**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO MÔN HỌC: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI 25:**

**Ứng dụng hệ quản trị cơ sở dữ liệu để xây dựng hệ thống bán thuốc online**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Giảng viên hướng dẫn** | **:** | **Phạm Trọng Huynh** | |  |
| **Lớp** | **:** | **09\_CNPM1** | |  |
| **Khóa** | **:** | **2020 – 2024** | |  |
| **Nhóm thực hiện** | **:** | **Nguyễn Anh Tới** | | **0950080156** |
|  |  | **Nguyễn Công Trường** | | **0950080039** |
| **Trần Quốc Thắng** | | **0950080038** |
|  |  |  | |
|  |  |  |  | |

**TP. Hồ Chí Minh, tháng 4 năm 2023**

**LỜI CẢM ƠN**

Xin chào,

Trước hết, em xin gửi lời cảm ơn đến các thầy giáo đã dành thời gian và tâm huyết để hướng dẫn và giảng dạy cho em trong suốt thời gian qua. Nhờ sự truyền đạt kiến thức và kinh nghiệm của thầy , em đã có được nhiều kiến thức và kỹ năng mới, từ đó phát triển bản thân và chuẩn bị tốt hơn cho tương lai.

Trong quá trình học tập và nghiên cứu, em đã gặp nhiều thách thức và khó khăn. Tuy nhiên, nhờ sự hướng dẫn và giúp đỡ của các thầy, em đã vượt qua được những khó khăn đó và tiến bộ hơn trong học tập và nghiên cứu. Em rất biết ơn và cảm kích tình cảm quan tâm và sự giúp đỡ của thầy .

Một lần nữa, em xin chân thành cảm ơn và kính chúc thầy luôn vui vẻ, hạnh phúc và thành công trong sự nghiệp giảng dạy. Em cũng hy vọng sẽ tiếp tục được học hỏi và trau dồi kiến thức từ thầy trong thời gian tới.

Trân trọng,

Top of Form

MỤC LỤC

[**Chương I: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 6](#_Toc132927743)

[**LỜI GIỚI THIỆU** 6](#_Toc132927744)

[**1. PHÂN TÍCH HỆ THỐNG ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ THÔNG BÁN THUỐC ONLINE** 7](#_Toc132927745)

[1.1 Thực trạng của hệ thống bán thuốc online 7](#_Toc132927746)

[1.1.1. Mô tả bài toán: 7](#_Toc132927747)

[1.1.2. Mô tả hoạt động của hệ thống: 7](#_Toc132927748)

[1.1.3. Quy trình xử lý nghiệp vụ của hệ thống: 9](#_Toc132927749)

[1.2. Biểu đồ phân cấp chức năng: 9](#_Toc132927750)

[1.3. Phân loại các thuộc tính vào một tập thực thể 9](#_Toc132927751)

[1.4. Xây dựng CSDL trên hệ QTCSDL 11](#_Toc132927752)

[**Chương II: ỨNG DỤNG LÝ THUYẾT** 12](#_Toc132927753)

[**1*.* Xây dựng CSDL trên HQT CSDL** 12](#_Toc132927754)

[1.1. Dữ liệu của các bảng 13](#_Toc132927755)

[1.3. Tạo và xóa tài khoản người dùng đăng nhập quyền SQL server bằng công cụ 21](#_Toc132927756)

[1.4. Tạo và xóa tài khoản người dùng đăng nhập quyền SQL server bằng câu lệnh 23](#_Toc132927757)

[***2. Thiết kế giao diện*** 24](#_Toc132927758)

[*2.1. Giao diện chính* 24](#_Toc132927759)

[*2.2. Các giao diện cập nhật dữ liệu* 24](#_Toc132927760)

[***a. Giao diện cập nhật thông tin Khách hàng*** 24](#_Toc132927761)

[***b. Giao diện cập nhật thông tin Sản Phẩm*** 28](#_Toc132927762)

[***c. Giao diện cập nhật thông tin Nhân Viên*** 32](#_Toc132927763)

[***d. Giao diện cập nhật thông tin Nhà cung cấp*** 36](#_Toc132927764)

[***e. Giao diện cập nhật thông tin Nhập hàng*** 40](#_Toc132927765)

[***g. Giao diện cập nhật thông tin đơn vị tính*** 48](#_Toc132927766)

[***h. Giao diện Bảng giá*** 52](#_Toc132927768)

[*2.3. Các tình huống tranh chấp* 56](#_Toc132927769)

[*2.4. Cài đặt các tình huống tranh chấp* 57](#_Toc132927770)

[**3. Store Procedure, Funtion, Trigger** 63](#_Toc132927771)

[3.1. Store Procedure 63](#_Toc132927772)

[3.2. Function 65](#_Toc132927773)

[3.2.1. Scalar valued Function: 65](#_Toc132927774)

[3.2.2. Tạo Function trả về giá trị loại Table-valued 66](#_Toc132927775)

[3.3. Trigger 68](#_Toc132927776)

[**4. Phân quyền người dùng** 69](#_Toc132927777)

[4.1 Nhân Viên 69](#_Toc132927778)

[4.2 Khách Hàng 77](#_Toc132927779)

[4.3 Người Quản Trị(ADMIN) 80](#_Toc132927780)

[**Chương 3: KẾT LUẬN** 82](#_Toc132927781)

[***1. Những kết quả đạt được của đồ án*** 82](#_Toc132927782)

[***2. Nhược điểm của đồ án*** 82](#_Toc132927783)

**Nhóm: 25**

**Phân công nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| Phân công công việc | | |
| 0950080156 Nguyễn Anh Tới  (33%) | 0950080039  Nguyễn Công Trường  (34%) | 0950080038  Trần Quốc Thắng  (33%) |
| - Phân tích nghiệp vụ CSDL  - Phân tích nghiệp vụ ERD  - Tạo các database  - Mô tả giao diện  - Phân quyền: Phân hệ quản lý | - Phân tích nghiệp vụ CSDL  - Phân tích nghiệp vụ ERD  - UI Thiết kế giao diện  - Tình huống tranh chấp | - Phân tích nghiệp vụ CSDL  - Phân tích nghiệp vụ ERD  - UI thêm Sản Phẩm..  - Insert dữ liệu cho các bảng  - Viết báo cáo |

# **Chương I: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

# **LỜI GIỚI THIỆU**

Trong thời đại số hóa hiện nay, việc sử dụng ứng dụng hệ quản trị cơ sở dữ liệu để xây dựng hệ thống bán thuốc online là một xu hướng không thể bỏ qua. Với sự phát triển của ngành dược phẩm và nhu cầu sử dụng thuốc của con người ngày càng tăng cao, việc cung cấp dịch vụ bán thuốc trực tuyến trở thành một lựa chọn hợp lý cho nhiều người.

Ứng dụng hệ quản trị cơ sở dữ liệu cho phép lưu trữ, quản lý và truy xuất thông tin liên quan đến thuốc một cách nhanh chóng và hiệu quả. Hệ thống này cho phép khách hàng dễ dàng tìm kiếm và mua các sản phẩm thuốc tại nhà một cách tiện lợi và an toàn. Bên cạnh đó, hệ thống cũng giúp cho các nhà cung cấp dược phẩm có thể quản lý được lượng hàng tồn kho, theo dõi các đơn đặt hàng và xử lý các giao dịch một cách nhanh chóng và chính xác.

Với sự phát triển của công nghệ, việc sử dụng ứng dụng hệ quản trị cơ sở dữ liệu không chỉ giúp cho các doanh nghiệp trong ngành dược phẩm có thể tối ưu hóa quy trình kinh doanh mà còn mang lại nhiều tiện ích cho người dùng. Đó là lý do tại sao ứng dụng hệ quản trị cơ sở dữ liệu đang trở thành một công cụ quan trọng trong việc xây dựng hệ thống bán thuốc online hiện nay.

# **1. PHÂN TÍCH HỆ THỐNG ỨNG DỤNG HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐỂ XÂY DỰNG HỆ THÔNG BÁN THUỐC ONLINE**

## 1.1 Thực trạng của hệ thống bán thuốc online

### 1.1.1. Mô tả bài toán:

Hệ thống quản lý bán thuốc online là một hệ thống phần mềm được sử dụng để quản lý các hoạt động kinh doanh bán hàng của một cửa hàng thuốc online .Hệ thống sẽ bao gồm các chức năng như Đăng Ký, Đăng Nhập, Xem danh sách các sản phẩm, Xem chi tiết sản phẩm, đặt hàng, chat với khách hàng,tìm kiếm sản phẩm, quản lý kho hàng, Quản lý báo cáo, Quản lý tài khoản, Quản lý thống kê.

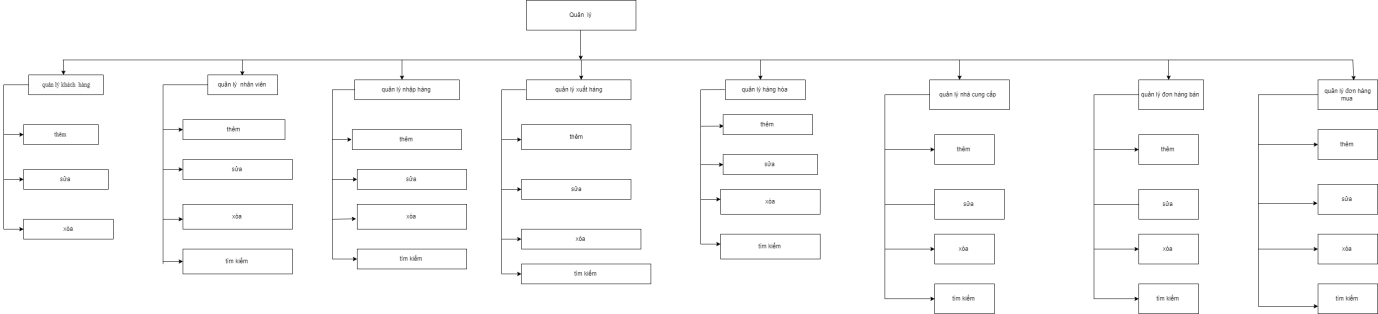
### 1.1.2. Mô tả hoạt động của hệ thống:

* Quản lý khách hàng: Chức năng cho phép quản lý thông tin các khách hàng, bao gồm tên, địa chỉ, số điện thoại, email và các thông tin khác liên quan đến giao dịch mua bán. Nhờ tính năng này, hệ thống có thể lưu trữ, tra cứu thông tin khách hàng và giúp các nhân viên chăm sóc khách hàng có thể đưa ra các giải pháp, giúp đỡ khách hàng một cách nhanh chóng và hiệu quả hơn.
* Quản lý nhân viên: Chức năng cho phép quản lý thông tin các nhân viên, bao gồm tên, chức vụ, lương, các thông tin liên quan đến giao dịch và hoạt động của nhân viên trong công ty. Chức năng này giúp cho quản lý có thể theo dõi tình trạng hoạt động của từng nhân viên và đưa ra các biện pháp quản lý tốt hơn.
* Quản lý nhập hàng: Chức năng cho phép quản lý việc nhập hàng từ các nhà cung cấp. Tính năng này cho phép lưu trữ thông tin về đơn hàng, số lượng hàng nhập, giá cả và các thông tin khác liên quan đến quá trình nhập hàng. Quản lý nhập hàng giúp cho công ty có thể quản lý tốt hơn việc nhập hàng và giảm thiểu sai sót trong quá trình này.
* Quản lý xuất hàng: Chức năng cho phép quản lý việc xuất hàng cho khách hàng. Tính năng này cho phép lưu trữ thông tin về đơn hàng, số lượng hàng xuất, giá cả và các thông tin khác liên quan đến quá trình xuất hàng. Quản lý xuất hàng giúp cho công ty có thể quản lý tốt hơn việc xuất hàng và giảm thiểu sai sót trong quá trình này.
* Quản lý hàng hóa: Chức năng cho phép quản lý thông tin về các mặt hàng được bán trong cửa hàng. Tính năng này cho phép lưu trữ thông tin về tên hàng hóa, mã hàng, số lượng, giá cả và các thông tin khác liên quan đến quản lý hàng hóa. Quản lý hàng hóa giúp cho công ty có thể quản lý tốt hơn việc nhập, xuất và bán hàng hóa của mình.
* Quản lý nhà cung cấp: Chức năng quản lý nhà cung cấp trong hệ thống quản trị cơ sở dữ liệu cho phép người dùng quản lý thông tin của các nhà cung cấp mà doanh nghiệp đang hợp tác với. Thông tin về nhà cung cấp bao gồm tên, địa chỉ, số điện thoại, email, thông tin tài khoản ngân hàng, danh sách sản phẩm mà nhà cung cấp cung cấp và giá cả. Chức năng này giúp cho doanh nghiệp dễ dàng quản lý được thông tin của các nhà cung cấp, tương tác với các nhà cung cấp và đưa ra quyết định hợp tác một cách chính xác.
* Quản lý đơn hàng bán: Chức năng quản lý đơn hàng bán cho phép người dùng quản lý thông tin của các đơn hàng mà khách hàng đã đặt mua sản phẩm. Thông tin của đơn hàng bao gồm thông tin khách hàng, số điện thoại, địa chỉ, sản phẩm được mua, số lượng, giá cả, phí vận chuyển và tổng giá trị của đơn hàng. Chức năng này giúp cho doanh nghiệp dễ dàng theo dõi các đơn hàng đã được bán ra, tình trạng đơn hàng, đưa ra các quyết định về vận chuyển và giao hàng cho khách hàng một cách nhanh chóng và chính xác.
* Quản lý đơn hàng mua: Chức năng quản lý đơn hàng mua cho phép người dùng quản lý thông tin của các đơn hàng mà doanh nghiệp đã đặt mua sản phẩm từ các nhà cung cấp. Thông tin của đơn hàng mua bao gồm thông tin nhà cung cấp, số điện thoại, địa chỉ, sản phẩm được đặt mua, số lượng, giá cả, phí vận chuyển và tổng giá trị của đơn hàng. Chức năng này giúp cho doanh nghiệp dễ dàng quản lý các đơn hàng đã đặt mua từ các nhà cung cấp, tình trạng đơn hàng, đưa ra các quyết định về vận chuyển và thanh toán cho các nhà cung cấp một cách nhanh chóng và chính xác.

### 1.1.3. Quy trình xử lý nghiệp vụ của hệ thống:

Khách hàng thực hiện mua sản phẩm trên phần mềm,đăng nhập tài khoản để thực hiện mua sắm sản phẩm,sau khi chọn sản phẩm ưng ý thì có thể kiểm tra giỏ hàng và thanh toán, hoặc có thể chat nhờ nhân viên tư vấn về sản phẩm.Nhân viên sẽ kiểm tra sản phẩm trong kho và gửi đơn hàng cho bạn.

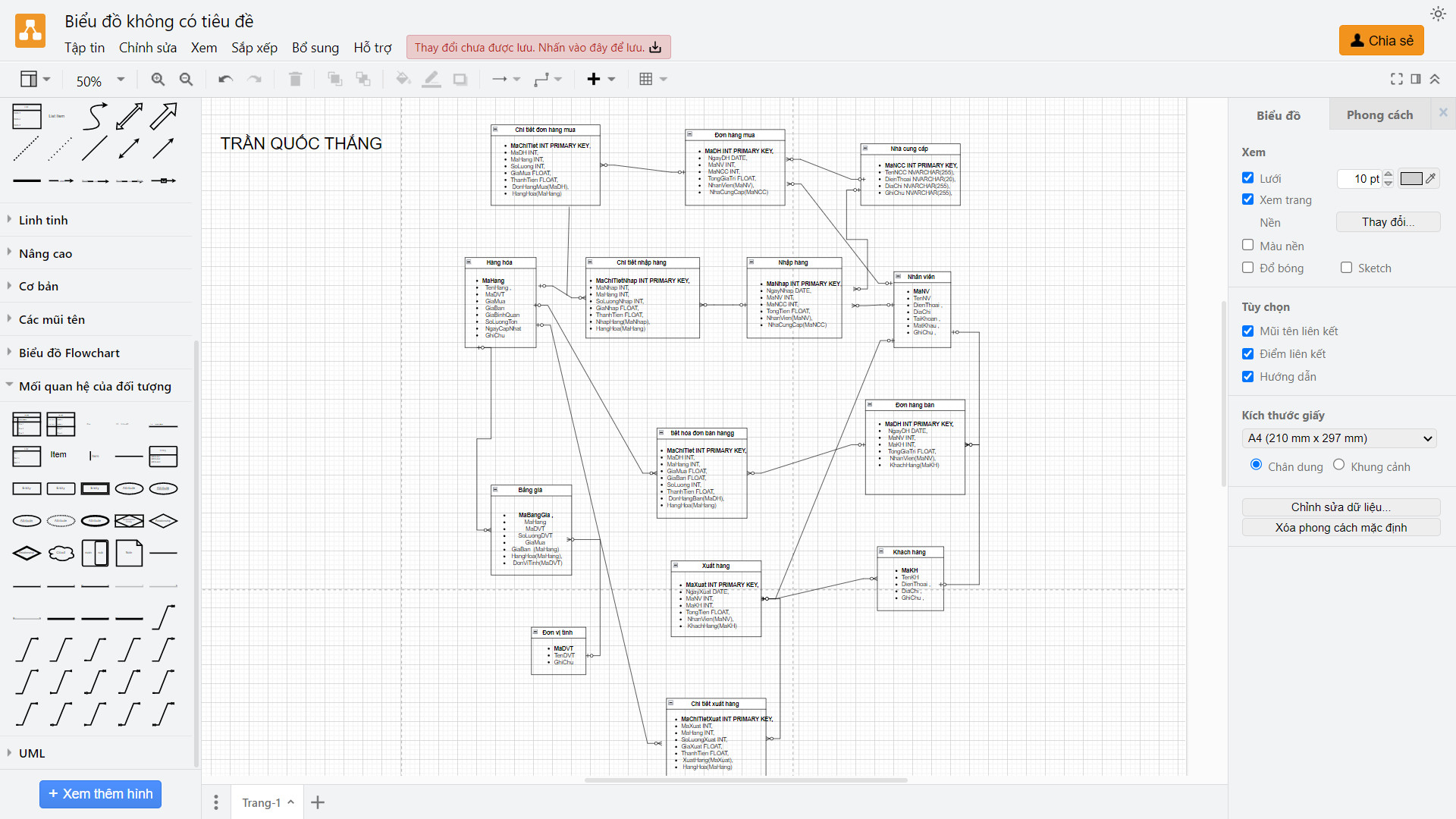
## 1.2. Biểu đồ phân cấp chức năng:



1.3. Phân loại các thuộc tính vào một tập thực thể

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Kiểu thực thể | Thuộc tính | Thuộc tính khóa |
| 1 | HangHoa | mã hàng, tên hàng, đơn vị tính, giá mua, giá bán, giá bình quân, số lượng tồn, ngày cập nhật và ghi chú. | Mã hàng |
| 2 | DonViTinh | mã đơn vị tính, tên đơn vị tính và ghi chú | Mã đơn vị tính |
| 3 | BangGia | mã bảng giá, mã hàng hóa, mã đơn vị tính, số lượng đơn vị tính, giá mua và giá bán | Mã bảng giá |
| 4 | NhanVien | mã nhân viên, tên nhân viên, số điện thoại, địa chỉ, tên tài khoản, mật khẩu và ghi chú | Mã nhân viên |
| 5 | KhachHang | mã khách hàng, tên khách hàng, số điện thoại, địa chỉ và ghi chú. | Mã Khách hàng |
| 6 | DonHangBan | mã đơn hàng, ngày đặt hàng, mã nhân viên, mã khách hàng, tổng giá trị | Mã đơn hàng |
| 7 | ChiTietDonHangBan | mã chi tiết, mã đơn hàng, mã hàng hóa, giá mua, giá bán, số lượng và thành tiền | Mã chi tiết |
| 8 | NhaCungCap | mã nhà cung cấp, tên nhà cung cấp, số điện thoại, địa chỉ và ghi chú. | Mã nhà cung cấp |
| 9 | DonHangMua | mã đơn hàng, ngày đặt hàng, mã nhân viên, mã nhà cung cấp, tổng giá trị | Mã đơn hàng |
| 10 | ChiTietDonHangMua | MaChiTiet ,MaDH ,MaHang, SoLuong, GiaMua, và ThanhTien | Mã chi tiết |
| 11 | NhapHang | MaNhap, NgayNhap, MaNV, MaNCC, và TongTien | Mã nhập |
| 12 | ChiTietNhapHang | MaChiTietNhap, MaNhap, MaHang, SoLuongNhap, GiaNhap, và ThanhTien | Mã chi tiết |
| 13 | XuatHang | MaXuat, NgayXuat, MaNV, MaKH, và TongTien | Mã xuất |
| 14 | ChiTietXuatHang | MaChiTietXuat, MaXuat,SoLuongXuat, GiaXuat, ThanhTien | Mã Chi tiết |

Mô hình Thực Thể Kết hợp (ERD)



## 1.4. Xây dựng CSDL trên hệ QTCSDL

1. Hàng Hóa (Mã Hàng, Tên Hfang,Mã DVT,Giá Mua,Giá Bán,Gía Bình Quân , Ngày Cập Nhật ,Ghi Chú).

2. Đơn Vị Tính: (Mã DVT,Tên DVT,Ghi Chú).

3. Bảng Giá: (Mã Bảng Giá, Mã Hàng, Mã DVT, Số Lượng DVT ,Giá Mua,Giá Bán).

4. Nhân Viên: (Mã NV,Tên NV,Điện Thọai, Địa Chỉ, Tài Khỏan, Mật Khẩu,Ghi Chú).

5. Khách Hàng: (Mã KH, Tên KH, Điện Thọai, Địa Chỉ,Ghi Chú).

6. Đơn Hàng Bán: (Mã DH, Ngày DH, Mã NV, Mã KH,Tổng Giá Trị).

7. Chi Tiết Đơn Hàng Bán: (Mã Chi Tiết, Mã DH, Mã Hàng, Giá Mua, Giá Bán, Số Lượng,Thành Tiền).

8. Nhà Cung Cấp: (Mã NCC, Tên NCC, Điện Thoại)

9. Đơn Hàng Mua: (Mã DH, Ngày DH, Mã NV, Mã NCC,Tổng Giá Trị)

10. Chi Tiết Đơn Hàng Mua : (Mã Chi Tiết, Mã DH, Mã Hàng, Số Lượng, Giá Mua,Thành Tiền)

11. Nhập Hàng: (Mã Nhập, Ngày Nhập, Mã NV, Mã NCC, Tổng Tiền)

12. Chi Tiết Nhập Hàng: (Mã Chi Tiết Nhập , Mã Nhập, Mã Hàng, Số Lượng Nhập, Giá Nhập, Thành Tiền)

13. Xuất Hàng: (Mã Xuất, Ngày Xuất, Mã NV, Mã KH, Tổng Tiền)

# **Chương II: ỨNG DỤNG LÝ THUYẾT**

# **1*.* Xây dựng CSDL trên HQT CSDL**

* Câu lệnh tạo CSDL

CREATE DATABASE Tên\_CSDL;

Ví dụ: CREATE DATABASE QLYBANHANGNOITHAT;

* Câu lệnh tạo bảng

CREATE TABLE <tên bảng> (

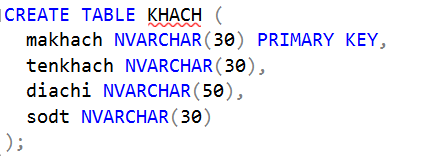
<tên cột 1> <kiểu dữ liệu 1> [RBTV],

<tên cột 2> <kiểu dữ liệu 2> [RBTV] ,

…

<tên cột n> <kiểu dữ liệu n> [RBTV];)

Ví dụ:



* Câu lệnh tạo khóa

**Cú pháp tạo khóa chính bằng lệnh CREATE TABLE**

CREATE TABLE ten\_bang

(

cot1 kieu\_du\_lieu [ NULL | NOT NULL ] [ PRIMARY KEY ],

cot2 kieu\_du\_lieu [ NULL | NOT NULL ],

...

);

**Cú pháp tạo khóa ngoài**

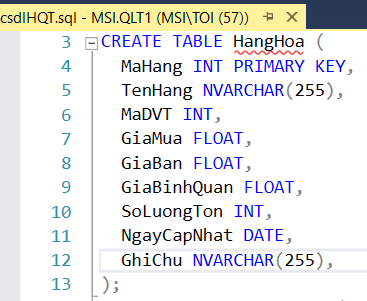
ALTER TABLE<tên bảng>

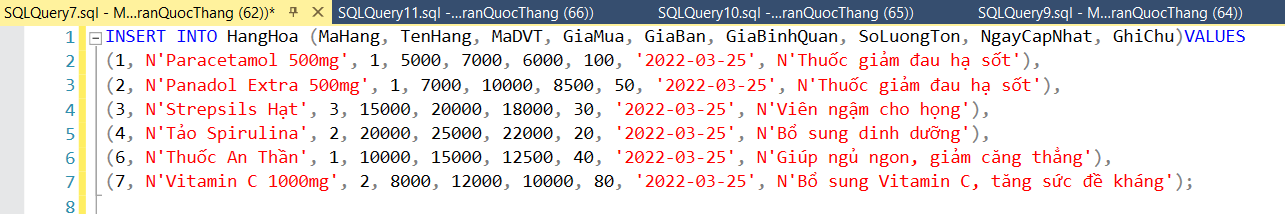
ADD FOREIGN KEY (Danh sách tên cột)

REFERENCES <tên bảng> (Danh sách tên cột);

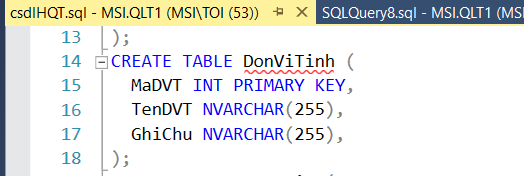
1.1. Dữ liệu của các bảng

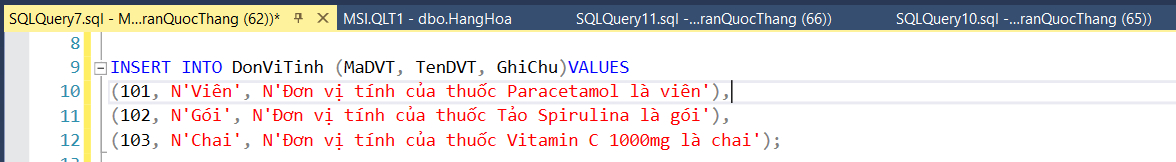
* Bảng Hàng Hóa

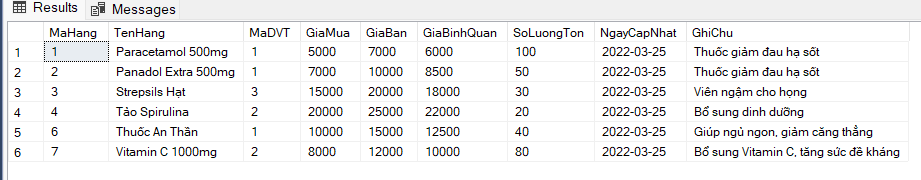




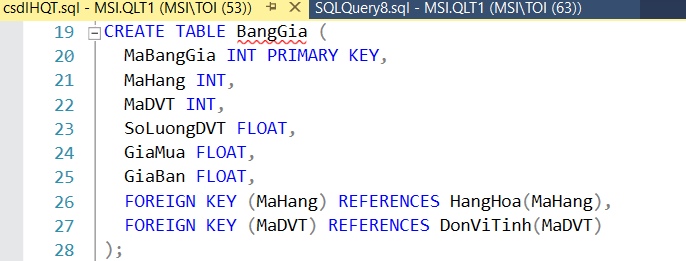
* Bảng Đơn Vị Tính

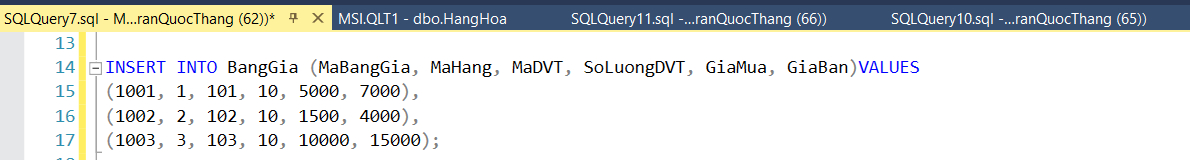


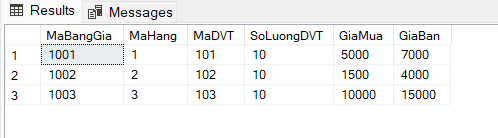




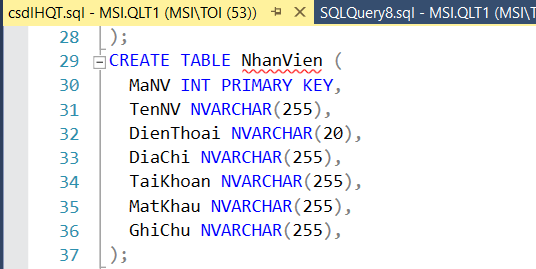
* Bảng Bảng giá sản phẩm

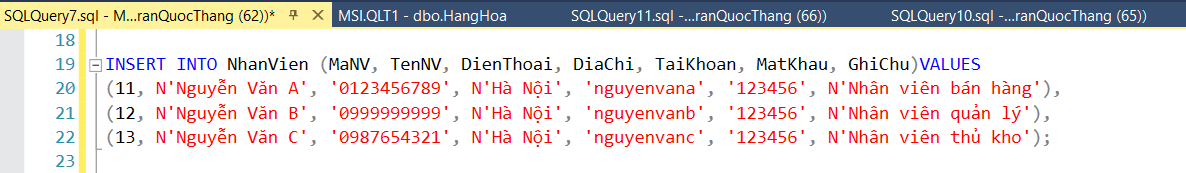


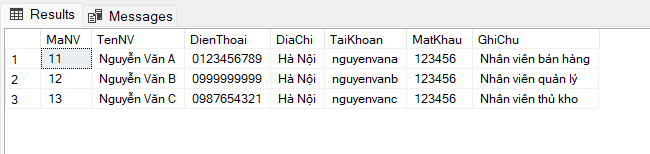




* Bảng Nhân Viên

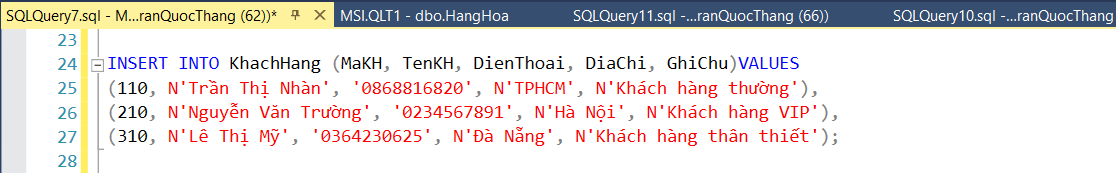
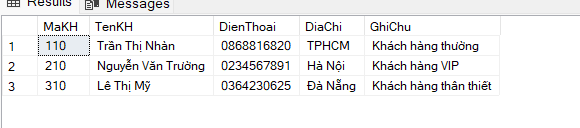




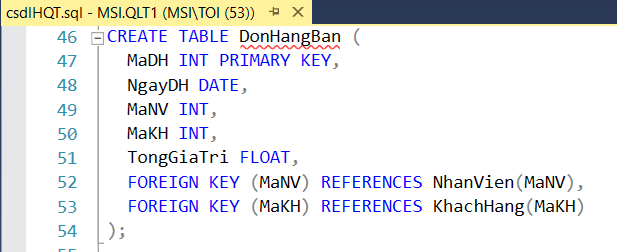
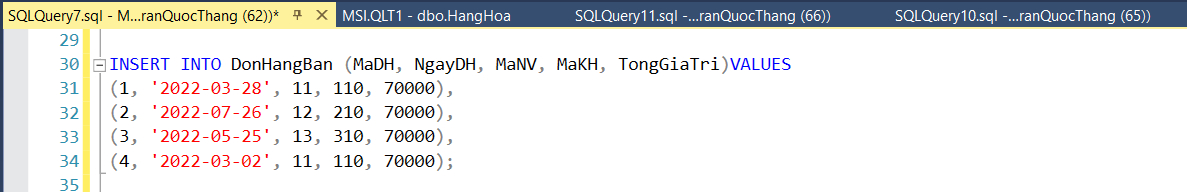


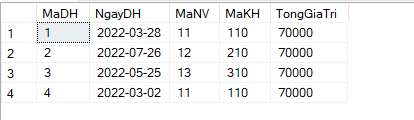
* Bảng khách Hàng



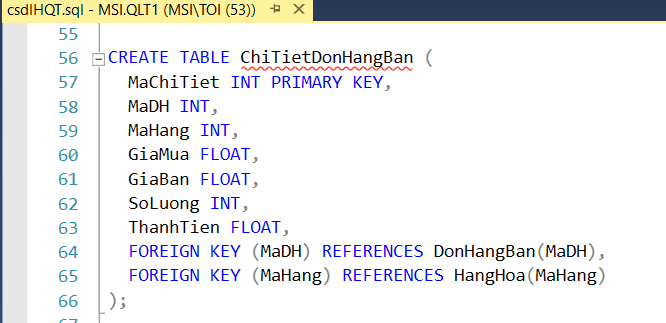
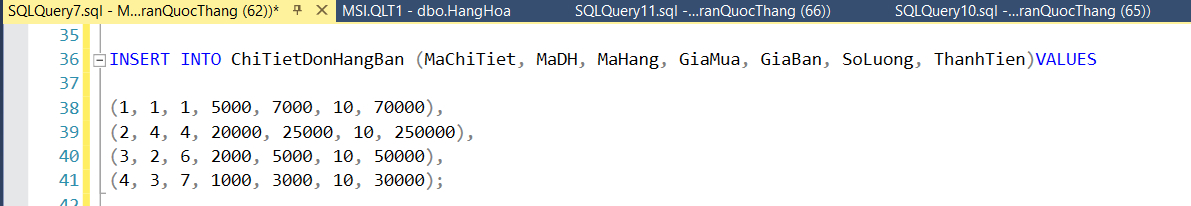
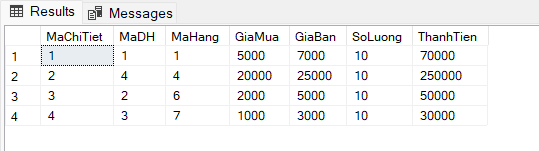
 

* Bảng Đơn hàng bán

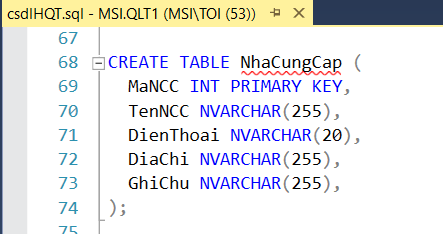
 

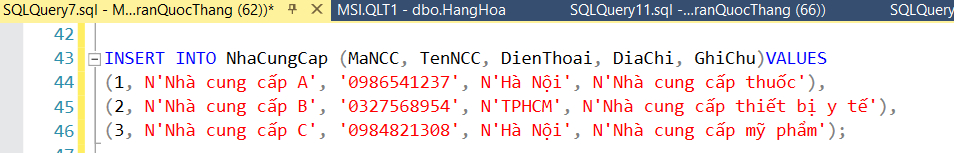
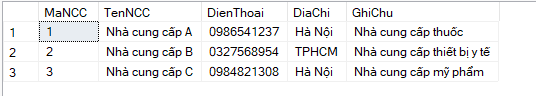


* Bảng chi tiết Đơn hàng bán

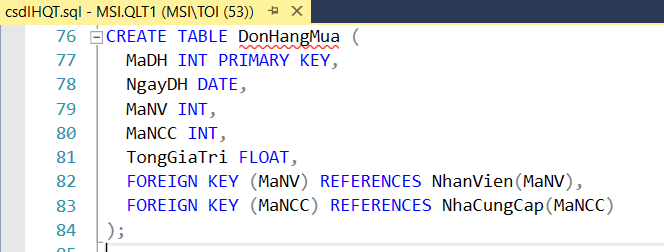
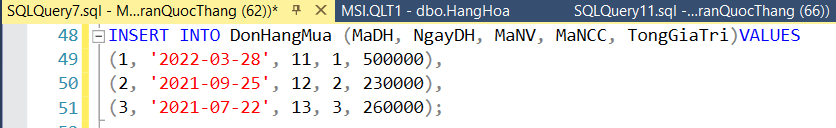
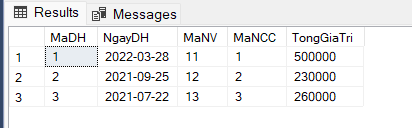
  

* Bảng Nhà cung cấp

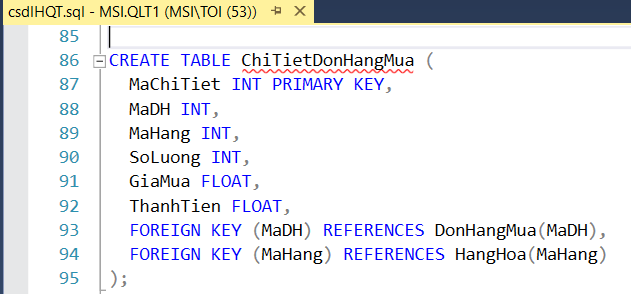
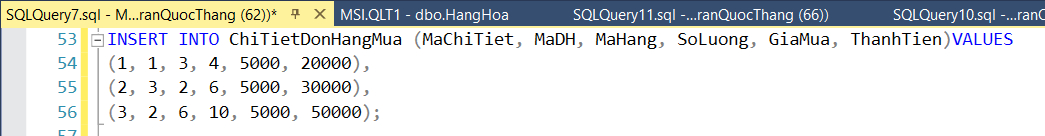
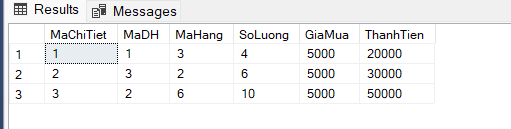


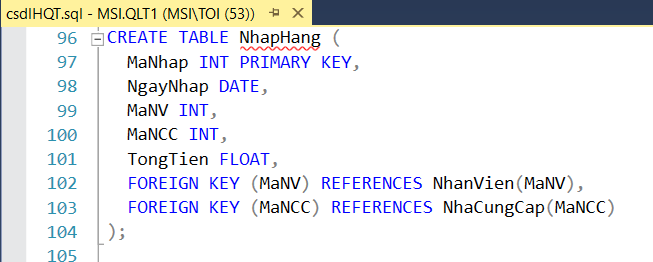
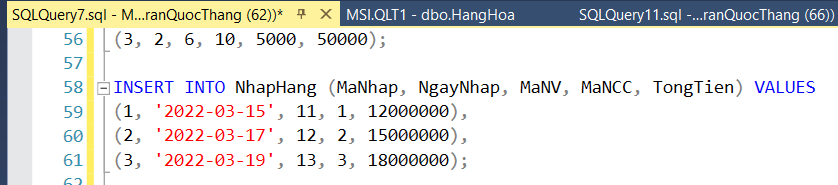
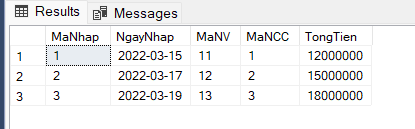
* Bảng Đơn Hàng Mua

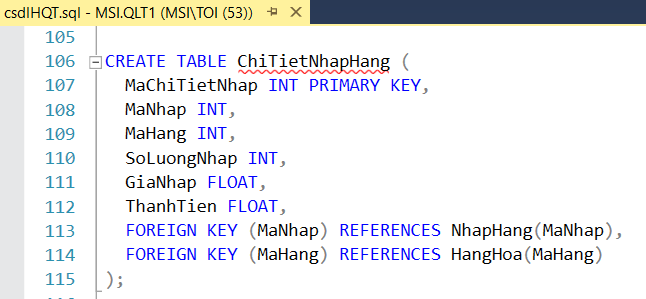
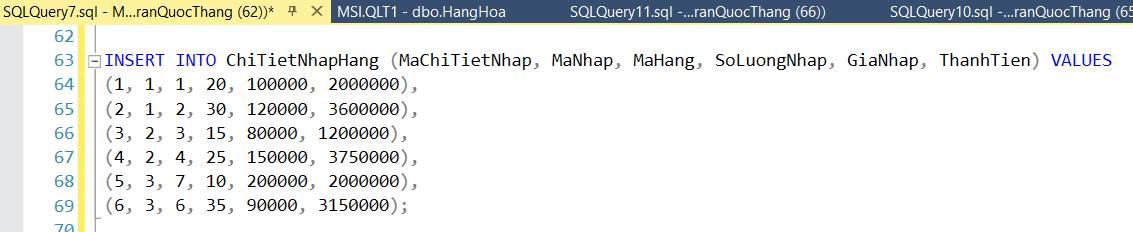
* Bảng chi tiết Đơn Hàng Mua

* Bảng Nhập Hàng

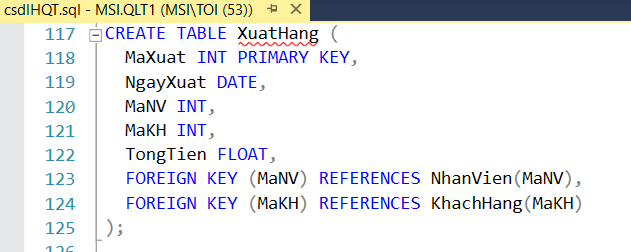
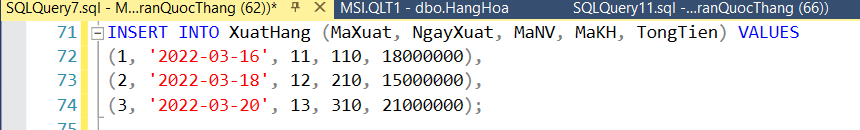
  

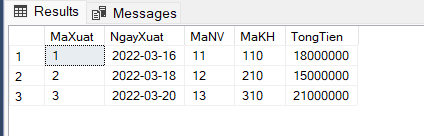
* Bảng chi tiết Nhập Hàng

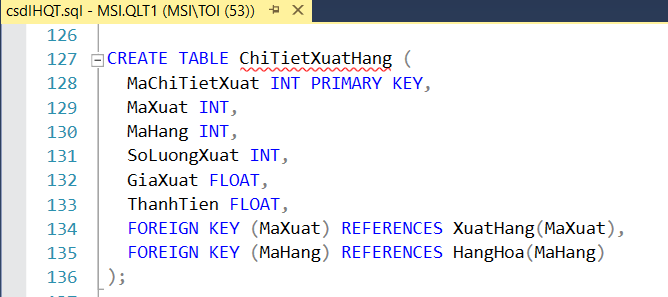


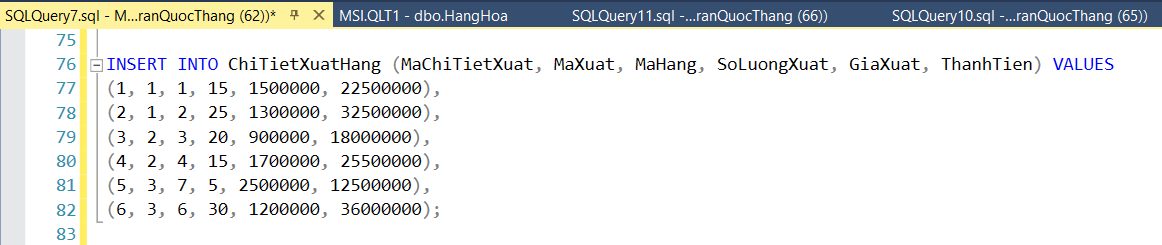
* Bảng Xuất hàng

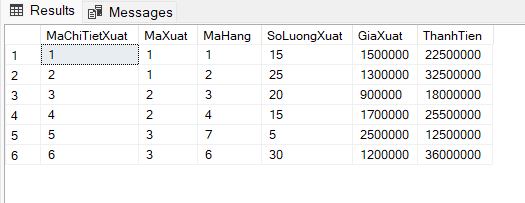
 



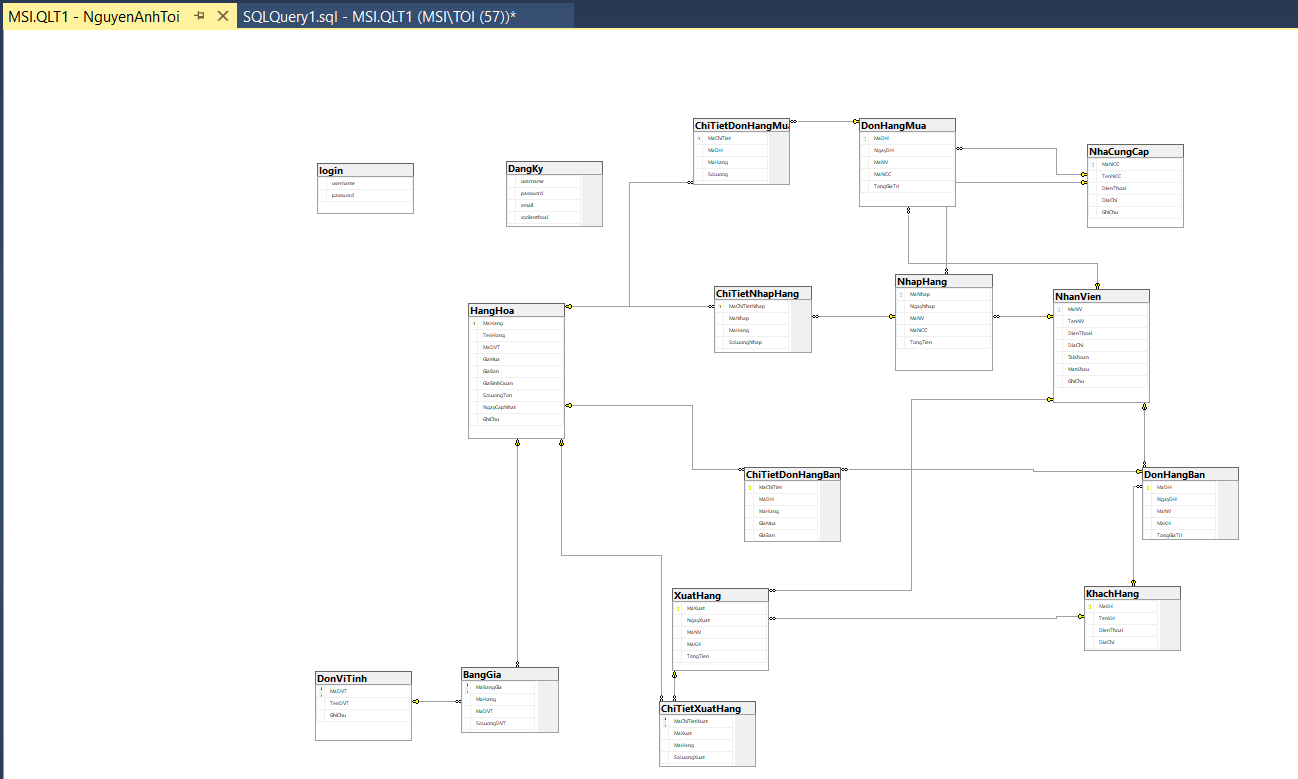
* Bảng chi tiết xuất hàng





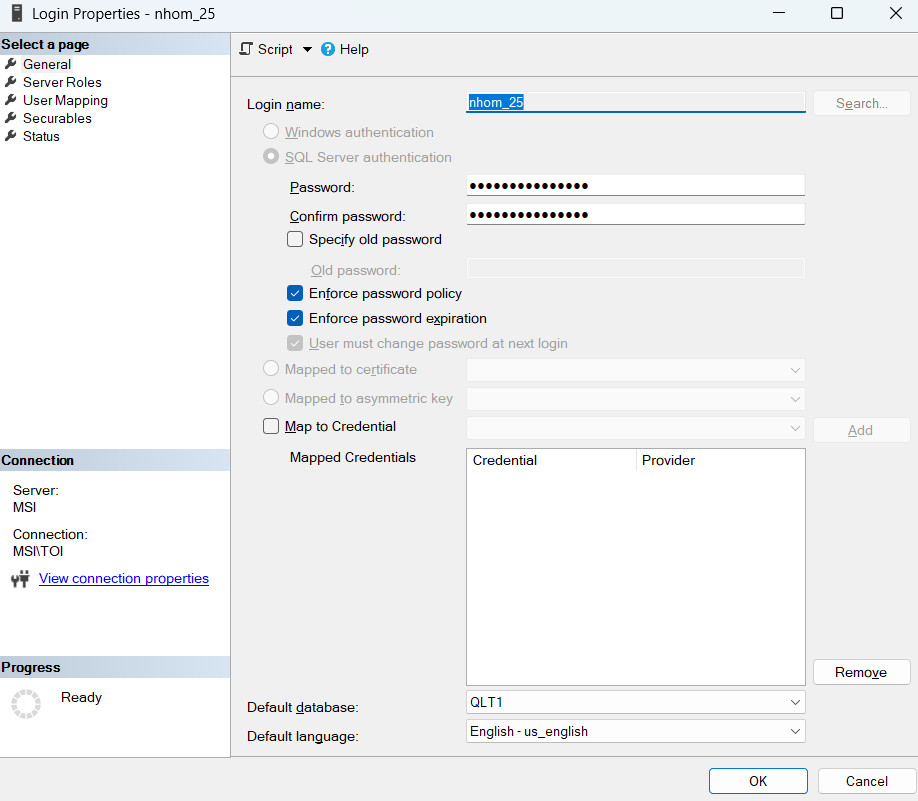


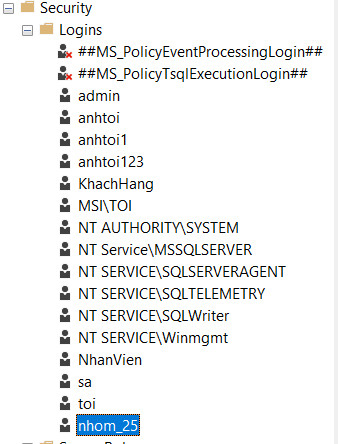
**1.2. Mô hình quan hệ**



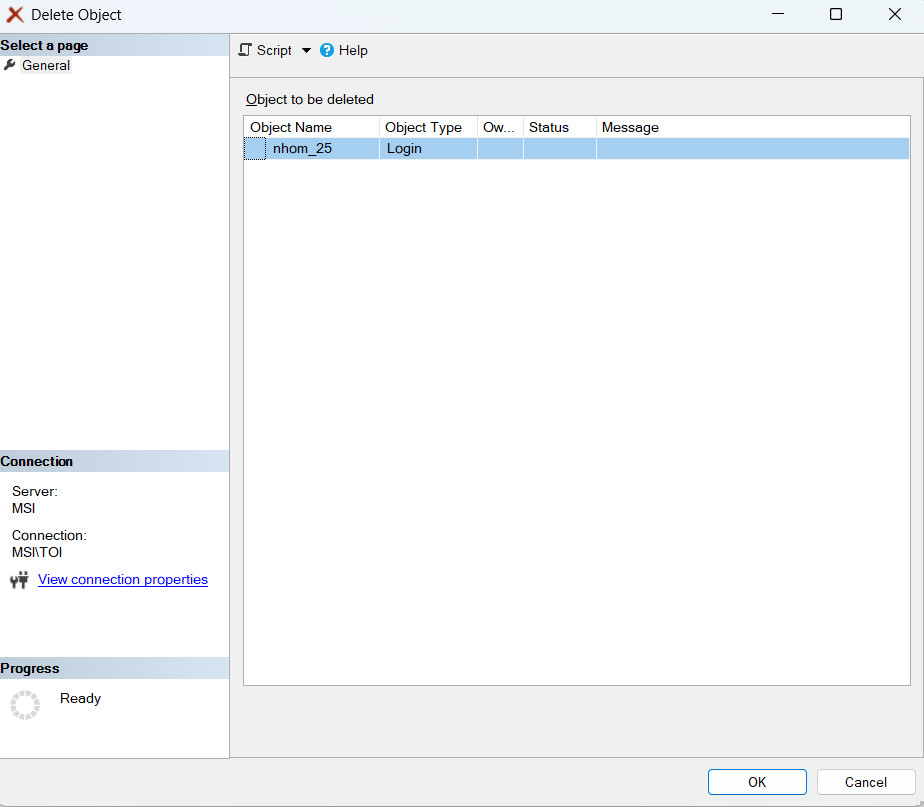
## 1.3. Tạo và xóa tài khoản người dùng đăng nhập quyền SQL server bằng công cụ

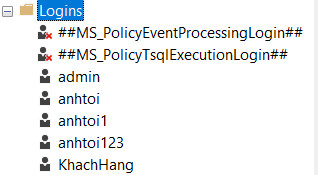
* Tạo người dùng user





* Xóa người dùng





## 1.4. Tạo và xóa tài khoản người dùng đăng nhập quyền SQL server bằng câu lệnh

* **Câu lệnh insert**

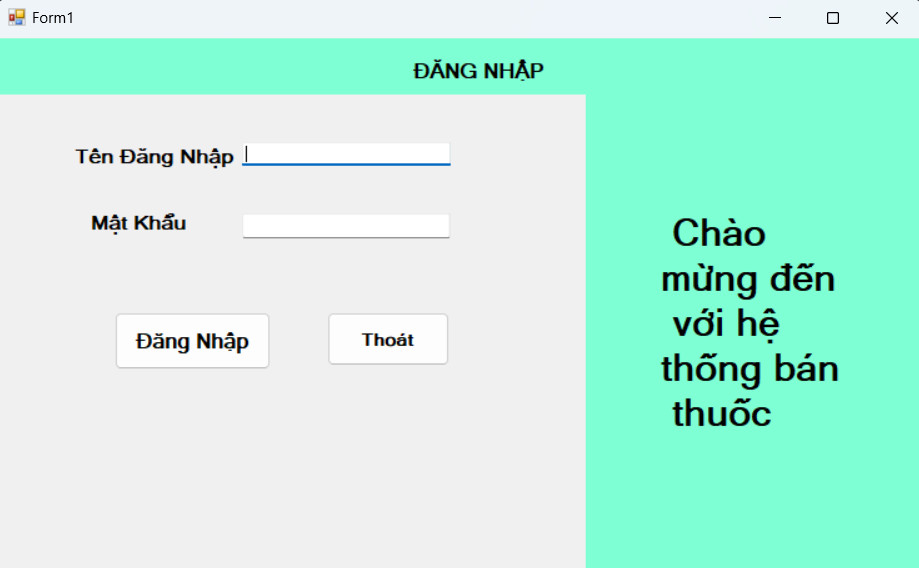
|  |
| --- |
| * INSERT INTO NhanVien (MaNV, TenNV, DienThoai, DiaChi, TaiKhoan, MatKhau, GhiChu)VALUES (11, N'Nguyễn Anh Tới', '0123456789', N'Hà Nội', 'nguyenanhtoi', '123456', N'Nhân viên bán hàng'), |

* **Câu lệnh Xóa**

|  |
| --- |
| DELETE From NhanVien where MaNV = 11 |

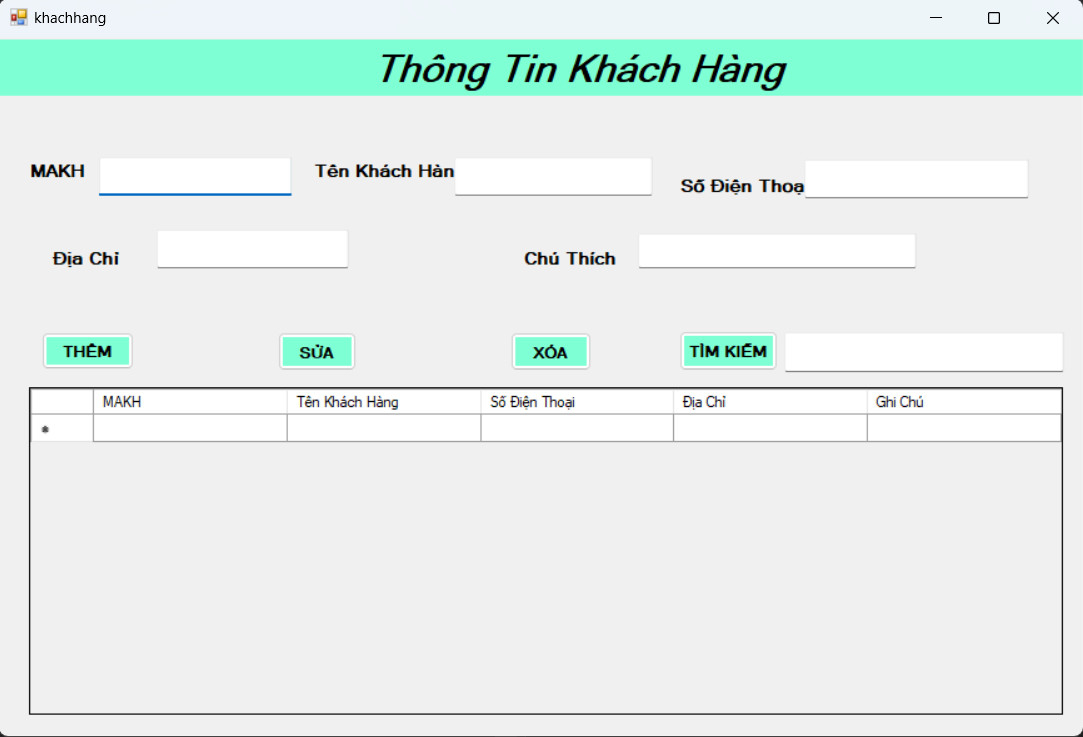
***2. Thiết kế giao diện***

## *2.1. Giao diện chính*



## *2.2. Các giao diện cập nhật dữ liệu*

### ***a. Giao diện cập nhật thông tin Khách hàng***



Mô tả giao diện:

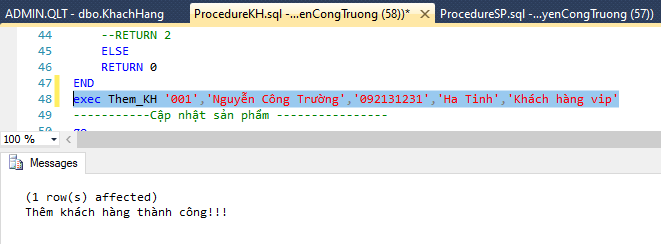
* Form dùng để câp nhật dữ liệu cho bảng KHACH trong cơ sở dữ liệu

Sau đây là phần query và kết quả thêm, sửa, xóa của KhachHang

* Thêm KHÁCH HÀNG

|  |
| --- |
| CREATE PROCEDURE [dbo].[Them\_KH]  @MaKH INT ,  @TenKH NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= dbo.KT\_TonTai\_KH @MaKH  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã khách hàng đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO KhachHang ( MaKH ,TenKH , DienThoai ,DiaChi ,GhiChu)  VAlUES(@MaKH , @TenKH ,@DienThoai , @DiaChi , @GhiChu )  PRINT N'Thêm khách hàng thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_KH]  @MaKhachHang char(10)  AS  BEGIN  IF EXISTS (SELECT MaKH FROM KhachHang WHERE MaKH=@MaKhachHang)  RETURN 1    ELSE  RETURN 0  END |

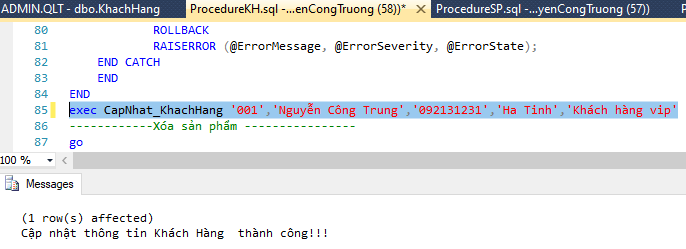
Kết quả:



* Sửa KHÁCH HÀNG

|  |
| --- |
| go  CREATE PROCEDURE [dbo].[CapNhat\_KhachHang]  @MaKH INT ,  @TenKH NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_KH @MaKH  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin khác hàng KHÔNG thành công. Mã khách hàng không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE KhachHang  SET TenKH=@TenKH , DienThoai=@DienThoai ,DiaChi=@DiaChi, GhiChu =@GhiChu  WHERE MaKH=@MaKH  PRINT N'Cập nhật thông tin Khách Hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END |
|  |

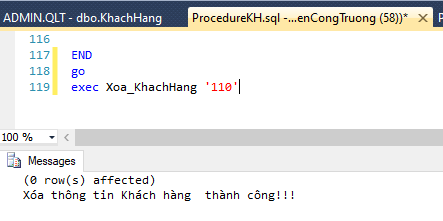
Kết quả:



* Xóa KHÁCH HÀNG

|  |
| --- |
| ------------Xóa khách hàng ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_KhachHang]  @MaHang char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @TenHang nvarchar(50)  EXEC @check=KT\_TonTai\_KH @MaHang, @TenHang  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin khách hàng KHÔNG thành công. Mã khách hàng không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM HangHoa WHERE MaHang=@MaHang  PRINT N'Xóa thông tin Khách hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END  go  exec Xoa\_KhachHang '110' |

Kết quả:



### ***b. Giao diện cập nhật thông tin Sản Phẩm***



Mô tả giao diện:

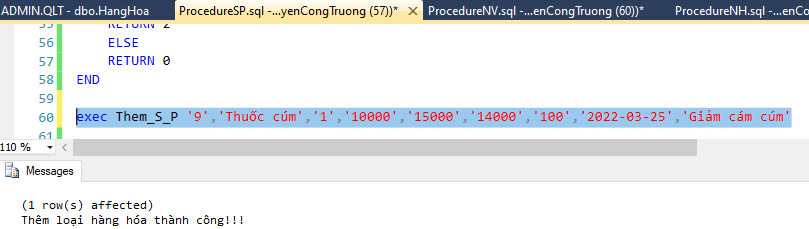
* Form dùng để câp nhật dữ liệu cho bảng HangHoa trong cơ sở dữ liệu
* Đối với mã loại hàng là một hộp lựa chọn, lấy nguồn từ cột mã loại hàng trong bảng HangHoa

Sau đây là phần query và kết quả thêm, sửa, xóa của HangHoa

* Thêm Sản phẩm

|  |
| --- |
| --------Thêm Sản Phẩm -------------  CREATE PROCEDURE [dbo].[Them\_S\_Pham]  @MaHang char(10),  @TenHang NVARCHAR(255),  @MaDVT nvarchar(10),  @GiaMua INT ,  @GiaBan FLOAT,  @GiaBinhQuan FLOAT,  @SoLuongTon INT,  @NgayCapNhat DATE,  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_SPham @MaHang, @TenHang  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã sản phẩm đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE IF @check=2  BEGIN  RAISERROR ( N'Thêm không thành công. Tên sản phẩm đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO HangHoa(MaHang,TenHang,MaDVT,GiaMua,GiaBan,GiaBinhQuan,SoLuongTon,NgayCapNhat,GhiChu)  VAlUES(@MaHang ,@TenHang ,@MaDVT , @GiaMua , @GiaBan, @GiaBinhQuan ,@SoLuongTon , @NgayCapNhat ,@GhiChu)  PRINT N'Thêm loại hàng hóa thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_SPham]  @MaLoaiHH char(10),  @TenLoaiHH nvarchar(50)  AS  BEGIN  IF EXISTS (SELECT MaHang FROM HangHoa WHERE MaHang=@MaLoaiHH)  RETURN 1  ELSE IF EXISTS (SELECT TenHang FROM HangHoa WHERE TenHang=@TenLoaiHH)  RETURN 2  ELSE  RETURN 0  END |

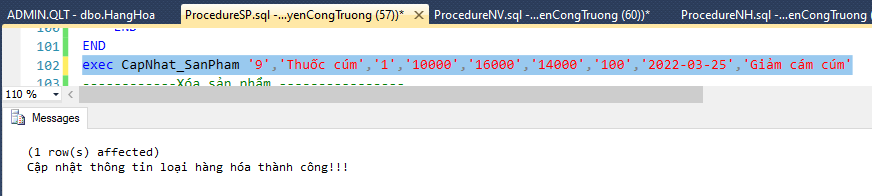
Kết quả:



* Sửa Sản Phẩm

|  |
| --- |
| -----------Cập nhật sản phẩm ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_SanPham]  @MaHang char(10),  @TenHang NVARCHAR(255),  @MaDVT nvarchar(10),  @GiaMua INT ,  @GiaBan FLOAT,  @GiaBinhQuan FLOAT,  @SoLuongTon INT,  @NgayCapNhat DATE,  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_SPham @MaHang, @TenHang  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin loại hàng hóa KHÔNG thành công. Mã loại hàng hóa không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE HangHoa  SET TenHang=@TenHang, MaDVT = @MaDVT, GiaMua= @GiaMua, GiaBan =@GiaBan, GiaBinhQuan =@GiaBinhQuan, SoLuongTon =@SoLuongTon, NgayCapNhat =@NgayCapNhat ,GhiChu=@GhiChu  WHERE MaHang=@MaHang  PRINT N'Cập nhật thông tin loại hàng hóa thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

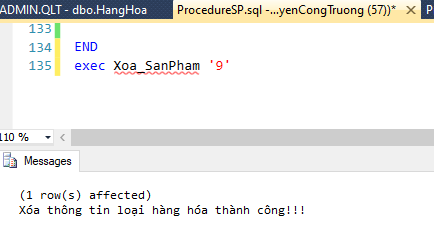
Kết quả:



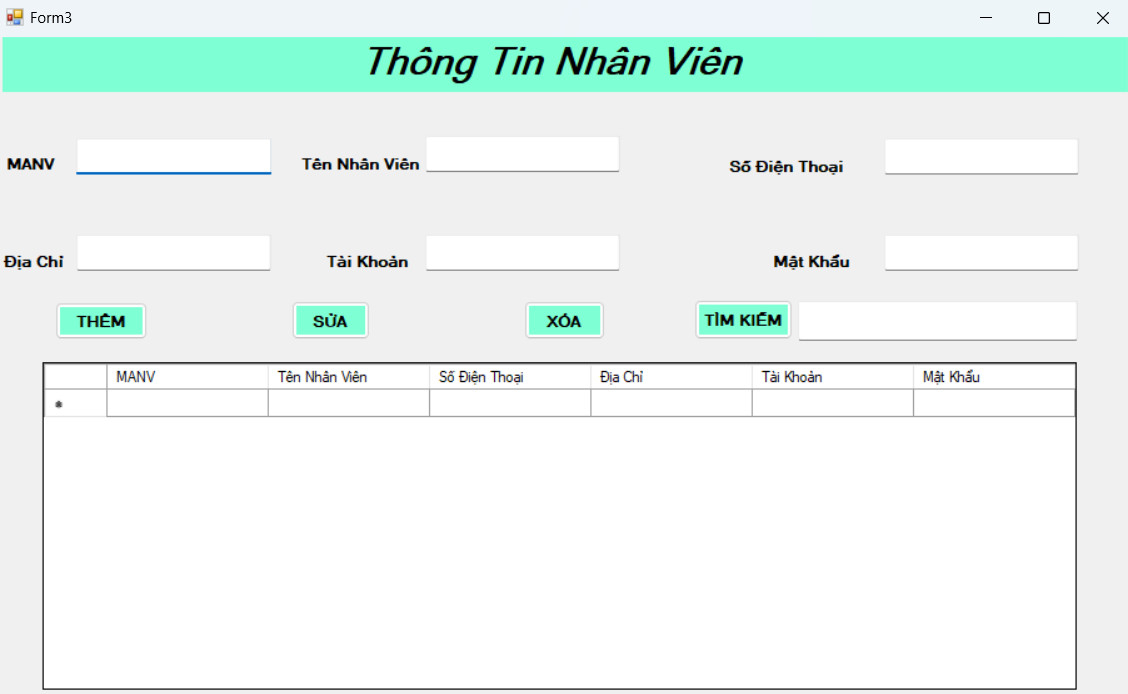
* Xóa Sản Phẩm

|  |
| --- |
| ------------Xóa sản phẩm ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_SanPham ]  @MaHang char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @TenHang nvarchar(50)  EXEC @check=KT\_TonTai\_SPham @MaHang, @TenHang  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin loại hàng hóa KHÔNG thành công. Mã loại hàng hóa không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM HangHoa WHERE MaHang=@MaHang  PRINT N'Xóa thông tin loại hàng hóa thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

Kết quả:



### ***c. Giao diện cập nhật thông tin Nhân Viên***



Mô tả giao diện:

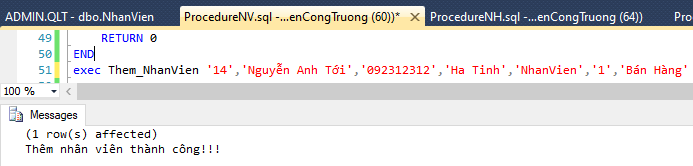
* Form dùng để câp nhật dữ liệu cho bảng NhanVien trong cơ sở dữ liệu

Sau đây là phần query và kết quả thêm, sửa, xóa của NhanVien

* Thêm Nhân Viên

|  |
| --- |
| --------Thêm Nhân viên -------------  CREATE PROCEDURE [dbo].[Them\_NhanVien]  @MaNV INT ,  @TenNV NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @TaiKhoan NVARCHAR(255),  @MatKhau NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_NhanVien @MaNV  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã nhân viên đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO NhanVien (MaNV ,TenNV , DienThoai ,DiaChi ,TaiKhoan , MatKhau , GhiChu )  VAlUES(@MaNV ,@TenNV , @DienThoai ,@DiaChi ,@TaiKhoan , @MatKhau , @GhiChu )  PRINT N'Thêm nhân viên thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_NhanVien]  @MaNV char(10),  @TenNV nvarchar (255)  AS  BEGIN  IF EXISTS (SELECT MaNV FROM NhanVien WHERE MaNV=@MaNV)  RETURN 1  ELSE IF EXISTS (SELECT TenNV FROM NhanVien WHERE TenNV=@TenNV)  RETURN 2  ELSE  RETURN 0  END  exec Them\_NhanVien '14','Nguyễn Anh Tới','092312312','Ha Tinh','NhanVien','1','Bán Hàng' |

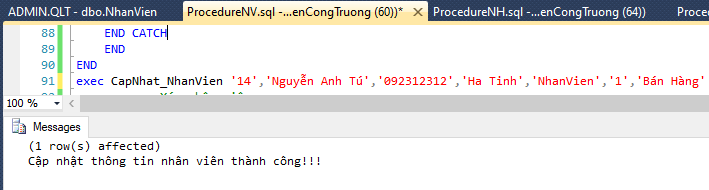
Kết quả:



* Sửa Nhân Viên

|  |
| --- |
| -----------Cập nhật nhân viên ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_NhanVien]  @MaNV INT ,  @TenNV NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @TaiKhoan NVARCHAR(255),  @MatKhau NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_NhanVien @MaNV  IF @check=0  BEGIN  RAISERROR (N'Cập nhật thông tin nhân viên KHÔNG thành công. Mã nhân viên không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE NhanVien  SET TenNV=@TenNV ,DienThoai=@DienThoai ,DiaChi= @DiaChi,TaiKhoan=@TaiKhoan,MatKhau=@MatKhau ,GhiChu=@GhiChu  WHERE MaNV=@MaNV  PRINT N'Cập nhật thông tin nhân viên thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

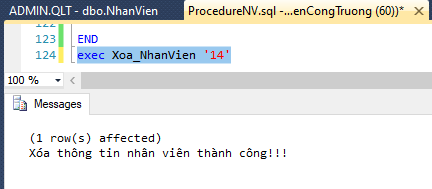
Kết quả:



* Xóa Nhân Viên

|  |
| --- |
| ------------Xóa nhân viên ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_NhanVien]  @MaNV char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @TenNV nvarchar(50)  EXEC @check=KT\_TonTai\_NhanVien @MaNV, @TenNV  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin nhân viên KHÔNG thành công. Mã nhân viên không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM NhanVien WHERE MaNV=@MaNV  PRINT N'Xóa thông tin nhân viên thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

Kết quả:



### ***d. Giao diện cập nhật thông tin Nhà cung cấp***



Mô tả giao diện:

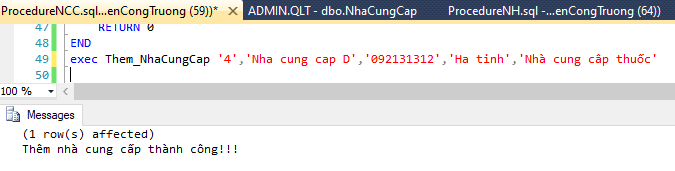
* Form dùng để câp nhật dữ liệu cho bảng NhaCungCap trong cơ sở dữ liệu

Sau đây là phần query và kết quả thêm, sửa, xóa của NhaCungCap

* Thêm nhà cung cấp

|  |
| --- |
| --------Thêm Nhà cung cấp -------------  CREATE PROCEDURE [dbo].[Them\_NhaCungCap]  @MaNCC INT ,  @TenNCC NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_NhaCungCap @MaNCC  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã nhà cung cấp đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO NhaCungCap (MaNCC , TenNCC,DienThoai ,DiaChi ,GhiChu )  VAlUES( @MaNCC , @TenNCC ,@DienThoai ,@DiaChi ,@GhiChu )  PRINT N'Thêm nhà cung cấp thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_NhaCungCap]  @MaNCC char(10),  @TenNCC nvarchar (255)  AS  BEGIN  IF EXISTS (SELECT MaNCC FROM NhaCungCap WHERE MaNCC=@MaNCC)  RETURN 1  ELSE IF EXISTS (SELECT TenNCC FROM NhaCungCap WHERE TenNCC=@TenNCC)  RETURN 2  ELSE  RETURN 0  END |

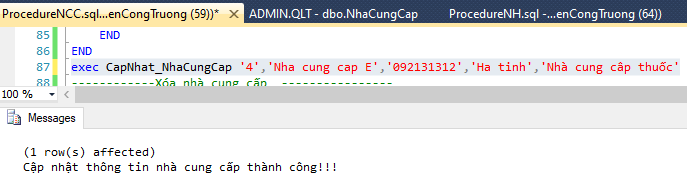
Kết quả:



* Sửa nhà cung cấp

|  |
| --- |
| -----------Cập nhật nhà cung cấp ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_NhaCungCap]  @MaNCC INT ,  @TenNCC NVARCHAR(255),  @DienThoai NVARCHAR(20),  @DiaChi NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_NhaCungCap @MaNCC  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin nhà cung cấp KHÔNG thành công. Mã nhà cung cấp không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE NhaCungCap  SET TenNCC=@TenNCC ,DienThoai=@DienThoai ,DiaChi= @DiaChi ,GhiChu=@GhiChu  WHERE MaNCC=@MaNCC  PRINT N'Cập nhật thông tin nhà cung cấp thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

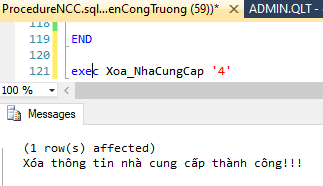
Kết quả:



* Xóa nhà cung cấp

|  |
| --- |
| ------------Xóa nhà cung cấp ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_NhaCungCap]  @MaNCC char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @TenNCC nvarchar(50)  EXEC @check=KT\_TonTai\_NhaCungCap @MaNCC, @TenNCC  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin NHÀ CUNG CẤP KHÔNG thành công. Mã nhà cung cấp không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM NhaCungCap WHERE MaNCC=@MaNCC  PRINT N'Xóa thông tin nhà cung cấp thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

Kết quả:



### ***e. Giao diện cập nhật thông tin Nhập hàng***



Mô tả giao diện:

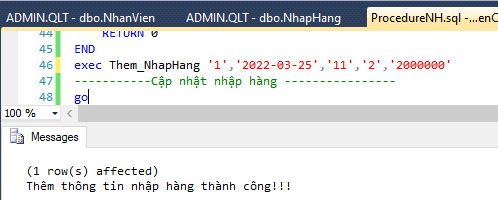
* Form dùng để câp nhật dữ liệu cho bảng NhapHang trong cơ sở dữ liệu

**Sau đây là phần query và kết quả thêm, sửa, xóa của NhapHang**

* Thêm nhập hàng

|  |
| --- |
| --------Thêm Nhập hàng -------------  CREATE PROCEDURE [dbo].[Them\_NhapHang]  @MaNhap INT,  @NgayNhap DATE,  @MaNV INT,  @MaNCC INT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_NH @MaNhap  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã nhập hàng đã tồn tại!!!', 16, 1)  ROLLBACK  END    ELSE  BEGIN  BEGIN TRY  INSERT INTO NhapHang (MaNhap ,NgayNhap ,MaNV ,MaNCC,TongTien)  VAlUES(@MaNhap ,@NgayNhap ,@MaNV ,@MaNCC,@TongTien )  PRINT N'Thêm thông tin nhập hàng thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_NH]  @MaNhap char(10)  AS  BEGIN  IF EXISTS (SELECT MaNhap FROM NhapHang WHERE MaNhap =@MaNhap)  RETURN 1  ELSE  RETURN 0  END |

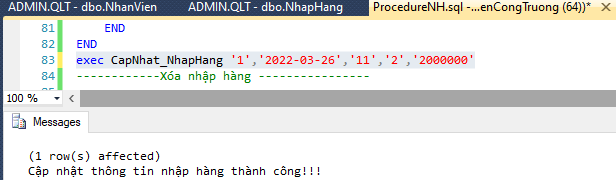
Kết quả:



* Sửa nhập hàng

|  |
| --- |
| -----------Cập nhật nhập hàng ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_NhapHang]  @MaNhap INT,  @NgayNhap DATE,  @MaNV INT,  @MaNCC INT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_NH @MaNhap  IF @check=0  BEGIN  RAISERROR (N'Cập nhật thông tin nhập hàng KHÔNG thành công. Mã nhập hàng không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE NhapHang  SET NgayNhap=@NgayNhap ,MaNV=@MaNV ,MaNCC=@MaNCC,TongTien=@TongTien  WHERE MaNhap=@MaNhap  PRINT N'Cập nhật thông tin nhập hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

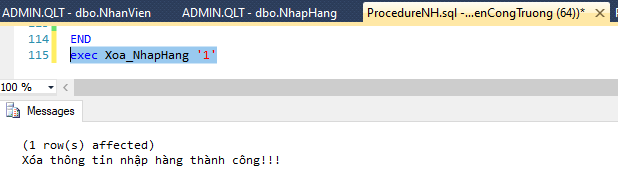
Kết quả:



* Xóa nhập hàng

|  |
| --- |
| ------------Xóa nhập hàng ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_NhapHang]  @MaNhap char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_NH @MaNhap  IF @check=0  BEGIN  RAISERROR ( N'Xóa thông tin nhập hàng KHÔNG thành công. Mã nhập hàng không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM NhapHang WHERE MaNhap =@MaNhap  PRINT N'Xóa thông tin nhập hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

Kết quả:



***f. Giao diện cập nhật thông tin Xuất hàng***



Mô tả giao diện:

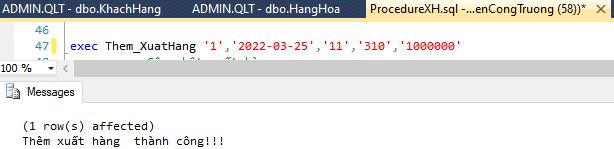
* Form dùng để câp nhật dữ liệu cho bảng XuatHang trong cơ sở dữ liệu

**Sau đây là phần query và kết quả thêm, sửa, xóa của XuatHang**

* Thêm Xuất hàng

|  |
| --- |
| CREATE PROCEDURE [dbo].[Them\_XuatHang]  @MaXuat INT ,  @NgayXuat DATE,  @MaNV INT,  @MaKH INT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_XH @MaXuat  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã xuất hàng đã tồn tại!!!', 16, 1)  ROLLBACK  END    ELSE  BEGIN  BEGIN TRY  INSERT INTO XuatHang (MaXuat ,NgayXuat ,MaNV ,MaKH,TongTien)  VAlUES(@MaXuat ,@NgayXuat ,@MaNV ,@MaKH,@TongTien )  PRINT N'Thêm xuất hàng thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_XH]  @MaXuat char(10)  AS  BEGIN  IF EXISTS (SELECT MaXuat FROM XuatHang WHERE MaXuat=@MaXuat)  RETURN 1  ELSE  RETURN 0  END |

Kết quả:



* Sửa Xuất hàng

|  |
| --- |
| -----------Cập nhật xuất hàng ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_XuatHang]  @MaXuat INT ,  @NgayXuat DATE,  @MaNV INT,  @MaKH INT,  @TongTien FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_XH @MaXuat  IF @check=0  BEGIN  RAISERROR (N'Cập nhật thông tin đơn vị tính KHÔNG thành công. Mã đơn vị tính không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE XuatHang  SET NgayXuat=@NgayXuat ,MaNV=@MaNV ,MaKH=@MaKH,TongTien=@TongTien  WHERE MaXuat=@MaXuat  PRINT N'Cập nhật thông tin xuất hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

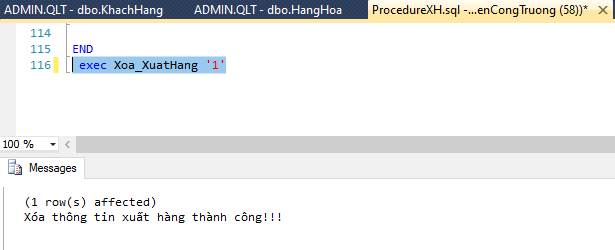
Kết quả:



* Xóa Xuất hàng

|  |
| --- |
| ------------Xóa xuất hàng ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_XuatHang]  @MaXuat char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_XH @MaXuat  IF @check=0  BEGIN  RAISERROR ( N'Xóa thông tin xuất hàng KHÔNG thành công. Mã xuất hàng không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM XuatHang WHERE MaXuat =@MaXuat  PRINT N'Xóa thông tin xuất hàng thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

Kết quả:



### ***g. Giao diện cập nhật thông tin đơn vị tính***

### 

Mô tả giao diện:

- Form dùng để cập nhật dữ liệu cho bảng DonViTinh trong cơ sở dữ liệu

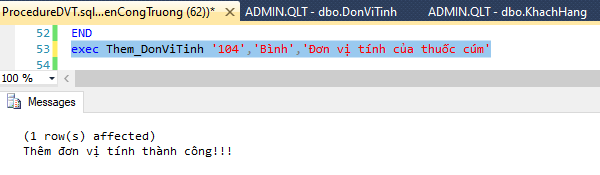
- Đối với mã bộ phận là một lựa chọn, lấy nguồn từ cột mã bộ phận trong bảng DonViTinh

Sau đây là phần query và kết quả thêm, sửa, xóa của DonViTinh

* Thêm đơn vị tính

|  |
| --- |
| --------Thêm đơn vị tính -------------  CREATE PROCEDURE [dbo].[Them\_DonViTinh]  @MaDVT INT ,  @TenDVT NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_DVT @MaDVT, @TenDVT  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã đơn vị tính đã tồn tại!!!', 16, 1)  ROLLBACK  END    ELSE IF @check=2  BEGIN  RAISERROR ( N'Thêm không thành công. Tên đơn vị tính đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO DonViTinh (MaDVT ,TenDVT ,GhiChu)  VAlUES(@MaDVT , @TenDVT ,@Ghichu )  PRINT N'Thêm đơn vị tính thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_DVT]  @MaDVT char(10),  @TenDVT nvarchar (50)  AS  BEGIN  IF EXISTS (SELECT MaDVT FROM DonViTinh WHERE MaDVT=@MaDVT)  RETURN 1  ELSE IF EXISTS (SELECT TenDVT FROM DonViTinh WHERE TenDVT =@TenDVT )  RETURN 2  ELSE  RETURN 0  END |

Kết quả:



* Sửa đơn vị tính

|  |
| --- |
| -----------Cập nhật đơn vị tính ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_DVT]  @MaDVT INT ,  @TenDVT NVARCHAR(255),  @GhiChu NVARCHAR(255)  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_DVT @MaDVT, @TenDVT  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin đơn vị tính KHÔNG thành công. Mã đơn vị tính không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE DonViTinh  SET TenDVT= @TenDVT ,GhiChu = @GhiChu  WHERE MaDVT=@MaDVT  PRINT N'Cập nhật thông tin đơn vị tính thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END |

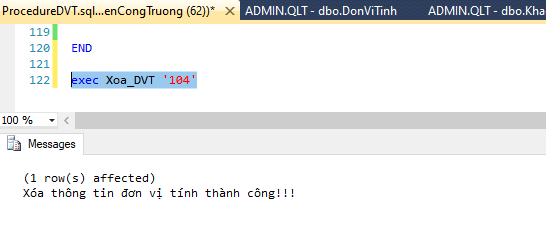
Kết quả:



* Xóa đơn vị tính

|  |
| --- |
| ------------Xóa đơn vị tính ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_DVT]  @MaDVT char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @TenDVT NVARchar(50)  EXEC @check=KT\_TonTai\_DVT @MaDVT, @TenDVT  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin đơn vị tính KHÔNG thành công. Mã đơn vị tính không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM DonViTinh WHERE MaDVT =@MaDVT  PRINT N'Xóa thông tin đơn vị tính thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END |

Kết quả:



### ***h. Giao diện Bảng giá***

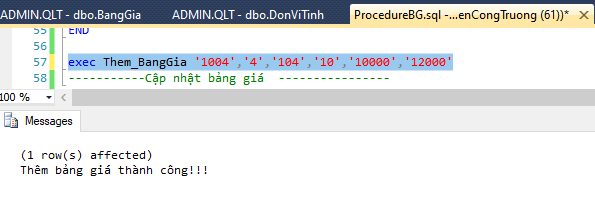


Sau đây là phần query và kết quả Bảng giá

* Thêm bảng giá

|  |
| --- |
| --------Thêm Bảng giá -------------  CREATE PROCEDURE [dbo].[Them\_BangGia]  @MaBangGia INT ,  @MaHang INT,  @MaDVT INT,  @SoLuongDVT FLOAT,  @GiaMua FLOAT,  @GiaBan FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check= KT\_TonTai\_BangGia @MaBangGia, @MaHang  IF @check=1  BEGIN  RAISERROR ( N'Thêm không thành công. Mã bảng giá đã tồn tại!!!', 16, 1)  ROLLBACK  END    ELSE IF @check=2  BEGIN  RAISERROR ( N'Thêm không thành công. Mã sản phẩm đã tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  INSERT INTO BangGia (MaBangGia ,MaHang,MaDVT , SoLuongDVT , GiaMua ,GiaBan )  VAlUES(@MaBangGia ,@MaHang,@MaDVT , @SoLuongDVT , @GiaMua ,@GiaBan )  PRINT N'Thêm bảng giá thành công!!!'  COMMIT TRAN  END Try  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCh  END  END  go  CREATE PROCEDURE [dbo].[KT\_TonTai\_BangGia]  @MaBangGia char(10),  @MaHang char (10)  AS  BEGIN  IF EXISTS (SELECT MaBangGia FROM BangGia WHERE MaBangGia=@MaBangGia)  RETURN 1  ELSE IF EXISTS (SELECT MaHang FROM BangGia WHERE MaHang=@MaHang)  RETURN 2  ELSE  RETURN 0  END |

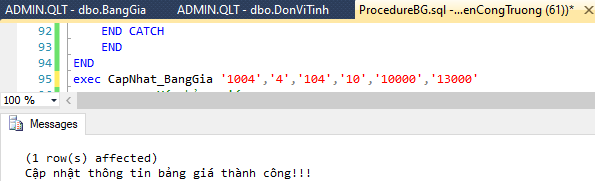
* Kết quả:



* Sửa bảng giá

|  |
| --- |
| -----------Cập nhật bảng giá ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_BangGia]  @MaBangGia INT ,  @MaHang INT,  @MaDVT INT,  @SoLuongDVT FLOAT,  @GiaMua FLOAT,  @GiaBan FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_BangGia @MaBangGia,@MaHang  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin bảng giá KHÔNG thành công. Mã bảng giá không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE BangGia  SET MaDVT=@MaDVT ,SoLuongDVT= @SoLuongDVT ,GiaMua = @GiaMua ,GiaBan=@GiaBan  WHERE MaBangGia=@MaBangGia  PRINT N'Cập nhật thông tin bảng giá thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |
|  |

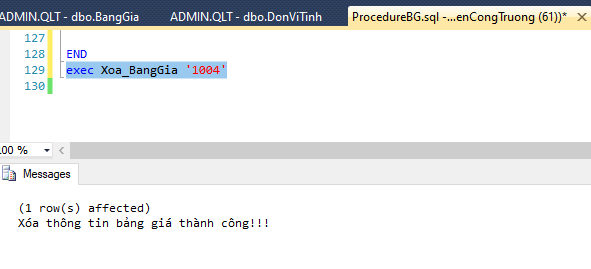
* Kết quả



* Xóa bảng giá

|  |
| --- |
| ------------Xóa bảng giá ----------------  go  CREATE PROCEDURE [dbo].[Xoa\_BangGia]  @MaBangGia char(10)  AS  BEGIN  BEGIN TRAN  DECLARE @check int, @MaHang char(10)  EXEC @check=KT\_TonTai\_BangGia @MaBangGia, @MaHang  IF @check=0 or @check=2  BEGIN  RAISERROR ( N'Xóa thông tin bảng giá KHÔNG thành công. Mã bảng giá không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  DELETE FROM BangGia WHERE MaBangGia =@MaBangGia  PRINT N'Xóa thông tin bảng giá thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

* Kết quả



## *2.3. Các tình huống tranh chấp*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Chức năng 1** | **Người dùng** | **Chức năng 2** | **Người dùng** | **Lỗi Tranh chấp** | **Mô tả** |
| 1 | tìm kiếm sản phẩm theo tên. | Khách hàng | Thêm sản phẩm | Nhân viên bán | Phantom Read | GT1: (User = khách hàng): khách hàng tìm kiếm sản phẩm theo tên GT1: Nhân viên thêm sản phẩm 2(Bảng HangHoa ) |
| 2 | Cập nhật thông tin | Nhân viên bán hàng | Cập nhật thông tin | Nhân viên bán hàng | Cycle Deadlock | - GT1: Nhân viên bán hàng cập nhật tên  - GT1: Nhân viên bán hàng cập nhật số điện thoại |
| 3 | Tạo thông tin nhập hàng | Nhân viên | Cập nhật thông tin nhập hàng | Nhân viên | Unrepeatable Read | - GT1: Tạo danh thông tin nhập hàng  - GT1: Cập nhật thông tin nhập hàng |

## *2.4. Cài đặt các tình huống tranh chấp*

**ERR01: Phantom Read**

T1 (User = khách hàng): khách hàng tìm kiếm sản phẩm theo tên.

T2 (User = Nhân viên): thêm một sản phẩm mới cùng tên với sản phẩm khách hàng đang tìm

Input: Tên sản phẩm

Output: Số lượng tìm được và danh sách phẩm tương ứng

Input: Mã chi nhánh, mã sản phẩm, số lượng

Output: Thêm sản phẩm thành công

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

BEGIN TRAN;

**B1: -- Truy vấn sản phẩm mà khách hàng đang tìm kiếm**

SELECT \* FROM HangHoa1 WHERE TenHang = TenHang;

HangHoa1: là một view để xem tên sản phẩm

**B1: --Kiểm tra xem sản phẩm mới đã tồn tại hay chưa**

go CREATE PROCEDURE [dbo].[KT\_TonTai\_SPham]

@MaLoaiHH char(10),

@TenLoaiHH nvarchar(50)

AS BEGIN IF EXISTS (SELECT MaHang FROM HangHoa WHERE MaHang=@MaLoaiHH) RETURN 1

ELSE

IF EXISTS (SELECT TenHang FROM HangHoa WHERE TenHang=@TenLoaiHH) RETURN 2

ELSE

RETURN 0

**B2:-- Nếu sản phẩm đã tồn tại, hủy giao dịch**

BEGIN TRAN

DECLARE @check int

EXEC @check=KT\_TonTai\_SPham @MaHang, @TenHang

IF @check=1

BEGIN

RAISERROR ( N'Mã sản phẩm đã tồn tại!!!', 16, 1)

ROLLBACK

END

ELSE IF @check=2

BEGIN

RAISERROR ( N'Tên sản phẩm đã tồn tại!!!', 16, 1)

ROLLBACK

END

**B3**:-- **Nếu sản phẩm chưa tồn tại, thêm sản phẩm mới vào cơ sở dữ liệu**

BEGIN

INSERT INTO Them\_S\_Pham (MaHang, TenHang,MaDVT, GiaMua, GiaBan, GiaBinhQuan, SoLuongTon, NgayCapNhat, GhiChu)

VAlUES(@MaHang ,@TenHang ,@MaDVT , @GiaMua , @GiaBan, @GiaBinhQuan ,@SoLuongTon , @NgayCapNhat ,@GhiChu)

PRINT N'Thêm loại hàng hóa thành công!!!'

COMMIT TRAN

**ERR02: Cycle Deadlock**

T1 (User = nhân viên): cập nhật thông tin của nhân viên.

T2 (User = nhân viên ): cập nhật thông tin của nhân viên.

Input: Tên nhân viên ,điện thoại ,Đia chỉ,Tài Khoản,Mật Khẩu,Ghi Chú

Output: Deadlock

Input: Tên nhân viên ,điện thoại ,Đia chỉ,Tài Khoản,Mật Khẩu,Ghi Chú

Output: Deadlock

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

BEGIN TRAN;

**B1:-- T1 kiểm tra thông tin nhân viên trước khi cập nhật**

go

CREATE PROCEDURE [dbo].[KT\_TonTai\_NhanVien]

@MaNV char(10),

AS

BEGIN

IF EXISTS (SELECT MaNV FROM NhanVien WHERE MaNV=@MaNV)

RETURN 1

ELSE

RETURN 0

ENDBEGIN TRAN

DECLARE @check int

EXEC @check= KT\_TonTai\_NhanVien @MaNV

IF @check=1

BEGIN

RAISERROR ( N'Mã nhân viên đã tồn tại!!!', 16, 1)

ROLLBACK

END

**B2:-- T1 cập nhật thông tin của nhân viên**

UPDATE NhanVien

SET TenNV=@TenNV

WHERE MaNV=@MaNV

PRINT N'Cập nhật thông tin nhân viên thành công!!!'

COMMIT TRAN

END

**B1:-- T2 kiểm tra thông tin nhân viên trước khi cập nhật**

go

CREATE PROCEDURE [dbo].[KT\_TonTai\_NhanVien]

@MaNV char(10),

AS

BEGIN

IF EXISTS (SELECT MaNV FROM NhanVien WHERE MaNV=@MaNV)

RETURN 1

ELSE

RETURN 0

ENDBEGIN TRAN

DECLARE @check int

EXEC @check= KT\_TonTai\_NhanVien @MaNV

IF @check=1

BEGIN

RAISERROR ( N'Mã nhân viên đã tồn tại!!!', 16, 1)

ROLLBACK

END

**B2:-- T2 cập nhật thông tin của nhân viên**

UPDATE NhanVien

SET DienThoai=@DienThoai

WHERE MaNV=@MaNV

PRINT N'Cập nhật thông tin nhân viên thành công!!!'

COMMIT TRAN

END

COMMIT;

**ERR03: Unrepeatable Read**

T1 (User = nhân viên): Tạo thôn tin nhập hàng

T2 (User = nhân viên ): cập nhật thông tin nhập hàng.

B1 -- T1 Tạo thông tin nhập hàng

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

BEGIN TRAN;

CREATE PROCEDURE [dbo].[Them\_NhapHang]

@MaNhap INT,

@NgayNhap DATE,

@MaNV INT,

@MaNCC INT,

@TongTien FLOAT

AS

BEGIN

BEGIN TRAN

DECLARE @check int

EXEC @check= KT\_TonTai\_NH @MaNhap

IF @check=1

BEGIN

RAISERROR ( N'Thêm không thành công. Mã nhập hàng đã tồn tại!!!', 16, 1)

ELSE

BEGIN

BEGIN TRY

INSERT INTO NhapHang (MaNhap ,NgayNhap ,MaNV ,MaNCC,TongTien)

VAlUES(@MaNhap ,@NgayNhap ,@MaNV ,@MaNCC,@TongTien )

PRINT N'Thêm thông tin nhập hàng thành công!!!'

COMMIT TRAN

go

CREATE PROCEDURE [dbo].[KT\_TonTai\_NH]

@MaNhap char(10)

AS

BEGIN

IF EXISTS (SELECT MaNhap FROM NhapHang WHERE MaNhap =@MaNhap)

RETURN 1

ELSE

RETURN 0

END

**B1-- T2 Kiểm tra thông tin nhập hàng**

go

CREATE PROCEDURE [dbo].[CapNhat\_NhapHang]

@MaNhap INT,

@NgayNhap DATE,

@MaNV INT,

@MaNCC INT,

@TongTien FLOAT

AS

BEGIN

BEGIN TRAN

DECLARE @check int

EXEC @check= KT\_TonTai\_NH @MaNhap

IF @check=0

BEGIN

RAISERROR (N'Mã nhập hàng không tồn tại!!!', 16, 1)

ROLLBACK

END

**B2-- T2 Cập nhật thông tin nhập hàng**

BEGIN

UPDATE NhapHang

SET NgayNhap=@NgayNhap ,MaNV=@MaNV ,MaNCC=@MaNCC,TongTien=@TongTien

WHERE MaNhap=@MaNhap

PRINT N'Cập nhật thông tin nhập hàng thành công!!!'

COMMIT TRAN

END TRY

BEGIN CATCH

END

-- Nếu T2 thực hiện trước T1, T1 sẽ không thể đọc được thông tin mới của T2, nếu T1 thực hiện đọc lại dữ liệu, nó sẽ đọc được dữ liệu mới đã được cập nhật bởi T2 và không bị Unrepeatable Read.

# **3. Store Procedure, Funtion, Trigger**

## 3.1. Store Procedure

Stored procedure(SP) là tập hợp một hoặc nhiều câu lệnh T-SQL thành một nhóm đơn vị xử lý logic và được lưu trữ trên Database Server. SP được chuẩn bị sẵn có thể lưu, vì vậy mã này có thể được sử dụng lại nhiều lần.

Vì vậy, một truy vấn SQL có thể viết đi viết lại nhiều lần, hãy lưu nó dưới dạng một thủ tục được lưu trữ và sau đó chỉ cần gọi nó để thực thi nó.

Có thể truyền tham số cho thủ tục được lưu trữ để thủ tục được lưu trữ có thể hoạt động dựa trên (các) giá trị tham số được truyền.

* Cú pháp:

CREATE PROCEDURE procedure\_name

AS

sql\_statement

GO;

* Thực hiện một thủ tục lưu trữ

EXEC procedure\_name;

**Các loại Store Procedure bao gồm:**

Stored Procedure System: Là các SP được xây dựng sẵn bởi nhà sản xuất cơ sở dữ liệu. Chúng có thể được sử dụng để thực hiện các tác vụ quản trị cơ sở dữ liệu như tạo, xóa hoặc sao lưu cơ sở dữ liệu.

Stored Procedure Extended: Là các SP được mở rộng bằng cách sử dụng các lệnh SQL hoặc các công cụ khác để thực hiện các tác vụ phức tạp.

Stored Procedure User-defined: Là các SP được tạo bởi người dùng để thực hiện các tác vụ cụ thể trên cơ sở dữ liệu. Chúng được tạo và lưu trữ trong cơ sở dữ liệu và có thể được gọi từ các ứng dụng hoặc SP khác.

**Các ưu điểm của Store Procedure bao gồm:**

Tăng hiệu suất: SP được biên dịch một lần và lưu trữ trong cơ sở dữ liệu, giúp giảm thời gian thực thi và tăng hiệu suất của hệ thống.

Tính bảo mật: Các quyền truy cập vào cơ sở dữ liệu có thể được kiểm soát thông qua việc sử dụng SP.

Tính đồng nhất: SP giúp đảm bảo tính đồng nhất của dữ liệu và hạn chế sự xuất hiện của lỗi trong cơ sở dữ liệu.

**Các khuyết điểm của Store Procedure là:**

Khó bảo trì: Khi một SP được tạo ra và lưu trữ trong cơ sở dữ liệu, nó trở nên khó khăn hơn để thực hiện các thay đổi hoặc bảo trì.

Khó để kiểm tra: Các SP cần được kiểm tra thủ công để đảm bảo tính đúng đắn và hiệu quả của chúng.

Khó để phát triển: Việc phát triển một SP có thể tốn nhiều thời gian hơn so với việc sử dụng các câu lệnh SQL.

* Ví dụ 1 code Store Procedure trên CSDL

|  |
| --- |
| -----------Cập nhật bảng giá ----------------  go  CREATE PROCEDURE [dbo].[CapNhat\_BangGia]  @MaBangGia INT ,  @MaHang INT,  @MaDVT INT,  @SoLuongDVT FLOAT,  @GiaMua FLOAT,  @GiaBan FLOAT  AS  BEGIN  BEGIN TRAN  DECLARE @check int  EXEC @check=KT\_TonTai\_BangGia @MaBangGia,@MaHang  IF @check=0 or @check=2  BEGIN  RAISERROR (N'Cập nhật thông tin bảng giá KHÔNG thành công. Mã bảng giá không tồn tại!!!', 16, 1)  ROLLBACK  END  ELSE  BEGIN  BEGIN TRY  UPDATE BangGia  SET MaDVT=@MaDVT ,SoLuongDVT= @SoLuongDVT ,GiaMua = @GiaMua ,GiaBan=@GiaBan  WHERE MaBangGia=@MaBangGia  PRINT N'Cập nhật thông tin bảng giá thành công!!!'  COMMIT TRAN  END TRY  BEGIN CATCH  DECLARE @ErrorSeverity INT, @ErrorState INT, @ErrorMessage NVARCHAR(100);  SELECT @ErrorSeverity = ERROR\_SEVERITY(), @ErrorState = ERROR\_STATE(), @ErrorMessage = ERROR\_MESSAGE();  ROLLBACK  RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END  END |

## 3.2. Function

Function (Hàm) là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh được nhóm lại với nhau và được tạo ra với mục đích sử dụng lại. Trong SQL Server, hàm được lưu trữ và bạn có thể truyền các tham số vào cũng như trả về các giá trị.

**Đặc điểm:**

Luôn trả về giá trị

- Có 2 loại Function:

✓ Hàm trả về giá trị kiểu vô hướng (Scalar valued Function)

✓ Hàm trả về giá trị kiểu bảng (Table valued Function)

### 3.2.1. Scalar valued Function:

**- Cú pháp hàm:**

create function Tenham(@Thambien1 Kieudl1,,,,)

returns kieudltrave

as

begin --bắt đầu khối lệnh

declare @bien kieudltrave --Khai báo biến cục bộ

-- Xử lý thay đổi trên hàm

return @bien

end --kết thúc khối lệnh

- Gọi hàm: select dbo.Tenham(doiso1,doiso2, , ,)

Các hàm vô hướng hoạt động trên một giá trị duy nhất và trả về một giá trị duy nhất. Dưới đây là danh sách một số hàm SQL Server Scalar

|  |  |
| --- | --- |
| **Scalar Function(Hàm vô hướng)** | Sự miêu tả |
| abs(-10,67) | Điều này trả về một số tuyệt đối của số đã cho có nghĩa là 10,67. |
| rand(10) | Điều này sẽ tạo ra một số ngẫu nhiên gồm 10 ký tự. |
| round(17.56719,3) | Điều này sẽ làm tròn số đã cho đến 3 chữ số thập phân có nghĩa là 17,567 |
| upper ('dotnet') | Điều này sẽ trả về chữ hoa của chuỗi đã cho có nghĩa là 'DOTNET' |
| lower ('DOTNET') | Điều này sẽ trả về chữ thường của chuỗi đã cho có nghĩa là 'dotnet' |
| ltrim(' dotnet') | Thao tác này sẽ xóa khoảng trắng ở phía bên trái của chuỗi 'dotnet'. |
| convert (int, 15,56) | Điều này sẽ chuyển đổi giá trị float đã cho thành số nguyên có nghĩa là 15. |

* Chức năng tổng hợp

Các hàm tổng hợp hoạt động trên một tập hợp các giá trị và trả về một giá trị duy nhất. Dưới đây là danh sách một số hàm SQL Server Aggregate hữu ích.

|  |  |
| --- | --- |
| Chức năng tổng hợp | Sự miêu tả |
| Max() | Điều này trả về giá trị tối đa từ một tập hợp các giá trị. |
| Min() | Điều này trả về giá trị tối thiểu từ một tập hợp các giá trị. |
| Avg() | Điều này trả về giá trị trung bình của tất cả các giá trị trong một bộ sưu tập. |
| Count() | Điều này không trả về số lượng từ một tập hợp các giá trị. |

### 3.2.2. Tạo Function trả về giá trị loại Table-valued

Function Table-valued có 2 loại:

Hàm giá trị bảng đơn giản: Trả về bảng, là kết quả của một câu lệnh SELECT đơn

Hàm giá trị bảng đa câu lệnh: Trả về bảng, là kết quả của nhiều câu lệnh

a) Hàm giá trị bảng đơn giản

CREATE FUNCTION <Tên function>

([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], …,[...]])

RETURNS TABLE

[WITH ENCRYPTION]

[AS]

RETURN <Câu lệnh SQL>

END

Lưu ý Hàm giá trị bảng đơn còn được gọi là hàm giá trị bảng nội tuyến. Có thể được dùng trong câu lệnh truy vấn thay thế cho tên bảng hoặc tên View.

b) Hàm giá trị bảng đa câu lệnh

CREATE FUNCTION <Tên function>

([@<tên tham số> <kiểu dữ liệu> [= <giá trị mặc định>], …,[...]])

RETURNS @<tên biến trả về> TABLE (<tên cột 1> <kiểu dữ liệu> [tùy chọn thuộc tính], ..., <tên cột n> <kiểu dữ liệu> [tùy chọn thuộc tính])

[AS]

BEGIN

<Câu lệnh SQL>

RETURN

END

**- Xóa hàm:**

Drop function tenham

✓ Hoặc: click chuột phải trên tên hàm -> Delete

**- Sửa lại nội dung hàm:**

Alter function Tenham... -- Các thay đổi nội dung hàm

**2. Sự khác biệt cơ bản giữa Stored Procedure và Function**

Thủ tục lưu trữ có thể trả về giá trị zero, một hoặc nhiều giá trị. Trong khi hàm phải trả về một giá trị duy nhất (có thể là bảng).

Các hàm chỉ có thể có các tham số đầu vào cho nó trong khi thủ tục lưu trữ có thể có các tham số đầu vào hoặc đầu ra.

Hàm có thể được gọi từ thủ tục lưu trữ trong khi thủ tục lưu trữ không thể được gọi từ hàm.

**3. Sự khác biệt nâng cao giữa Stored Procedure và Function**

Các thủ tục không thể được sử dụng trong câu lệnh SELECT trong khi hàm có thể được nhúng trong câu lệnh SELECT. Bởi vì một thủ tục có thể trả về nhiều tập kết quả nên nó không phù hợp để sử dụng trong câu lệnh SELECT.

Các thủ tục lưu trữ không thể được sử dụng trong các câu lệnh WHERE / HAVING / SELECT trong khi hàm thì có thể.

Một ngoại lệ có thể được xử lý bằng try-catch trong thủ tục lưu trữ, đối với hàm thì không thể.

Có thể sử dụng Transactions trong thủ tục lưu trữ, với hàm thì không thể.

* Ví dụ 1 code Function trên CSDL, theo loại Table-valued function

|  |
| --- |
| --Tạo hàm đọc danh sách nhân viên theo mã nhân viên truyền vào  CREATE FUNCTION Danh\_sach\_nhan\_vien(@MaNhanVien NVARCHAR(30))  RETURNS TABLE  AS  RETURN  (  SELECT \*  FROM NHANVIEN  WHERE CONVERT(NVARCHAR(30), MaNV) = @MaNhanVien  )  GO  --Gọi hàm  SELECT \* FROM Danh\_sach\_nhan\_vien('11') |

## 3.3. Trigger

**Khái niệm**

Lệnh CREATE TRIGGER dùng để tạo Trigger trong CSDL.

Trigger là một thủ tục nội tại (Stored Procedure) đặc biệt, được thực thi tự động mỗi khi có một câu lệnh INSERT/ UPDATE/ DELETE làm thay đổi dữ liệu trên bảng.

– Phải được liên kết với một bảng/ bảng ảo.

– Không thể gọi mà được thực hiện tự động.

– Không có tham số.

Lệnh cập nhật dữ liệu trên bảng > xảy ra biến cố (event) > Trigger sẽ được tự động gọi thực hiện.

**Cú pháp:**

|  |
| --- |
| CREATE TRIGGER tên\_trigger ON tên\_bảng  FOR {DELETE, INSERT, UPDATE}  AS  câu\_lệnh\_sql |

**Các kiểu Trigger trong SQL**

Trong SQL, bạn sẽ gặp phải 3 loại Trigger như sau:

Trigger cho DML (ngôn ngữ thao tác dữ liệu): bao gồm 3 kiểu là INSERT, UPDATE và DELETE.

Trigger cho DDL (ngôn ngữ định nghĩa dữ liệu): sử dụng cho các lệnh CREATE, ALTER và DROP.

Trigger cho LOGON: sử dụng để kiểm soát các lệnh khi đăng nhập vào máy chủ.

Trong đó, SQL Trigger cho DML được sử dụng nhiều nhất và có ứng dụng cao nhất

**Tại sao phải sử dụng Trigger**

Để kiểm tra các ràng buộc dữ liệu phức tạp.

Tính toán và tự động cập nhật giá trị.

**Các hạn chế khi viết Trigger**

Không được tạo bảng tạm.

Không được tham chiếu đến bảng tạm hoặc bảng hệ thống.

Không tạo/ sửa/ xoá cấu trúc các đối tượng sẵn có trong CSDL: CREATE/ ALTER/ DROP.

Không gán/ cấp quyền cho người dùng: GRANT/ REVOKE.

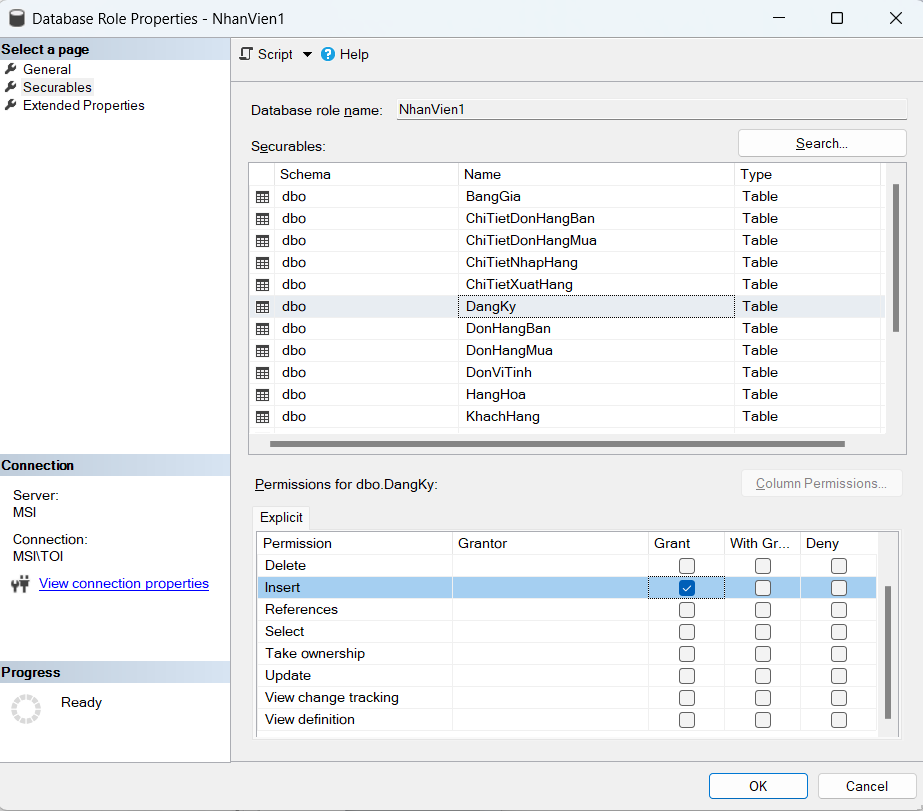
* Ví dụ 1 code Trigger trên CSDL

|  |
| --- |
| --Tạo Trigger để delete mới một nv vào bảng nhanvien  go  CREATE TRIGGER trg\_delete\_NHANVIEN  ON NHANVIEN  FOR DELETE  AS  BEGIN  -- Xóa dữ liệu nhân viên trong bảng nhanvien\_backup  DELETE NHANVIEN  FROM NHANVIEN  INNER JOIN deleted ON NHANVIEN.MaNV = deleted.MaNV;  END; |

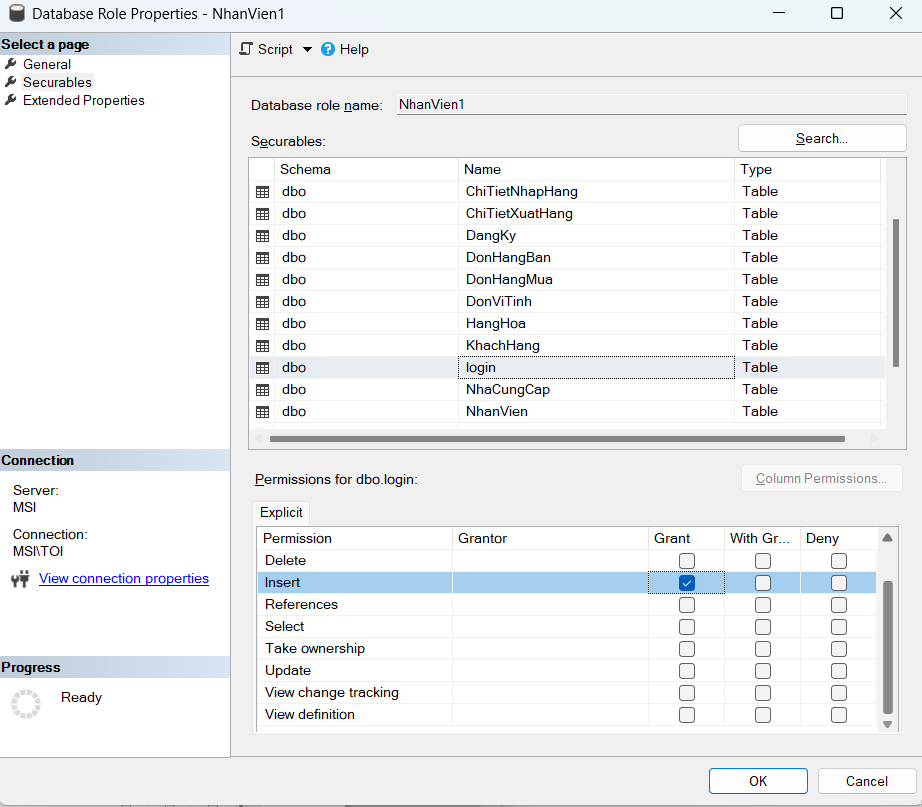
# **4. Phân quyền người dùng**

## 4.1 Nhân Viên

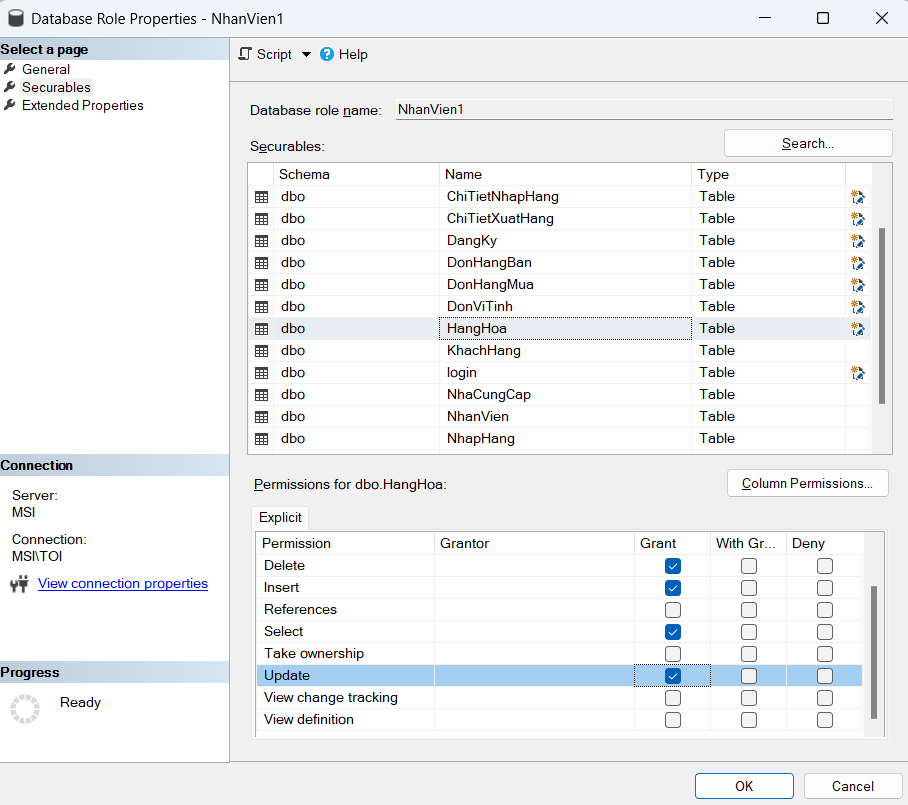
* (Chức Năng tạo tài khoản)



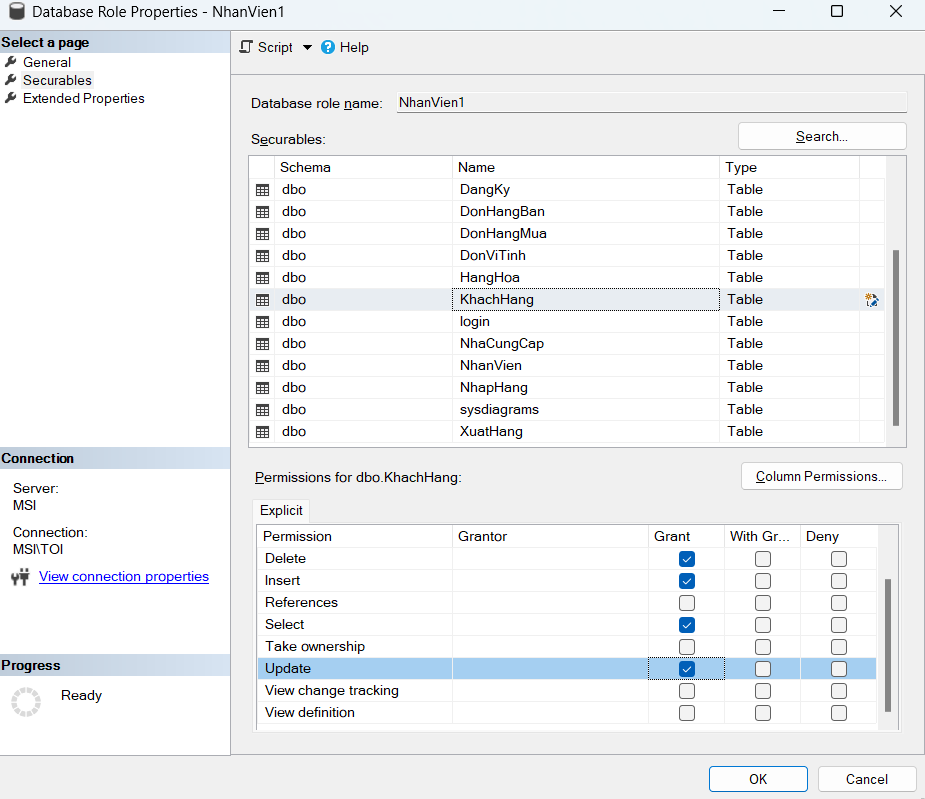
* (Chức Năng đăng nhập)



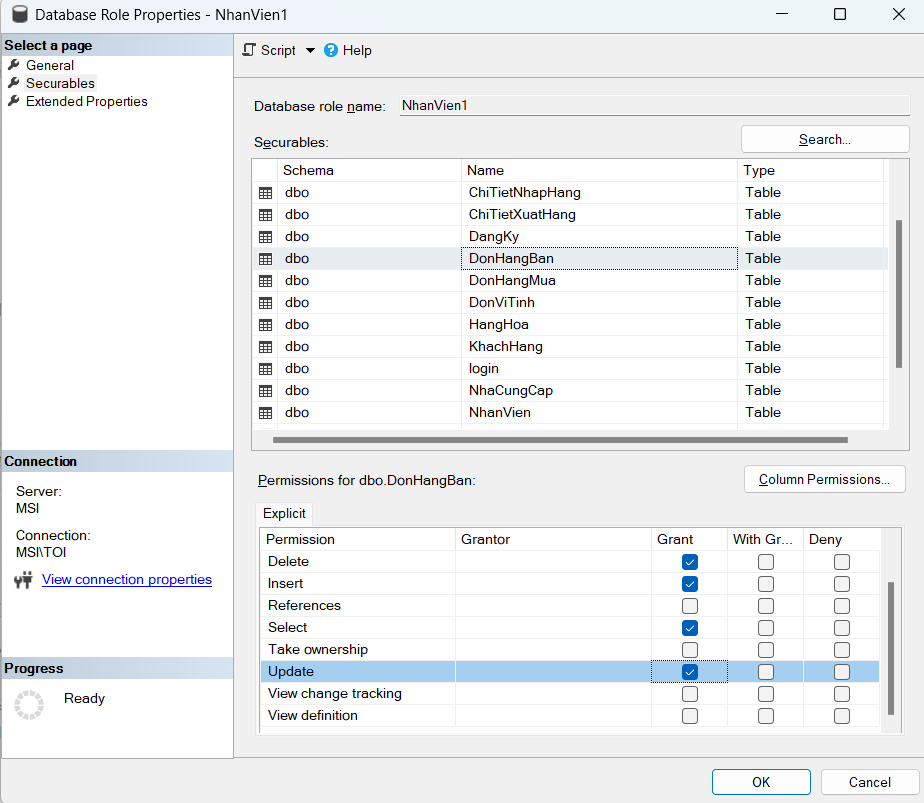
* (Chức Năng thêm, sửa, xóa, hiển thị hàng hoá )



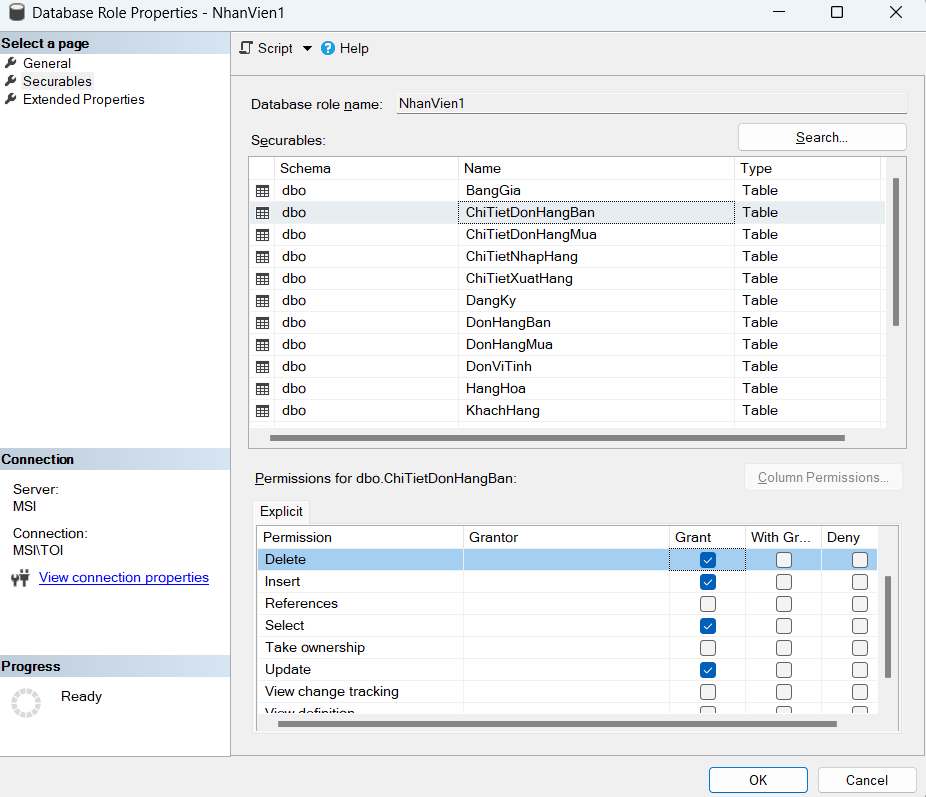
* (Chức Năng thêm, sửa, xóa, hiển thị khách hàng )



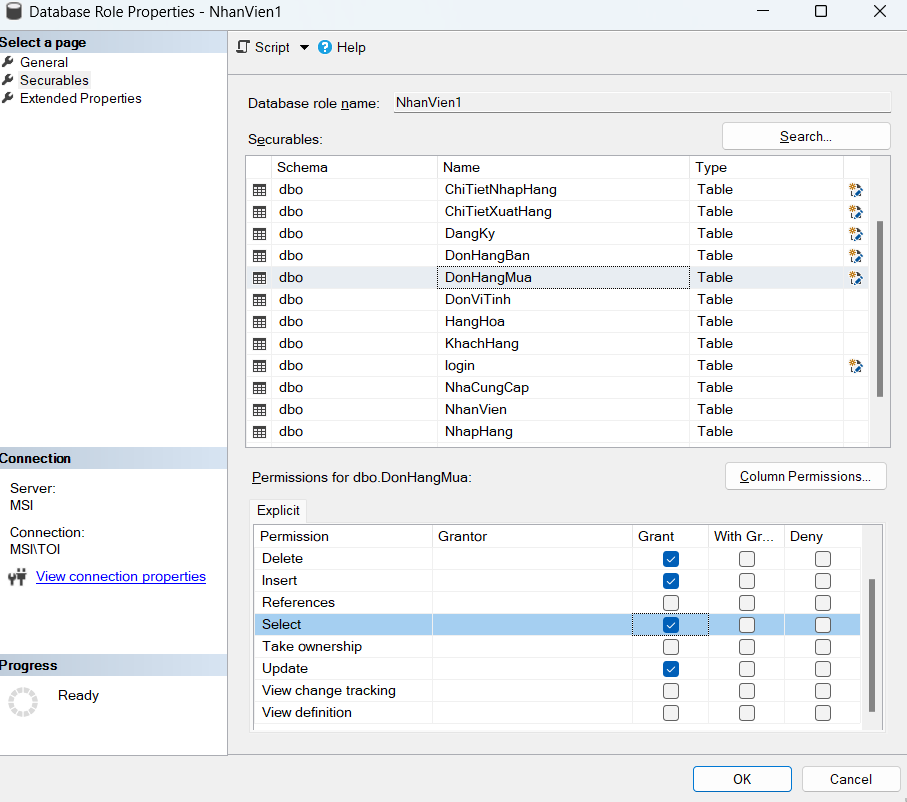
* (Chức Năng thêm, sửa, xóa, hiển thị đơn hàng bán)



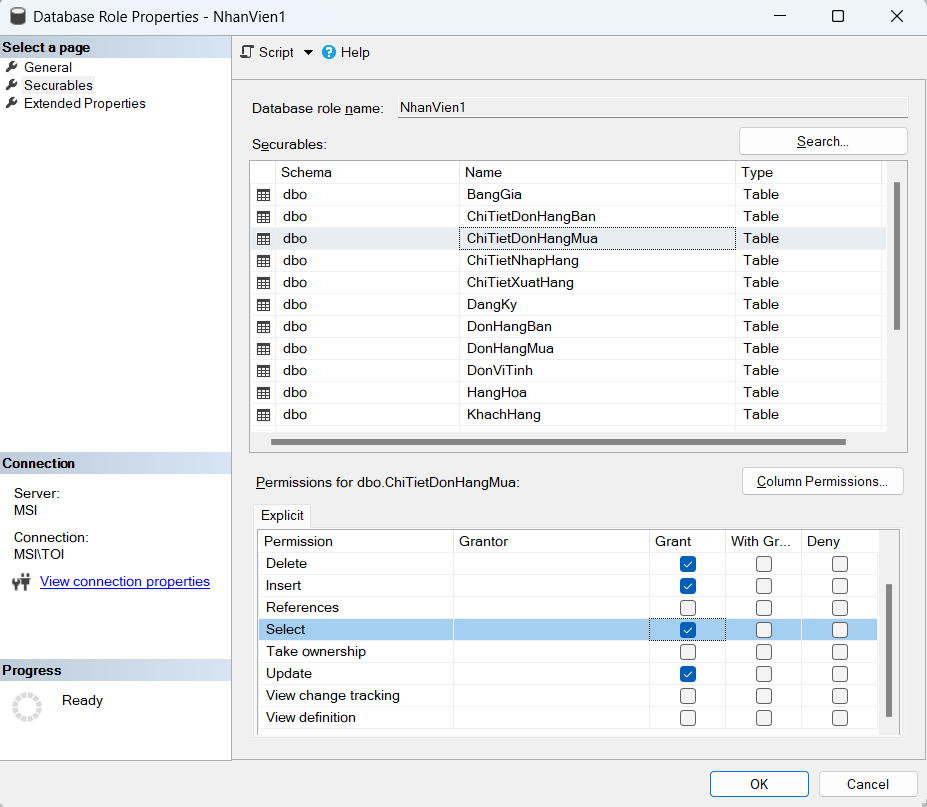
* (Chức Năng thêm, sửa, xóa,hiển thị chi tiết đơn hàng bán)



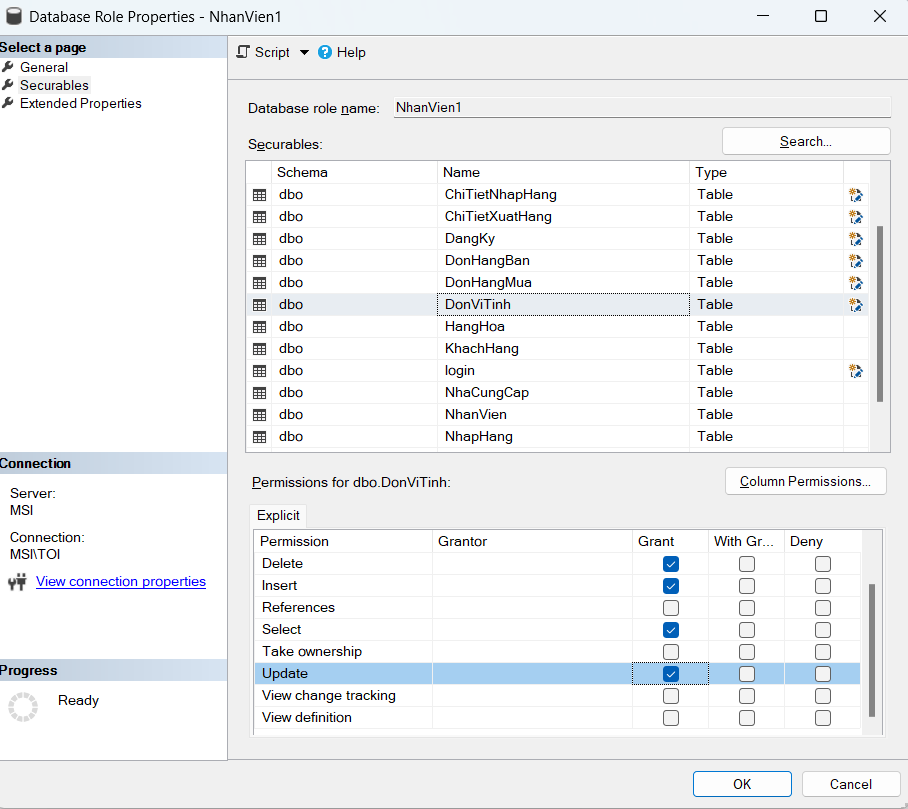
* (Chức Năng thêm, sửa, xóa, hiển thị đơn hàng mua)



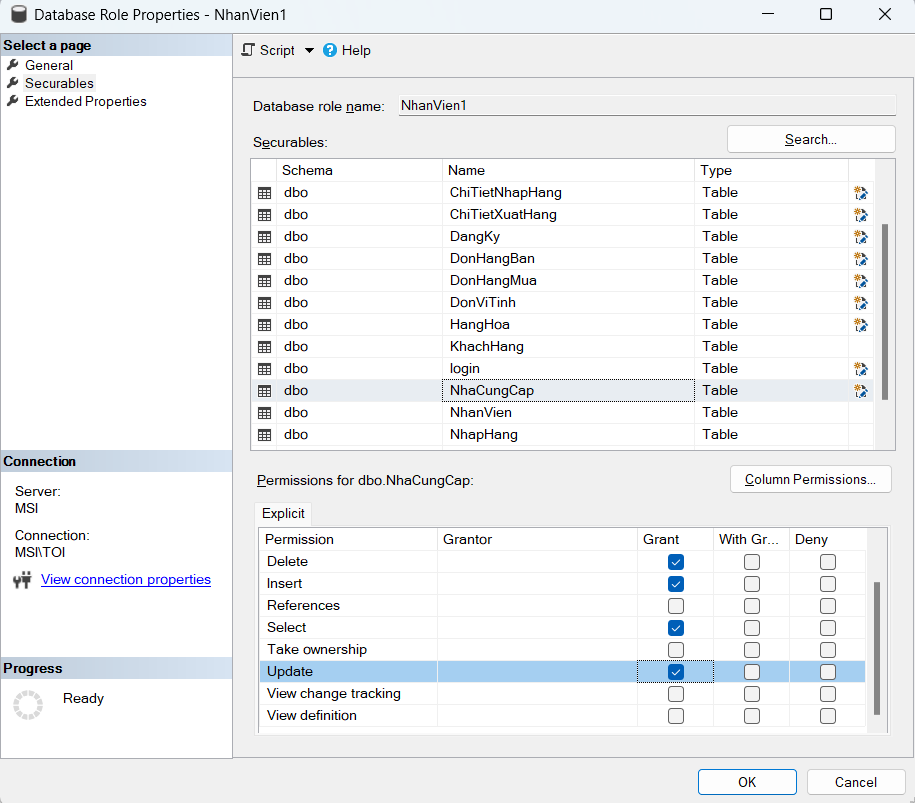
* (Chức Năng thêm, sửa, xóa,hiển thị chi tiết đơn hàng mua)



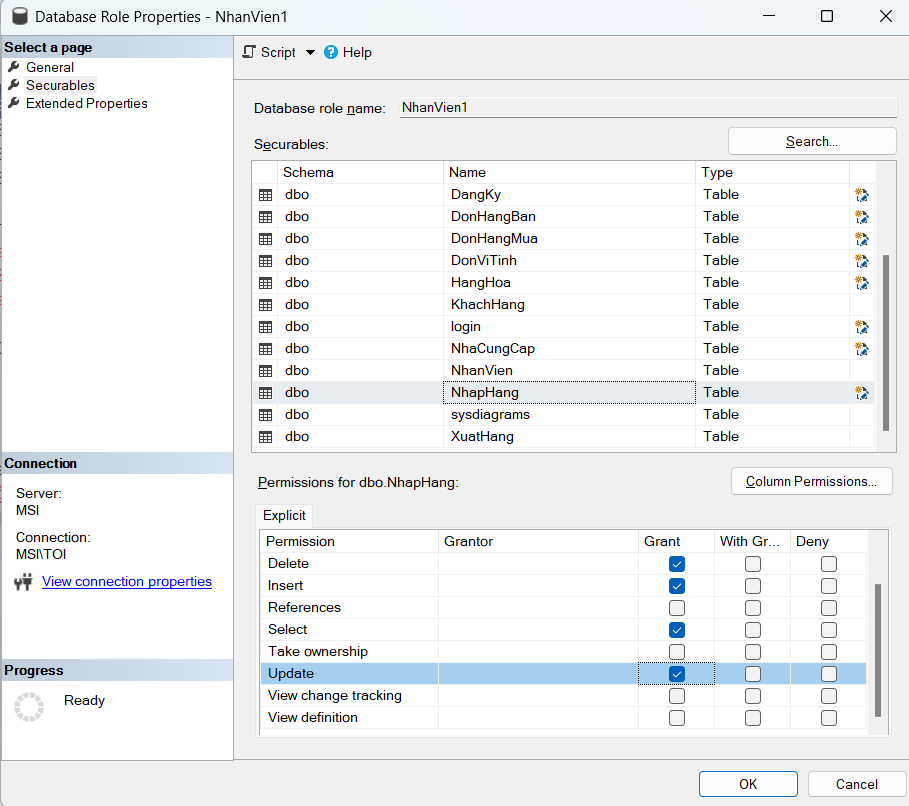
* (Chức Năng thêm, sửa, xóa,hiển thị chi tiết đơn vị tính )



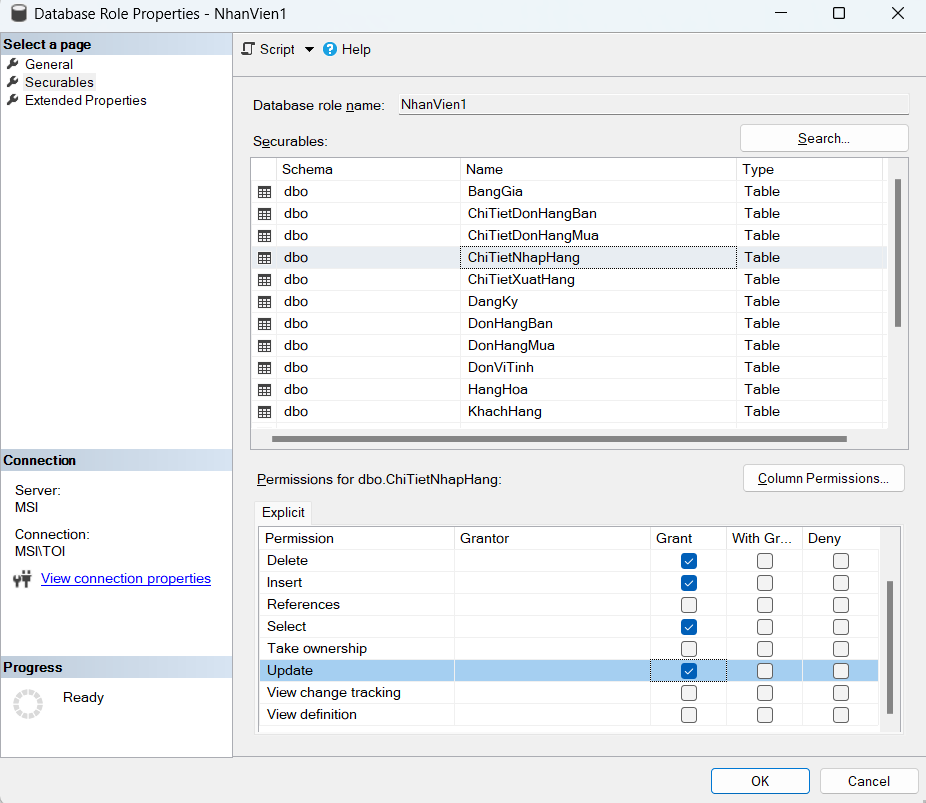
* (Chức Năng thêm, sửa, xóa,hiển thị Nhà cung cấp )



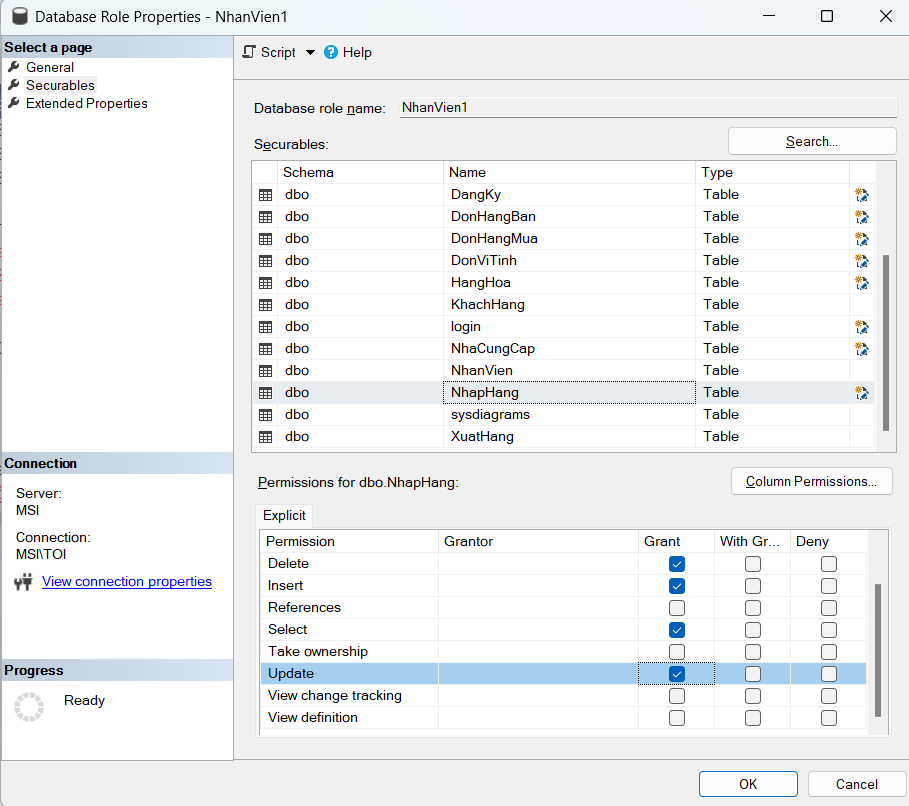
* (Chức Năng thêm, sửa, xóa,hiển thị nhập hàng )



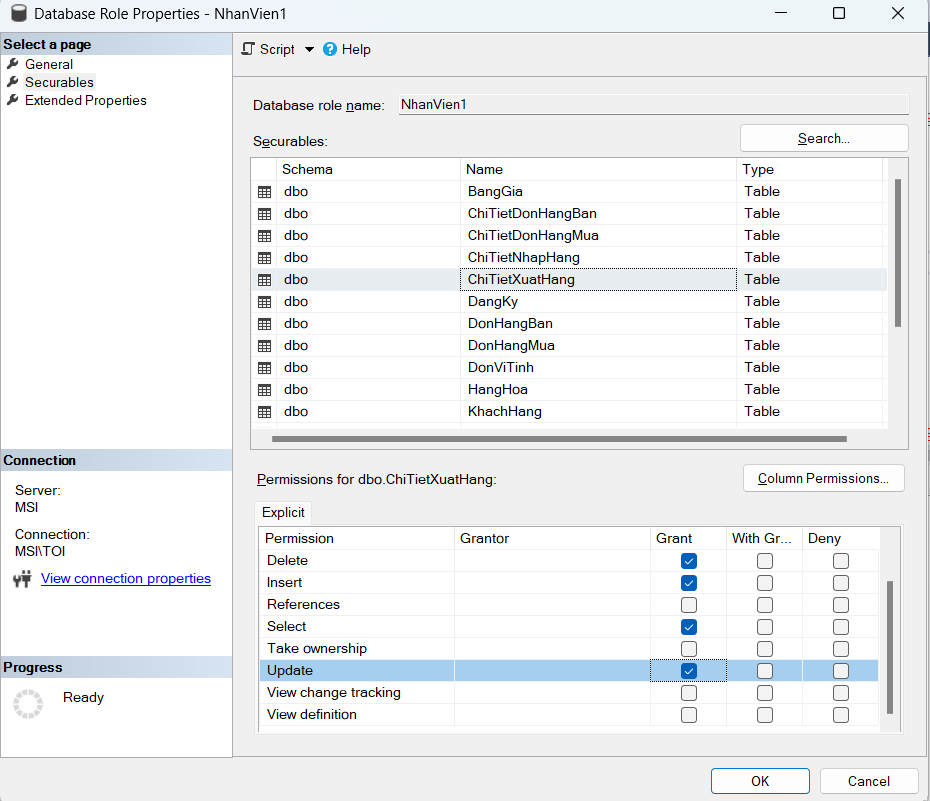
* (Chức Năng thêm, sửa, xóa,hiển thị chi tiết nhập hàng)



* (Chức Năng thêm, sửa, xóa,hiển thị xuất hàng)

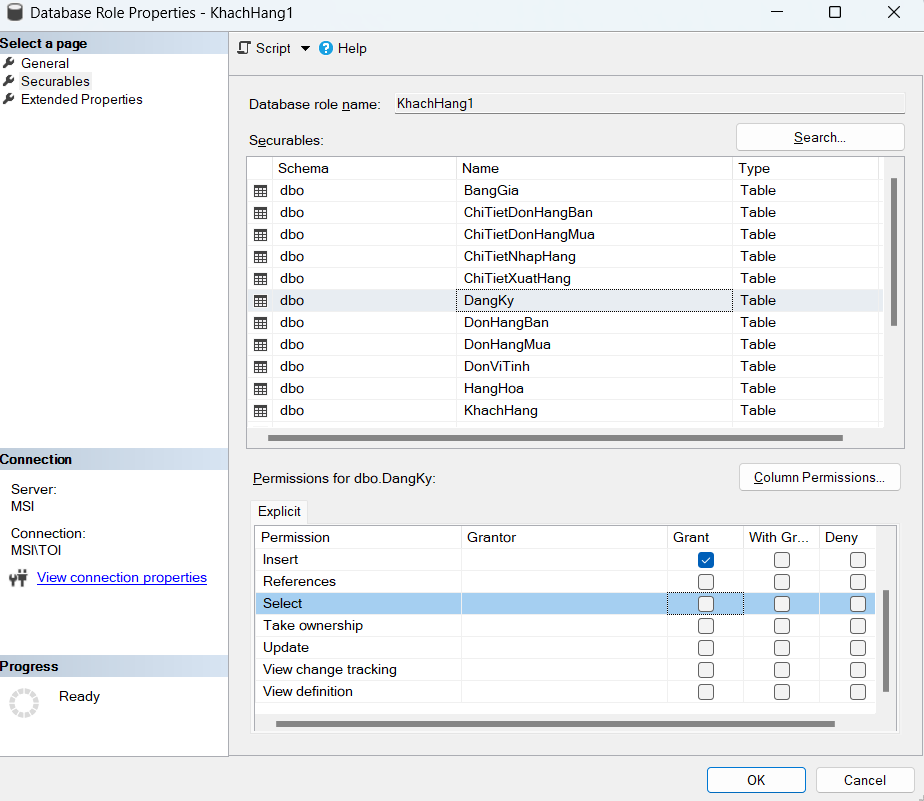


* (Chức Năng thêm, sửa, xóa,hiển thị chi tiết xuất hàng )

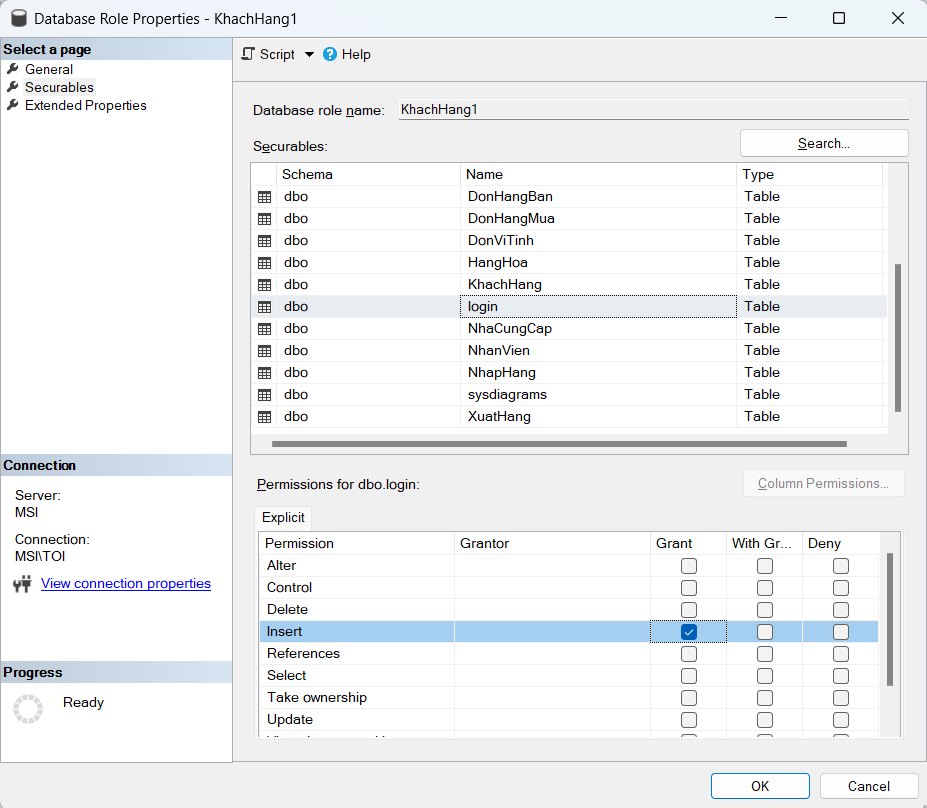


## 4.2 Khách Hàng

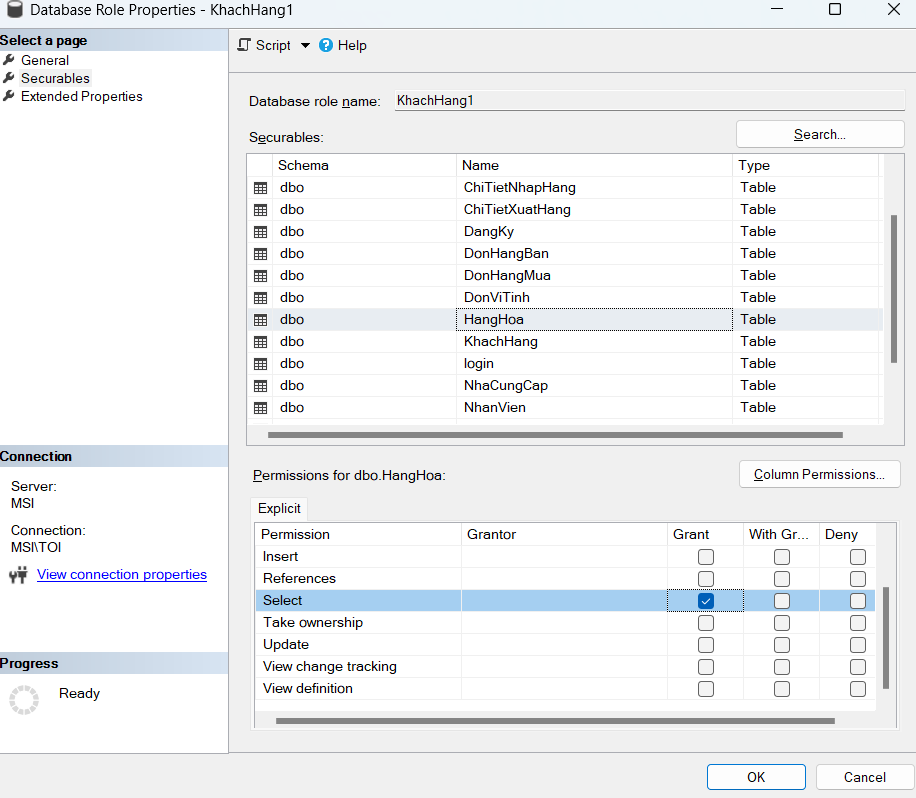
* (Chức Năng tạo tài khoản)



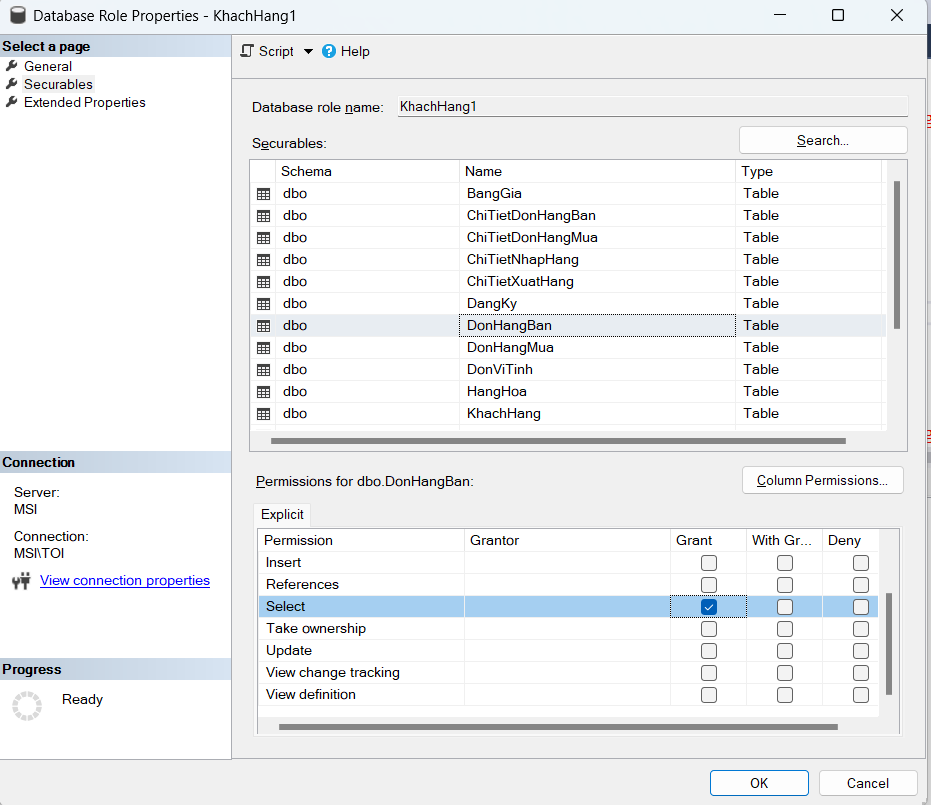
* (Chức Năng đăng nhập tài khoản)



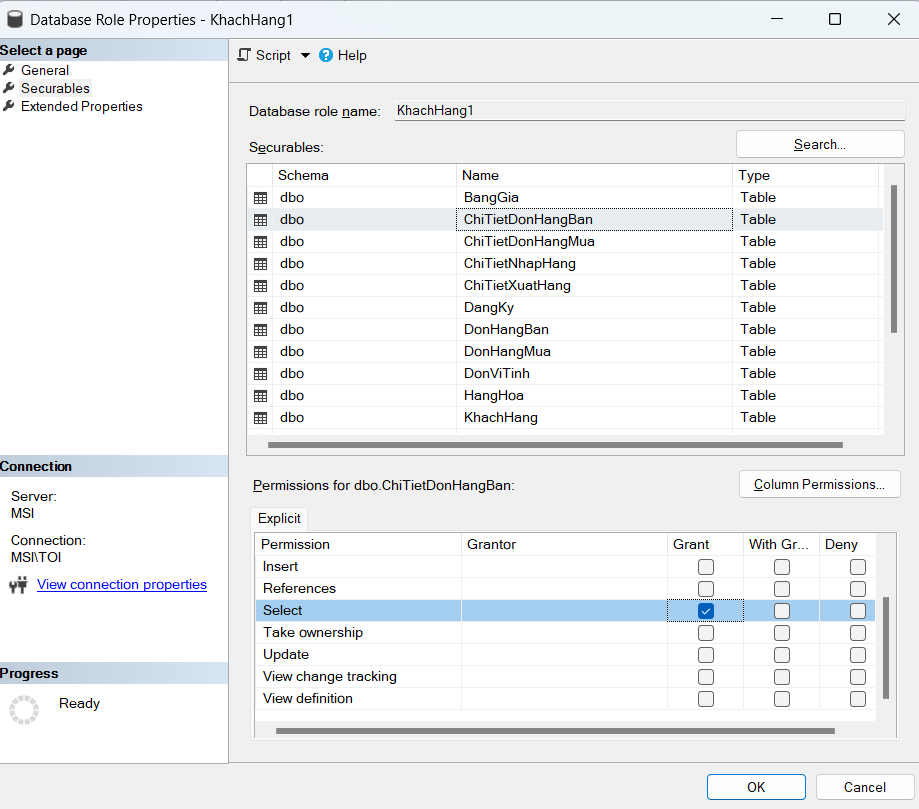
* (Chức Năng hiển thị hàng hoá)



* (Chức Năng hiển thị Đơn Hàng)

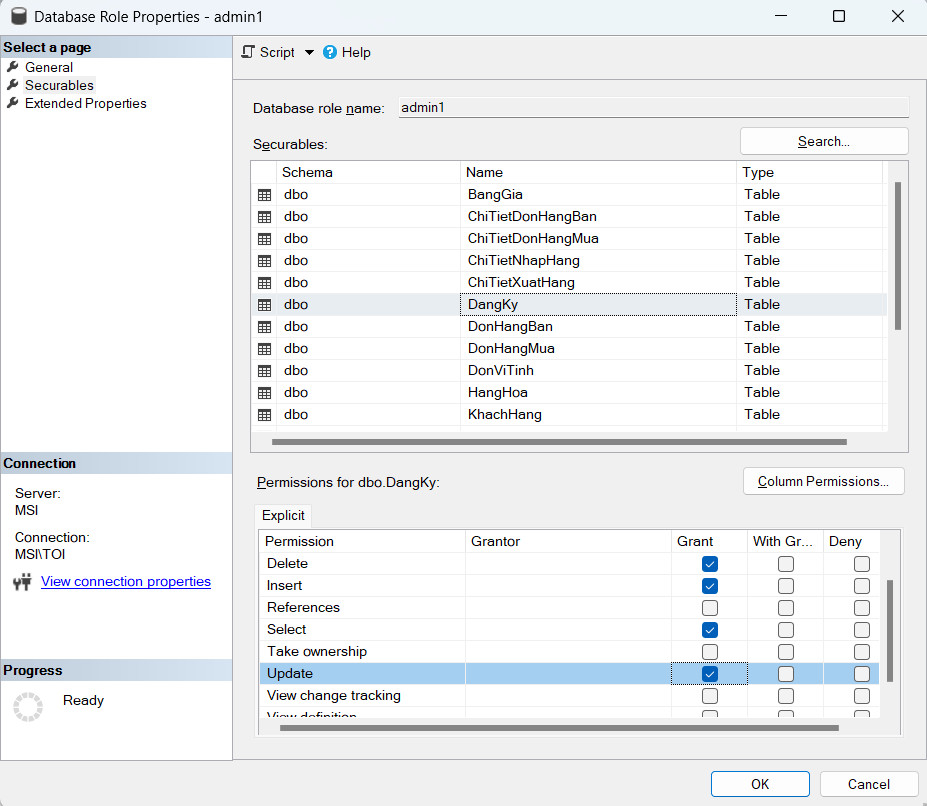


* (Chức Năng hiển thị Chi tiết đơn hàng)

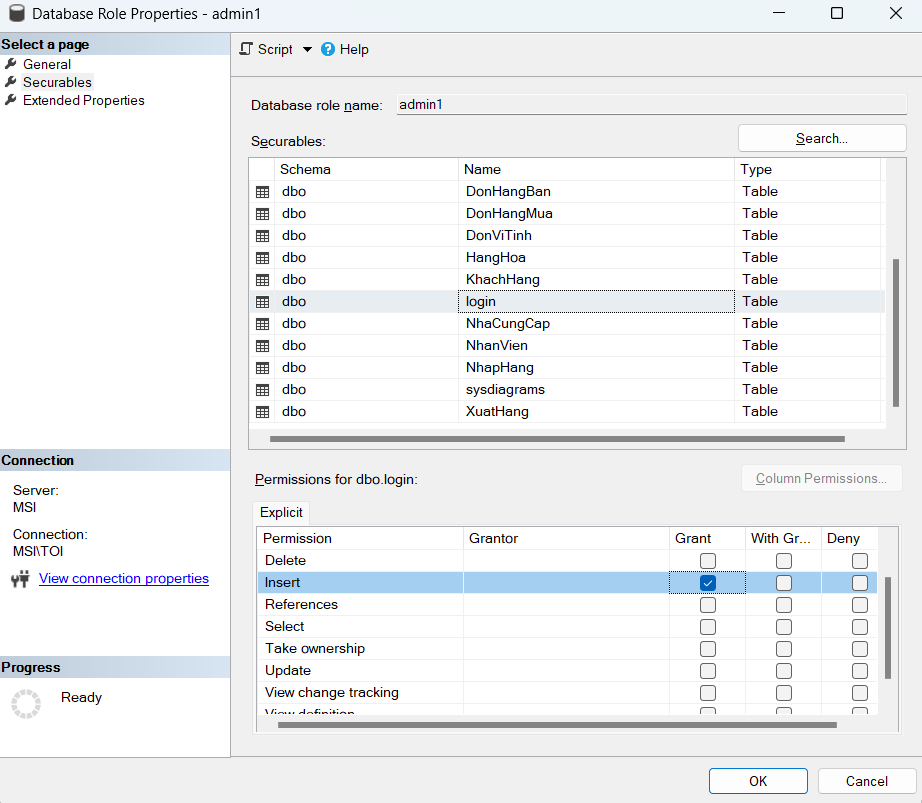


4.3 Người Quản Trị(ADMIN)

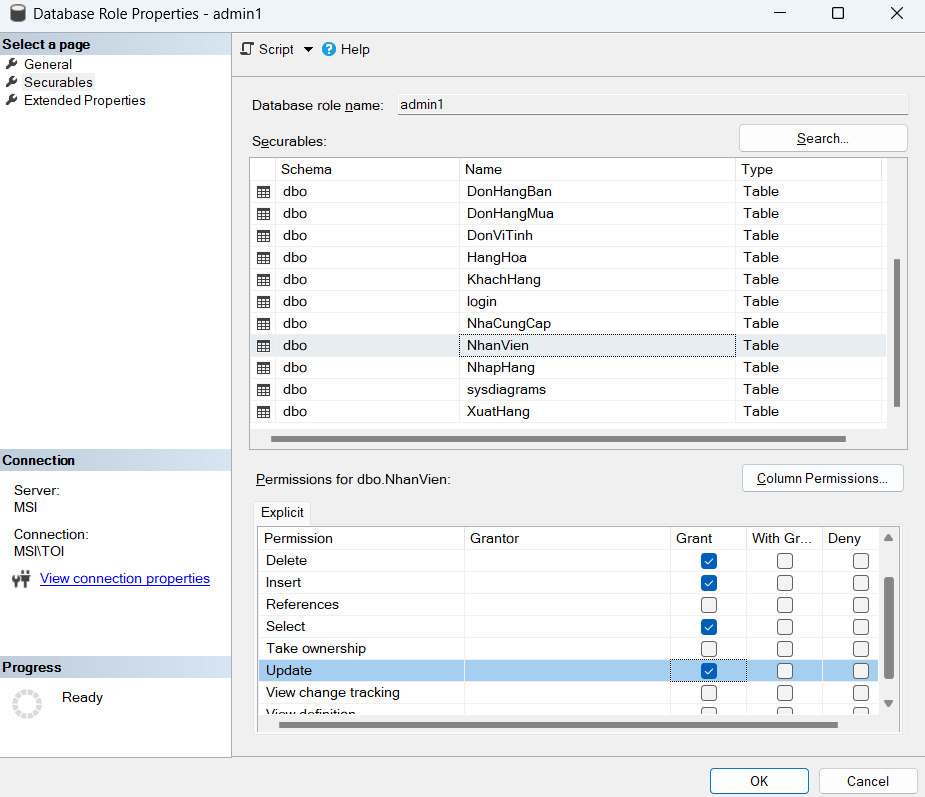
* (Chức Năng thêm, sửa, xóa,hiển thị Đăng ký)



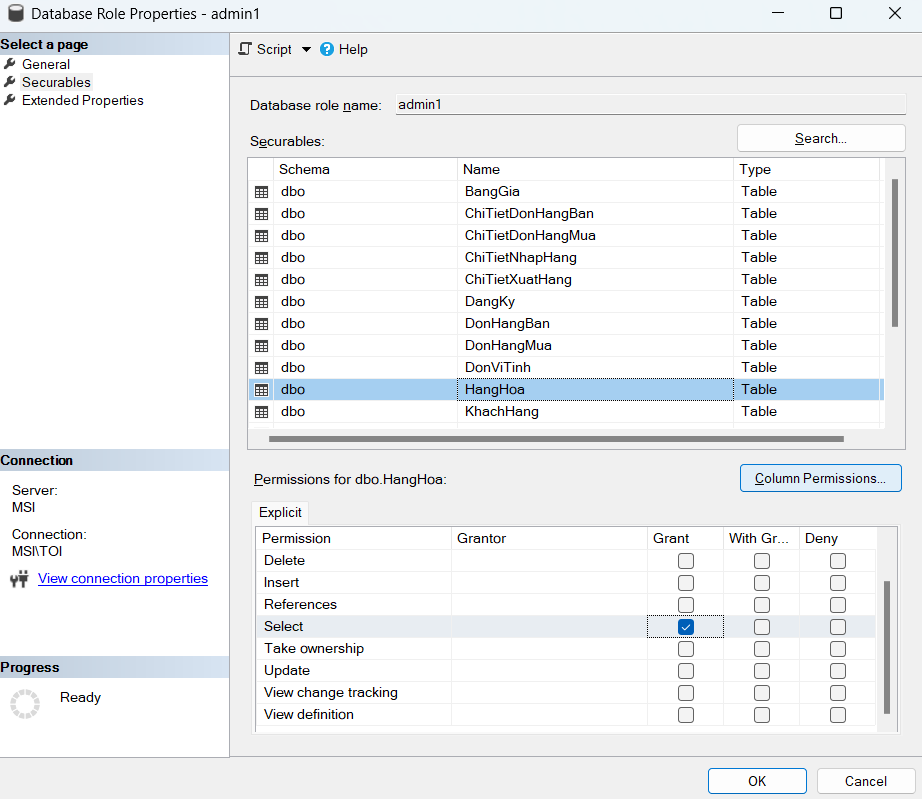
* (Chức Năng hiển thị Đăng nhập)



* (Chức Năng thêm, sửa, xóa,hiển thị danh sách nhân viên)



* (Chức Năng hiển thị hàng hoá)



# **Chương 3: KẾT LUẬN**

# ***1. Những kết quả đạt được của đồ án***

Trong đồ án này, nhóm em đã vận dụng phương pháp phân tích thiết kế và hệ quản trị cơ sở dữ liệu để xây dựng quản lý bán hàng nội thất. Kết quả đạt được của đồ án bao gồm:

* Phát biểu và mô tả được nghiệp vụ bài toán
* Biểu diễn các nghiệp vụ bài toán bằng các sơ đồ
* Phân tích thiết kế bài toán theo đúng quy trình đã được học bằng cá phương pháp
* Thiết kế được cơ sở dữ liệu quan hệ để lưu trữ dữ liệu
* Thiết kế được các câu lệnh để truy vấn cho các bảng
* Thiết kế được các giao diện cập nhật dữ liệu

# ***2. Nhược điểm của đồ án***

* Đồ án của nhóm em còn một số thiếu sót, cũng như một số tính năng chưa tối ưu hóa và xử lý được
* Giao diện chương trình chưa được đẹp mắt
* Xây dựng được hệ thống nhưng chỉ với các chức năng chính như: thêm, sửa, xóa.Cần thêm nhiều tính năng để chương trình hoạt động tốt và hoàn chỉnh hơn

Nếu có điều kiện, sau này nhóm em sẽ hoàn thiện chương trình được đầy đủ hơn và dễ dàng hơn trong việc quản trị