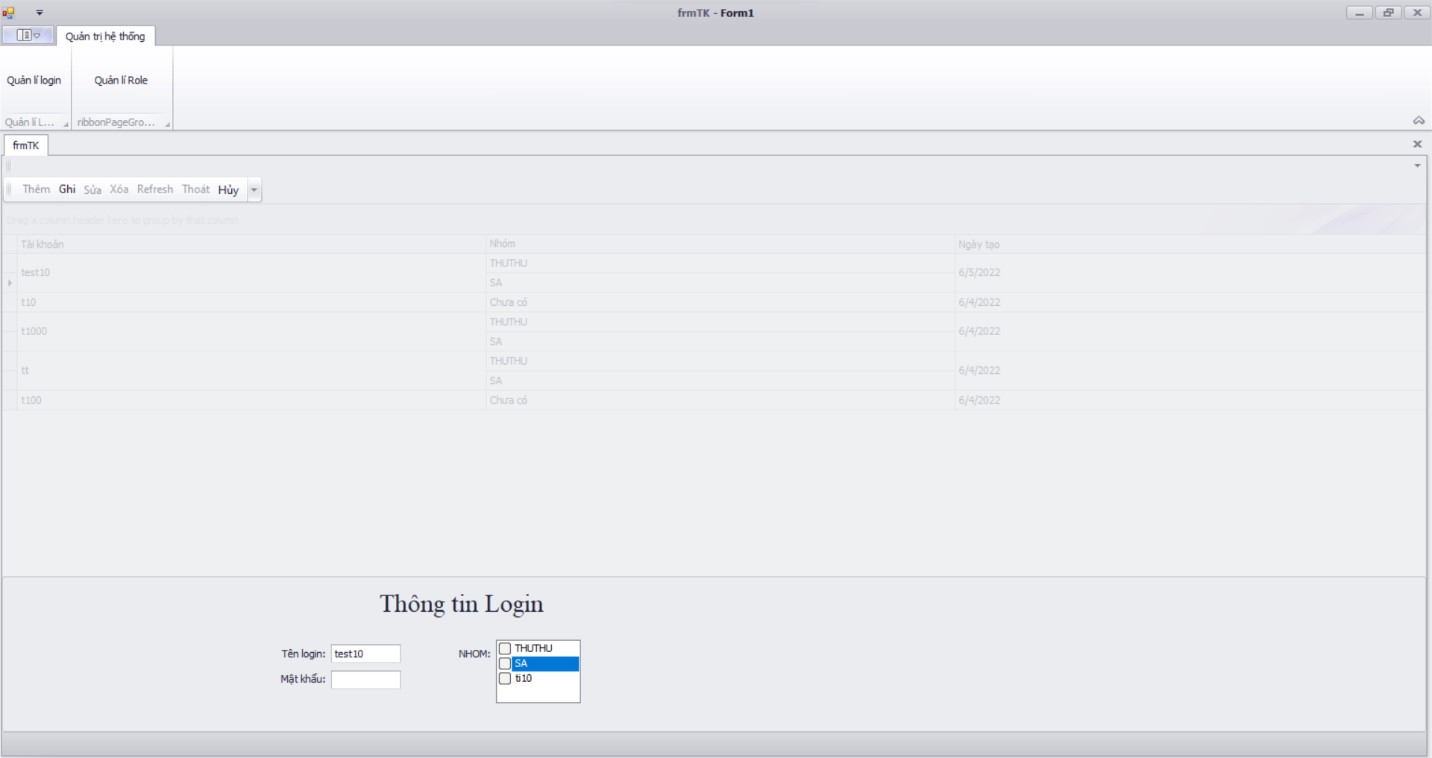
*1.1.g. Login đã được thêm vào*



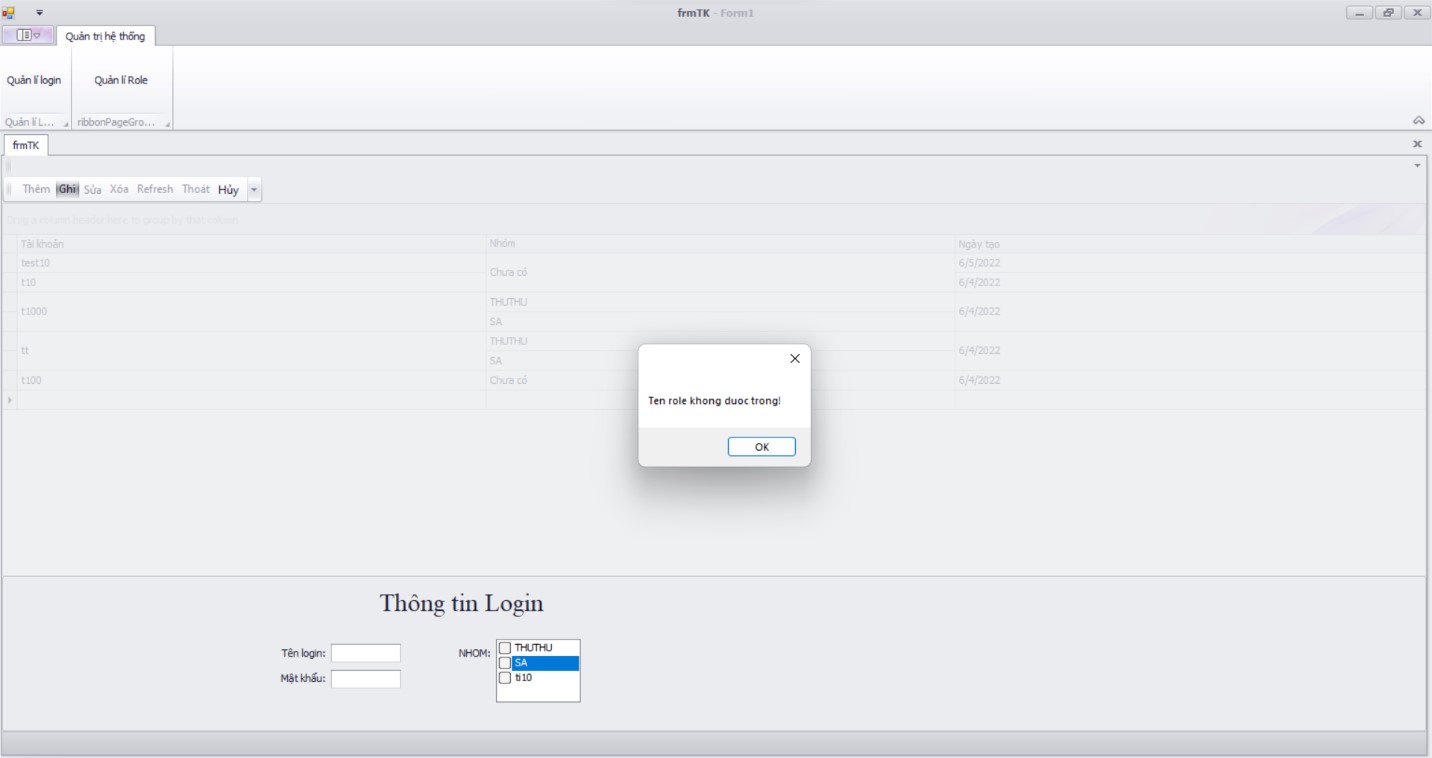
*1.1.g.1. Login đã xuất hiện trong database*



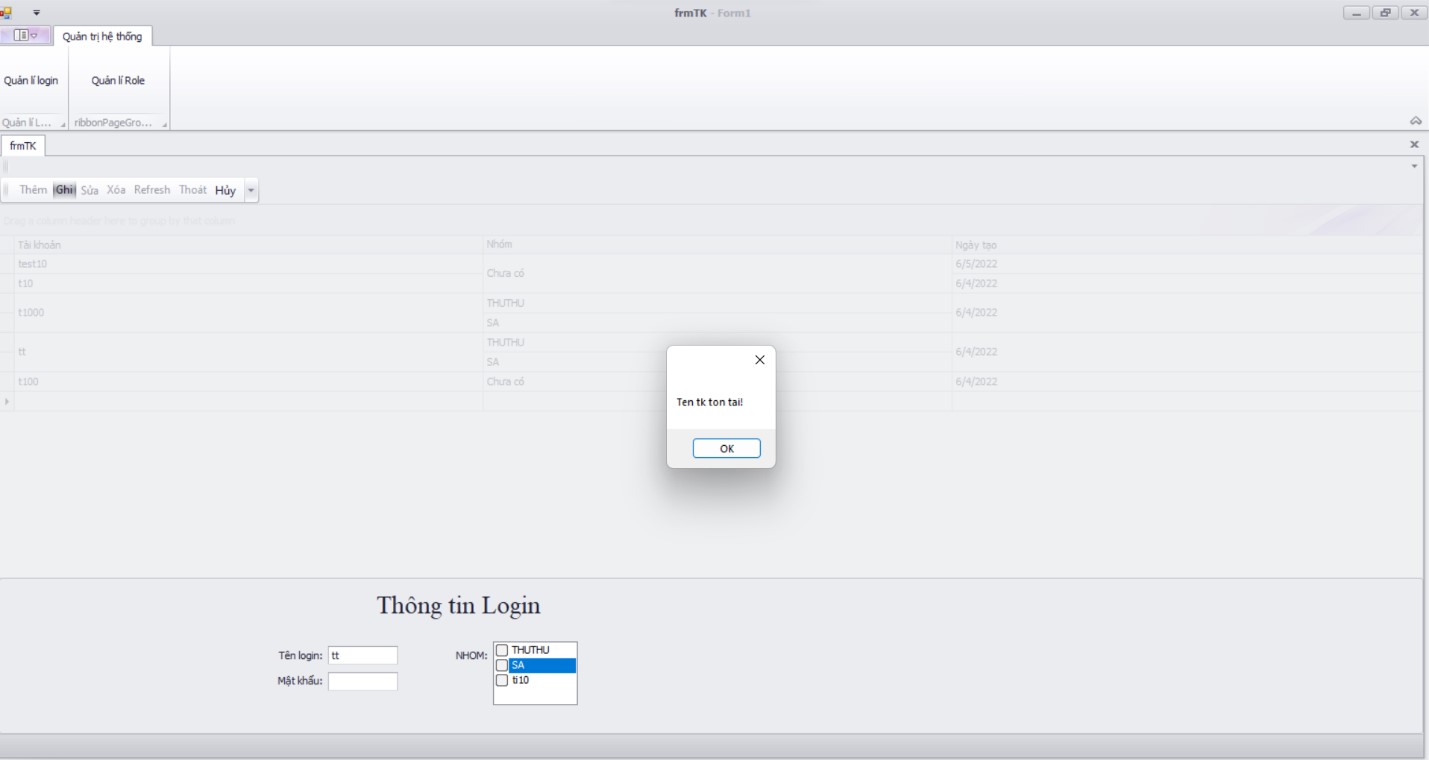
*1.1.g.2. Login đã xuất hiện trong nhóm*



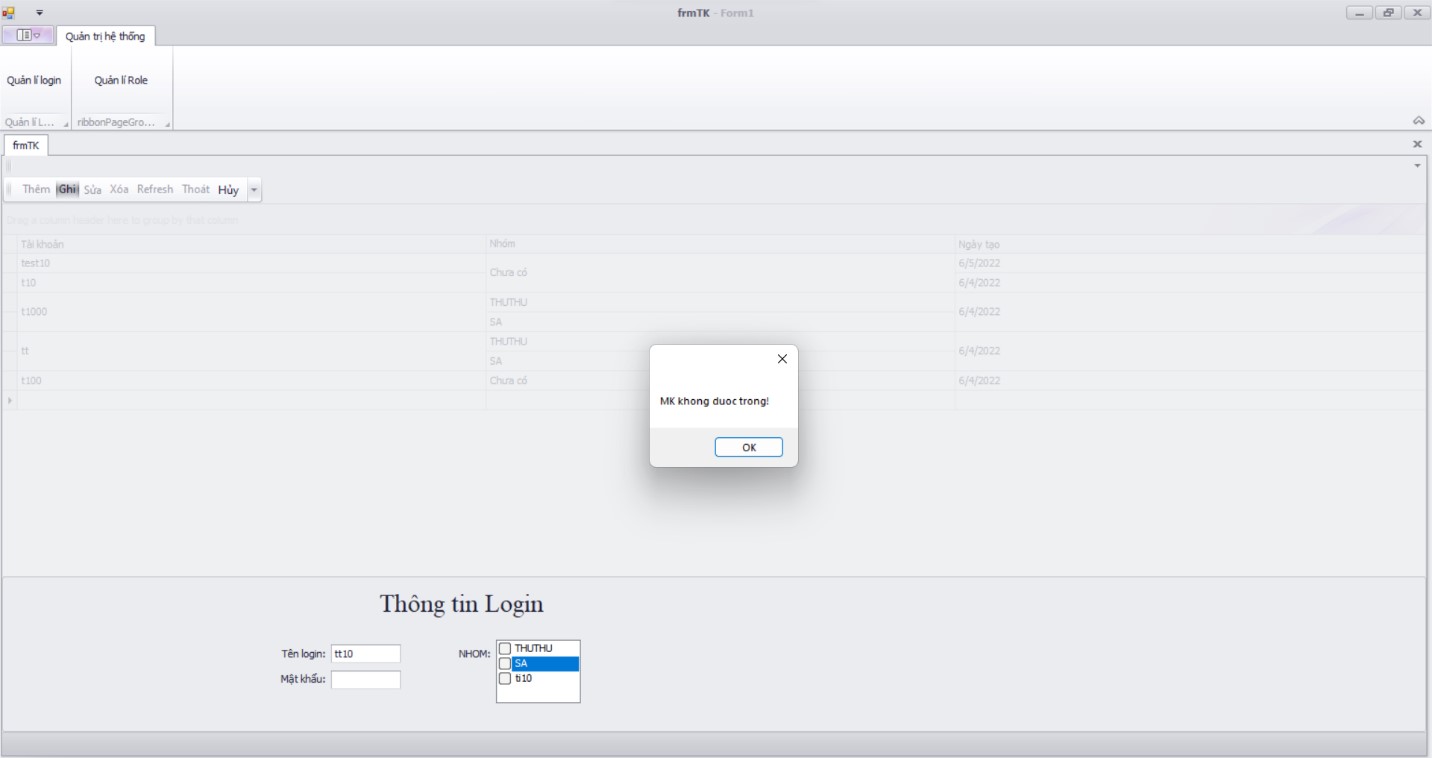
*1.1.h. Sửa login*



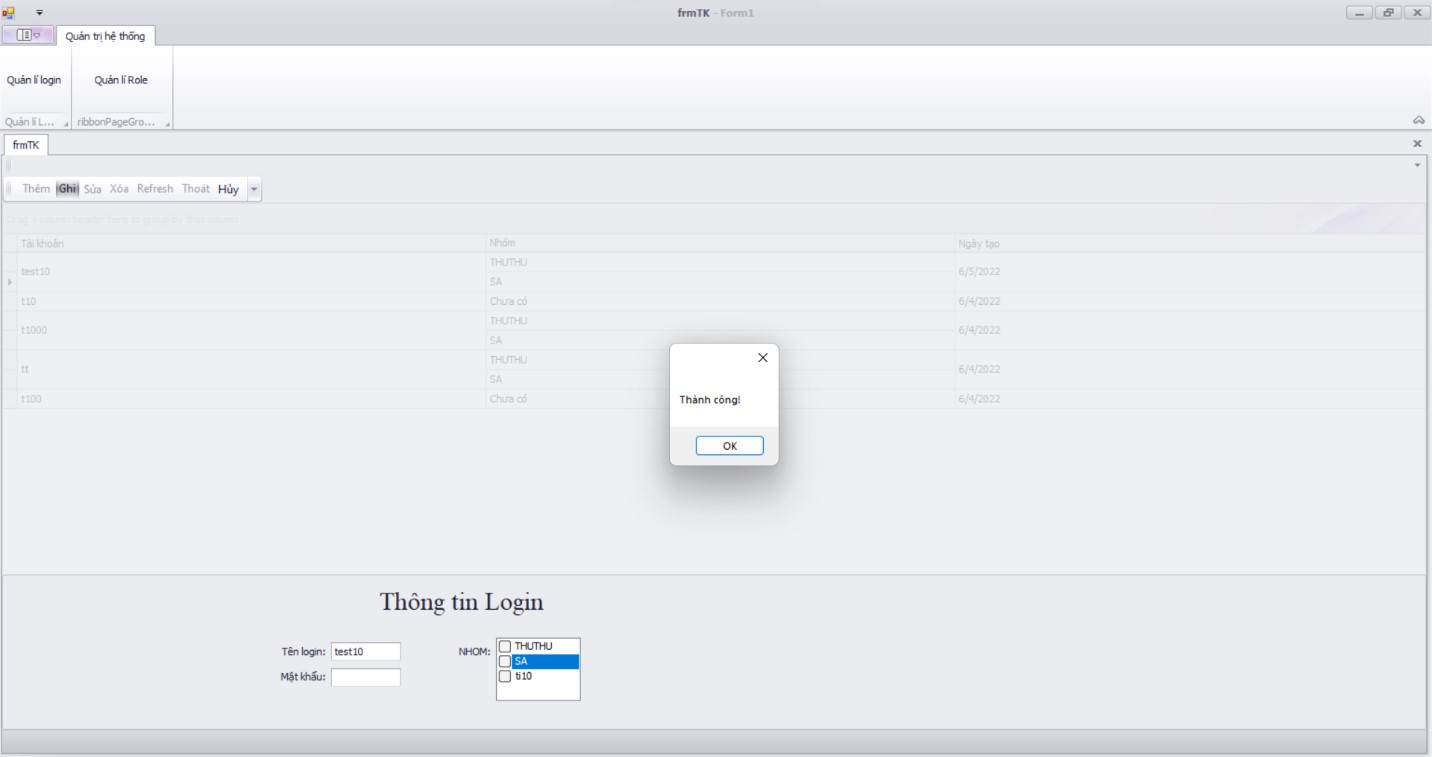
*1.1.i. Vi phạm tên login trống*



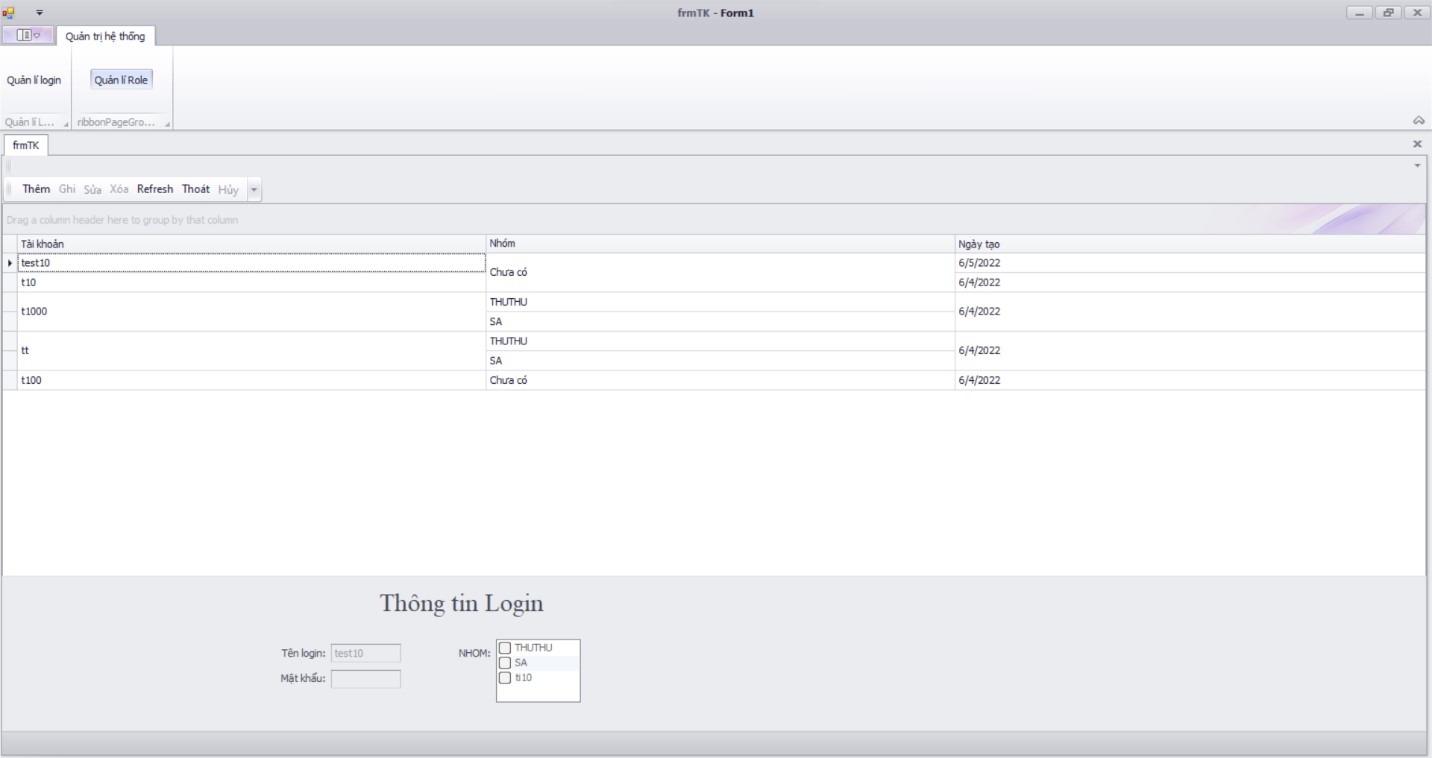
*1.1.j. Vi phạm tên login trùng*



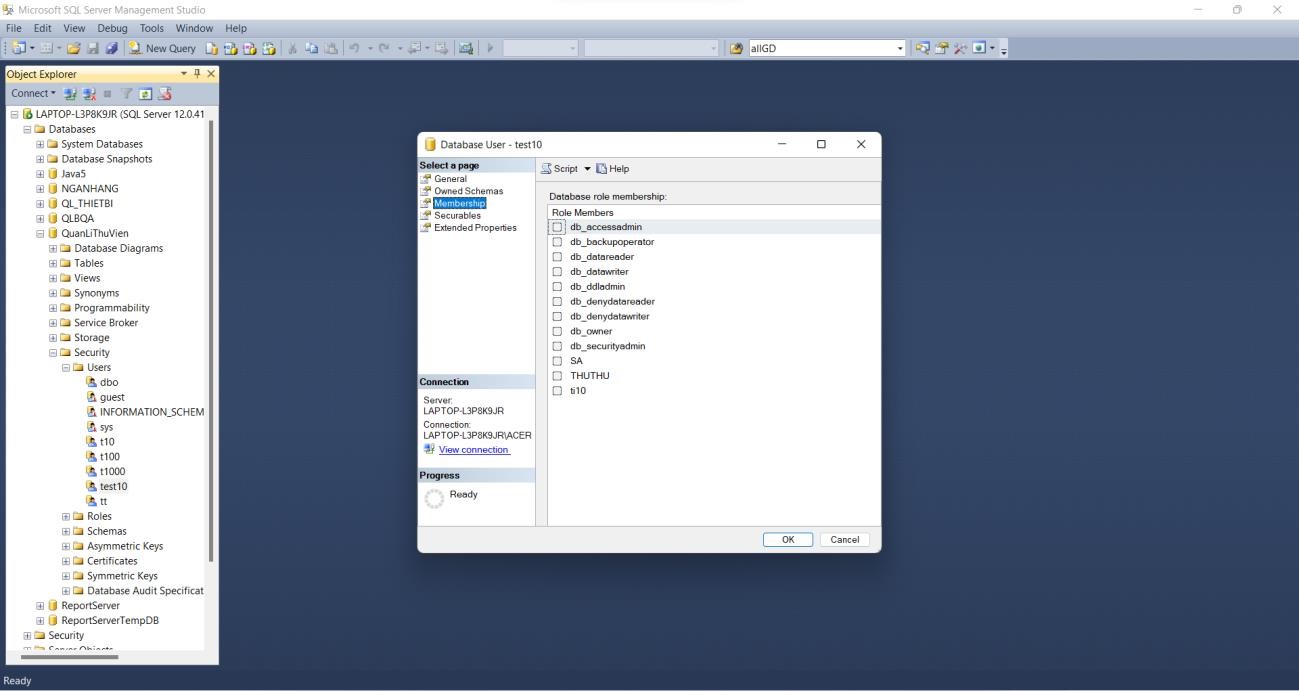
*1.1.k. Vi phạm mật khẩu trống*



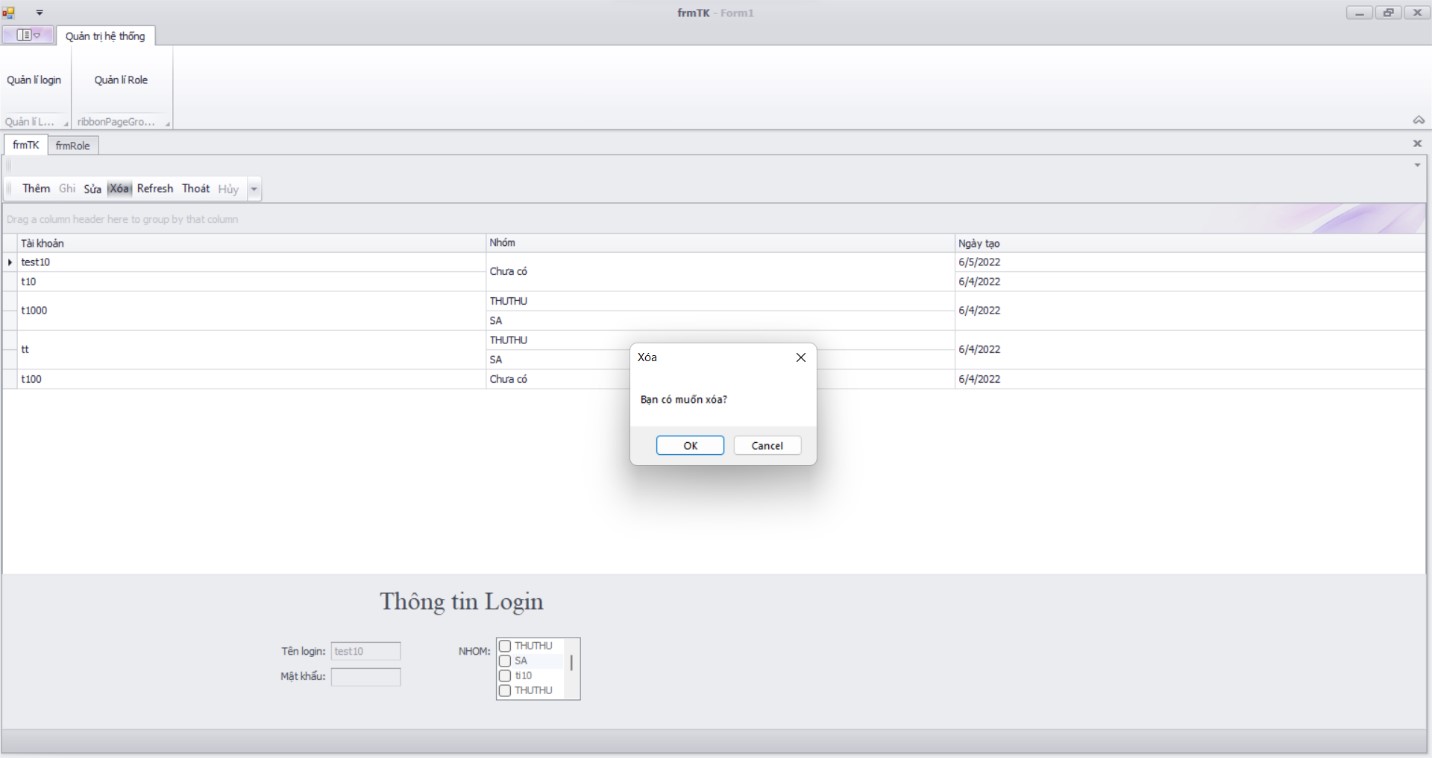
*1.1.l. Sửa thành công*



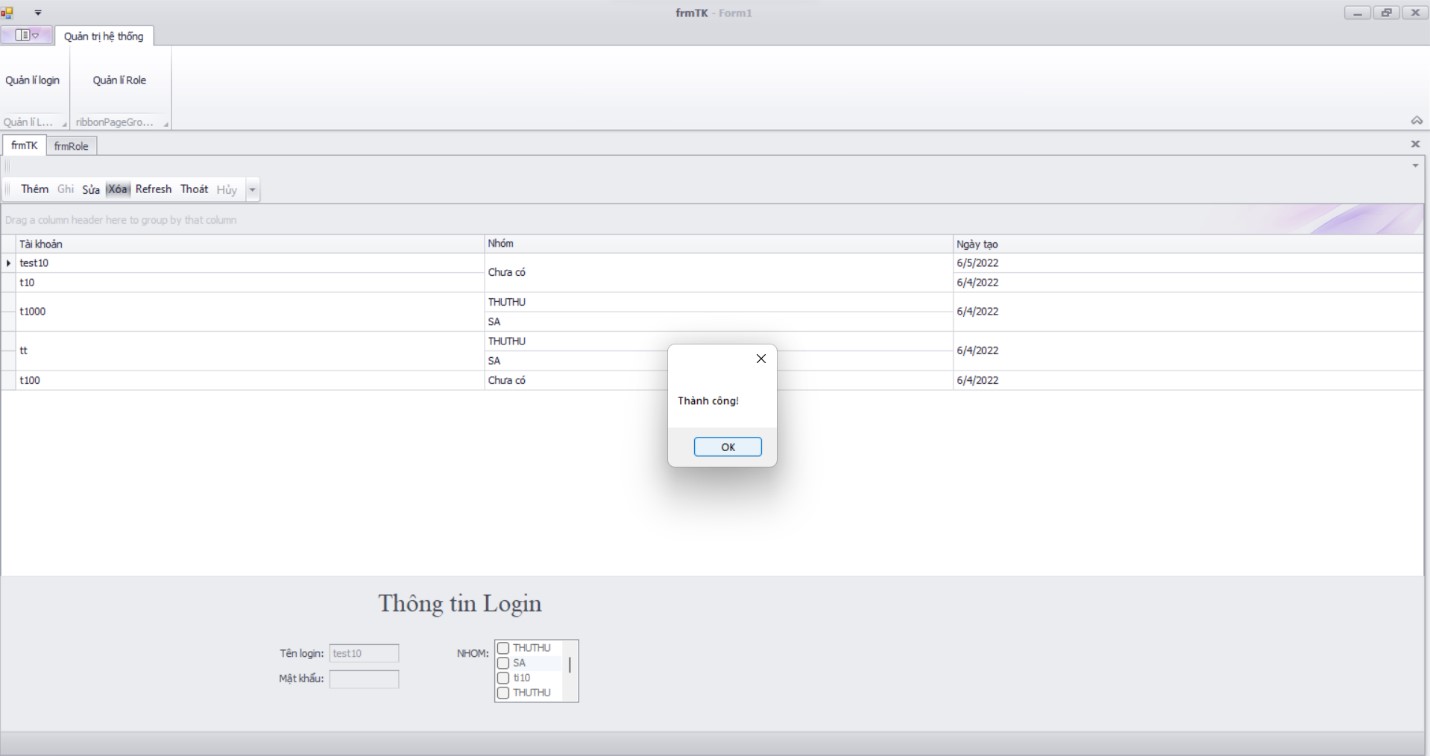
*1.1.m. Login đã được sửa lại thành chưa nhóm*



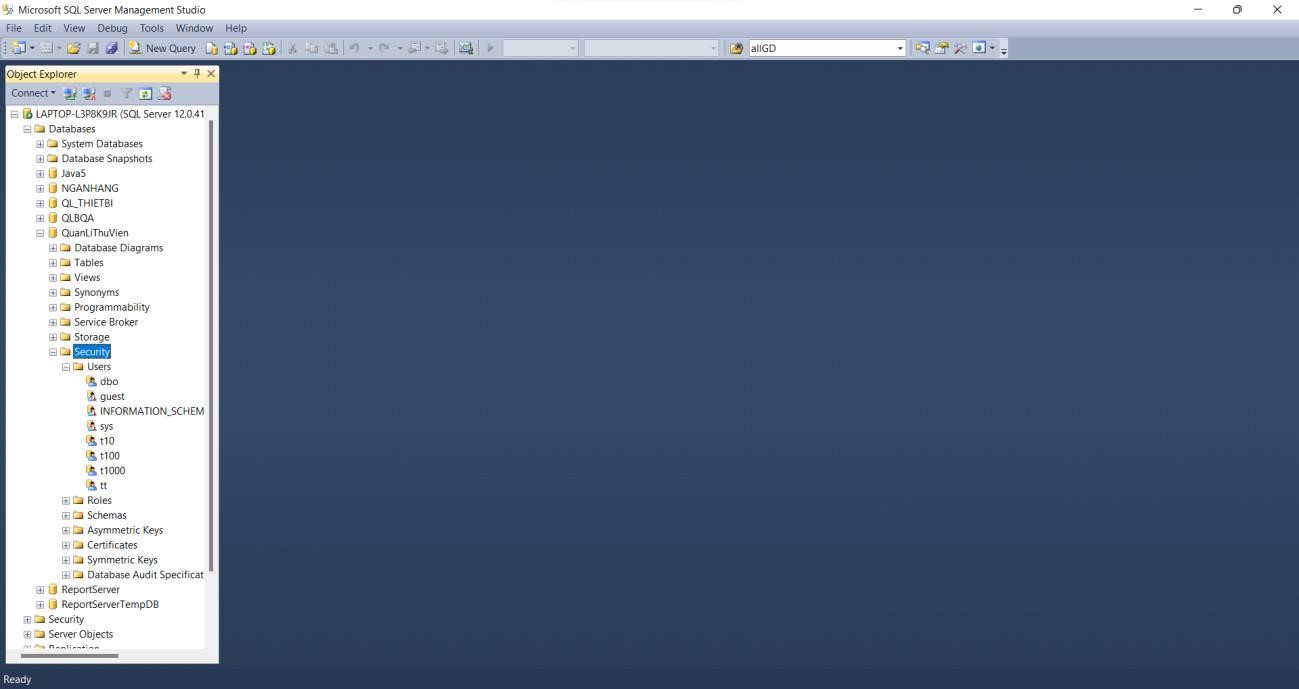
*1.1.n. Thay đổi trong database*



*1.1.o. Xóa login*

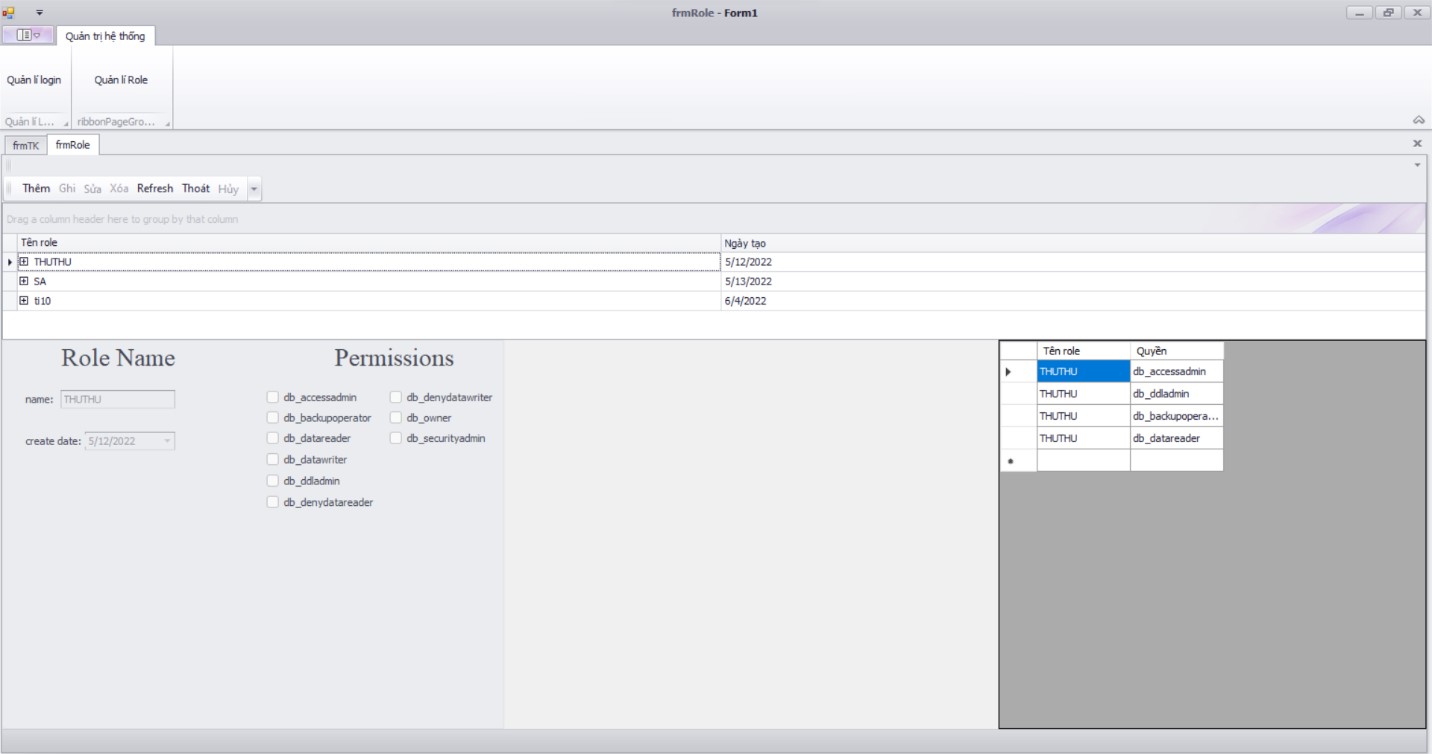


*1.1.p. Xóa thành công*

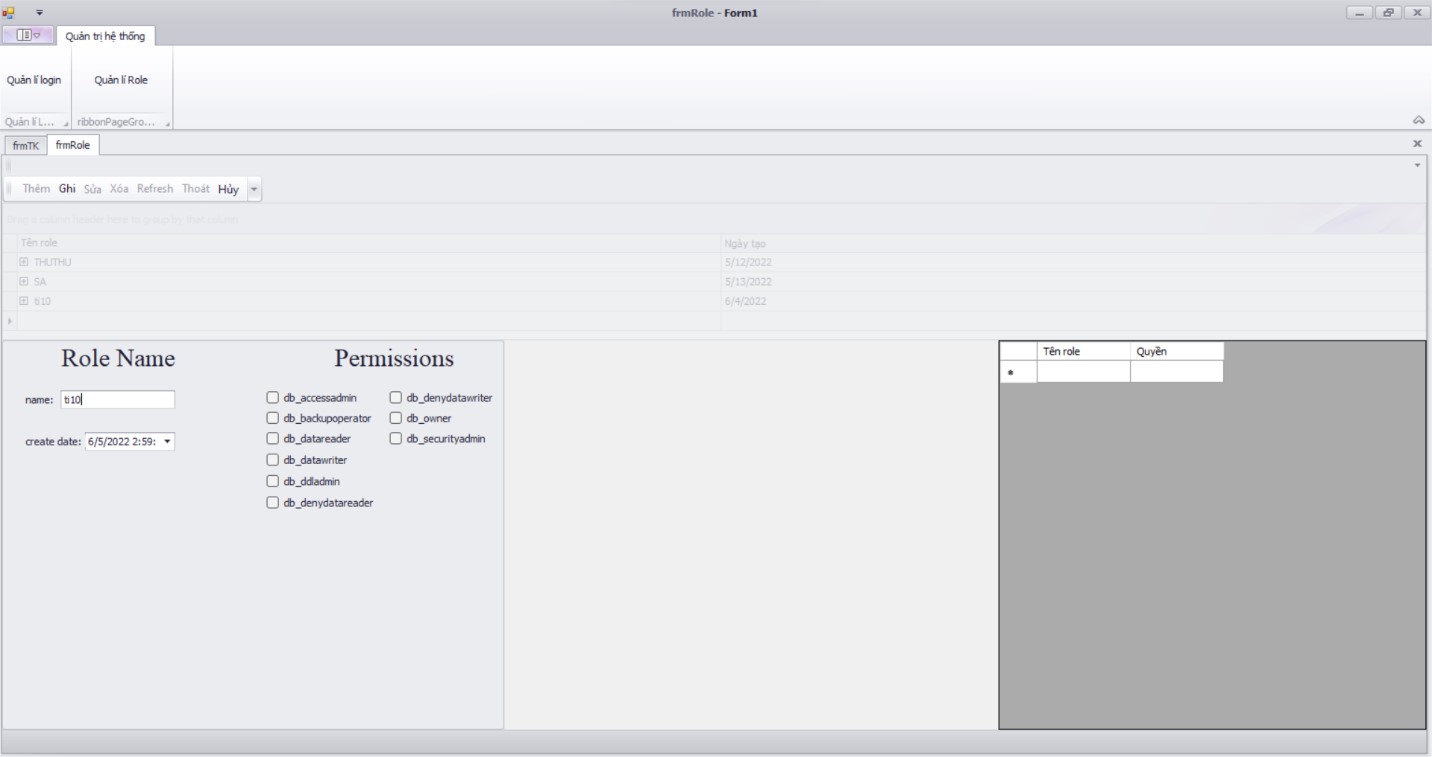


*1.1.p.1. Login đã được xóa trong database*

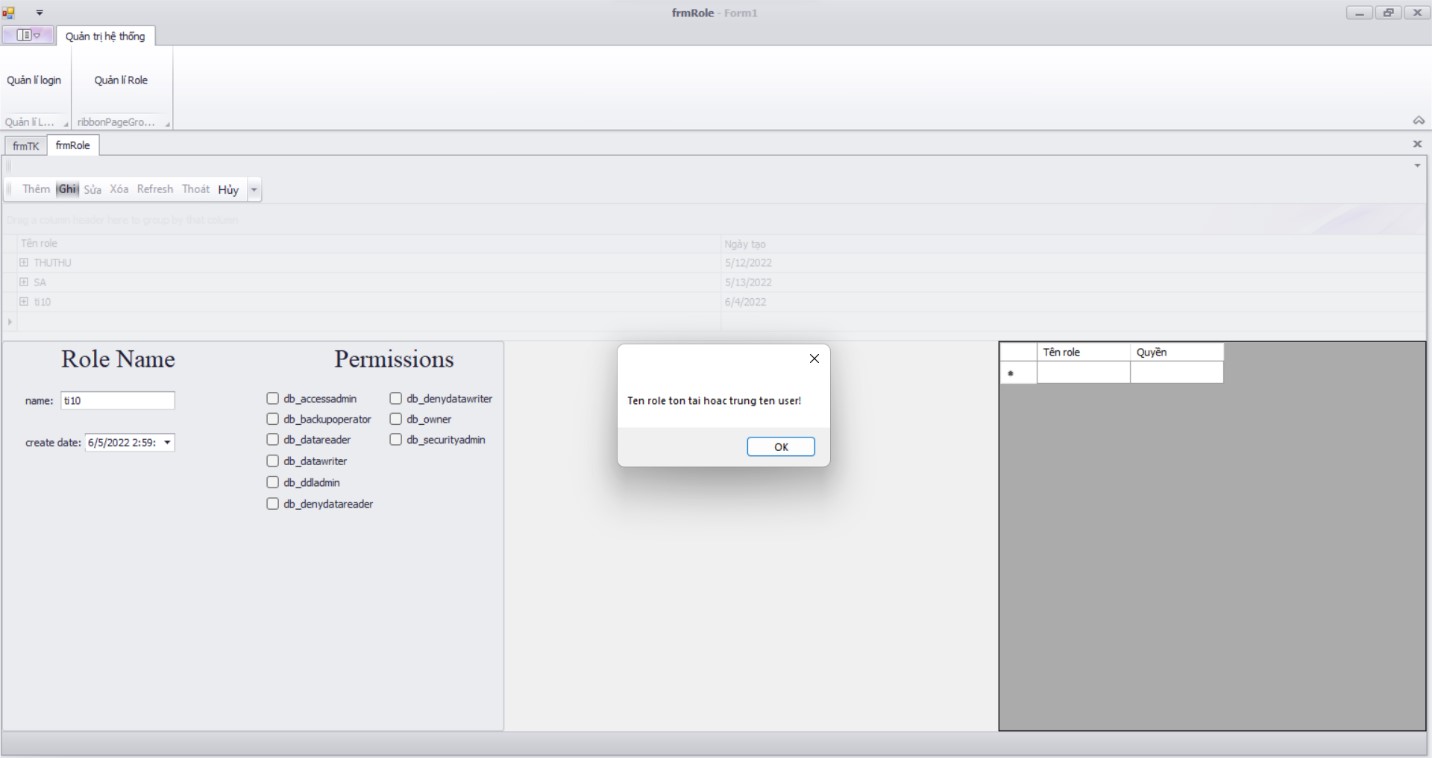
**1.2. Quản lí Role**



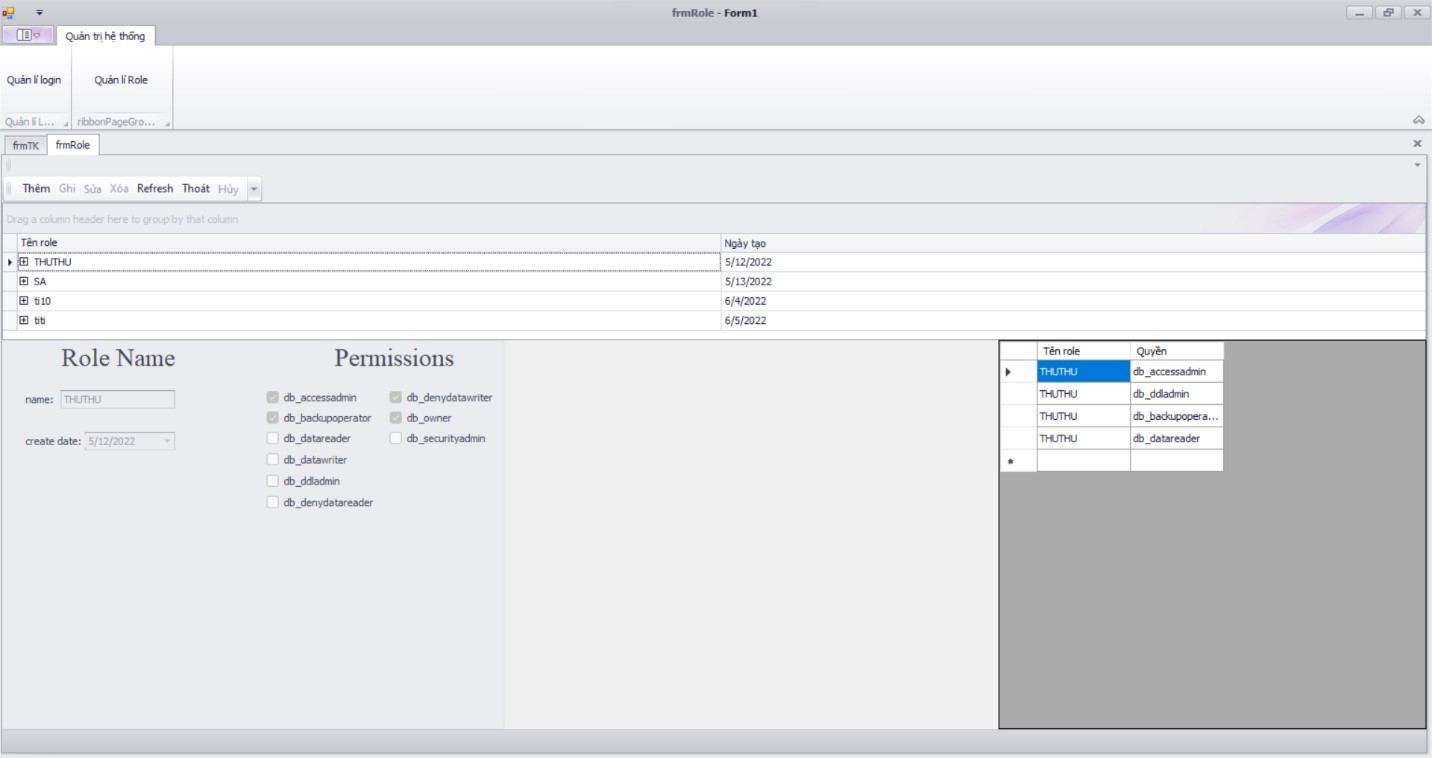
*1.2.a. Màn hình quản lí role*



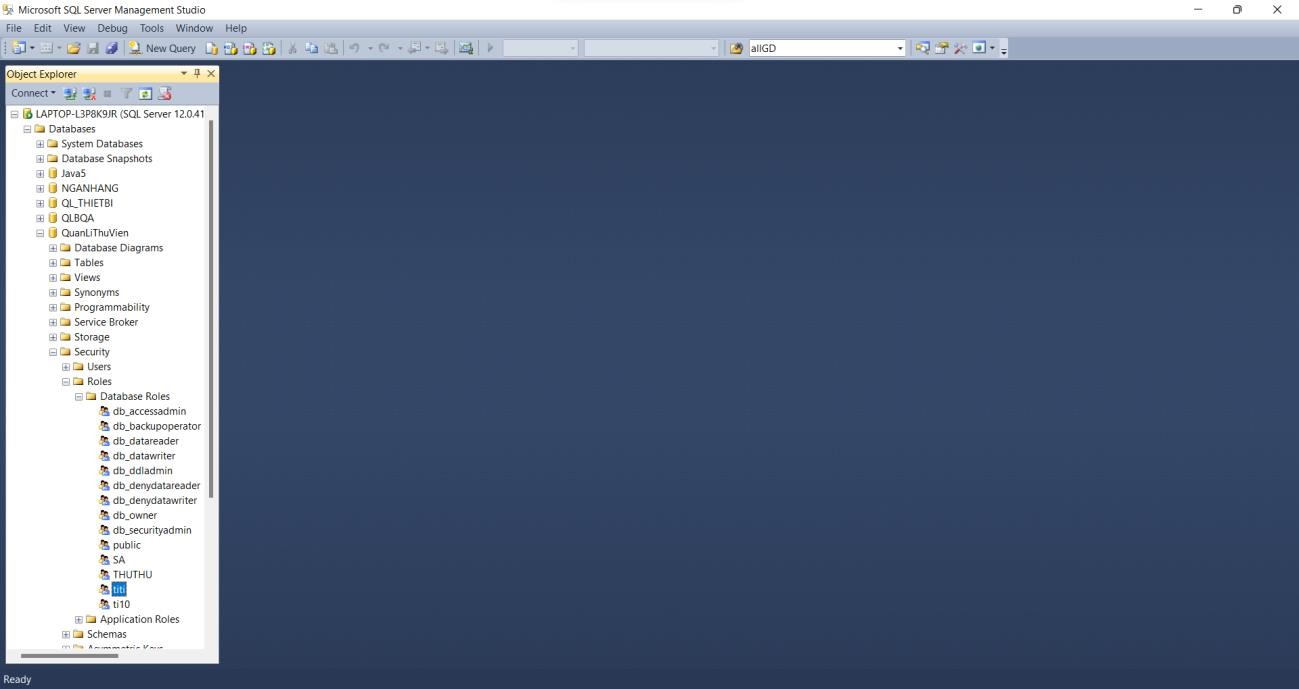
*1.2.b. Thêm role*



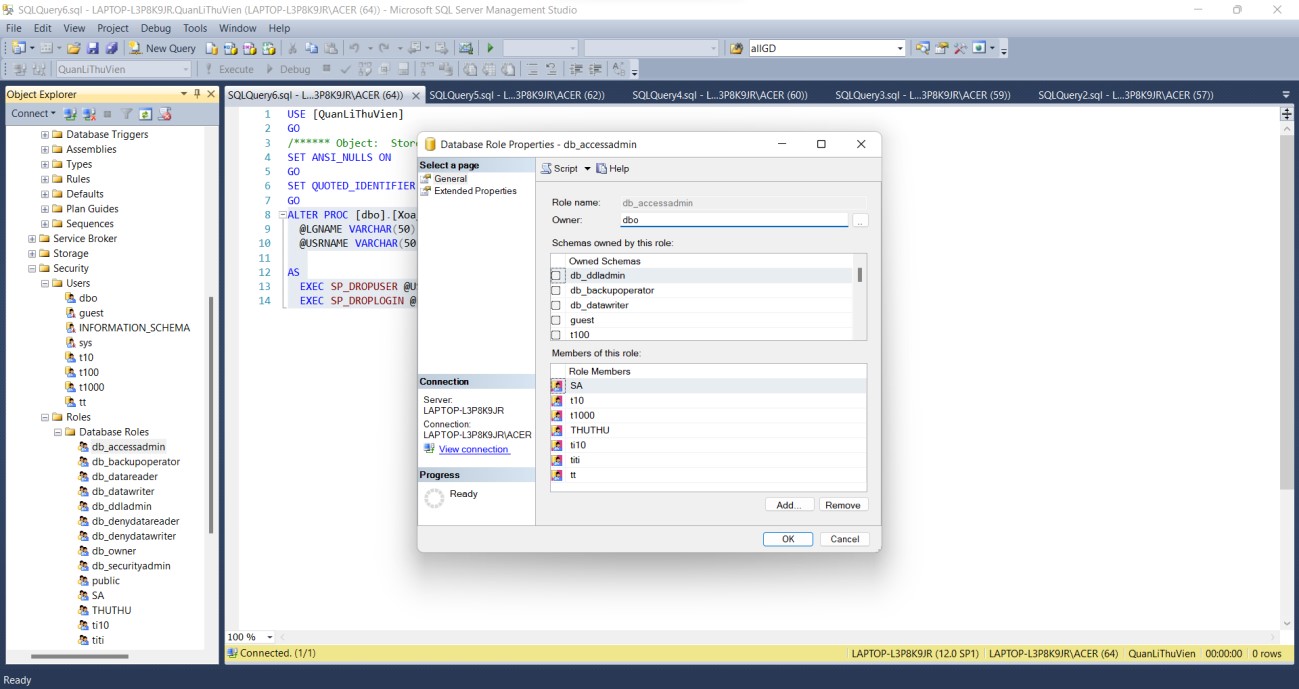
*1.2.c. Thông báo tên role trùng*



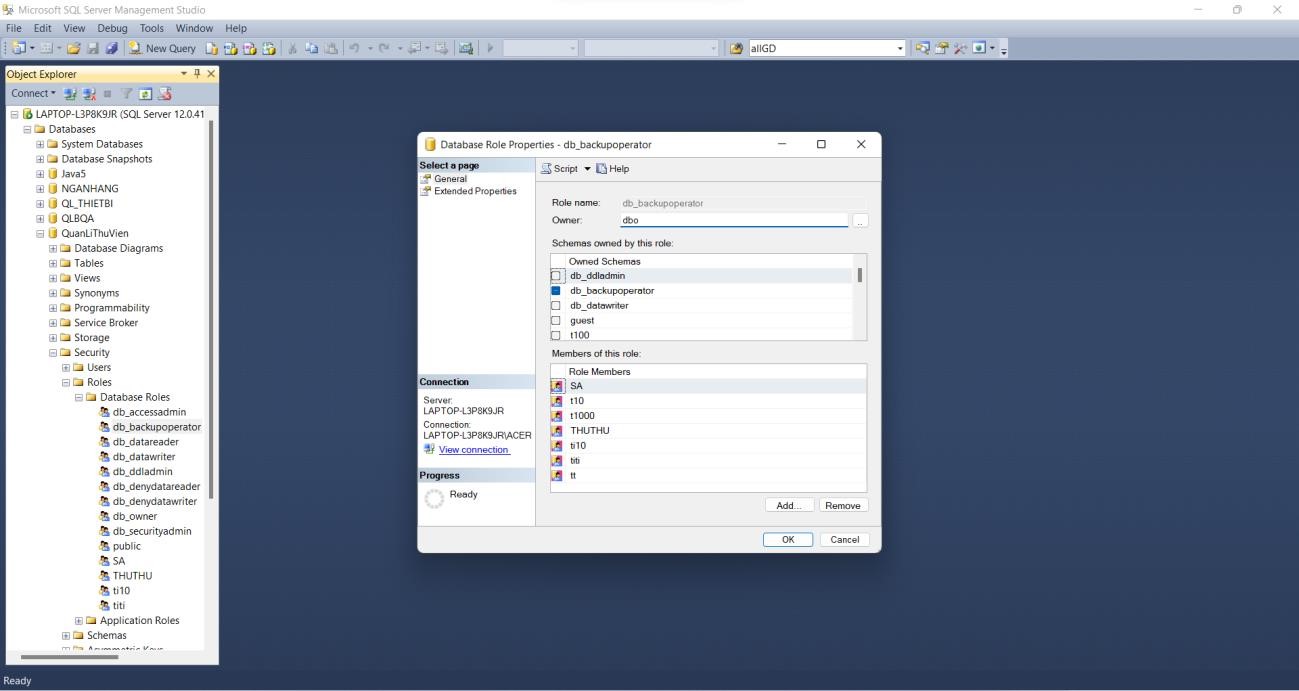
*1.2.d. Thêm thành công*



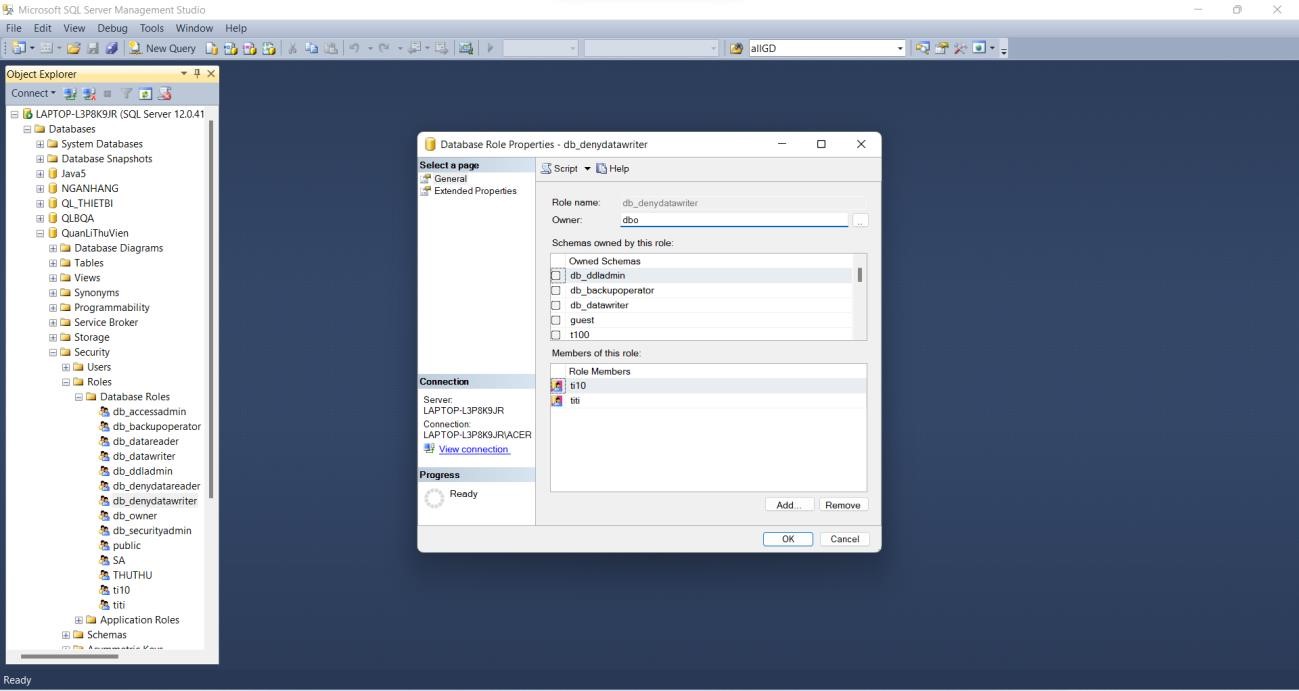
*1.2.d.1. Thêm thành công trong database*



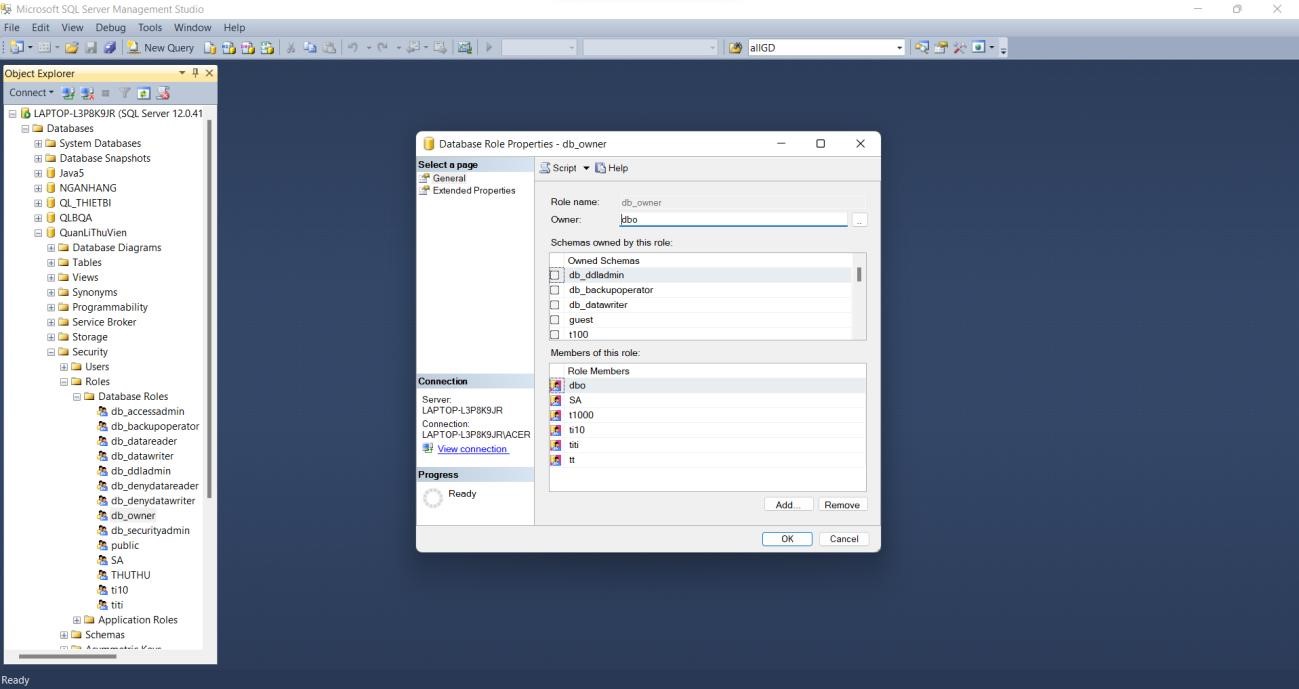
*1.2.d.2. Add thành công role vào db\_accessadmin*



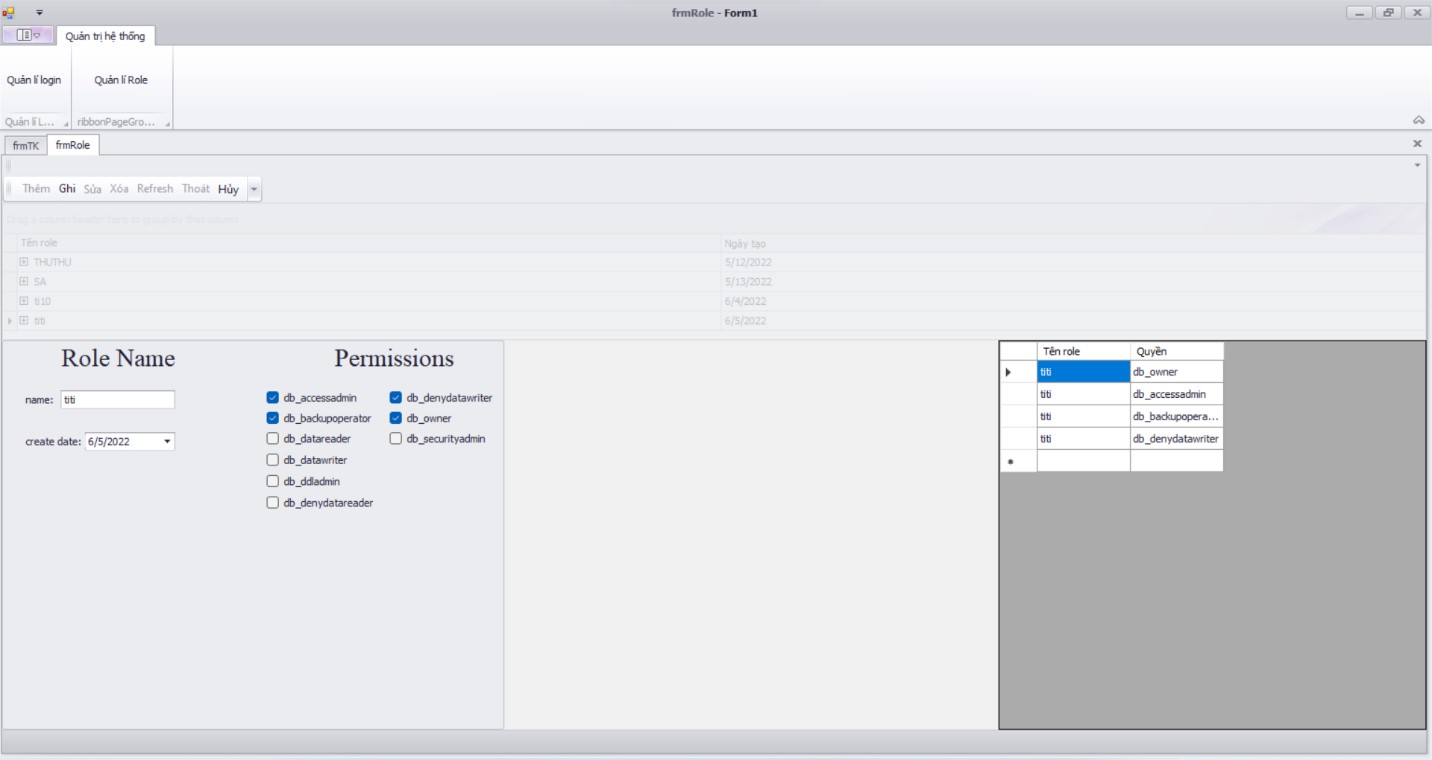
*1.2.d.3. Add thành công role vào db\_backupoperator*



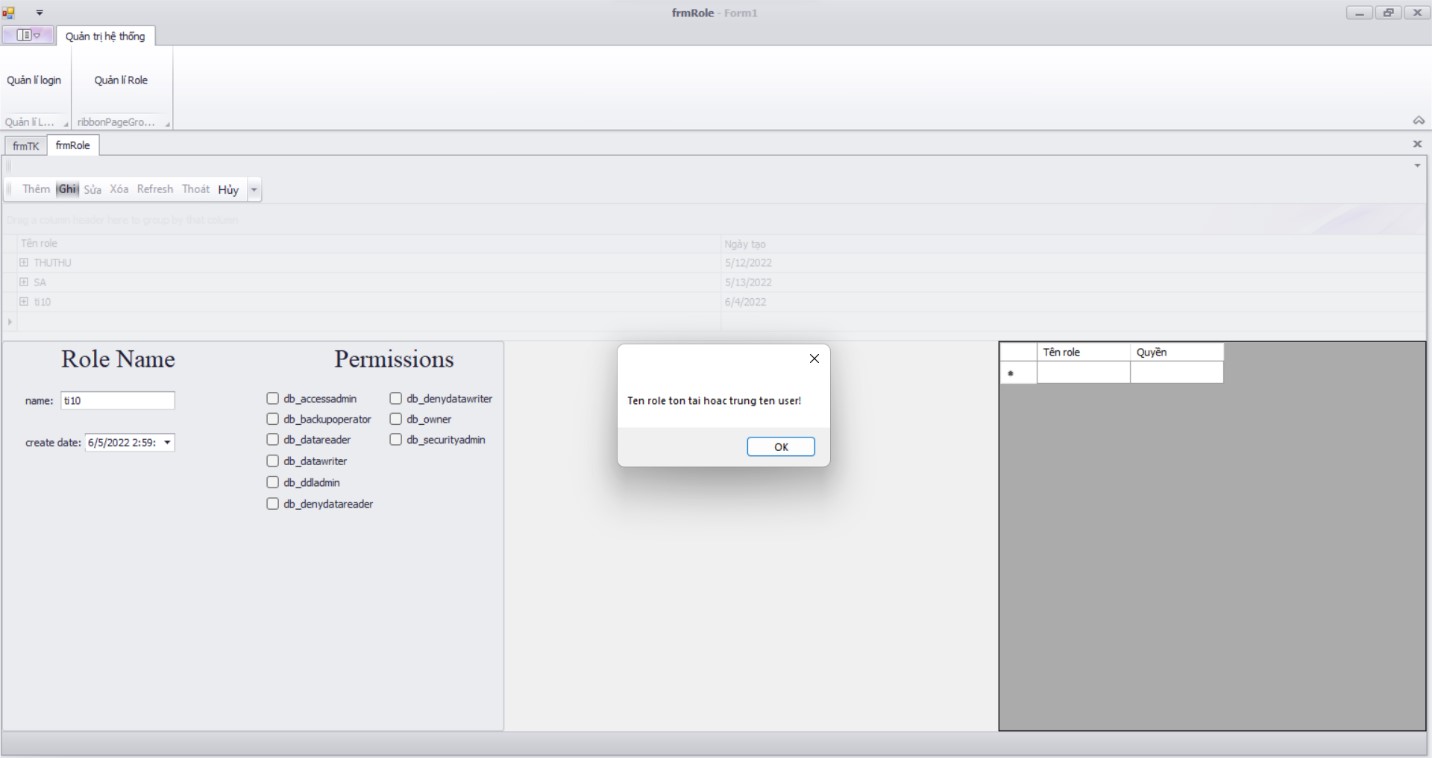
*1.2.d.4. Add thành công role vào db\_denydatawriter*



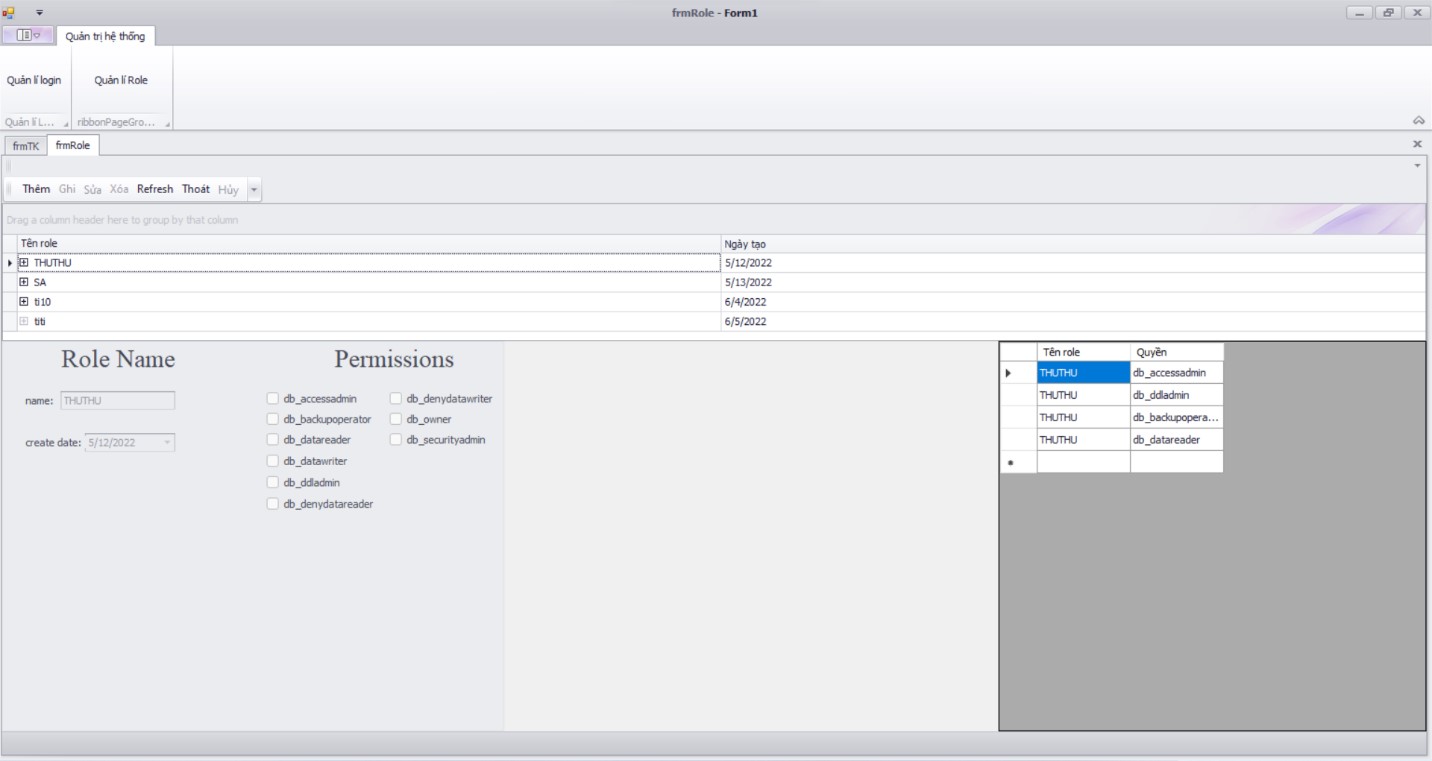
*1.2.d.5. Add thành công role vào db\_owner*



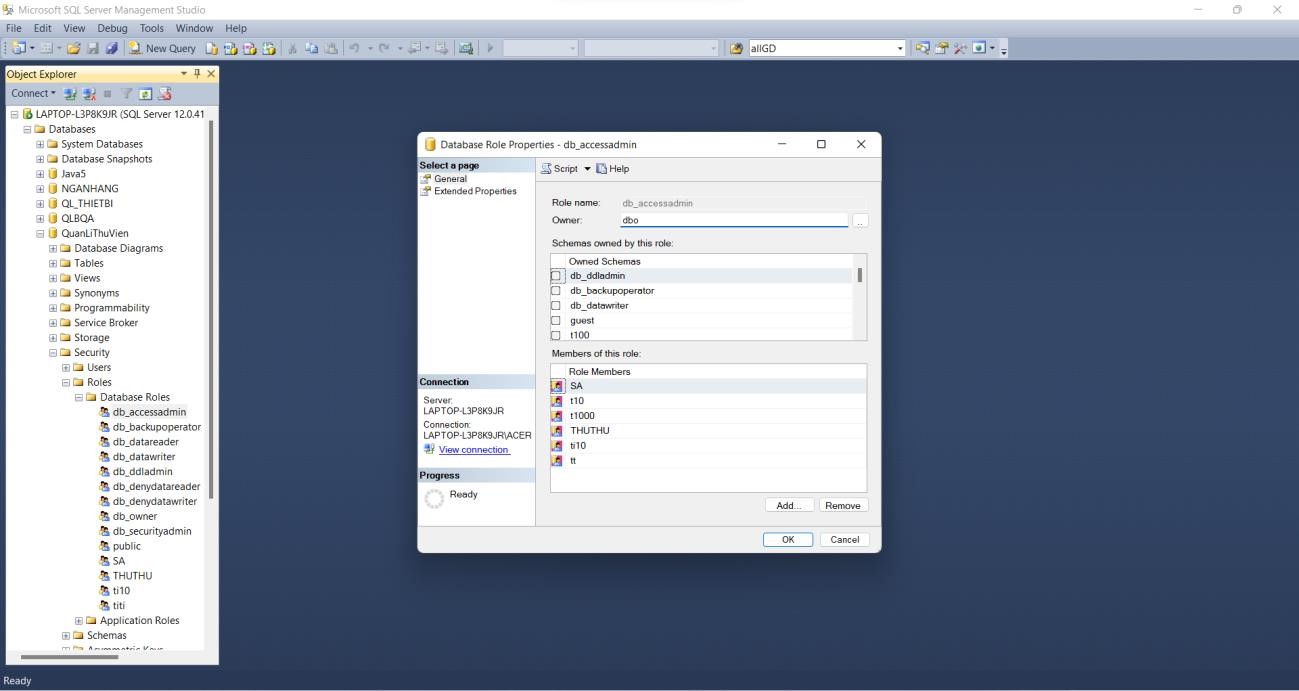
*1.2.e. Màn hình sửa*



*1.2.f. Thông báo tên role trùng*



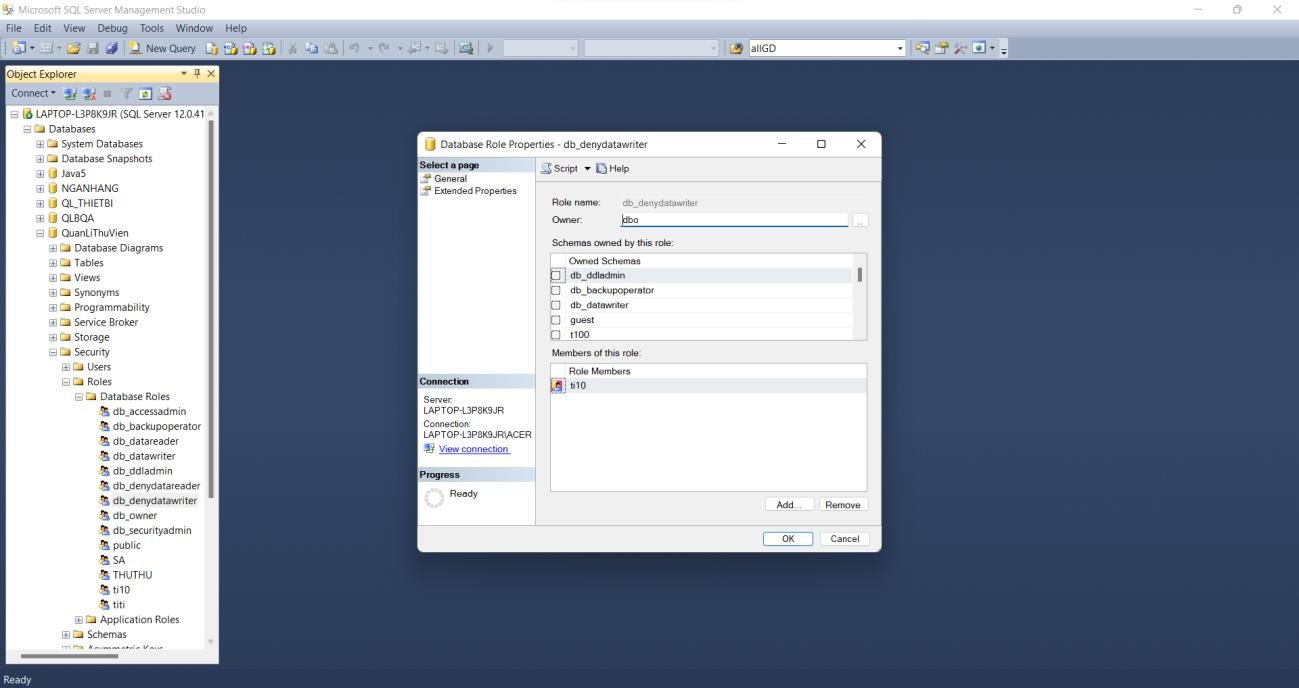
*1.2.g. Sửa thành công*



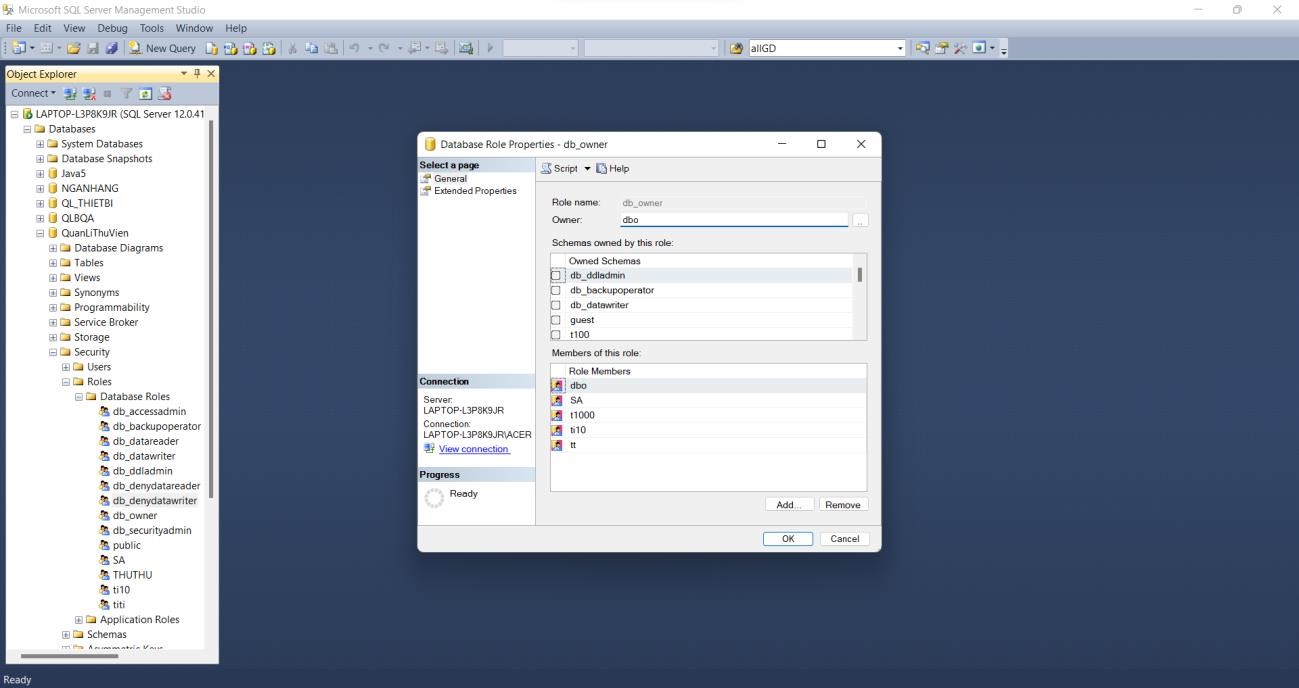
*1.2.g.1. Xóa role ra khỏi db\_accessadmin*



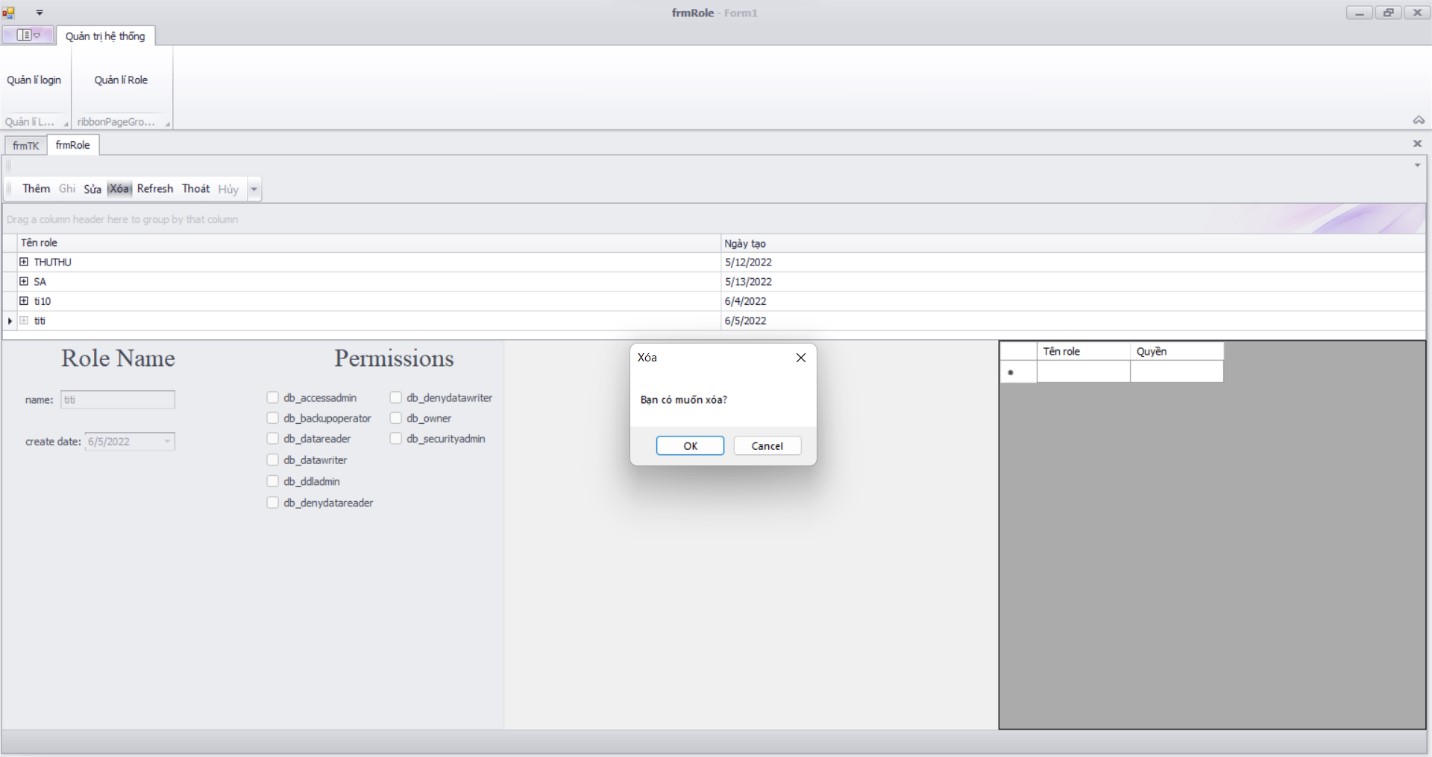
*1.2.g.2. Xóa role ra khỏi db\_backupoperator*



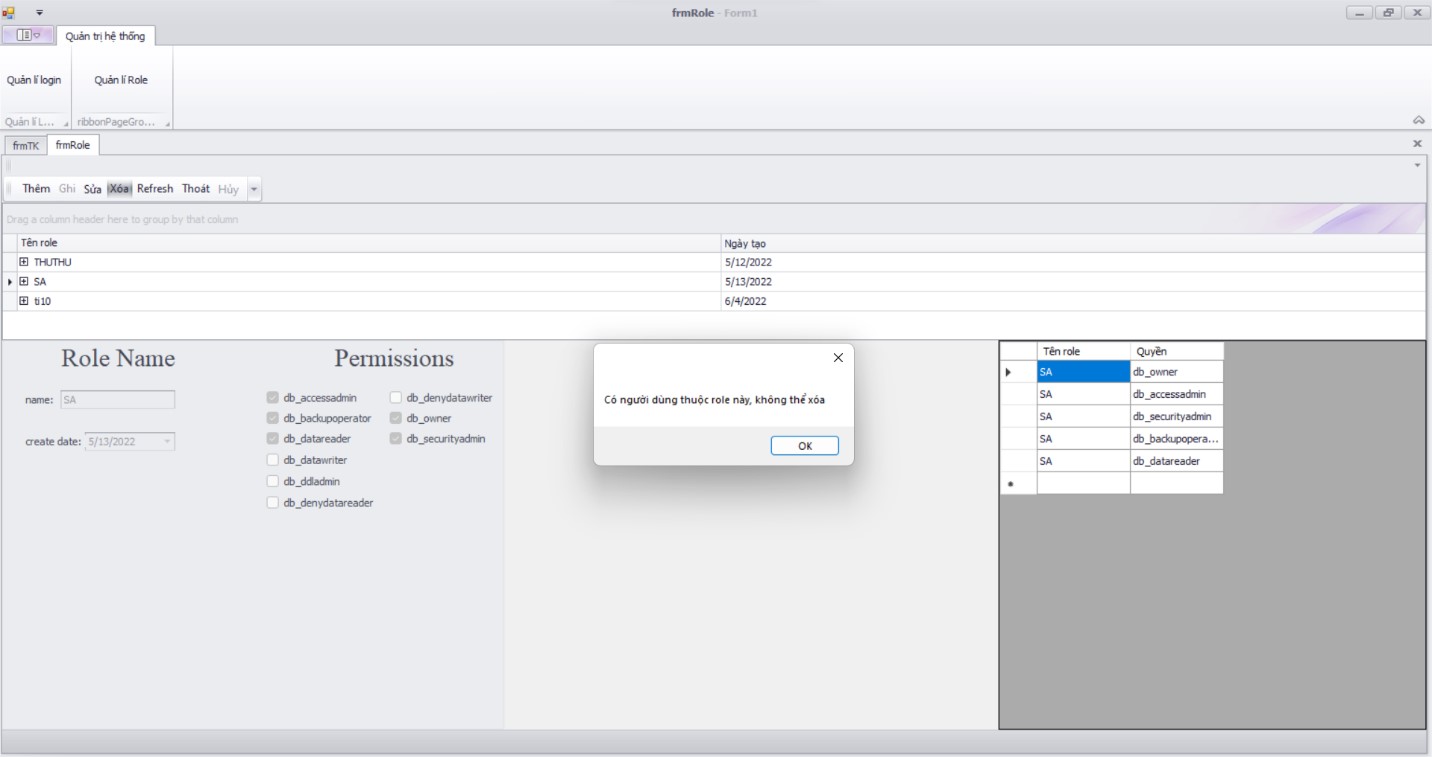
*1.2.g.3. Xóa role ra khỏi db\_denydatawriter*



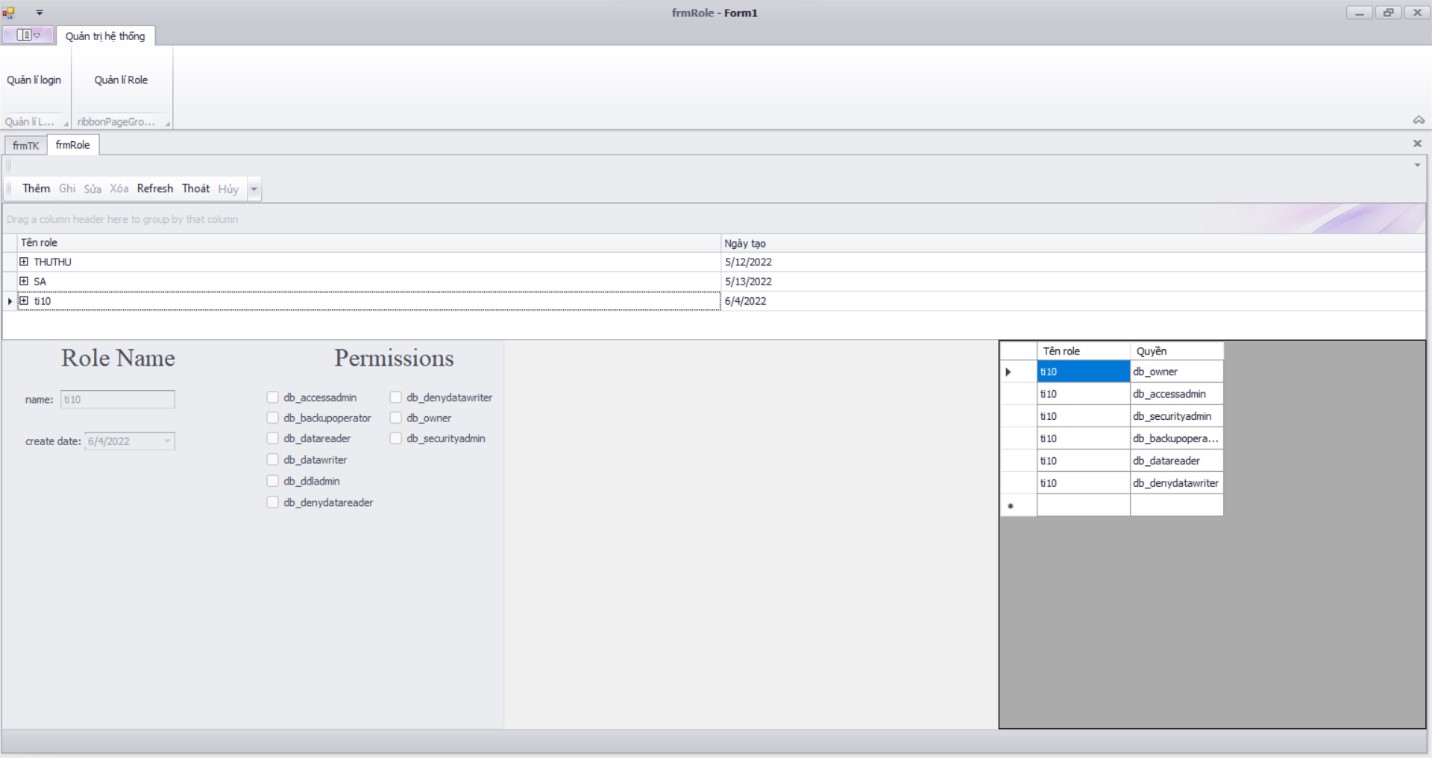
*1.2.g.4. Xóa role ra khỏi db\_owner*



*1.2.h. Thông báo xóa*



*1.2.i. Vi phạm do có login thuộc role muốn xóa*



*1.2.j. Xóa thành công*

## III.Trigger

-Trigger trong chi tiết phiếu nhập tự sinh số lượng mã sách tương ứng với số lượng đầu sách đã nhập trong chi tiết phiếu nhập.

-Lưu ý khi chỉnh sửa, xóa chỉ có thể chỉnh sửa số lượng sách trong chi tiết phiếu nhập khi sách chưa được đặt lên kệ.

-Khi trigger thực thi theo các sự kiện Insert hoặc Delete thì dữ liệu khi thay đổi sẽ được chuyển sang các bảng INSERTED và DELETED, là 2 bảng tạm thời chỉ chứa trong bộ nhớ, các bảng này chỉ được sử dụng với các lệnh trong trigger. Các bảng này thường được sử dụng để khôi phục lại phần dữ liệu đã thay đổi (roll back).

-Trigger chia thành 2 loại INSTEAD OF và AFTER: INSTEAD OF là loại trigger mà hoạt động của sự kiện gọi trigger sẽ bị bỏ qua và thay vào đó là các lệnh trong trigger được thực hiện. AFTER trigger là loại ngầm định, khác với loại INSTEAD OF thì loại trigger này sẽ thực hiện các lệnh bênh trong sau khi đã thực hiện xong sự kiện kích hoạt trigger.

### 1. TRIGGER tự sinh mã đầu sách khi thêm chi tiết phiếu nhập

CREATE TRIGGER [dbo].[TR\_TUSINH\_DAUSACH\_THEM\_CTPHIEUNHAP]

ON [dbo].[CT\_PHIEUNHAP]

AFTER INSERT

AS

BEGIN

DECLARE @SOLUONG INT

DECLARE @MACTPN INT

SELECT @SOLUONG = SOLUONG, @MACTPN = MACTPN FROM inserted

SET XACT\_ABORT ON

BEGIN TRANSACTION

BEGIN TRY

WHILE(@SOLUONG !=0)

BEGIN

DECLARE @MASACH NVARCHAR(10)

EXEC SP\_TUSINH\_MACUONSACH\_TANGDAN @MASACH OUTPUT

INSERT INTO CUONSACH (MAS,MANGAN,MACTPN)

VALUES(@MASACH, NULL, @MACTPN)

SET @SOLUONG = @SOLUONG -1

END

COMMIT

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage VARCHAR(2000)

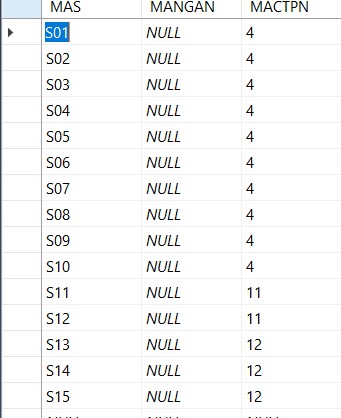
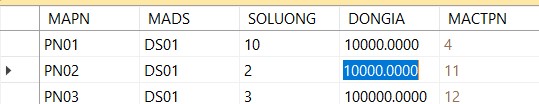
SELECT @ErrorMessage = 'Lỗi: ' + ERROR\_MESSAGE()

RAISERROR(@ErrorMessage, 16, 1)

END CATCH

END

Thêm chi tiết phiếu nhập cho PN03 với DS01 nhập 3 quyển



### 2. TRIGGER sửa chi tiết phiếu nhập

CREATE TRIGGER [dbo].[TR\_TUSINH\_DAUSACH\_SUA\_CTPHIEUNHAP]

ON [dbo].[CT\_PHIEUNHAP]

AFTER UPDATE

AS

BEGIN

DECLARE @SOLUONGNHAP INT

DECLARE @SOLUONGXOA INT

DECLARE @MACTPN INT

SELECT @SOLUONGNHAP = SOLUONG, @MACTPN = MACTPN FROM inserted

SELECT @SOLUONGXOA = SOLUONG FROM deleted

SET XACT\_ABORT ON

BEGIN TRANSACTION

BEGIN TRY

IF (@SOLUONGNHAP - @SOLUONGXOA > 0)

BEGIN

SET @SOLUONGNHAP = @SOLUONGNHAP - @SOLUONGXOA;

WHILE(@SOLUONGNHAP !=0)

BEGIN

DECLARE @MASACH NVARCHAR(10)

EXEC SP\_TUSINH\_MACUONSACH\_TANGDAN

@MASACH OUTPUT

INSERT INTO CUONSACH (MAS,MANGAN,MACTPN)

VALUES(@MASACH, NULL, @MACTPN)

SET @SOLUONGNHAP = @SOLUONGNHAP -1

END

END

ELSE IF (@SOLUONGNHAP - @SOLUONGXOA < 0)

BEGIN

DELETE CUONSACH WHERE MACTPN = @MACTPN

WHILE(@SOLUONGNHAP !=0)

BEGIN

DECLARE @MASACH1 NVARCHAR(10)

EXEC SP\_TUSINH\_MACUONSACH\_TANGDAN

@MASACH1 OUTPUT

INSERT INTO CUONSACH

(MAS,MANGAN,MACTPN) VALUES(@MASACH1, NULL, @MACTPN)

|  |  |
| --- | --- |
|  | SET @SOLUONGNHAP = @SOLUONGNHAP -1 |
| END  COMMIT  END TRY  BEGIN CATCH  ROLLBACK | END |

DECLARE @ErrorMessage VARCHAR(2000)

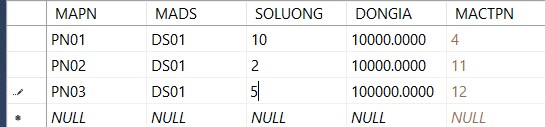
SELECT @ErrorMessage = 'Lỗi: ' + ERROR\_MESSAGE()

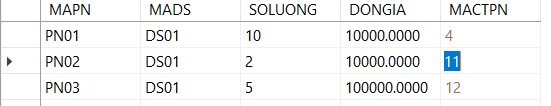
RAISERROR(@ErrorMessage, 16, 1)

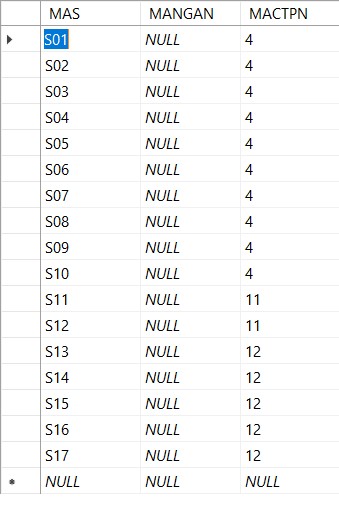
END CATCH

END

Sửa số lượng trong PN03 đầu sách DS01 thành 5 và xem kết quả







### 3. TRIGGER xóa chi tiết phiếu nhập

CREATE TRIGGER [dbo].[TR\_TUSINH\_DAUSACH\_XOA\_CTPHIEUNHAP]

ON [dbo].[CT\_PHIEUNHAP]

INSTEAD OF DELETE

AS

BEGIN

DECLARE @MACTPN INT

SELECT @MACTPN = MACTPN FROM deleted

SET XACT\_ABORT ON

BEGIN TRANSACTION

BEGIN TRY

DELETE CUONSACH WHERE MACTPN = @MACTPN

DELETE CT\_PHIEUNHAP WHERE MACTPN = @MACTPN

COMMIT

END TRY

BEGIN CATCH

ROLLBACK

DECLARE @ErrorMessage VARCHAR(2000)

SELECT @ErrorMessage = 'Lỗi: ' + ERROR\_MESSAGE()

RAISERROR(@ErrorMessage, 16, 1)

END CATCH

END

Xóa chi phiếu của nhập của phiếu nhập có mã phiếu nhập PN03

