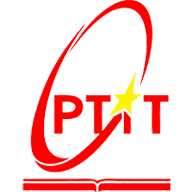
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**CƠ SỞ DỮ LIỆU PHÂN TÁN**

**ĐỀ TÀI**

**QUẢN LÝ CHUỖI NHÀ SÁCH**

**GIẢNG VIÊN: PHAN THỊ HÀ**

**NHÓM: 04**

**SINH VIÊN THỰC HIỆN:**

|  |  |
| --- | --- |
|  |  |
|  |

**Mục lục**

[I. Đặt vấn đề 3](#_Toc89205758)

[1. Nhu cầu và tầm quan trọng của dự án: 3](#_Toc89205759)

[2. Sơ lược về dự án: 3](#_Toc89205760)

[3. Kịch bản: 4](#_Toc89205761)

[II. Phân tích 5](#_Toc89205762)

[1. Các chức năng chính của hệ thống trong dự án 5](#_Toc89205763)

[2. Phân quyền cho các nhóm đối tượng thực hiện dự án 6](#_Toc89205764)

[3. Phân tích chức năng của từng vị trí thực hiện dự án 6](#_Toc89205765)

[4. Phân tích CSDL ( Mô hình thực thể liên kết, quan hệ) 7](#_Toc89205766)

[4.1. Mô hình thực thể liên kết (E-R) 8](#_Toc89205767)

[4.2. Lược đồ quan hệ 9](#_Toc89205768)

[5. Bảng tần suất truy cập tại các vị trí 10](#_Toc89205769)

[III. Thiết kế 10](#_Toc89205770)

[1. Thiết kế hệ thống mạng tổng quan 10](#_Toc89205771)

[2. Thiết kế CSDL của hệ thống 11](#_Toc89205772)

[2.1. Các bảng 11](#_Toc89205773)

[2.2. Mô hình quan hệ giữa các bảng 14](#_Toc89205774)

[2.3. Thiết kế csdlpt 14](#_Toc89205775)

[IV. Cài đặt 18](#_Toc89205776)

[1. Cài đặt SQL Server 18](#_Toc89205777)

[2. Cài đặt VPN 25](#_Toc89205778)

[Cài đặt Radmin VPN 25](#_Toc89205779)

[Cấu hình tường lửa 26](#_Toc89205780)

[3. Tạo Publication 30](#_Toc89205781)

[4. Tạo Subscriptions 38](#_Toc89205782)

[5. Tạo Link server 44](#_Toc89205783)

[Tạo tài khoản 44](#_Toc89205784)

[Tạo Link server 46](#_Toc89205785)

[6. Trigger 48](#_Toc89205786)

[7. Website quản lý 53](#_Toc89205787)

[Web ở phía máy chủ 53](#_Toc89205788)

[Web tại phía chi nhánh cho nhân viên quản lý 56](#_Toc89205789)

[Web ở chi nhánh cho nhân viên bán hàng 59](#_Toc89205790)

[8. App 60](#_Toc89205791)

[App ở phía máy chủ 60](#_Toc89205792)

[App tại phía chi nhánh cho nhân viên quản lý 62](#_Toc89205793)

[App ở chi nhánh cho nhân viên bán hàng 64](#_Toc89205794)

[9. Procedures 65](#_Toc89205795)

# I. Đặt vấn đề

## Nhu cầu và tầm quan trọng của dự án:

**Nhu cầu:**

* Có một chuỗi nhà sách ở miền Bắc với trụ sở đặt tại Hà Nội. Các nhà sách chi nhánh khác đặt tại các tỉnh thành còn lại.
* Để quản lí các của hàng chi nhánh hiệu quả cần xây dựng hệ thống quản lý thông tin và hoạt động kinh doanh. Các chi nhánh được đặt tại các tỉnh khác nhau nên cần một hệ thống có thể liên kết các chi nhánh với máy chủ giúp đồng bộ và phân tán dữ liệu.

**Tầm quan trọng:**

* Việc mở rộng thành các chuỗi chi nhánh gây khó khăn cho việc quản lý khi số lượng các giao dịch và mặt hàng tăng lên, khi đó khối lượng công việc theo từng ngày sẽ rất lớn lên rất nhanh.
* Việc đặt cơ sở dữ liệu nội bộ tại các chi nhánh lại rất tốn kém và đem lại nhiều khó khăn. Do đó, ta sẽ sử dụng và xây dựng CSDL phân tán cho chuỗi cửa hàng để đơn giản nâng cao hiệu quả bán hàng.

## Sơ lược về dự án:

* Hệ thống gồm 1 máy chủ đặt tại Hà Nội. Và có 9 máy trạm đặt tại 9 tỉnh khác nhau bao gồm: Hòa Bình, Hải Phòng, Nam Định, Hưng Yên, Bắc Giang, Thái Nguyên, Ninh Bình, Quảng Ninh, Hải Dương, Bắc Kạn.
* Các chức năng chính của hệ thống:
  + Phía máy trụ sở
  + Quản lý nhân viên: thêm sửa, xóa nhân viên; quản lý nhân viên bán hàng, nhân viên nhập sách, nhân viên quản lý.
  + Quản lý khách hàng: thêm, sửa, xóa thông tin khách hàng.
  + Quản lý nhà cung cấp: thêm, sửa, xóa thông tin nhà cung cấp
  + Nhập sách
  + Phân phát sách cho các chi nhánh
  + Thống kê: thống kê doanh thu, lợi nhuận theo ngày/tuần/tháng; thống kê sách bán chạy; thống kê khách hàng thân thiết.
  + Phía máy chi nhánh
    - Quản lý nhân viên: thêm sửa, xóa nhân viên; quản lý nhân viên bán hàng, nhân viên nhập sách, nhân viên quản lý.
    - Quản lý khách hàng: thêm, sửa, xóa thông tin khách hàng.
    - Thêm, sửa, xóa hóa đơn bán hàng
* Đối tượng sử dụng: nhân viên tại các chi nhánh.
* Mô hình: client-server
* Nhu cầu cần phải sử dụng cơ sở dữ liệu phân tán:
* Số lượng chi nhánh lớn và cách xa nhau.
* Dể dàng thêm chi nhánh mới nếu có nhu cầu mở rộng.
* Giảm thiểu chi phí quản lý.
* Cung cấp thông tin nhanh chóng và chính xác.

## Kịch bản:

* Một chuỗi nhà sách gồm 10 chi nhánh đặt tại 10 tỉnh khác nhau bao gồm: Hải Phòng, Hòa Bình, Bắc Kạn, Nam Định, Hưng Yên, Bắc Giang, Thái Nguyên, Ninh Bình, Quảng Ninh, Hải Dương và một máy chủ đặt tại Hà Nội. Hệ thống được sử dụng bởi các nhân viên tại các chi nhánh phục vụ cho việc bán sách cho khách hàng.
* Máy chủ được đặt tại Hà Nội sẽ chịu trách nhiệm lưu lại toàn bộ dữ liệu của cả 9 chi nhánh bao gồm các dữ liệu về thông tin khách hàng, thông tin nhân viên, chi nhánh, thông tin sản phẩm, thông tin giao dịch, thông tin nhà cung cấp, thông tin nhập sách, số lượng sách, thống kê doanh thu theo định kỳ ..
* Máy chủ sẽ tự động đồng bộ dữ liệu từ các chi nhánh khác nhau và lưu lại toàn bộ dữ liệu đó khi dữ liệu ở máy trạm có sự thay đổi.
* Mỗi máy trạm tại các chi nhánh sẽ cập nhật, sửa đổi các thông tin về khách hàng, hóa đơn, nhân viên, khi cần đến dữ liệu của một chi nhánh khác, máy trạm sẽ kết nối server đến máy trạm của chi nhánh đó để lấy dữ liệu cần thiết.
* Khách hàng A ở tỉnh n muốn đặt mua sách sẽ đến chi nhánh cửa hàng n để đặt mua sách, khi đó nhân viên tại chi nhánh trạm n sẽ truy cập vào CSDL để lấy thông tin từ trạm tổng gửi xuống thông tin về khách hàng đó và thực hiện giao dịch đặt mua sách cho khách hàng. Kết thúc giao dịch, chi nhánh tại trạm n lưu thông tin giao dịch vào CSDL của mình và cập nhật lên server máy chủ.

# II. Phân tích

## Các chức năng chính của hệ thống trong dự án

- **Quản lý nhân viên:** Cho phép quản lý có thể quản lý tài khoản của các nhân viên trong nhà sách

+ Thêm nhân viên: Thêm nhân viên mới của nhà sách vào hệ thống.

+ Sửa nhân viên: Sửa đổi, cập nhật thông tin của nhân viên trong hệ thống

+ Vô hiệu hóa nhân viên: Vô hiệu hóa tài khoản đăng nhập của nhân viên

→ Được thực hiện tại các chi nhánh, mỗi chi nhánh quản lý thông tin nhân viên của chi nhánh mình

- **Quản lý khách hàng thành viên**

+ Thêm, sửa thông tin khách hàng

→ Thông tin khách hàng được quản lý tại mỗi chi nhánh và được đồng bộ hóa tại máy chủ, các máy trạm tại các chi nhánh sẽ nhận được thông tin nhân bản của khách hàng mới (trong trường hợp thêm) và thông tin cập nhật của khách hàng (trong trường hợp cập nhật)

- **Chức năng thanh toán**

+ Tạo hóa đơn bán: Cho phép nhân viên bán hàng tạo hóa đơn cho khách hàng khi thanh toán tại quầy khi có yêu cầu thanh toán của khách hàng

→ Được thực hiện tại các chi nhánh, mỗi chi nhánh quản lý thông tin hóa đơn bán tại chi nhánh của mình

**- Chức năng quản lý kho (Chức năng tại trụ sở)**

+ Tạo hóa đơn nhập: Cho phép nhân viên nhập sách vào kho

+ Thêm, sửa, xóa thông tin sách: Sửa đổi thông tin sách trong kho

→ Được thực hiện tại tại trụ sở, trụ sở là kho tổng, có trách nhiệm phân phát sách tới các chi nhánh

**- Chức năng thống kê**

+ Thống kê doanh thu, lợi nhuận theo ngày/tuần/tháng: Cho phép nhân viên quản lý tại mỗi chi nhánh có thể xem báo cáo thống kê doanh thu, lợi nhuận theo ngày/tuần/tháng

+ Thống kê sách bán chạy: Cho phép nhân viên quản lý tại mỗi chi nhánh có thể xem báo cáo thống kê về sách bạn chạy tại chi nhánh và tất cả các chi nhanh

+ Thống kê khách hàng thân thiết: Cho phép nhân viên quản lý tại mỗi chi nhánh có thể xem báo cáo thống kê về khách hàng thân thiết tại chi nhánh và tất cả các chi nhánh

## Phân quyền cho các nhóm đối tượng thực hiện dự án

Nhóm đối tượng quản lý

* Người quản lý:
* Quản lý nhân viên tại chi nhánh
* Quản lý thông tin khách hàng thành viên
* Quản lý nhà cung cấp
* Xem các báo cáo thống kê
* Nhân viên:
* Quản lý thông tin sách
* Thực hiện bán, nhập sách
* Thực hiện tạo hóa đơn bán, nhập hàng

Nhóm đối tượng cần được quản lý:

* Nhân viên: các thông tin của nhân viên
* Sách: các thông tin của sách
* Nhà cung cấp: các thông tin của nhà cung cấp
* Khách hàng thành viên: các thông tin của khách hàng
* Hóa đơn bán: các thông tin về tài liệu được bán, chi phí phải thanh toán,...

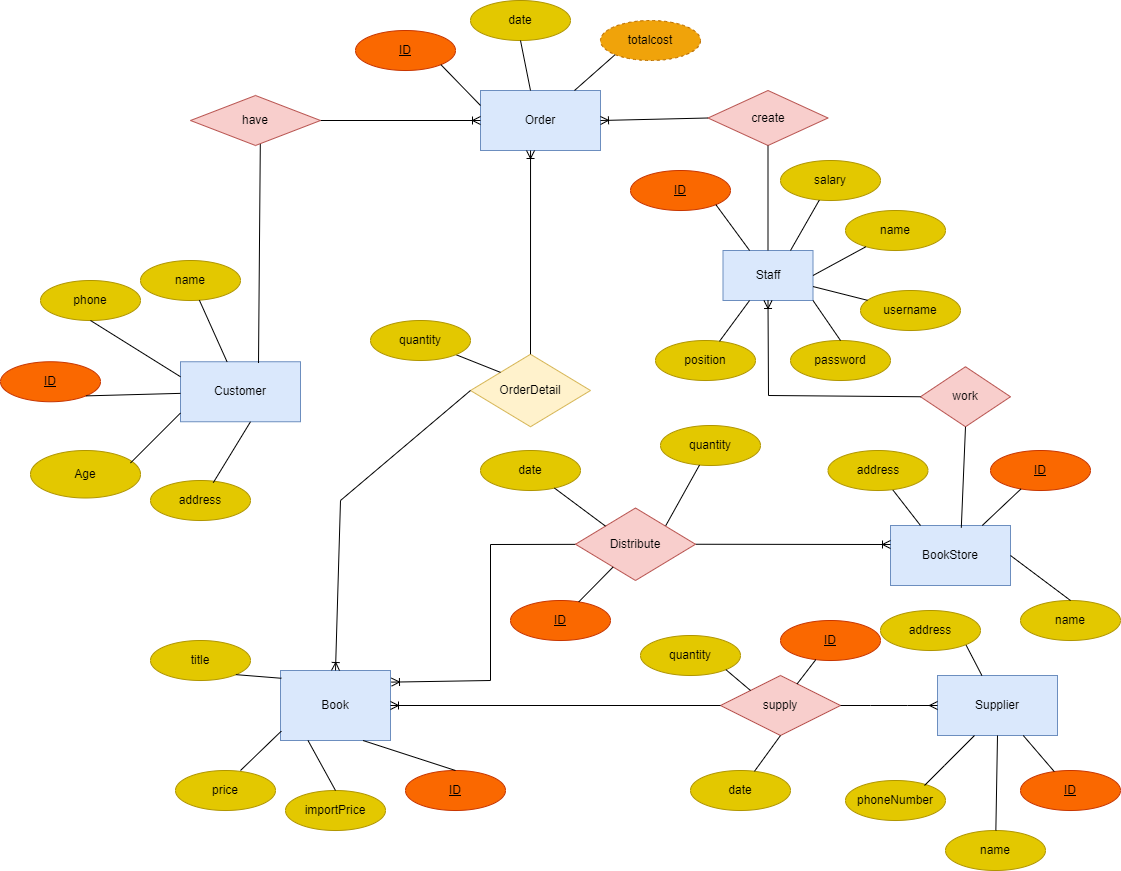
## Phân tích chức năng của từng vị trí thực hiện dự án

* Máy chủ:
  + Quản lý nhân viên: thêm sửa, xóa nhân viên; quản lý nhân viên bán hàng, nhân viên nhập sách, nhân viên quản lý.
  + Quản lý khách hàng: thêm, sửa, xóa thông tin khách hàng.
  + Quản lý nhà cung cấp: thêm, sửa, xóa thông tin nhà cung cấp
  + Nhập sách
  + Phân phát sách cho các chi nhánh
  + Thống kê: thống kê doanh thu, lợi nhuận theo ngày/tuần/tháng; thống kê sách bán chạy; thống kê khách hàng thân thiết.
* Chi nhánh:
  + - Quản lý nhân viên: thêm sửa, xóa nhân viên; quản lý nhân viên bán hàng, nhân viên nhập sách, nhân viên quản lý.
    - Quản lý khách hàng: thêm, sửa, xóa thông tin khách hàng.
    - Thêm, sửa, xóa hóa đơn bán hàng

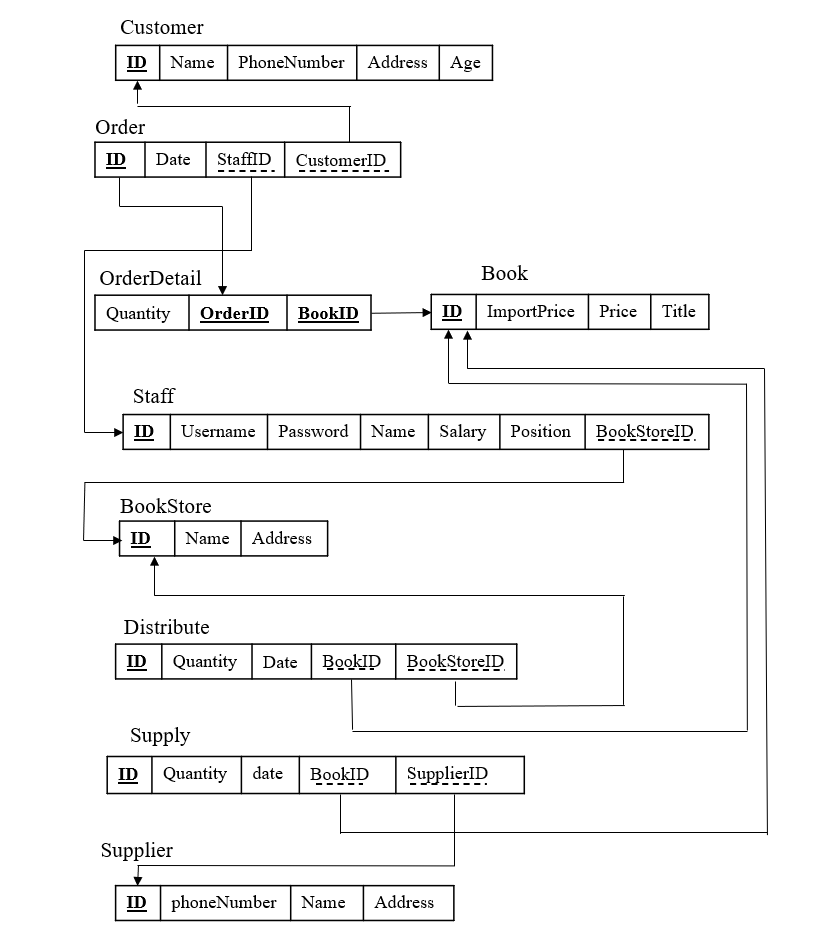
## Phân tích CSDL ( Mô hình thực thể liên kết, quan hệ)

* Yêu cầu về thông tin, dữ liệu được tra cứu trong csdl
* Lưu trữ thông tin về nhân viên làm việc: Mã nhân viên, tên, ngày sinh, lương, chức vụ
* Lưu trữ thông tin về sách: ID, tên sách, số lượng trong kho, giá bán, giá nhập
* Lưu trữ thông tin về nhà sách: ID, tên, địa chỉ
* Lưu trữ thông tin về hóa đơn bán: ID, thông tin của nhân viên, thông tin của khách hàng, thông tin các sản phẩm trong hóa đơn
* Mối quan hệ và các ràng buộc liên kết giữa các dữ liệu
* Một nhà sách có nhiều nhân viên
* Nhân viên phụ trách bán sách sách
* Sách được nhân viên bán
* Hóa đơn được tạo ra khi bán sách
* Mỗi hóa đơn chứa nhiều sách, mỗi sách trong hóa đơn có thêm số lượng
* Mối quan hệ số lượng giữa các dữ liệu
* Một nhà sách có thể có nhiều nhân viên, một nhân viên chỉ làm ở một nhà sách: Quan hệ giữa nhà sách - nhân viên là 1 - n
* Một sách có thể được nhập bởi nhiều nhà cung cấp, một nhà cung cấp có thể nhập được nhiều sách: Quan hệ giữa nhà cung cấp - sách là n - n
* Một hóa đơn bán có thể có nhiều sách, một sách có thể tham gia nhiều hóa đơn: Quan hệ giữa sách và hóa đơn bán là n – n
* Một loại sách có thể được phân phát tới chi nhánh nhiều lần, một chi nhánh được phân phát nhiều loại sách, quan hệ giữa chi nhánh và sách là n-n

### 4.1. Mô hình thực thể liên kết (E-R)



### 4.2. Lược đồ quan hệ



## Bảng tần suất truy cập tại các vị trí

|  |  |  |
| --- | --- | --- |
| Thực thể | Trụ sở chính | Chi nhánh |
| Customer | L.RWED | H.RWE |
| OrderDetail | L.R | H.RWED |
| Order | L.R | H.RWED |
| Distribute | H.RWED | H.R |
| Bookstore | L.RWED | H.R |
| Staff | L.RWED | H.RWED |
| Book | H.RWED | H.R |
| Supply | H.RWED | H.R |
| Supplier | H.RWED | H.R |

# III. Thiết kế

## Thiết kế hệ thống mạng tổng quan

* Máy chủ Hà Nội: Đặt máy Server thực hiện các chức năng và lưu CSDL của toàn hệ thống.
* Chi nhánh Hải Phòng: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Hải Phòng.
* Chi nhánh Hòa Bình: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Hòa Bình.
* Chi nhánh Bắc Kạn: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Bắc Kạn.
* Chi nhánh Nam Định: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Nam Định.
* Chi nhánh Hưng Yên: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Hưng Yên.
* Chi nhánh Bắc Giang: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Bắc Giang.
* Chi nhánh Thái Nguyên: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Thái Nguyên.
* Chi nhánh Ninh Bình : Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Ninh Bình.
* Chi nhánh Quảng Ninh: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Quảng Ninh.
* Chi nhánh Hải Dương: Đặt máy trạm thực hiện chức năng và lưu toàn bộ CSDL thuộc Hải Dương.

## Thiết kế CSDL của hệ thống

### 2.1. Các bảng

* **Bookstore**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã nhà sách (chi nhánh) |
| 2 | name | varchar | NOT NULL |  | Tên nhà sách |
| 3 | address | varchar | NOT NULL |  | Địa chỉ nhà sách |

* **Customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã khách hàng |
| 2 | name | varchar | NOT NULL |  | Tên khách hàng |
| 3 | phoneNumber | varchar | NOT NULL |  | SĐT khách hàng |
| 4 | address | varchar | NOT NULL |  | Địa chỉ khách hàng |
| 5 | age | int | NOT NULL |  | Tuổi khách hàng |

* **Staff**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã nhân viên |
| 2 | username | varchar | NOT NULL |  | Tên đăng nhập |
| 3 | password | varchar | NOT NULL |  | Mật khẩu |
| 4 | position | varchar | NOT NULL |  | Vị trí, chức vụ |
| 5 | salary | float | NOT NULL |  | Mức lương |
| 6 | BookStoreID | varchar | NOT NULL | FK | Mã nhà sách (chi nhánh) |

* **Supplier**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã nhà cung cấp |
| 2 | name | varchar | NOT NULL |  | Tên nhà cung cấp |
| 3 | phoneNumber | varchar | NOT NULL |  | SĐT nhà cung cấp |

* **Order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  |  | Mã hóa đơn bán |
| 2 | date | date | NOT NULL |  | Thời gian bán |
| 3 | StaffID | varchar | NOT NULL | FK | Mã nhân viên tham gia |
| 4 | CustomerID | varchar | NOT NULL | FK | Mã khách hàng mua |

* **OrderDetail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | quantity | int |  |  | Số lượng sách bán ra |
| 2 | BookID | varchar | NOT NULL | PK | Mã sách |
| 3 | OrderID | varchar | NOT NULL | PK | Mã hóa đơn bán |

* **Book**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã sách |
| 2 | title | varchar | NOT NULL |  | Tiêu đề sách |
| 3 | price | float | NOT NULL |  | Giá bán |
| 4 | importPrice | float | NOT NULL |  | Giá nhập |

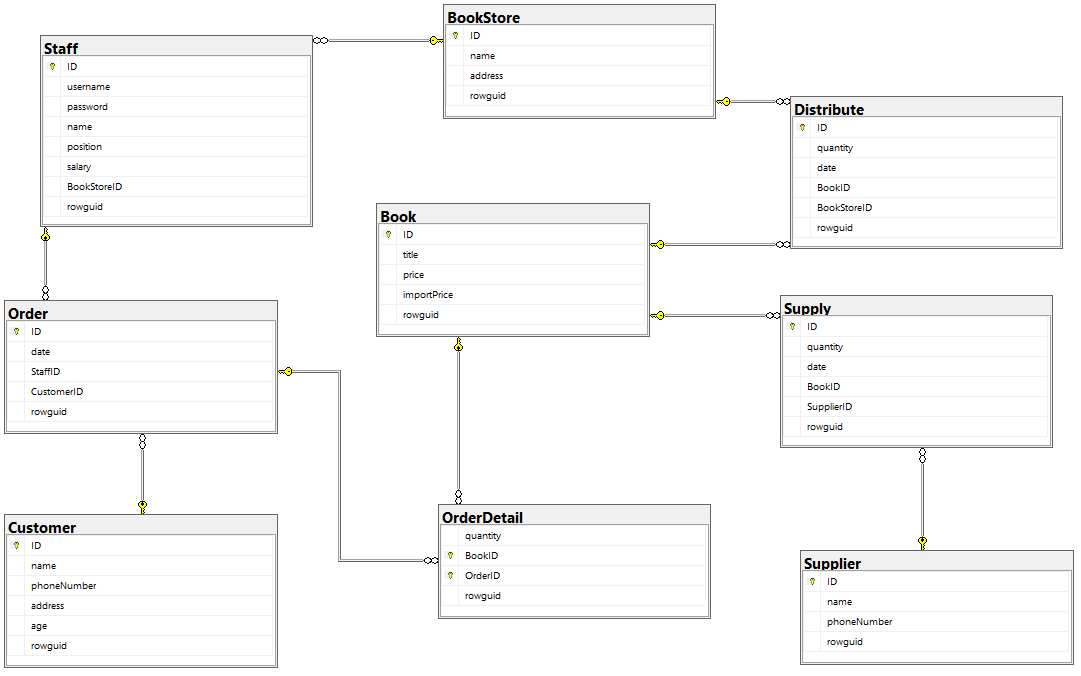
* **Distribute**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | ID | varchar |  | PK | Mã cung cấp sách từ trụ sở đến chi nhánh |
| 2 | quantity | int | NOT NULL |  | Số lượng cung cấp |
| 3 | Date | Date | NOT NULL |  | Ngày cung cấp |
| 3 | BookID | varchar | NOT NULL | FK | Mã sách |
| 4 | BookStoreID | varchar | NOT NULL | FK | Mã nhà sách |

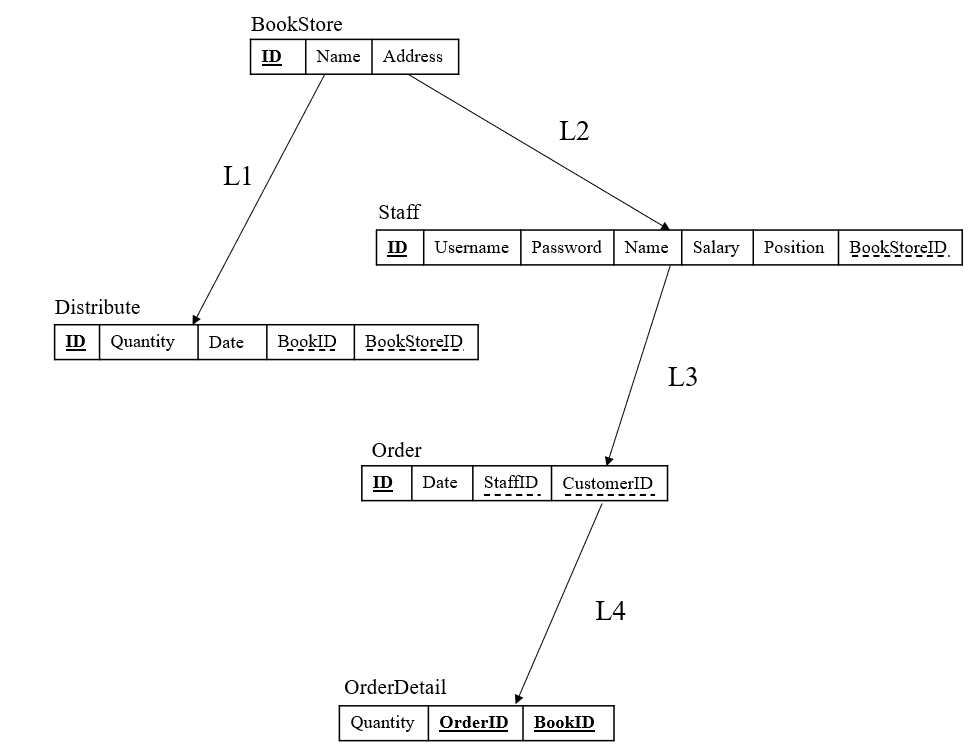
* **Supply**

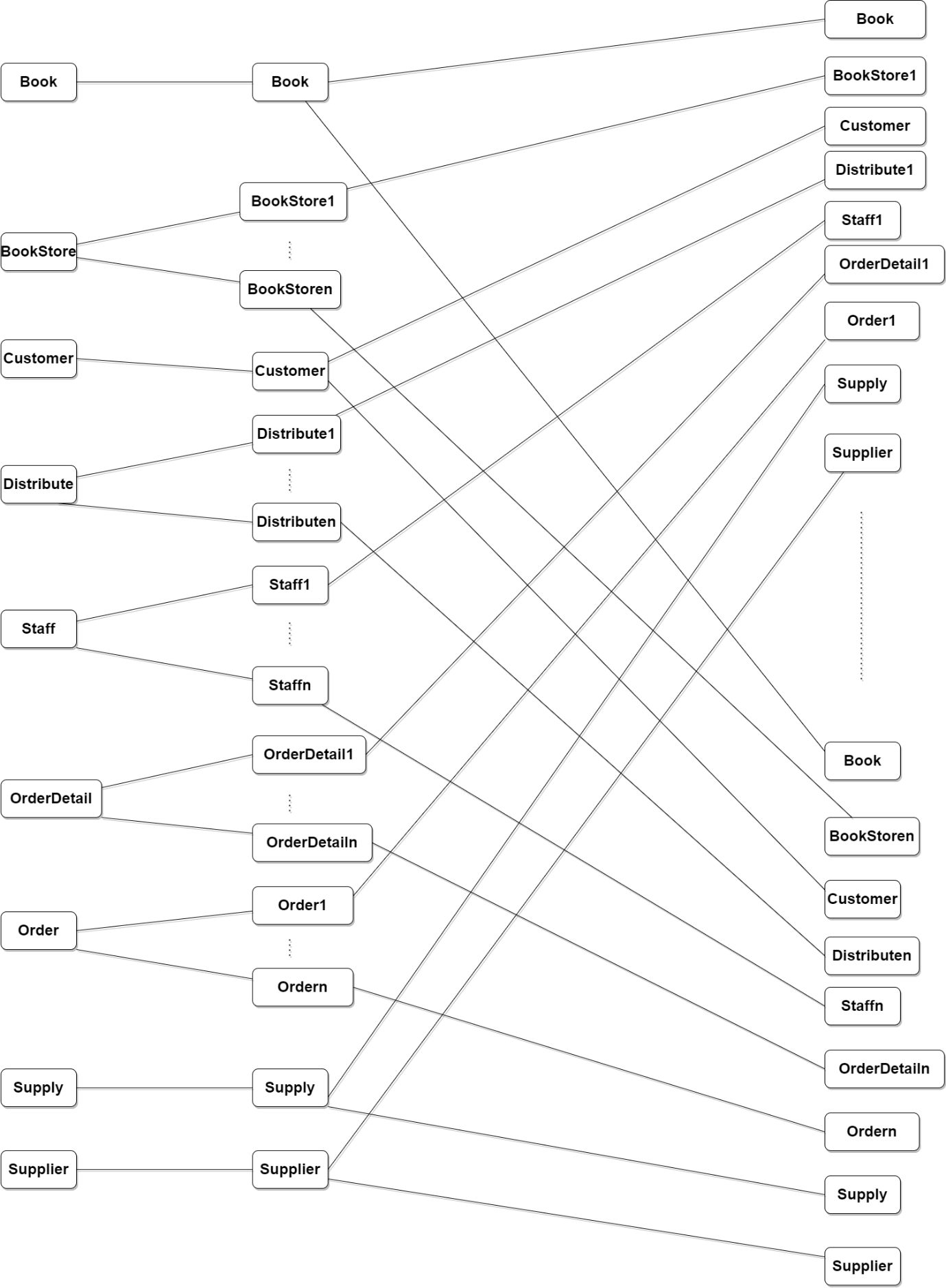
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ràng buộc | Khóa | Mô tả |
| 1 | quantity | int | NOT NULL |  | Số lượng nhập |
| 2 | date | date | NOT NULL |  | Thời gian nhập |
| 3 | BookID | varchar | NOT NULL | PK | Mã sách nhập vào |
| 4 | SupplierID | varchar | NOT NULL | PK | Mã nhà cung cấp |

### 2.2. Mô hình quan hệ giữa các bảng

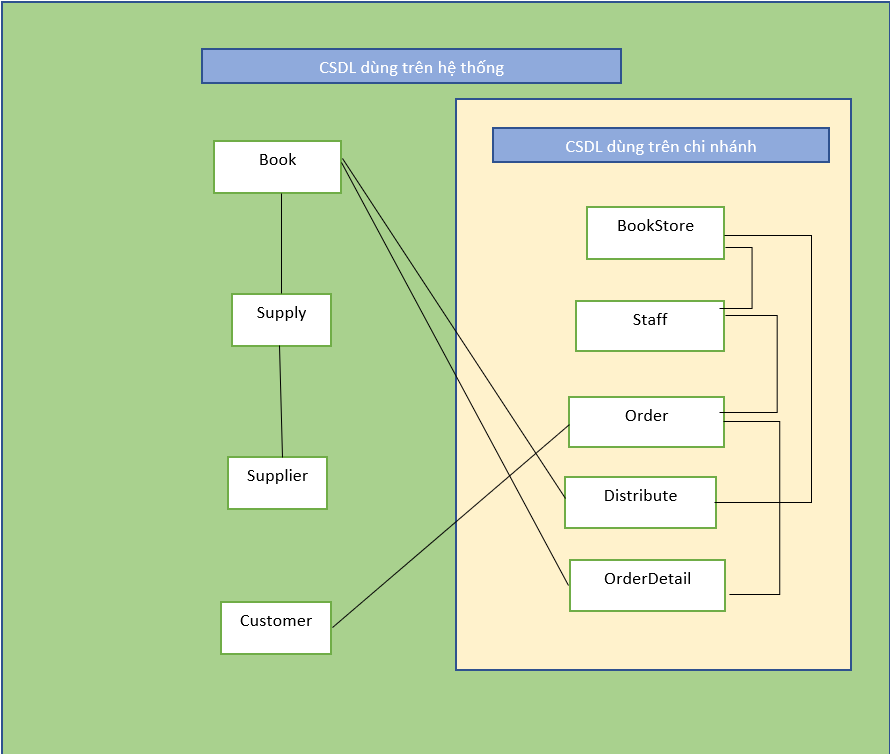


### 2.3. Thiết kế csdlpt

**2.3.1. Lược đồ phục vụ cho phân mảnh ngang dẫn xuất**

**2.3.2. Lược đồ ánh xạ phân mảnh**

**2.3.3. Thiết kế định vị và vẽ sơ đồ định vị**



**2.3.4. Thiết kế phân mảnh ngang, nhân bản**

Phân mảnh ngang nguyên thủy và phân mảnh ngang dẫn xuất để chia quan hệ tổng thể thành 9 mảnh đặt tại 9 vị trí như sau

* Vị trí 1: Dữ liệu đặt tại trạm 1 (chi nhánh Bắc Ninh).
* Vị trí 2: Dữ liệu đặt tại trạm 2 (chi nhánh Hòa Bình).
* Vị trí 3: Dữ liệu đặt tại trạm 3 (chi nhánh Hải Phòng).
* Vị trí 4: Dữ liệu đặt tại trạm 4 (chi nhánh Nam Định).
* Vị trí 5: Dữ liệu đặt tại trạm 5 (chi nhánh Hưng Yên).
* Vị trí 6: Dữ liệu đặt tại trạm 6 (chi nhánh Bắc Giang).
* Vị trí 7: Dữ liệu đặt tại trạm 7 (chi nhánh Thái Nguyên).
* Vị trí 8: Dữ liệu đặt tại trạm 8 (chi nhánh Ninh Bình).
* Vị trí 9: Dữ liệu đặt tại trạm 9 (chi nhánh Quảng Ninh).

Sử dụng phân mảnh ngang nguyên thủy phân rã quan hệ BookStore thành các mảnh ngang BookStoreBN, BookStoreHB, BookStoreHP, BookStoreND, BookStoreHY, …, BookStoreHD.

* BookStoreBN = 𝜎(ID)=”BN”( BookStore)
* BookStoreHB = 𝜎(ID)=”HB”( BookStore)
* BookStoreHP = 𝜎(ID)=”HP”( BookStore)
* BookStoreND = 𝜎(ID)=” ND”( BookStore)
* BookStoreHY = 𝜎(ID)=”HY”( BookStore)
* …
* BookStoreHD = 𝜎(BRANCH\_ID)=”HD”( BookStore)

Sử dụng phân mảnh ngang dẫn xuất phân rã quan hệ Staff thành các mảnh BookStoreBN\_Staff, BookStoreHB\_Staff, BookStoreHP\_Staff, BookStoreND\_Staff, BookStoreHY\_Staff, …, BookStoreHD\_Staff

* BookStoreBN\_Staff = Staff ⋉ BookStoreBN
* BookStoreHP\_Staff = Staff ⋉ BookStoreHP
* BookStoreND\_Staff = Staff ⋉ BookStoreND
* BookStoreHY\_Staff = Staff ⋉ BookStoreHY
* …
* BookStoreHD\_Staff = Staff ⋉ BookStoreHD

Sử dụng phân mảnh ngang dẫn xuất phân rã quan hệ Distribute thành các mảnh BookStoreBN\_ Distribute, BookStoreHB\_ Distribute, BookStoreHP\_ Distribute, BookStoreND\_ Distribute, BookStoreHY\_ Distribute, …, BookStoreHD\_ Distribute

* BookStoreBN\_ Distribute = Distribute ⋉ BookStoreBN
* BookStoreHP\_ Distribute = Distribute ⋉ BookStoreHP
* BookStoreND\_ Distribute = Distribute ⋉ BookStoreND
* BookStoreHY\_ Distribute = Distribute ⋉ BookStoreHY
* …
* BookStoreHD\_ Distribute = Distribute ⋉ BookStoreHD

Sử dụng phân mảnh ngang dẫn xuất phân rã quan hệ Order thành các mảnh BookStoreBN\_ Order, BookStoreHB\_ Order, BookStoreHP\_ Order, BookStoreND\_ Order, BookStoreHY\_ Order, …, BookStoreHD\_ Order

* BookStoreBN\_Order = Order ⋉ BookStoreBN\_Staff
* BookStoreHP\_Order = Order ⋉ BookStoreHP\_Staff
* BookStoreND\_Order = Order ⋉ BookStoreND\_Staff
* BookStoreHY\_Order = Order ⋉ BookStoreHY\_Staff
* …
* BookStoreHD\_Order = Order ⋉ BookStoreHD\_Staff

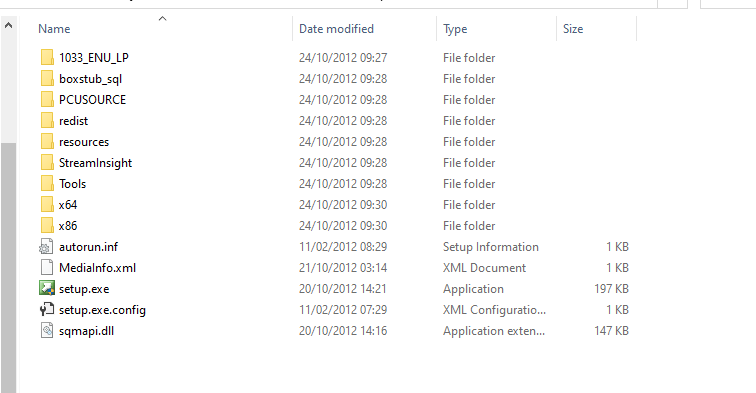
Sử dụng phân mảnh ngang dẫn xuất phân rã quan hệ OrderDetail thành các mảnh BookStoreBN\_ OrderDetail, BookStoreHB\_ OrderDetail, BookStoreHP\_ OrderDetail, BookStoreND\_ OrderDetail, BookStoreHY\_ OrderDetail, …, BookStoreHD\_ OrderDetail

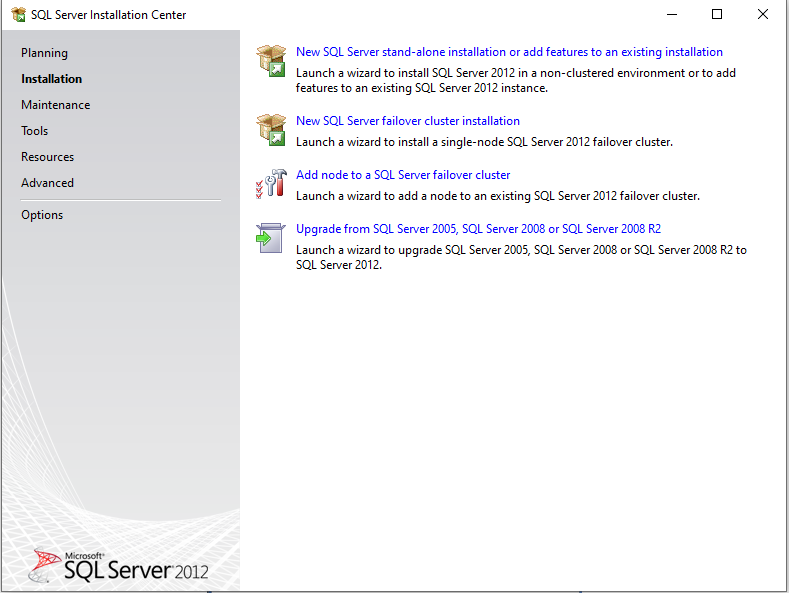
* BookStoreBN\_OrderDetail = OrderDetail ⋉ BookStoreBN\_Order
* BookStoreHP\_OrderDetail = OrderDetail ⋉ BookStoreHP\_Order
* BookStoreND\_OrderDetail = OrderDetail ⋉ BookStoreND\_Order
* BookStoreHY\_OrderDetail = OrderDetail ⋉ BookStoreHY\_Order
* …
* BookStoreHD\_OrderDetail = OrderDetail ⋉ BookStoreHD\_Order

# IV. Cài đặt

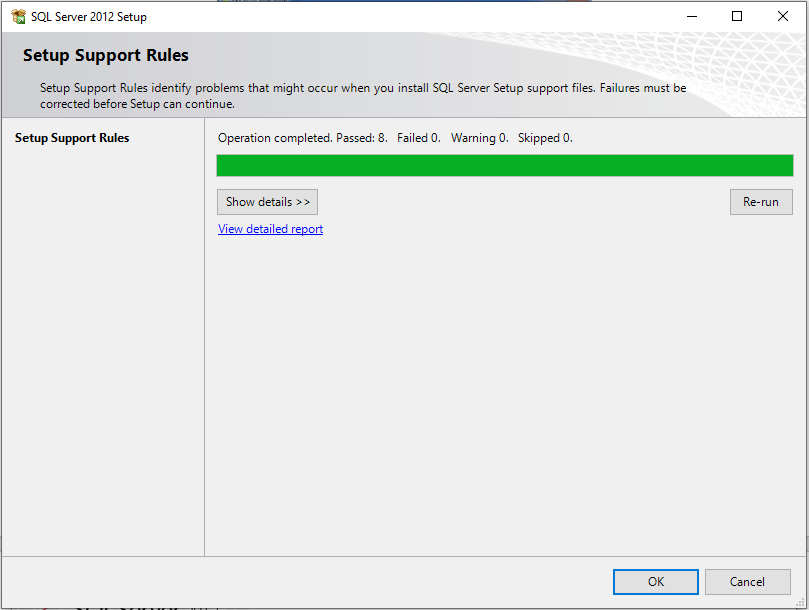
## Cài đặt SQL Server

Download file SQL Server 2012, sau đó mở file download, chọn setup.exe

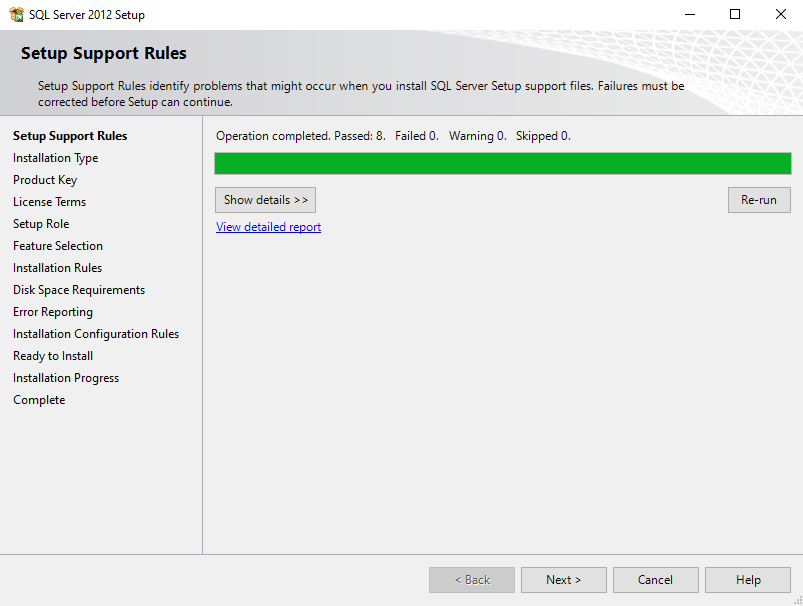


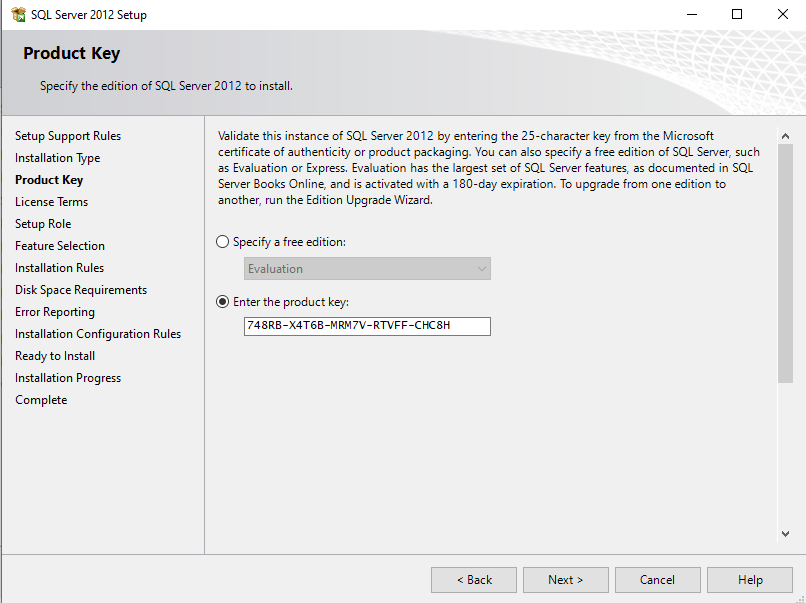
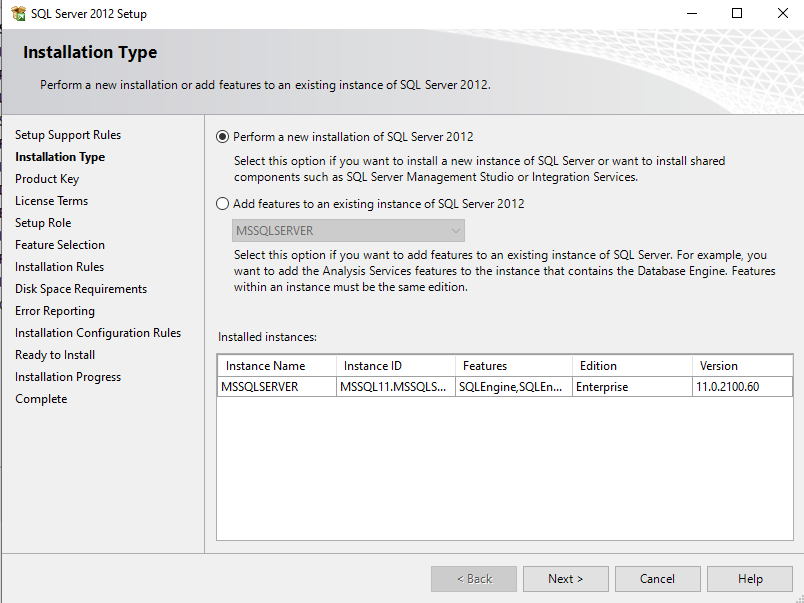


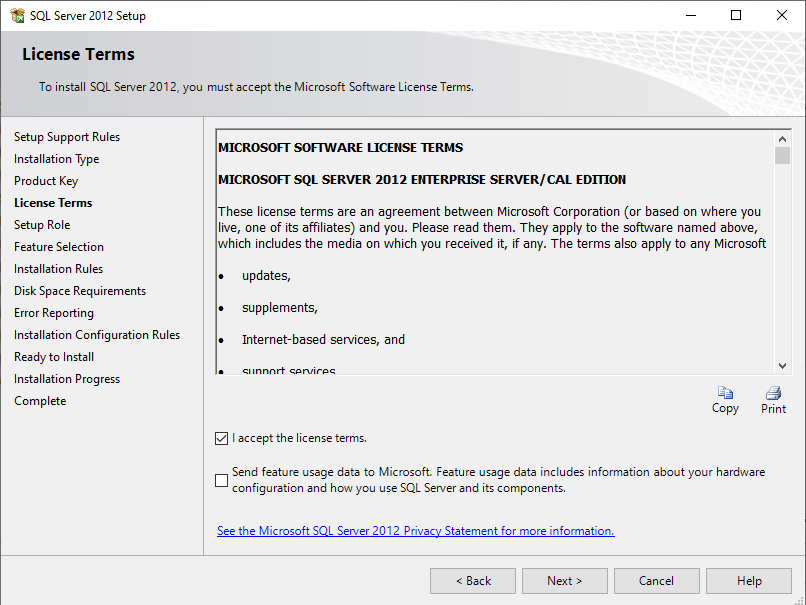
Chọn “Installation” -> “New SQL Server stand-alone installation or add features to an existing installation”.

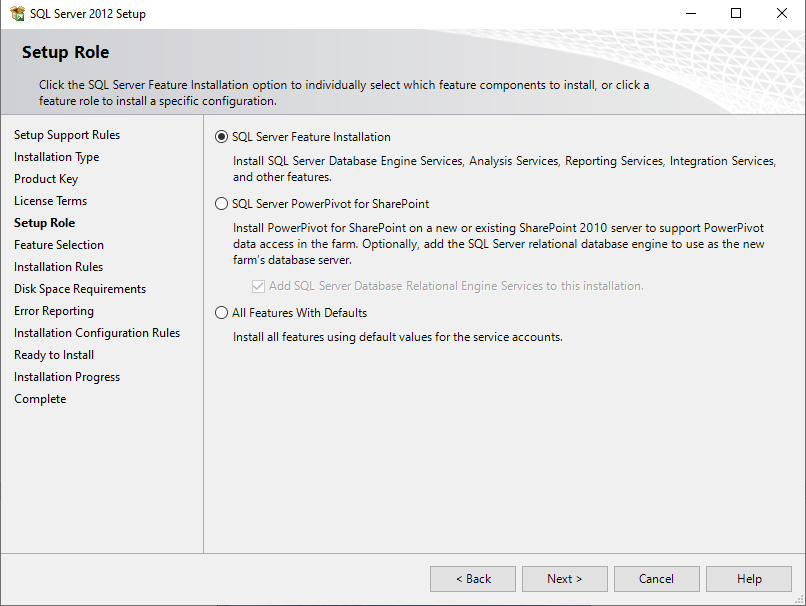


Chọn OK





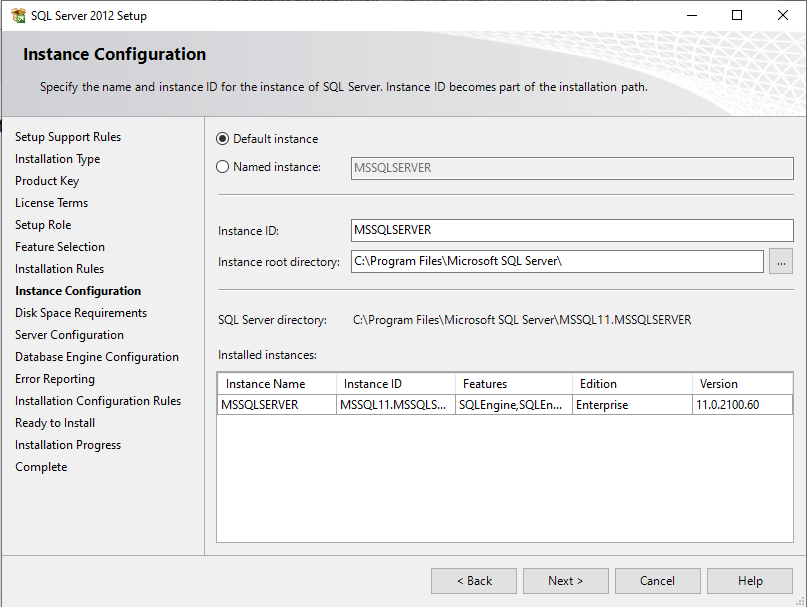




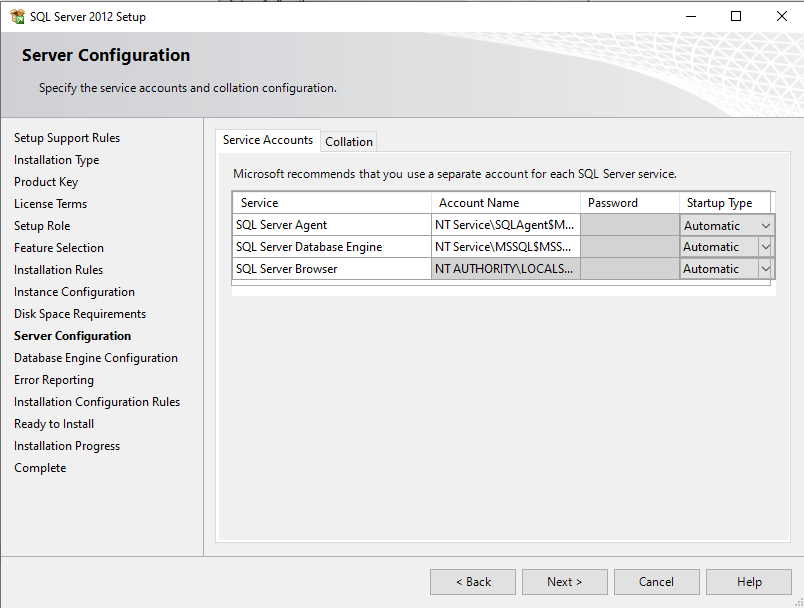
Chọn “Next” ở các cửa sổ tiếp theo



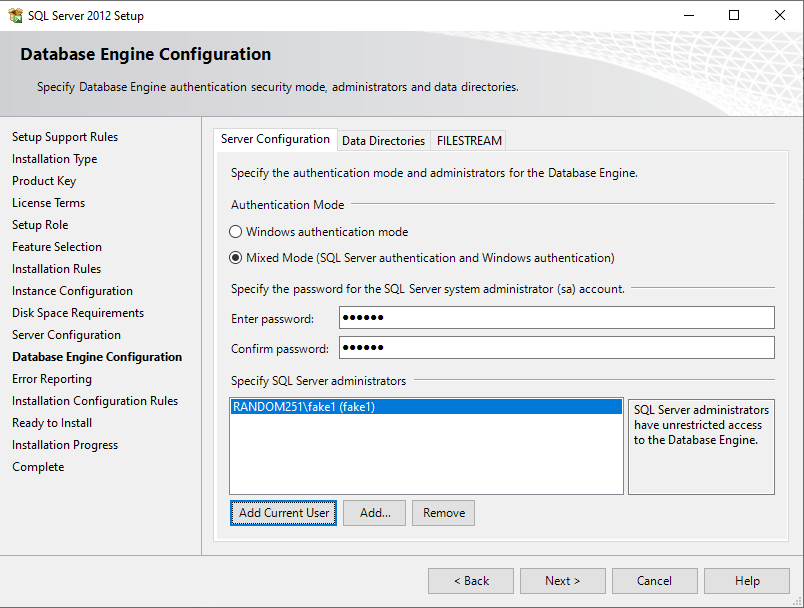
Tích chọn “SQL Server Replication”, “SQL Client Connectivity SDK”, “Management Studio”, sau đó chọn “Next”.



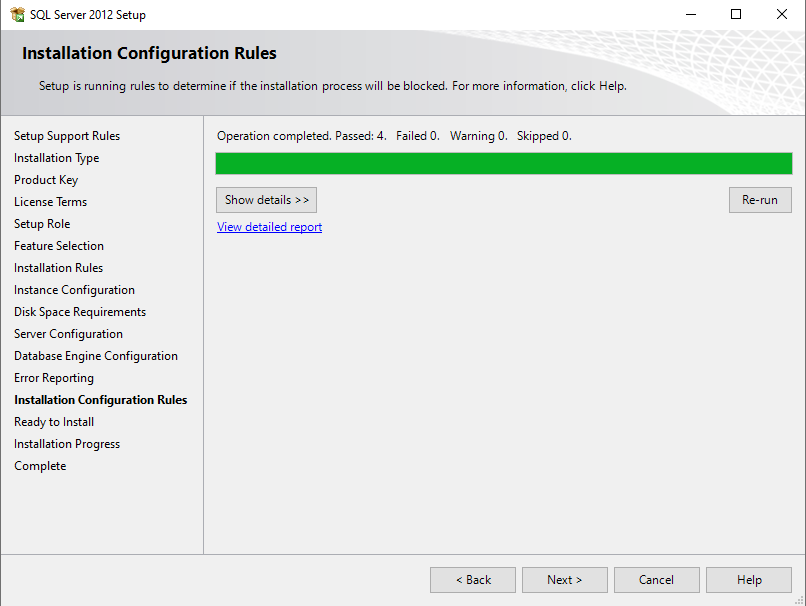
Đặt tên mặc định hoặc tùy chỉnh Instance và Instance ID

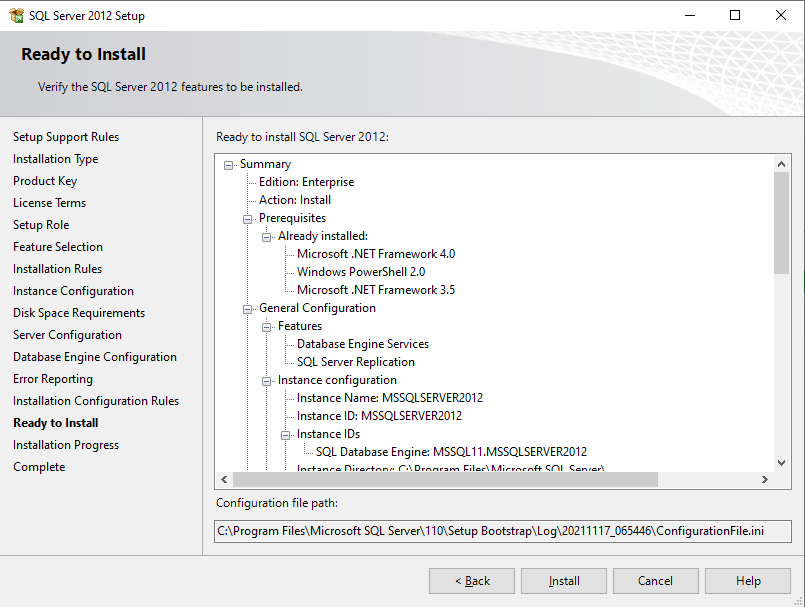


Tùy chỉnh “Startup Type” là “Automatic”



Chọn “Mixed Mode” -> Nhập mật khẩu tài khoản sa -> Chọn “Add Current User” sau đó chọn “Next”





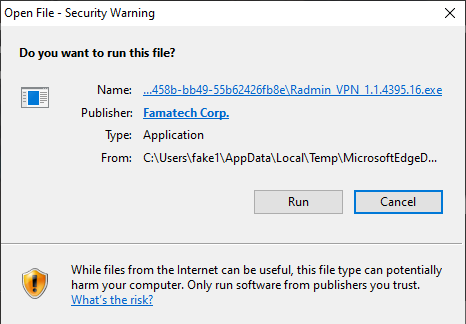
Chọn Install

## Cài đặt VPN

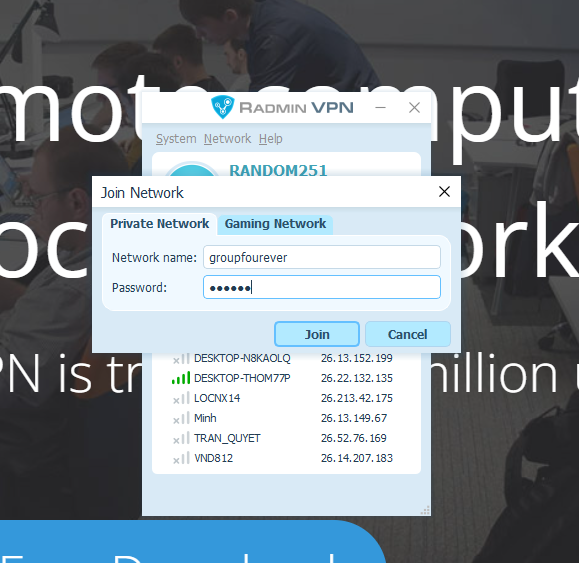
### Cài đặt Radmin VPN

Tải Radmin VPN từ trang <https://www.radmin-vpn.com/>





Mở file download, chọn “Run”

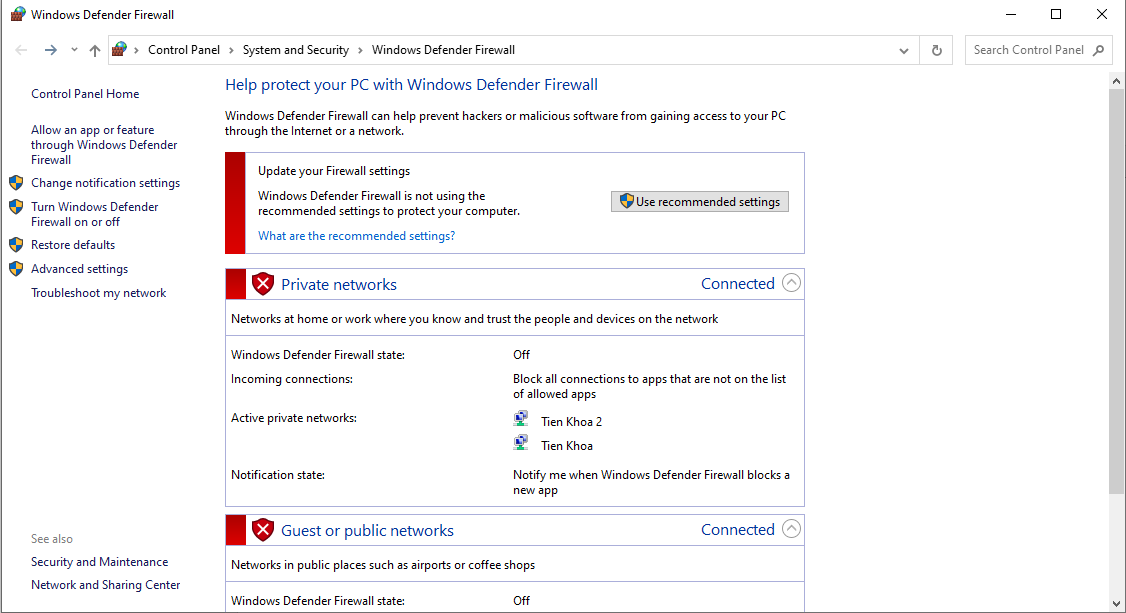


Chọn “Join an Existing Network” -> Nhập “Network name” và “Password”

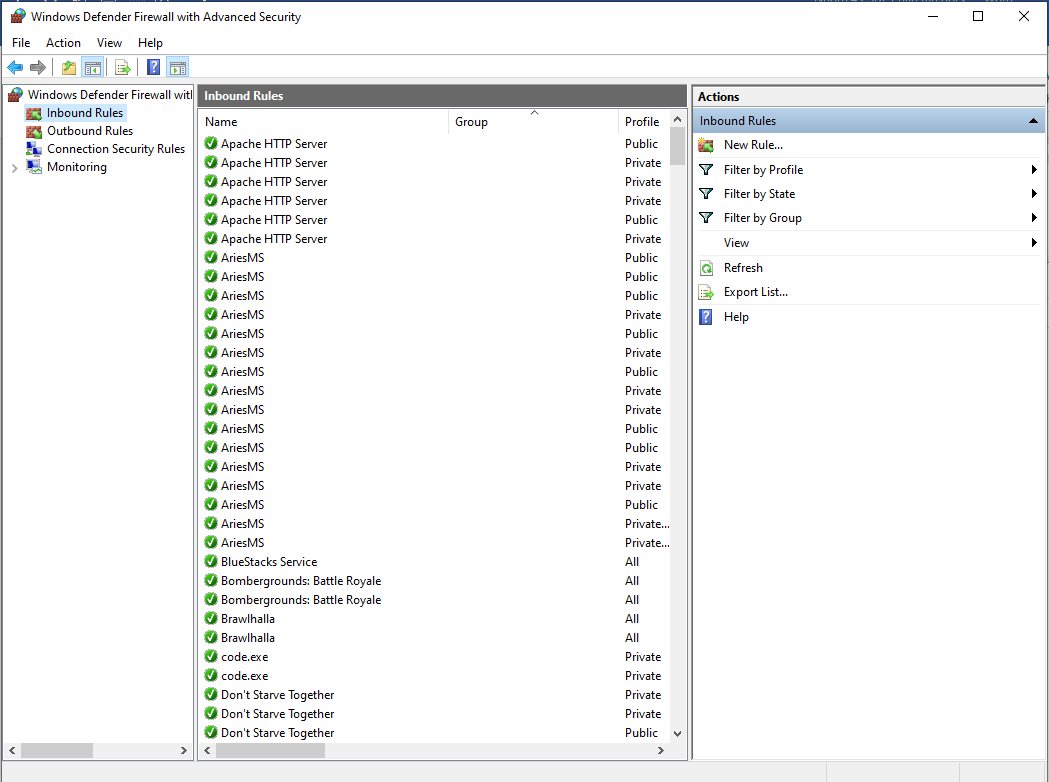
### Cấu hình tường lửa



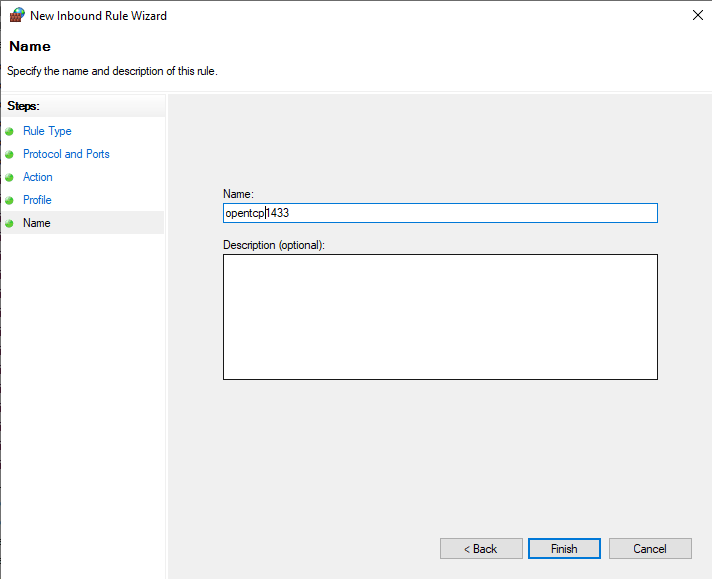
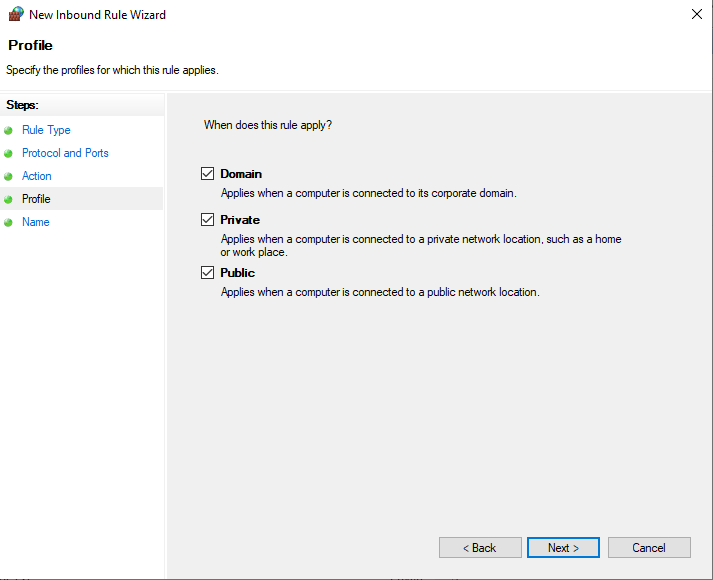
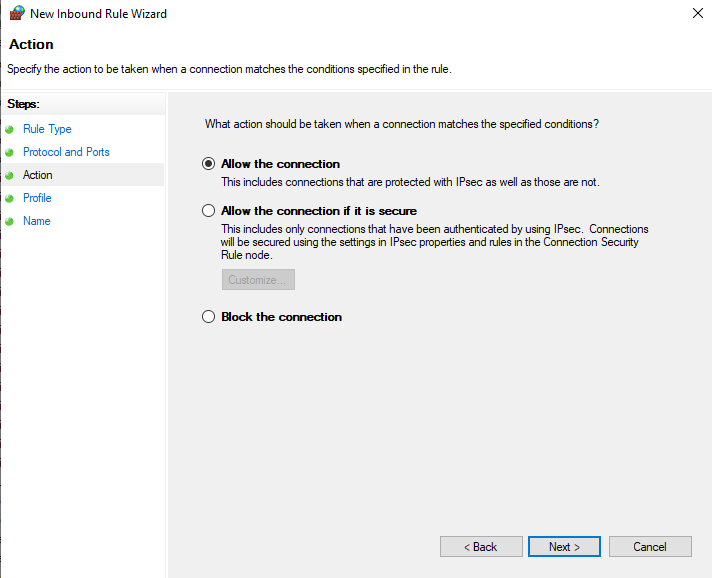
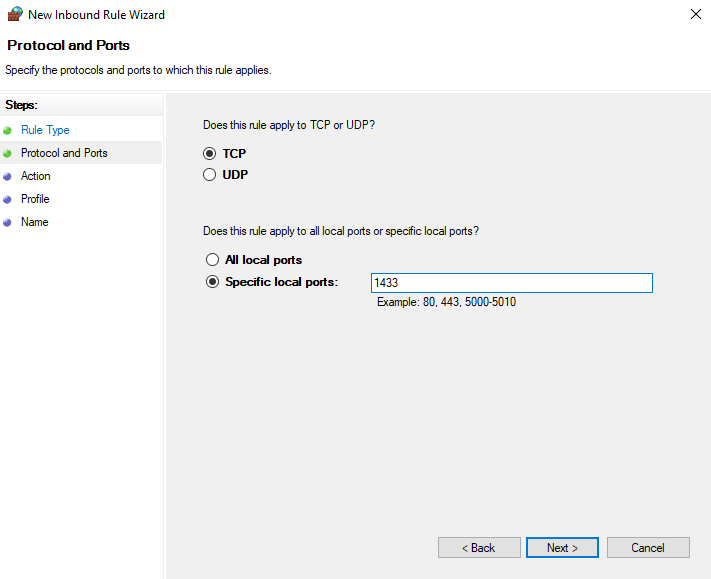
Tiến hành tắt tường lửa



Chọn “Advanced settings”

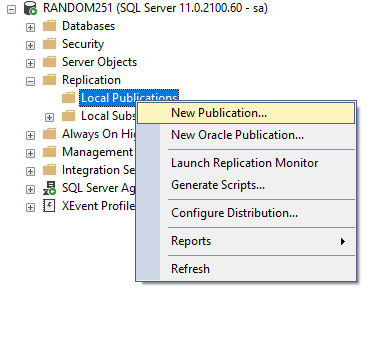


Chọn “Inbound Rules” -> chọn “New Rule…”

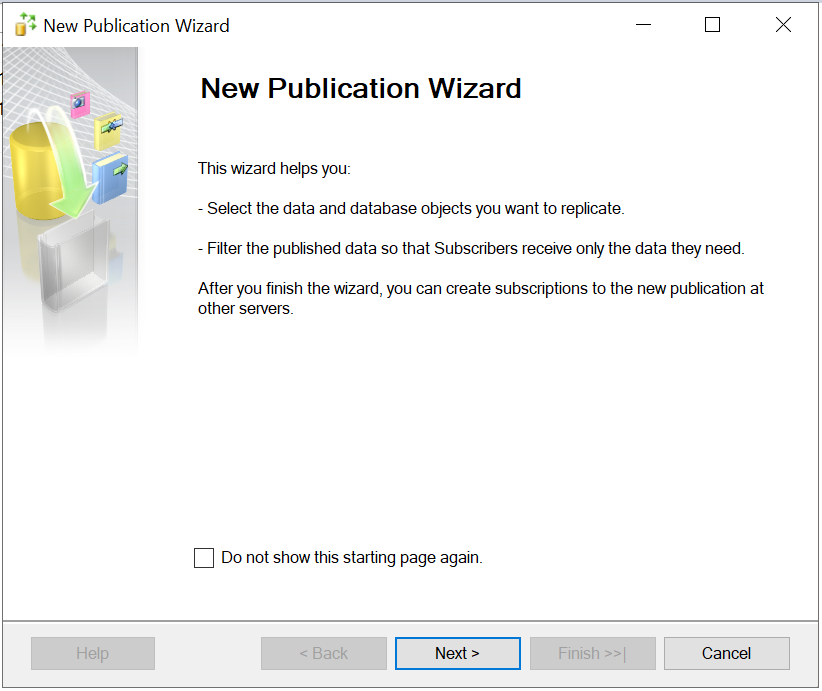


Cấu hình cổng TCP 1433

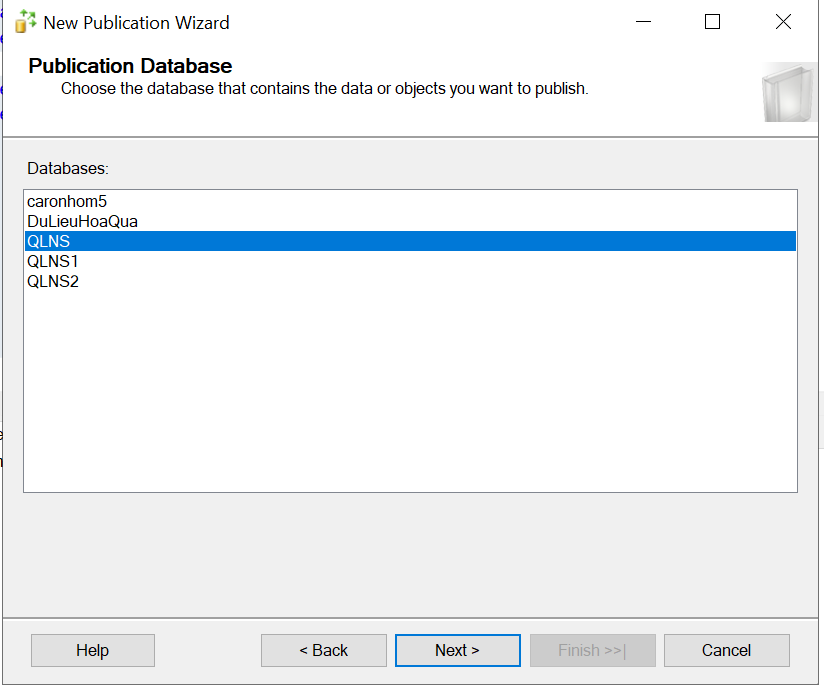
## Tạo Publication



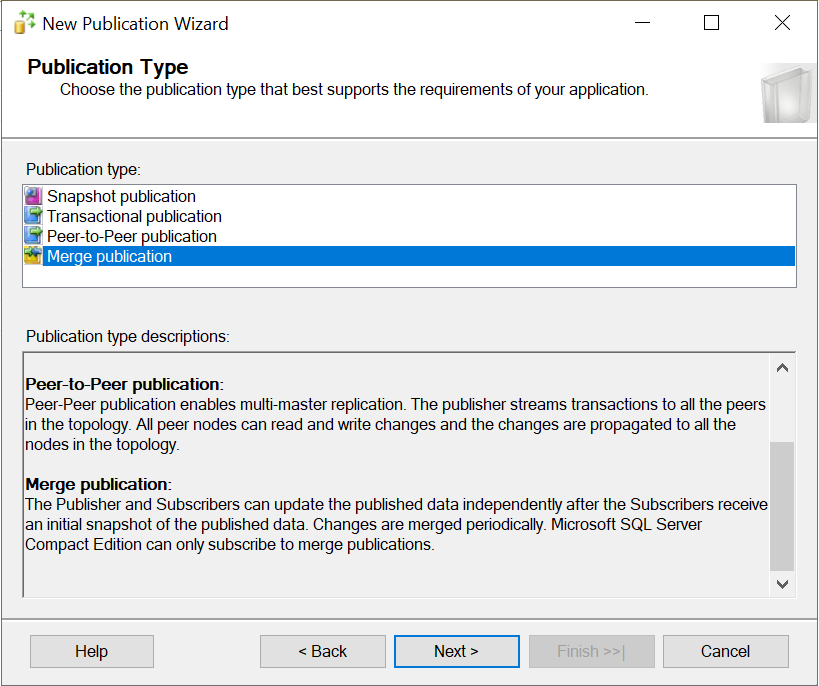
Chọn “Replication” -> chọn “Local Publications” -> chuột phải chọn “New Publication”



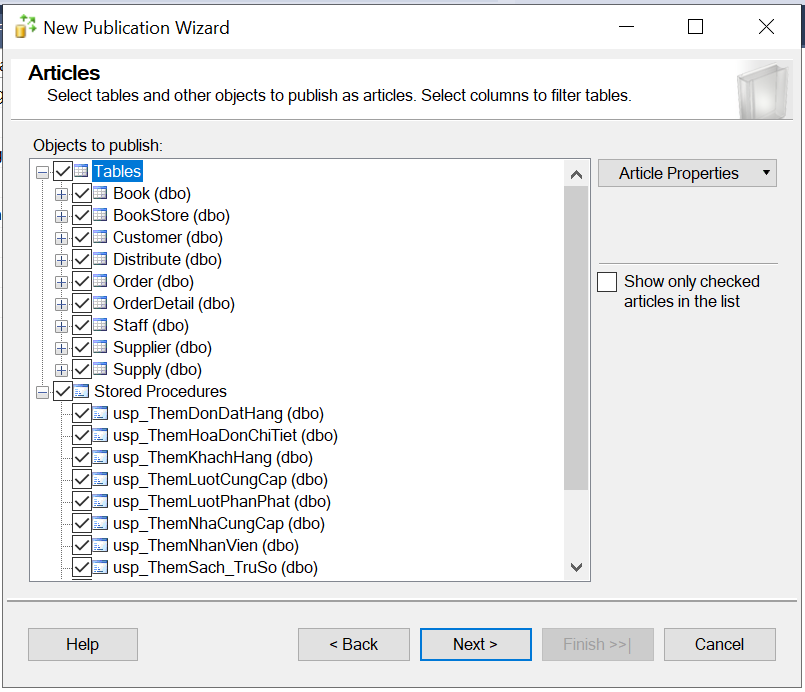
Chọn “Next”



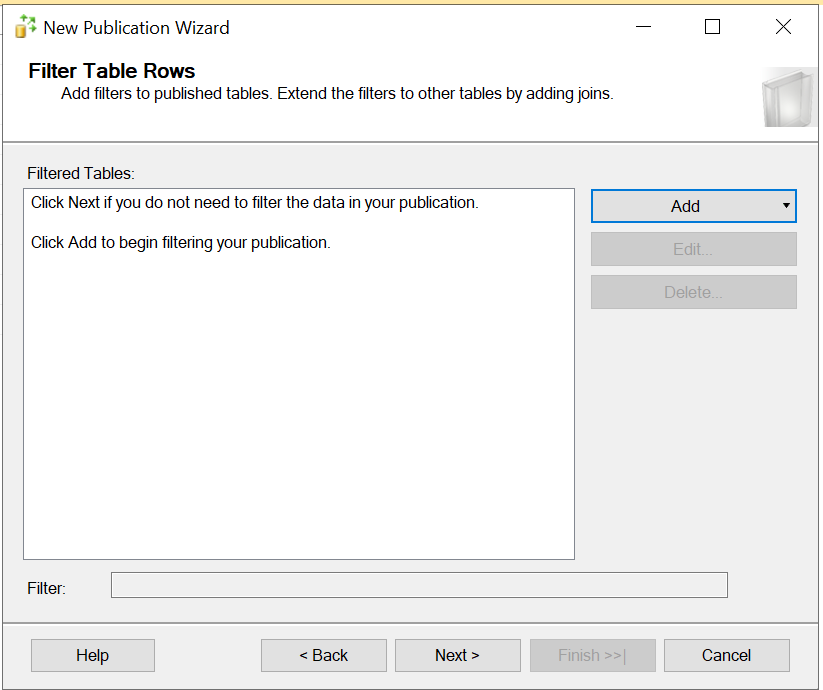
Chọn database muốn phân tán -> chọn “Next”



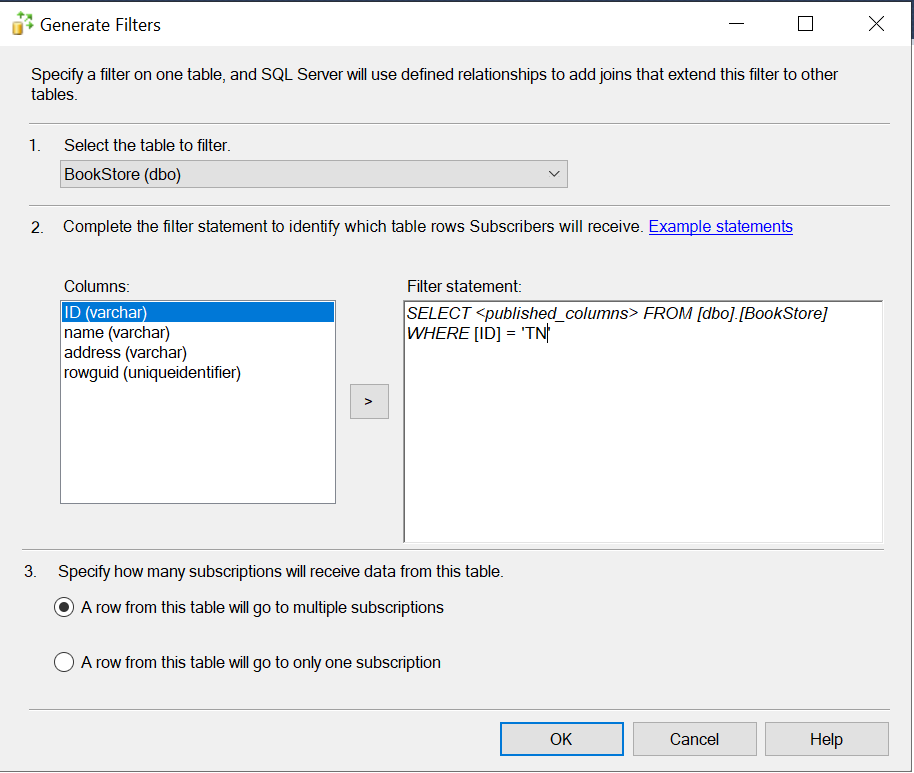
Chọn “Merge publication” -> chọn “Next”



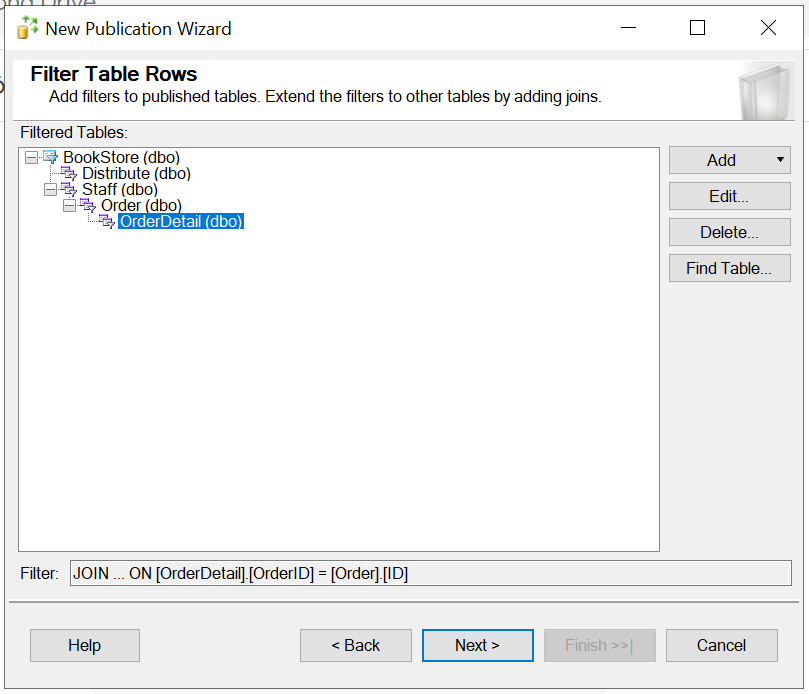
Chọn table và procedures -> chọn “Next”



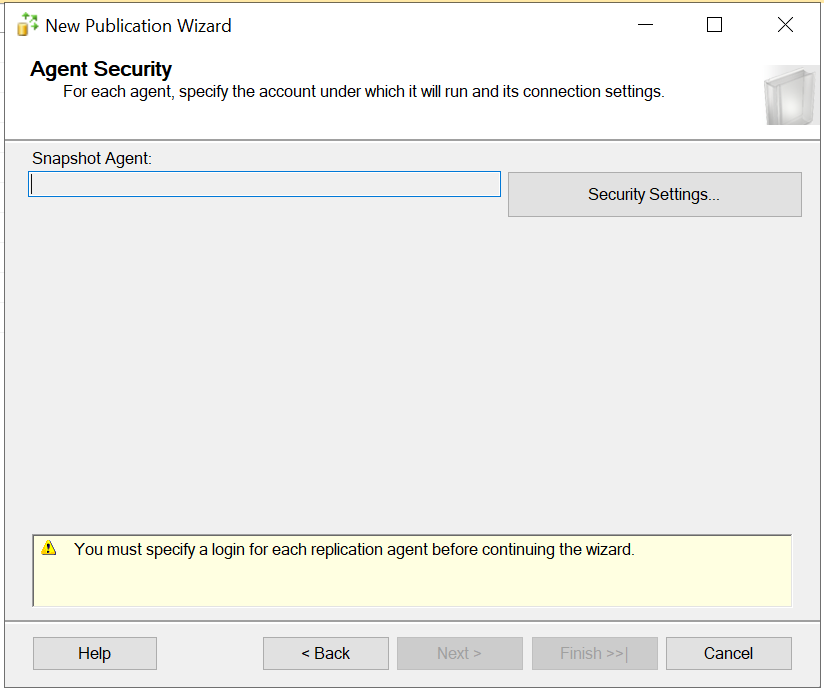
Chọn “Add”



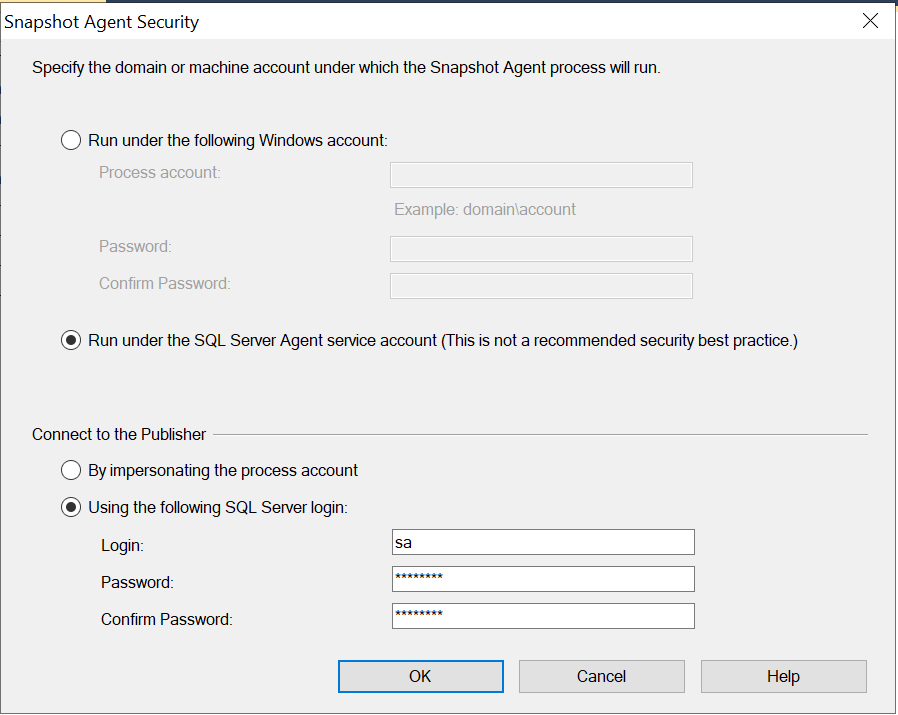
Tiến hành phân mảnh theo BookStore



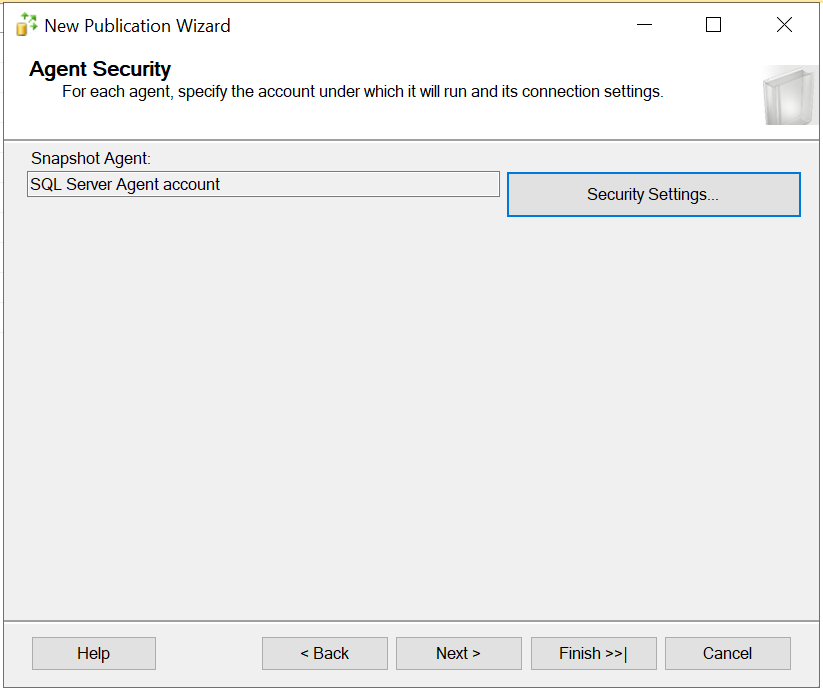
Tiến hành chọn phân mảnh xog -> chọn “Next”



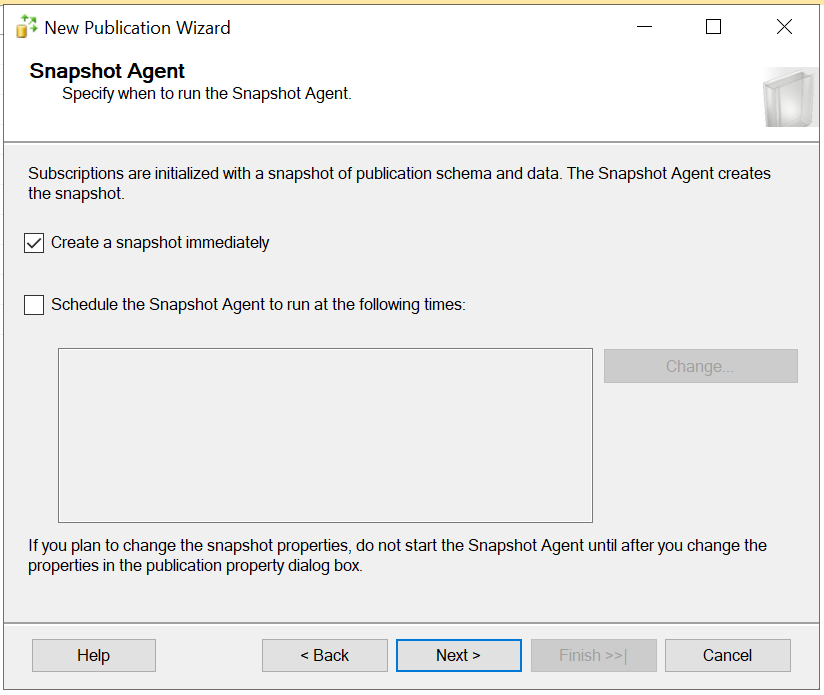
Chọn “Security Settings”



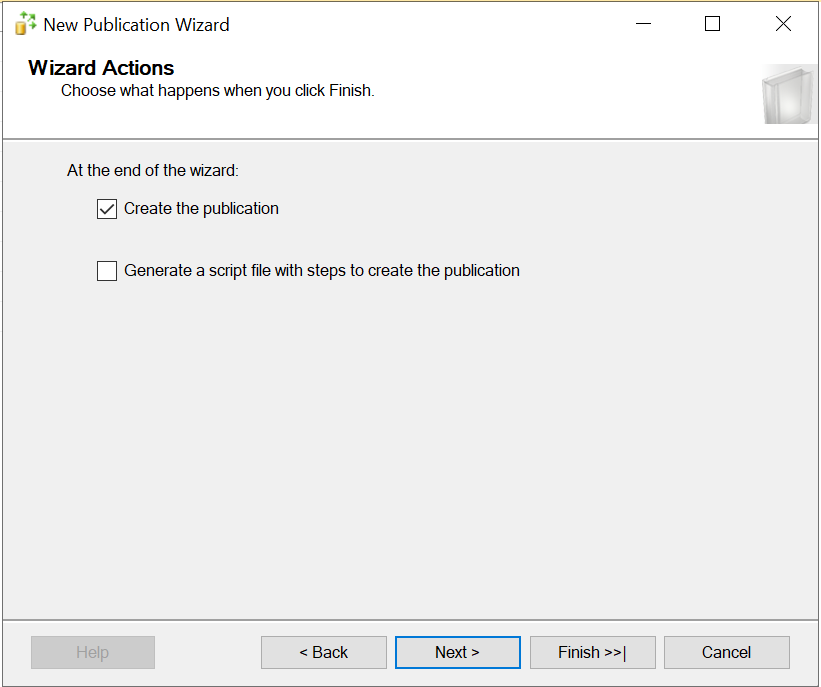
Chọn “Using the following SQL Server login” -> nhập tài khoản sa của máy chủ



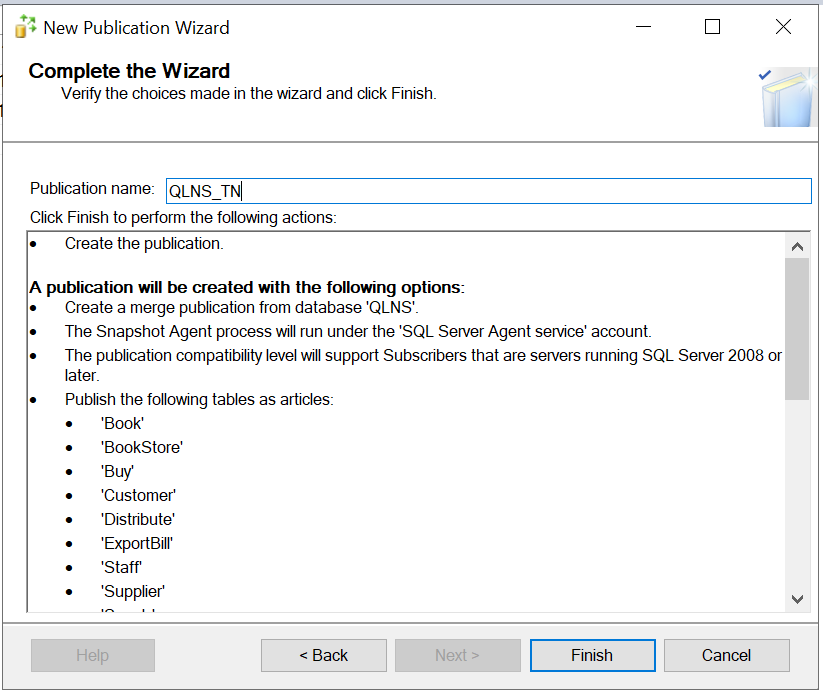
Chọn “Next”



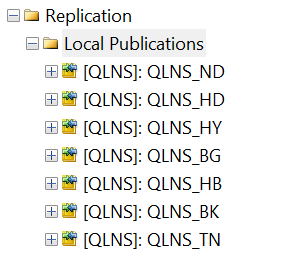
Chọn “Next”



Chọn “Next”

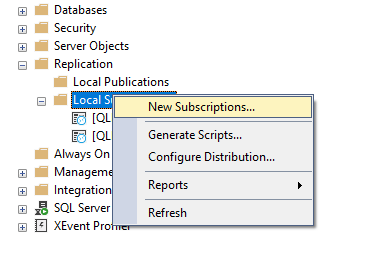


Đặt tên mảnh -> chọn “Finish”

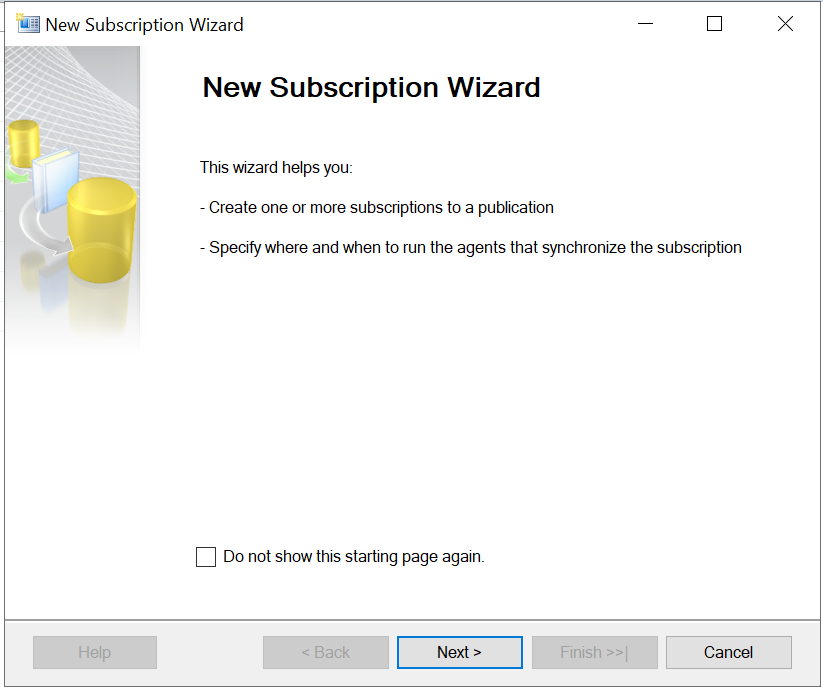


Kết quả

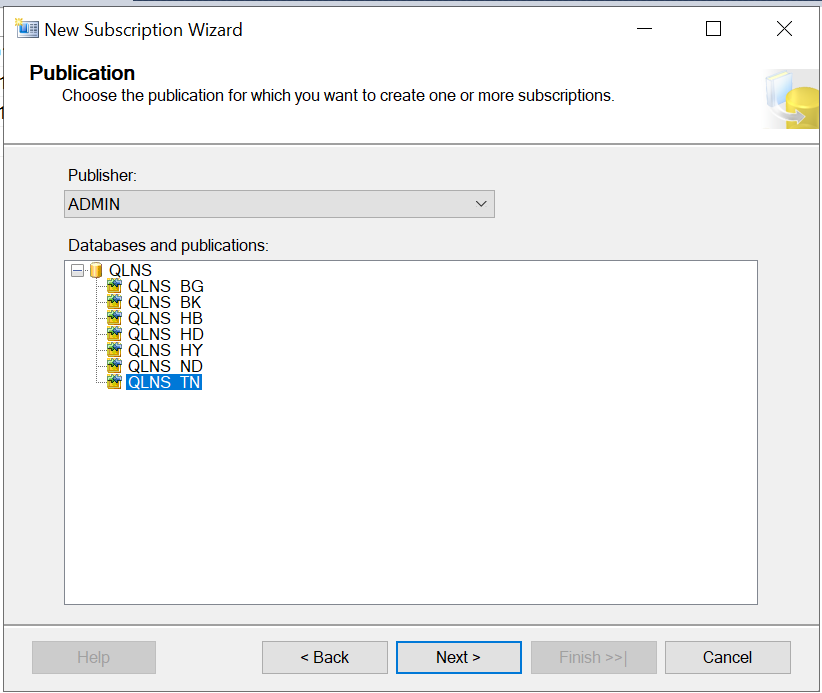
## Tạo Subscriptions



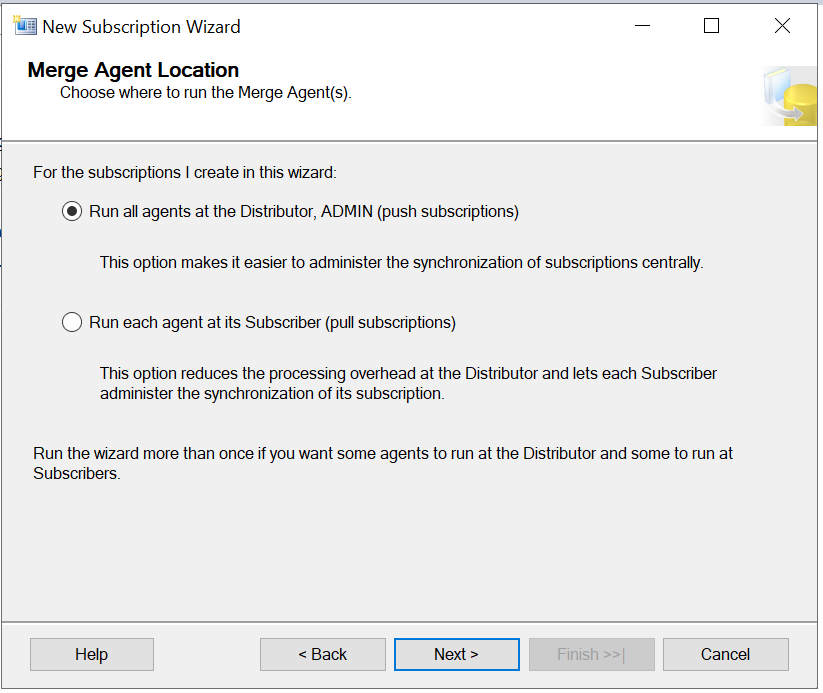
Chọn “Replication” -> chọn “New Subscriptions…”



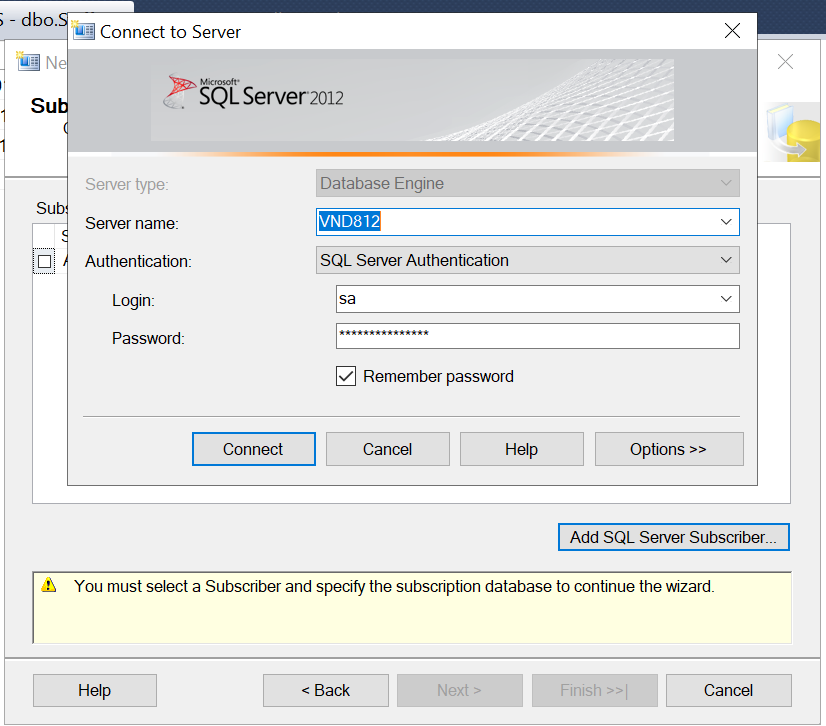
Chọn “Next”



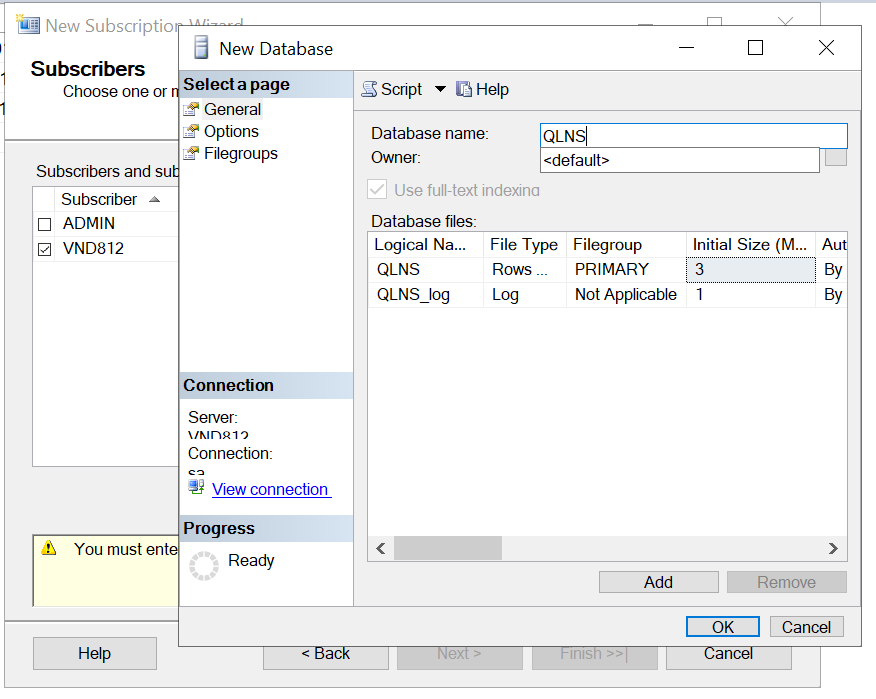
Chọn Publication



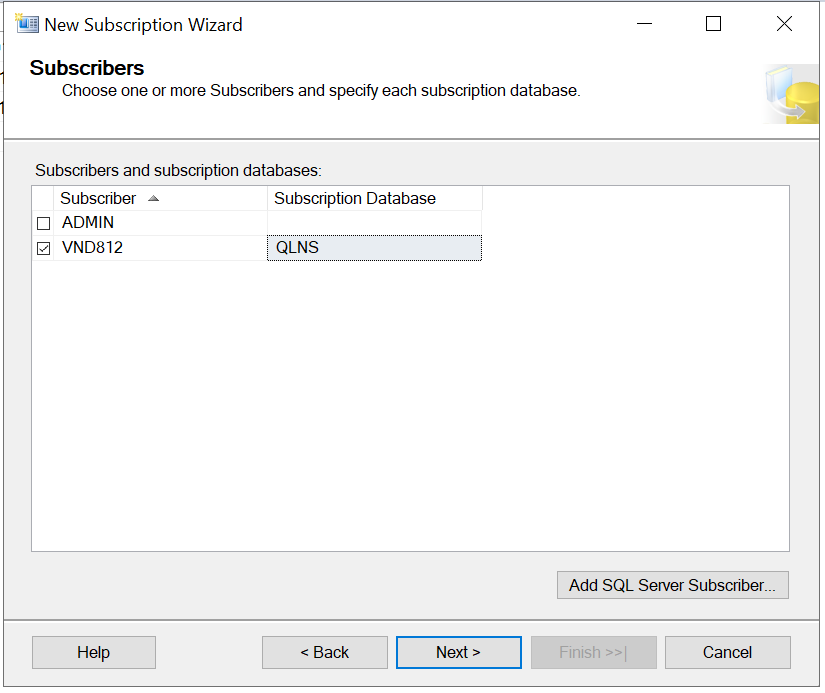
Chọn “Next”



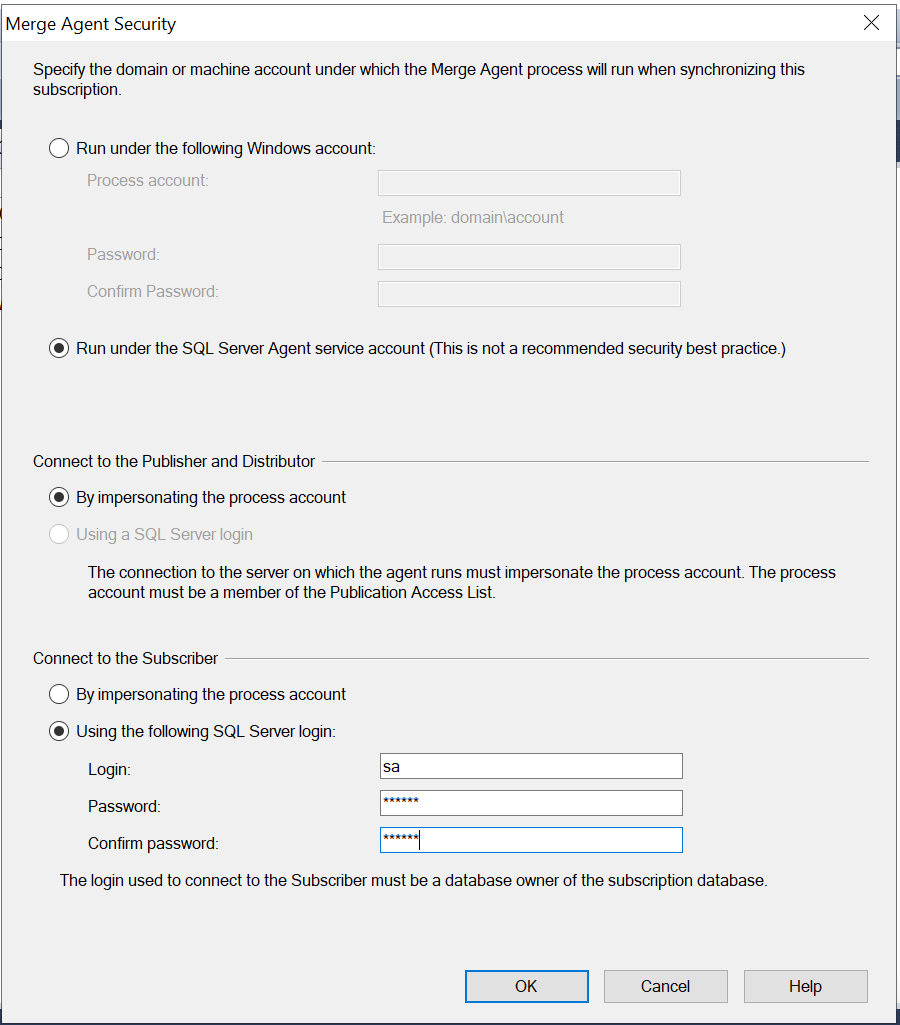
Chọn “Server name”, nhập tài khoản sa ở máy đó, chọn “Connect” -> chọn “Add SQL Server Subcriber…”



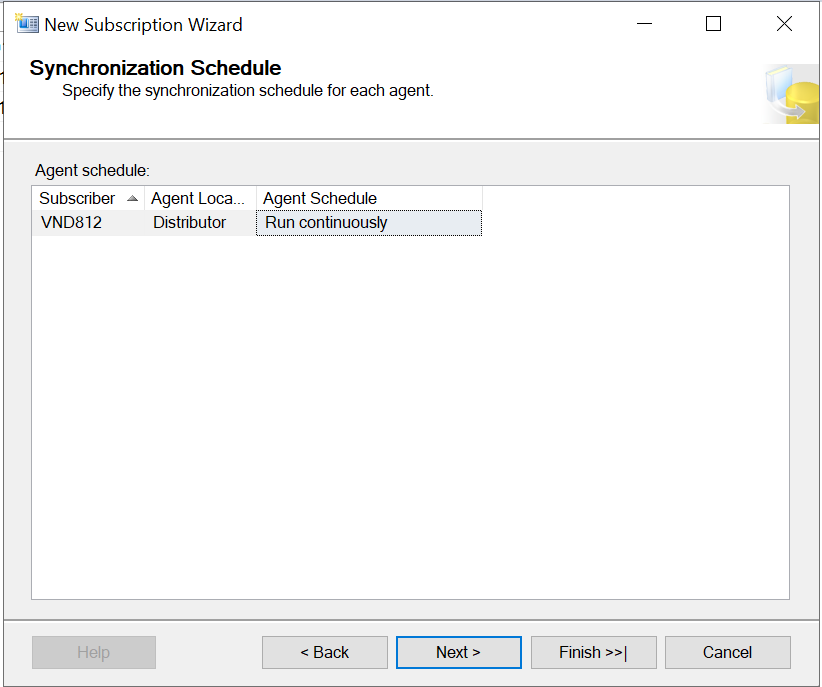
Chọn “OK”



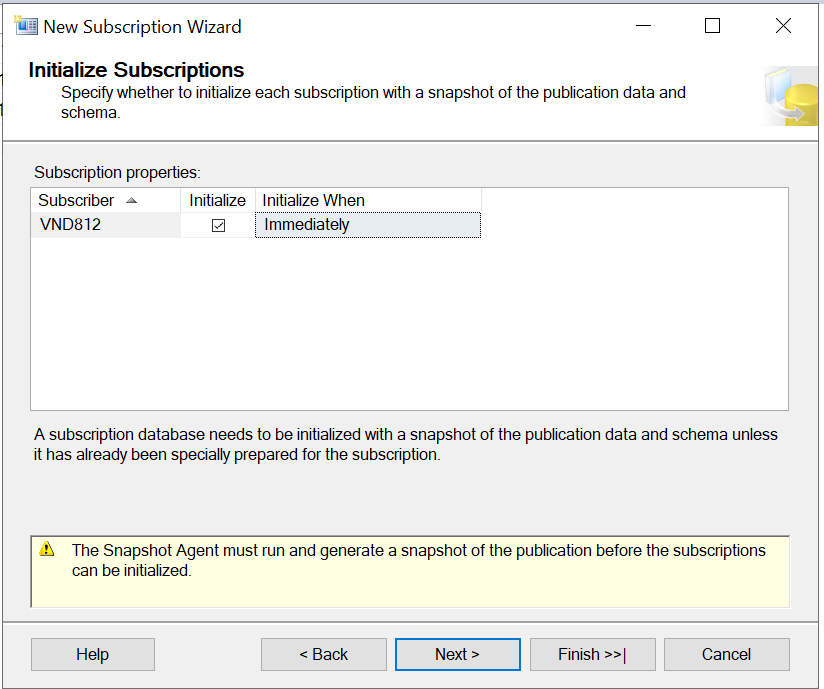
Chọn “Next”



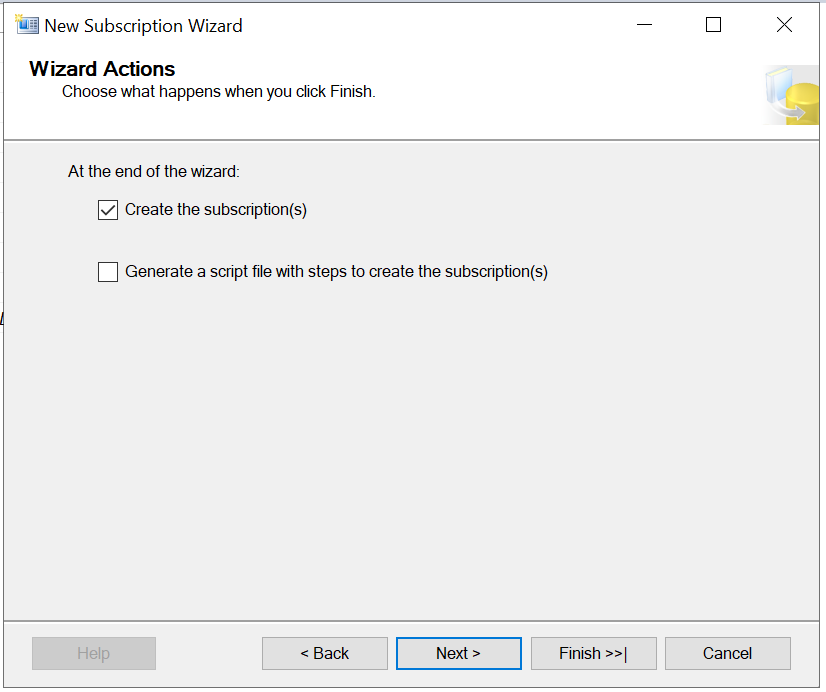
Chọn “Using the following SQL Server login” -> nhập tài khoản sa máy chủ -> chọn “OK”



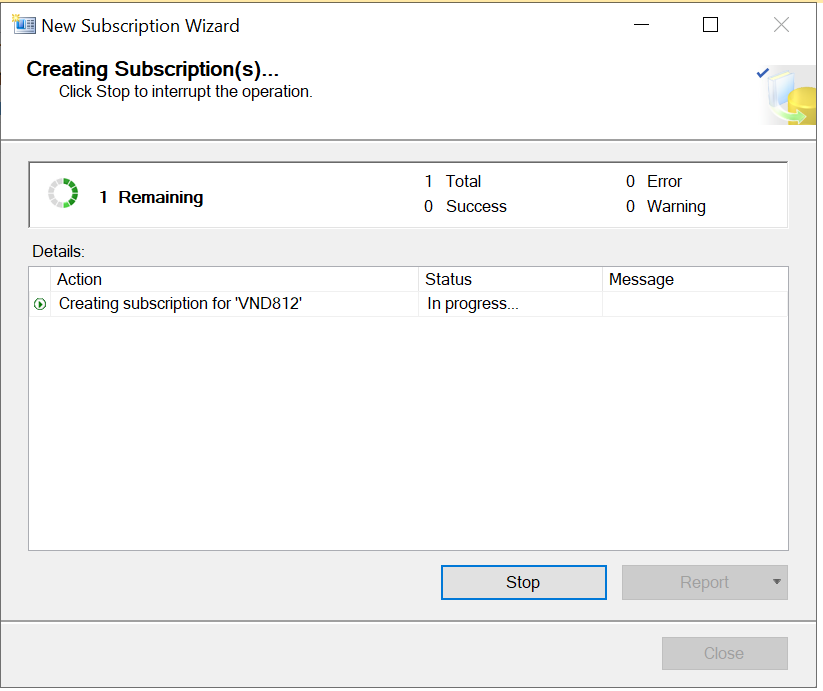
Cấu hình “Run continuously” -> chọn “Next”



Cấu hình “Immediately” -> chọn “Next”



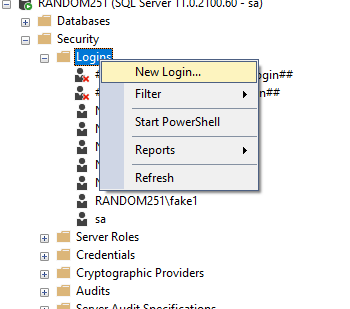
Chọn “Next”



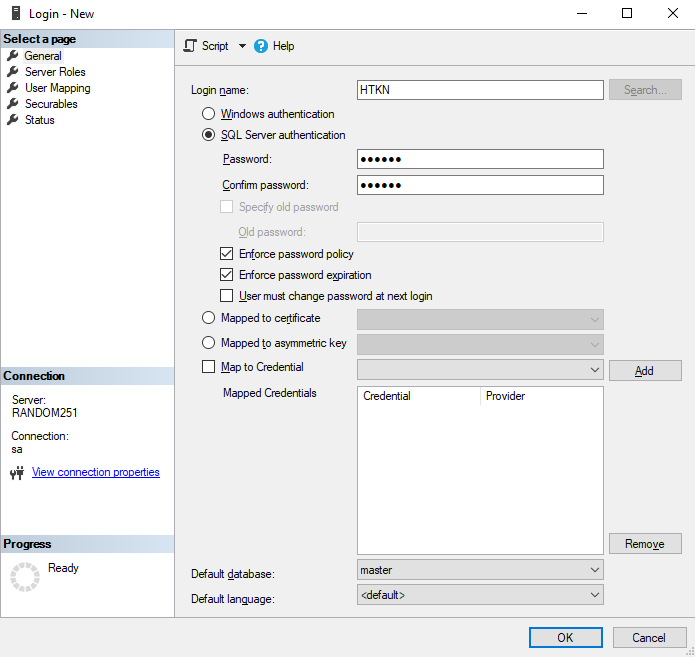
Hoàn tất

## Tạo Link server

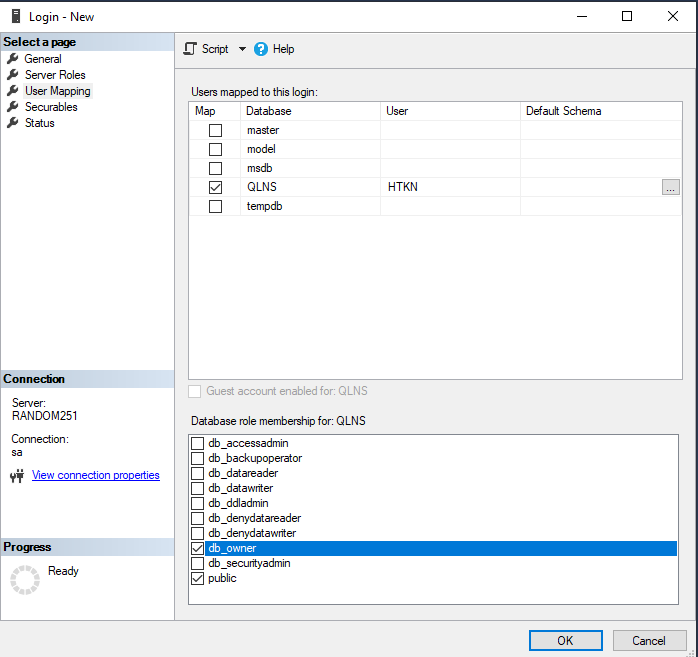
### Tạo tài khoản



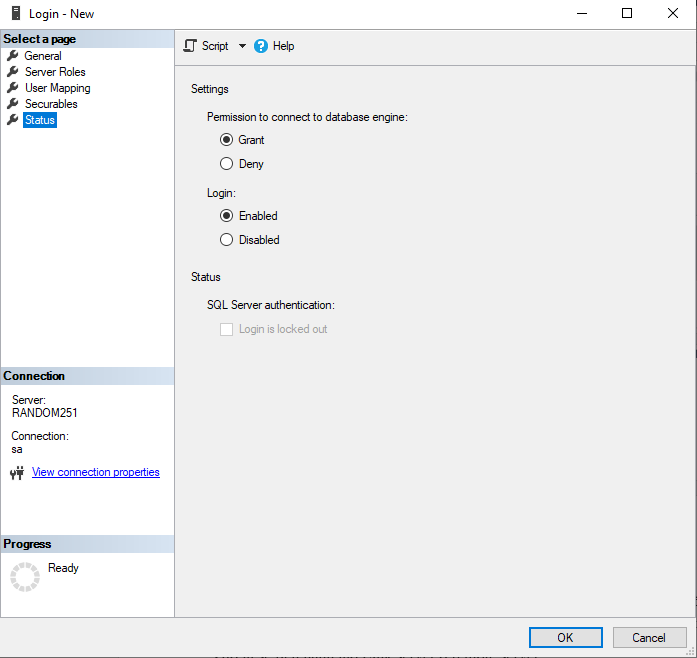
Tạo 1 tài khoản login mới tại server được link. Chuột phải vào thư mục “Logins” và chọn “New Login”



Đặt tên cho tài khoản, chọn SQL Server athentication và nhập mật khẩu

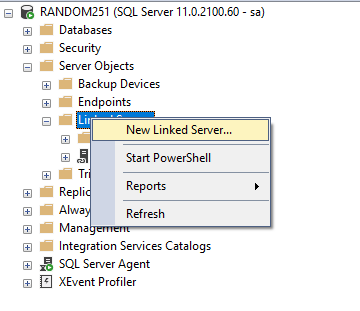


Chọn phần UserMapping -> chọn cơ sở dữ liệu cần kết nối tới -> chọn “db\_owner”

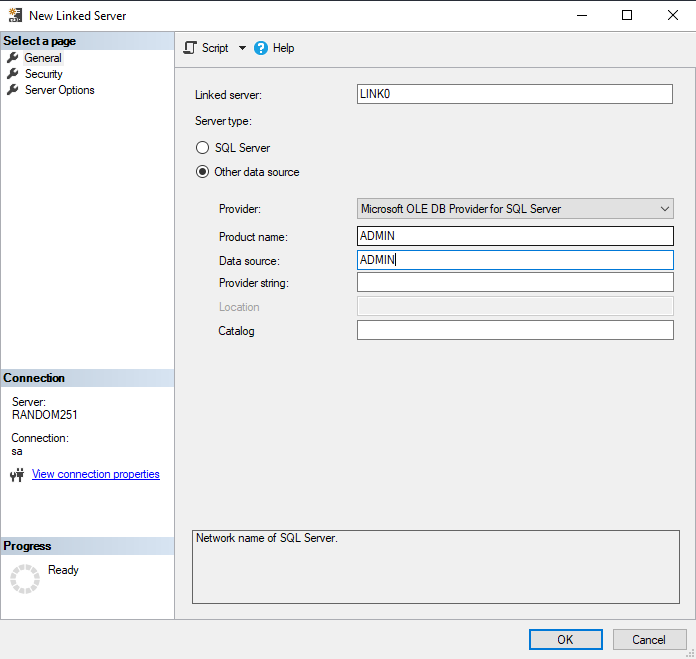


Chọn phần “Status” -> chọn “Grant” và “Enabled” -> chọn “OK” để hoàn tất

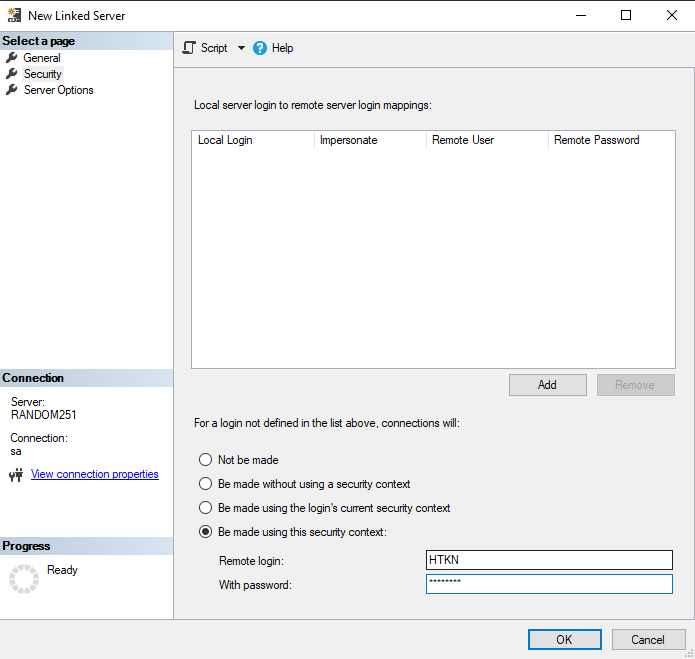
### Tạo Link server



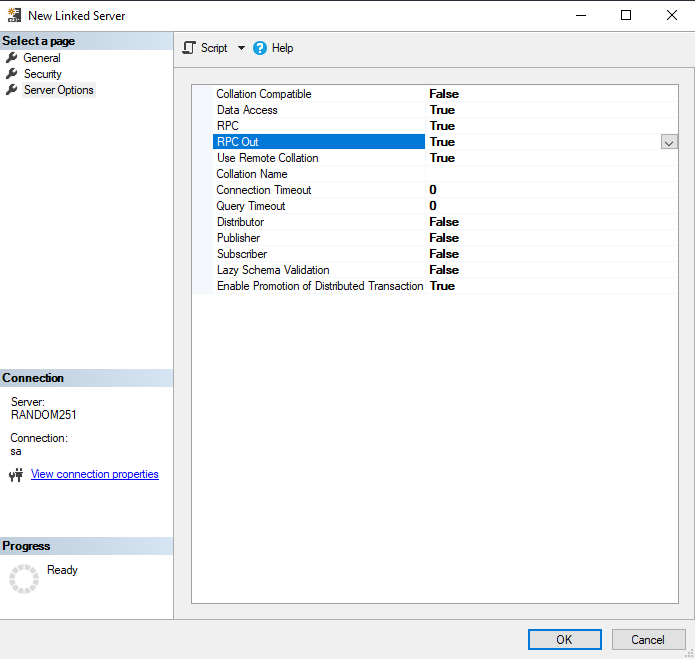
Chọn “Server Objects” -> chọn “Linked Servers” -> chuột phải chọn “New Linked Server…”



Đặt tên link server -> chọn “Other data source” -> gõ tên của server cần link vào phần “Product name” và “Data source”



Chọn phần “Sercurity” -> chọn “Be made using this sercurity context” -> Nhập tài khoản kết nối của server cần link



Chọn phần “Server Options” -> thiết lập “RPC” và “RPC Out” thành “True”

Nhấn OK để hoàn tất

## Trigger

**book\_trigger**

create trigger book\_trigger on dbo.Book

after insert, update, delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc them, sua, xoa sach tai tru so'

rollback tran

END

**bookstore\_trigger**

CREATE trigger bookstore\_trigger on dbo.BookStore

after insert, update, delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc them, sua, xoa chi nhanh tai tru so'

rollback tran

END

**create\_update\_distribute\_trigger**

CREATE trigger cre\_upd\_distribute\_trigger on dbo.Distribute

after insert,update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@ide like 'GG%')

BEGIN

declare @bookID varchar(50) = (select BookID from inserted)

declare @sumofdistributed int = (select sum(quantity) from Distribute where BookID = @bookID)

declare @sumofsupplied int = (select sum(quantity) from Supply where BookID = @bookID)

declare @avaiable int =@sumofsupplied- @sumofdistributed

print cast (@avaiable as varchar(50))

if(@avaiable >= 0)

begin

print('Ok')

end

else

begin

print('So luong phan phat vuot qua so luong con trong kho')

rollback tran

end

END

else

BEGIN

print 'ID phai bat dau bang GG'

rollback tran

END

**create\_update\_supplier\_trigger**

CREATE trigger cre\_upd\_supplier\_trigger on dbo.Supplier

after insert,update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@ide like 'GG%')

BEGIN

print 'OK'

END

else

BEGIN

print 'ID phai bat dau bang GG'

rollback tran

END

**create\_update\_supply\_trigger**

CREATE trigger cre\_upd\_supply\_trigger on dbo.Supply

after insert, update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@ide like 'GG%')

begin

print('Ok')

end

else

begin

print('ID phai bat dau bang GG')

rollback tran

end

**cutomer\_trigger**

create trigger [dbo].[customer\_trigger] on [dbo].[Customer]

after insert, update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@ide like 'GG%')

begin

print('Ok')

end

else

begin

print('ID Khach hang bat dau bang GG')

rollback tran

end

**del\_customer\_trigger**

create trigger [dbo].[del\_customer\_trigger] on [dbo].[Customer]

after delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc xoa khach hang tai tru so'

rollback tran

END

**distribute\_trigger**

CREATE trigger distribute\_trigger on dbo.Distribute

after insert, update, delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc them, sua, xoa phan phat tai tru so'

rollback tran

END

**order\_trigger**

Create trigger order\_trigger on dbo.[Order]

after insert, update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@@SERVERNAME = 'ADMIN')

begin

if(@ide like 'ND%' OR @ide like 'HD%' OR @ide like 'HY%' OR @ide like 'BG%' OR @ide like 'HB%' OR @ide like 'BK%' OR @ide like 'TN%' OR @ide like 'QN%' OR @ide like 'HP%')

begin

print 'Thanh cong'

end

else

begin

print 'Don hang khong thuoc bat ki chi nhanh nao'

rollback tran

end

end

else

BEGIN

declare @bID varchar(50)

if(@@SERVERNAME = 'DESKTOP-N8KAOLQ\MSSQLSERVER1')

begin

SET @bID = 'ND'

end

if(@@SERVERNAME = 'DESKTOP-67JPP3J\DUONGTB')

begin

SET @bID = 'HD'

end

if(@@SERVERNAME = 'LOCNX14')

begin

SET @bID = 'HY'

end

if(@@SERVERNAME = 'RANDOM251')

begin

SET @bID = 'BG'

end

if(@@SERVERNAME = 'DESKTOP-61JKQ7T')

begin

SET @bID = 'HB'

end

if(@@SERVERNAME = 'TRAN\_QUYET')

begin

SET @bID = 'BK'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'TN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'QN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'HP'

end

declare @check varchar(50)

set @check = @bID +'%'

if(@ide like @check)

begin

print('Thanh cong')

end

else

begin

print('ID phai bat dau la '+@bID)

rollback tran

end

END

**orderdetail\_trigger**

create trigger [dbo].[orderdetail\_trigger] on [dbo].[OrderDetail]

after insert, update

AS

declare @bookID varchar(50) = (select BookID from inserted)

declare @sumofbook int = (select sum(quantity) from Distribute where BookID = @bookID)

declare @sumofbuy int = (select sum(quantity) from OrderDetail where BookID = @bookID)

declare @avaiable int = @sumofbook - @sumofbuy

print cast (@avaiable as varchar(50))

if(@avaiable >= 0)

begin

print('Ok')

end

else

begin

print('So luong mua vuot qua so luong con trong kho')

rollback tran

end

**staff\_trigger**

CREATE trigger staff\_trigger on dbo.Staff

after insert, update

AS

declare @ide varchar(50)

set @ide = (select ID from inserted)

if(@@SERVERNAME = 'ADMIN')

begin

if(@ide like 'ND%' OR @ide like 'HD%' OR @ide like 'HY%' OR @ide like 'BG%' OR @ide like 'HB%' OR @ide like 'BK%' OR @ide like 'TN%' OR @ide like 'QN%' OR @ide like 'HP%')

begin

print 'Thanh cong'

end

else

begin

print 'Nhan vien khong thuoc bat ki chi nhanh nao'

rollback tran

end

end

else

BEGIN

declare @bID varchar(50)

if(@@SERVERNAME = 'DESKTOP-N8KAOLQ\MSSQLSERVER1')

begin

SET @bID = 'ND'

end

if(@@SERVERNAME = 'DESKTOP-67JPP3J\DUONGTB')

begin

SET @bID = 'HD'

end

if(@@SERVERNAME = 'LOCNX14')

begin

SET @bID = 'HY'

end

if(@@SERVERNAME = 'RANDOM251')

begin

SET @bID = 'BG'

end

if(@@SERVERNAME = 'DESKTOP-61JKQ7T')

begin

SET @bID = 'HB'

end

if(@@SERVERNAME = 'TRAN\_QUYET')

begin

SET @bID = 'BK'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'TN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'QN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'HP'

end

declare @check varchar(50)

set @check = @bID +'%'

if(@ide like @check)

begin

print('Thanh cong')

end

else

begin

print('ID phai bat dau la '+@bID)

rollback tran

end

END

**supplier\_trigger**

CREATE trigger supplier\_trigger on dbo.Supplier

after insert, update, delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc them, sua, xoa sach tai tru so'

rollback tran

END

**supply\_trigger**

CREATE trigger supply\_trigger on dbo.Supply

after insert, update, delete

AS

if(@@SERVERNAME <> 'ADMIN')

BEGIN

print 'Chi duoc them, sua, xoa chi nhanh tai tru so'

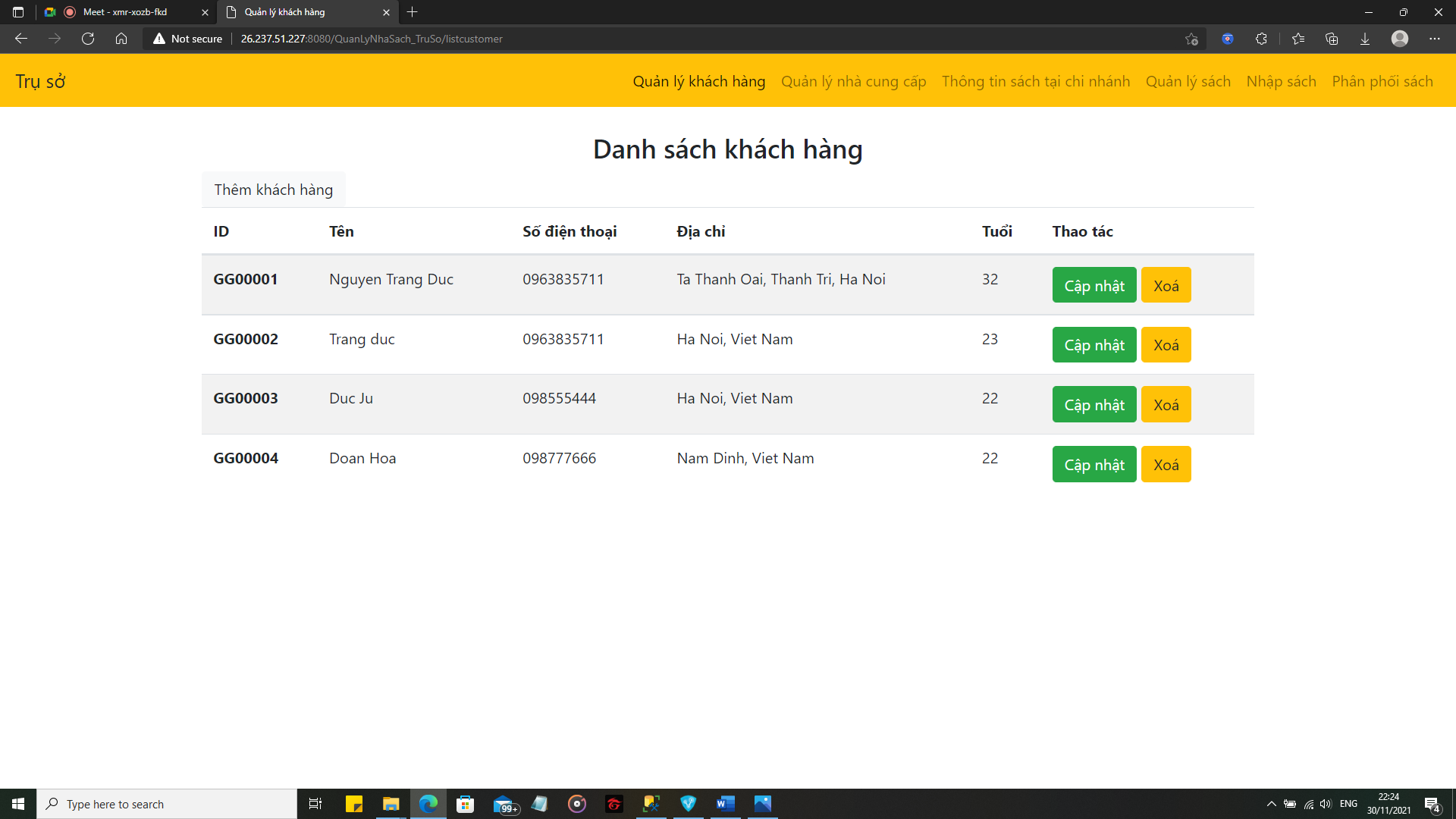
rollback tran

END

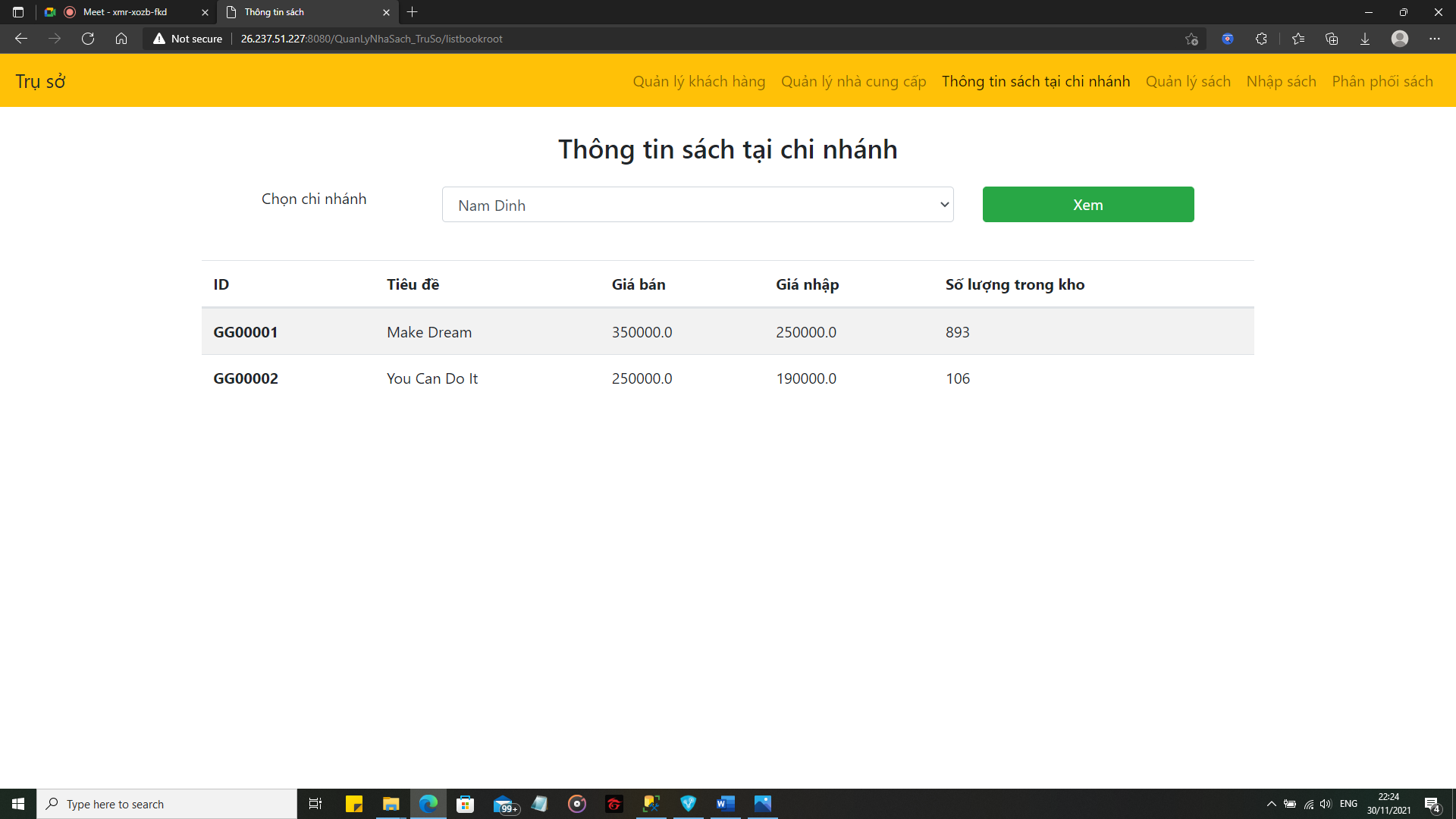
## Website quản lý

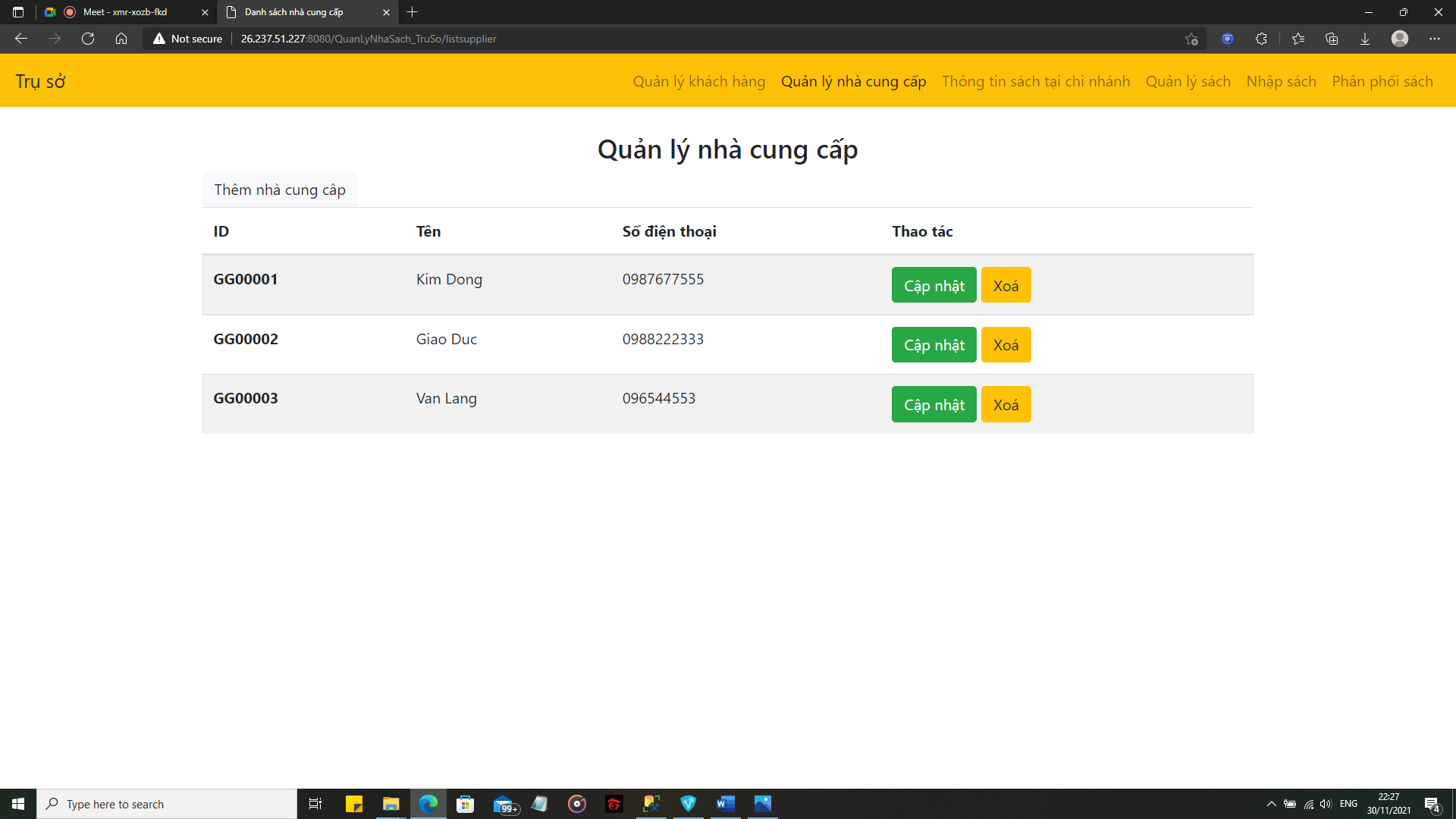
### Web ở phía máy chủ

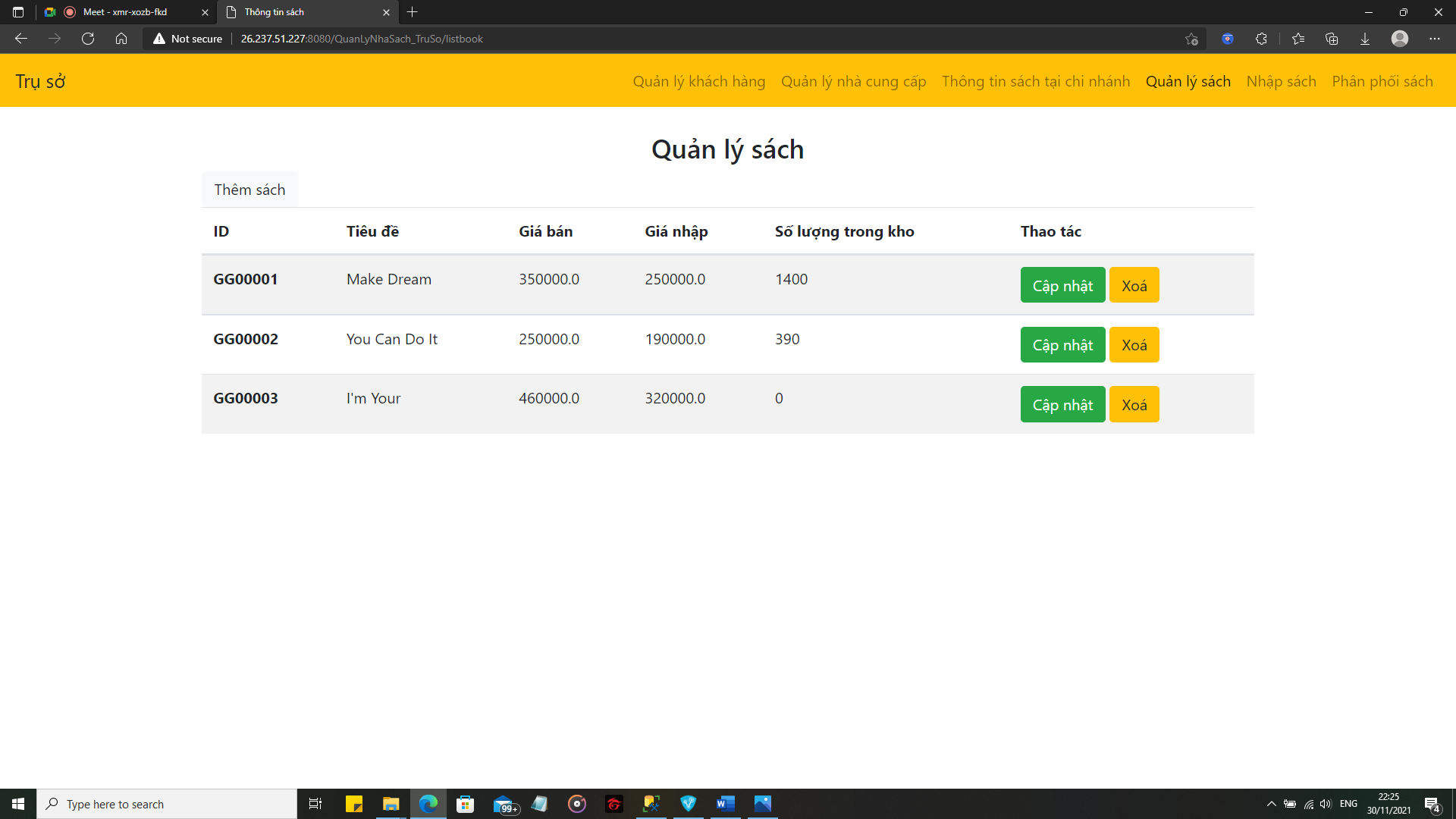
**Giao diện quản lý khách hàng**

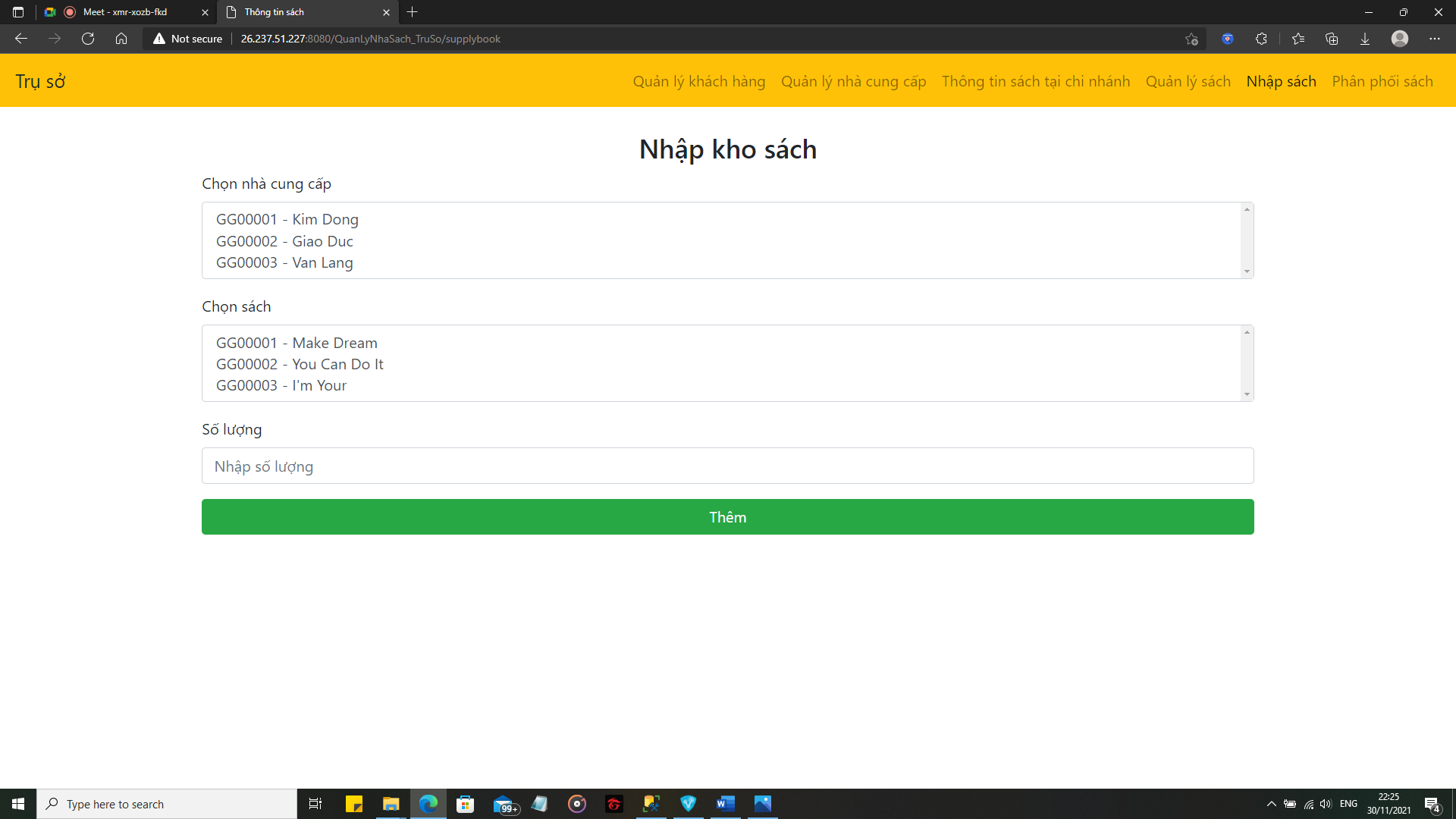


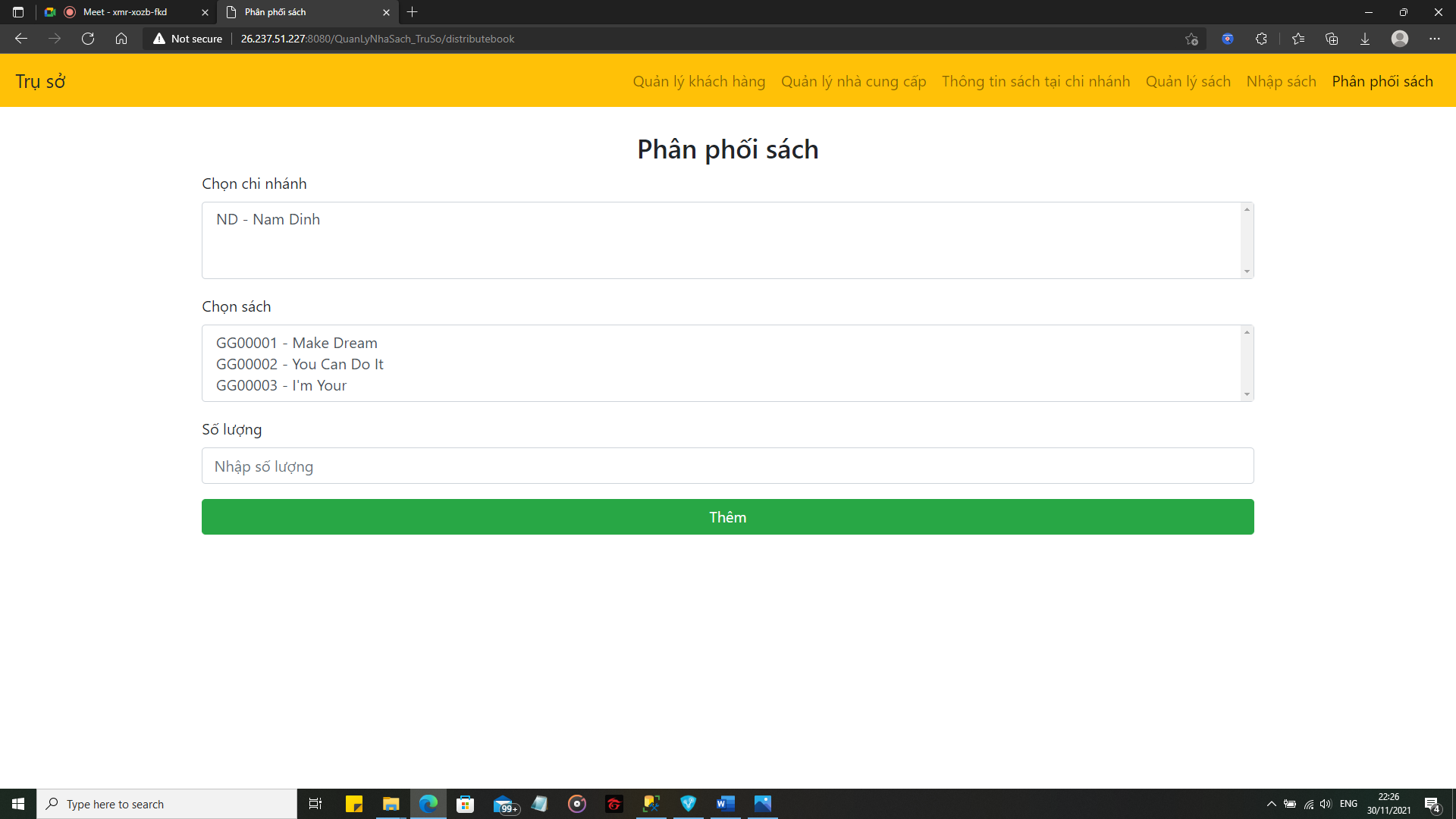
**Giao diện kiểm tra sách tại chi nhánh**



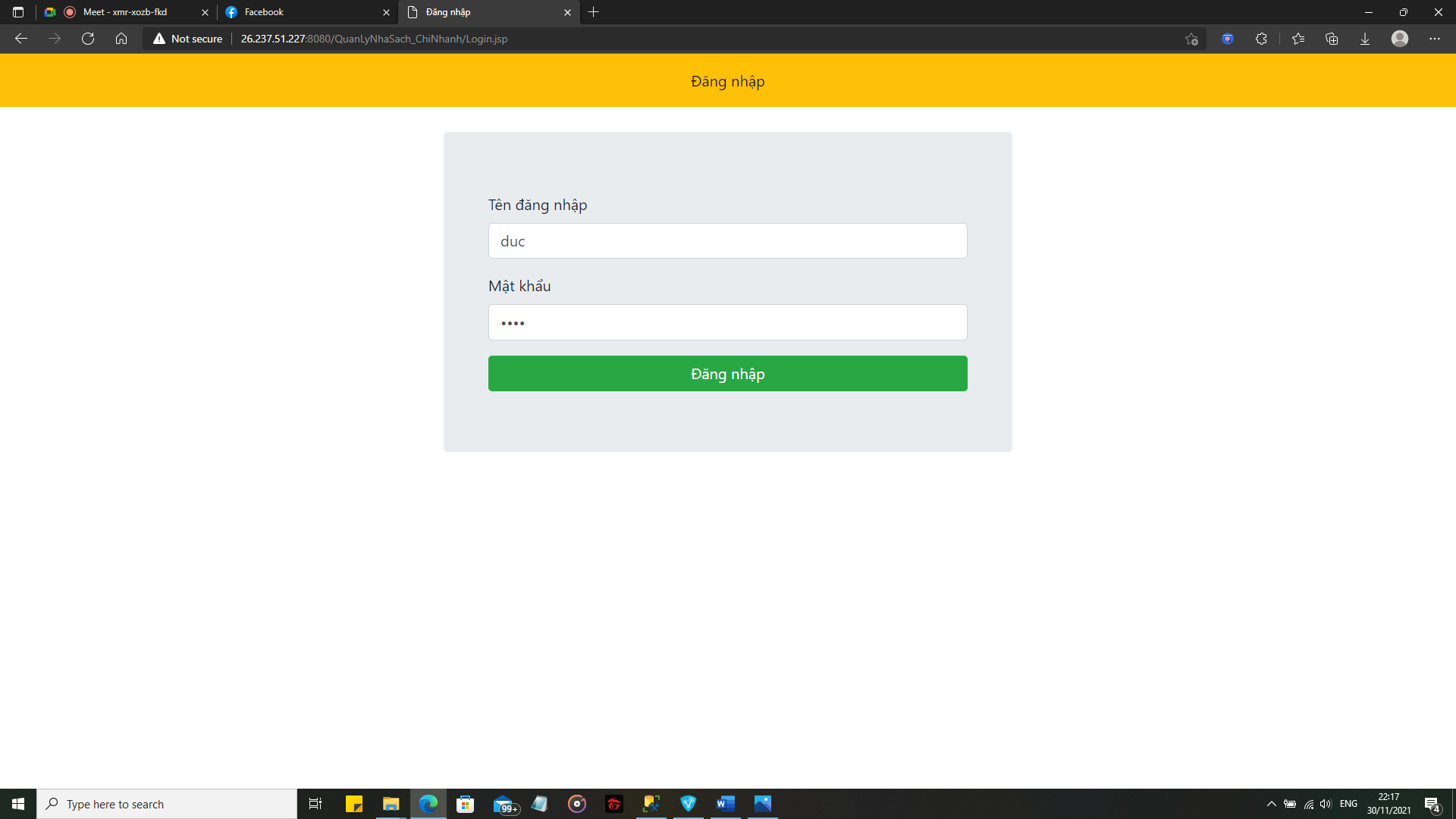
**Giao diện quản lý nhà cung cấp** 

**Giao diện quản lý sách** 

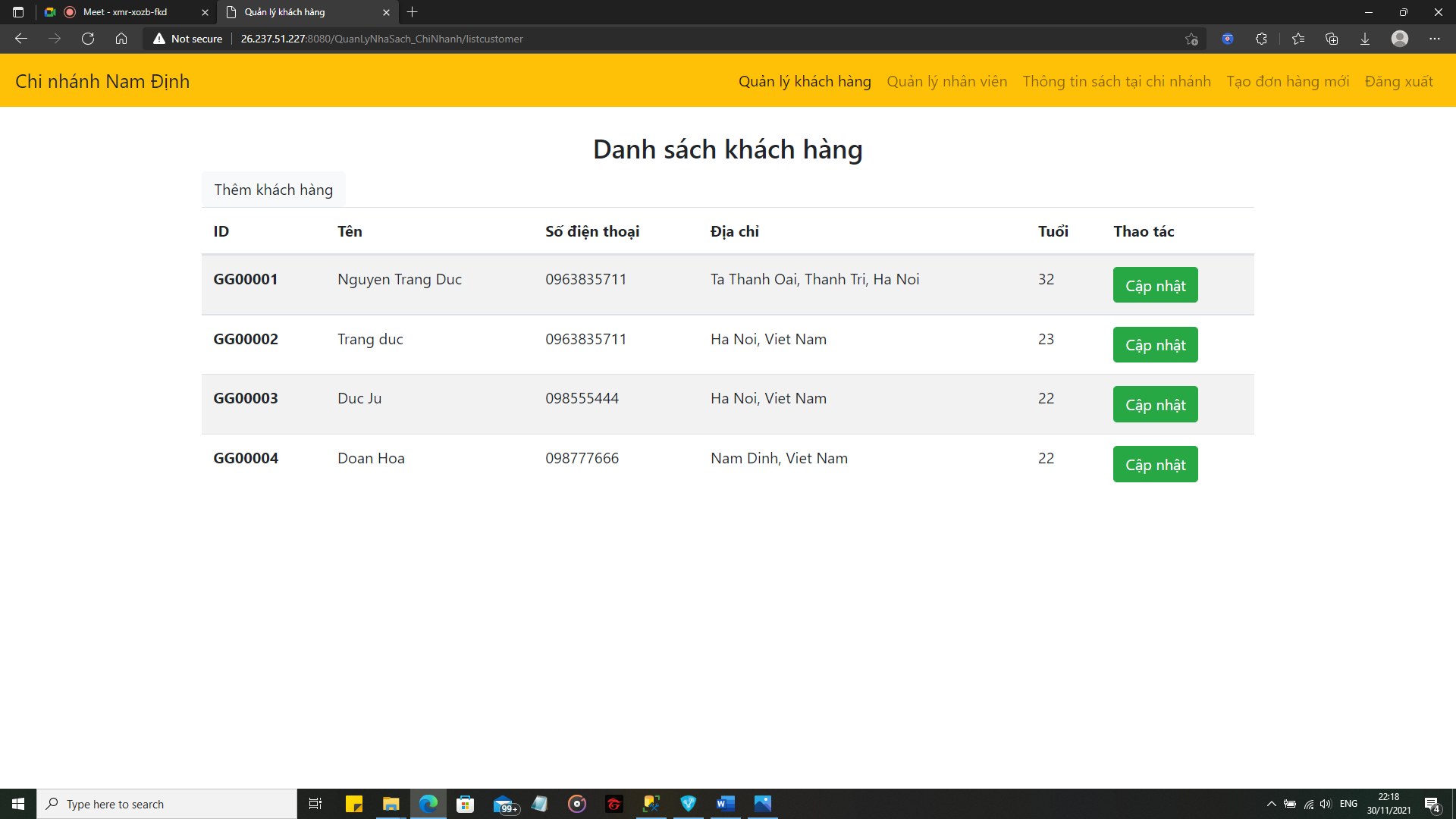
**Giao diện nhập sách** 

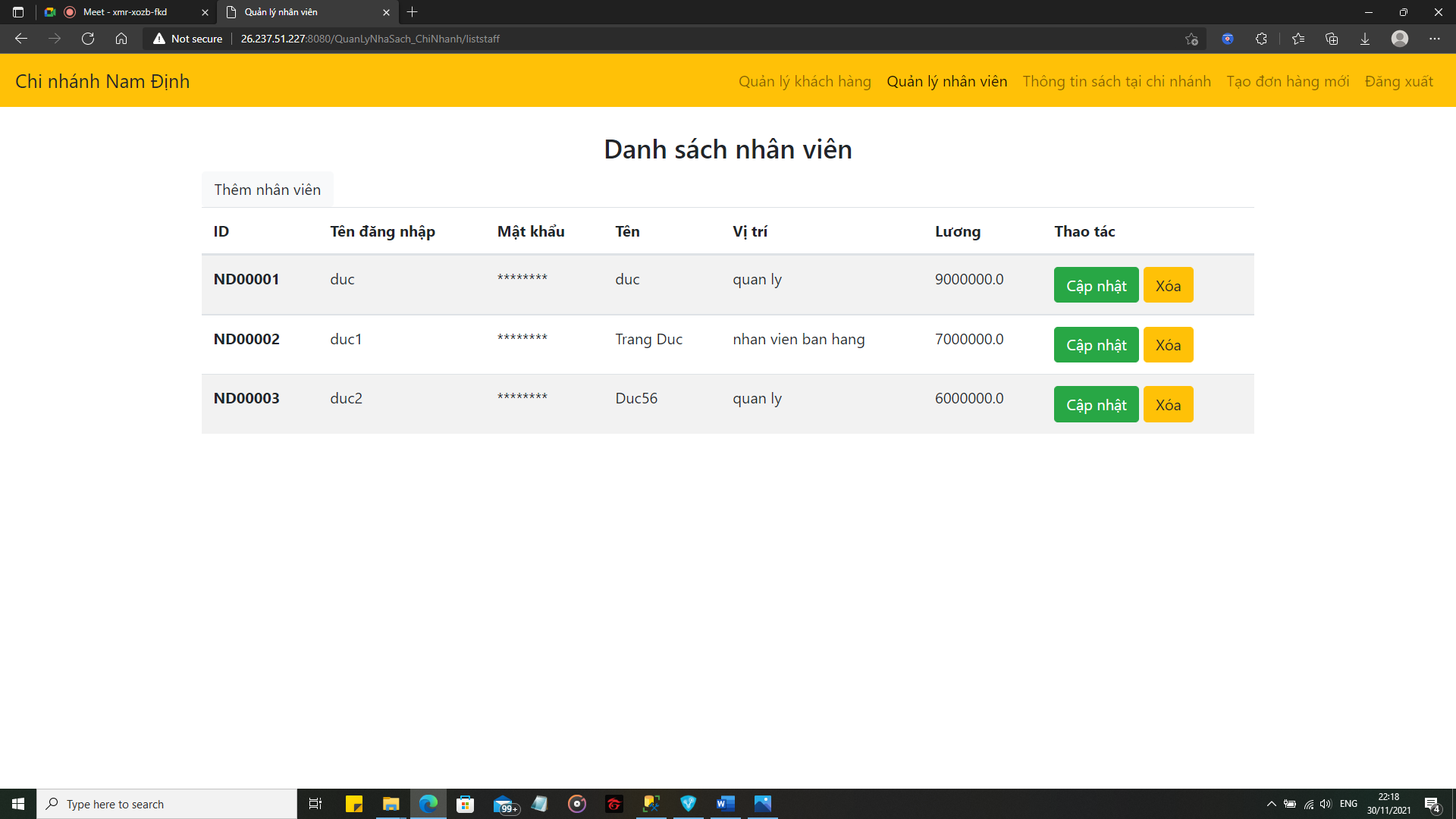
**Giao diện phân phối sách** 

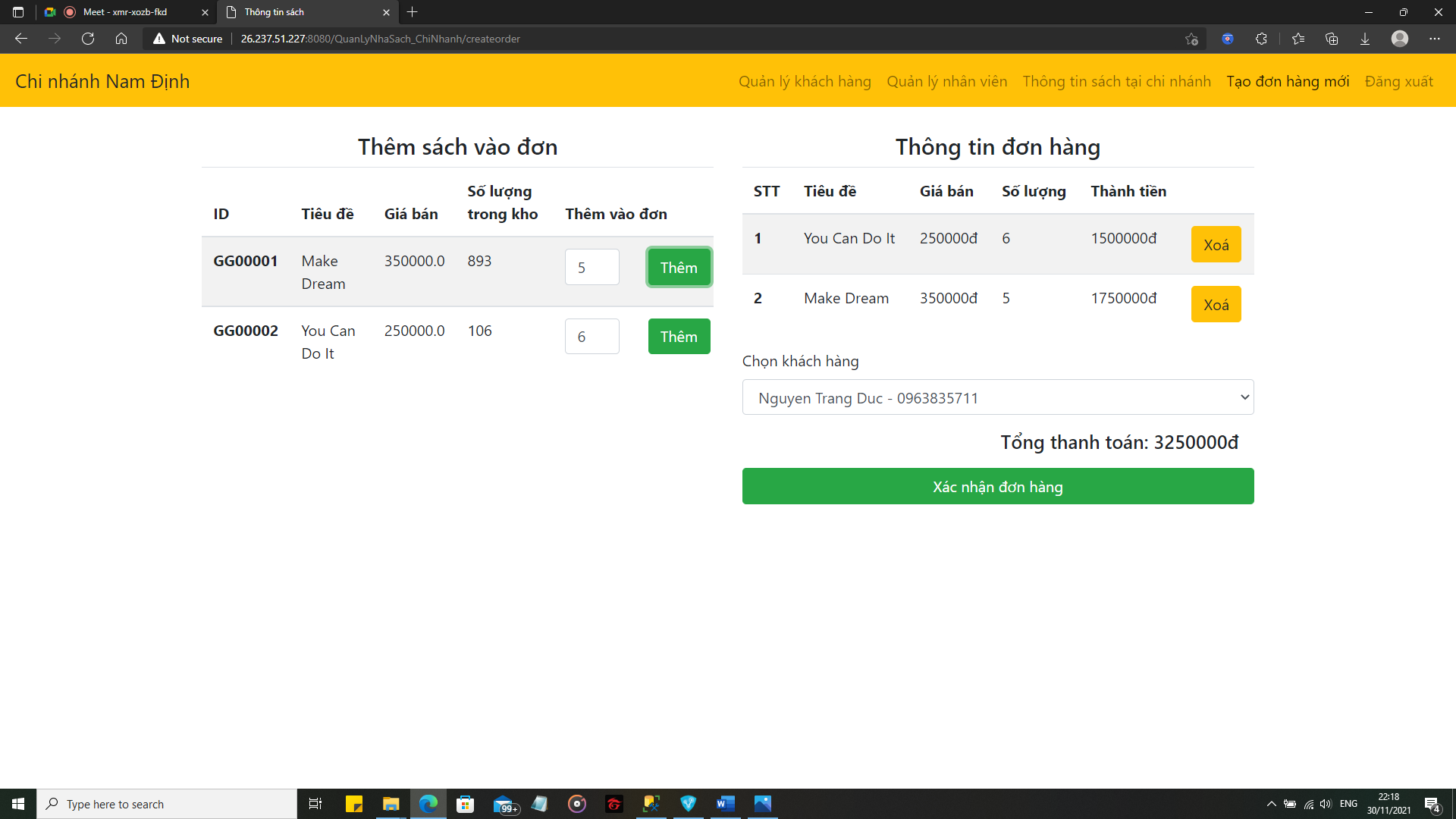
### Web tại phía chi nhánh cho nhân viên quản lý

**Giao diện đăng nhập** 

**Giao diện quản lý sách tại chi nhánh** 

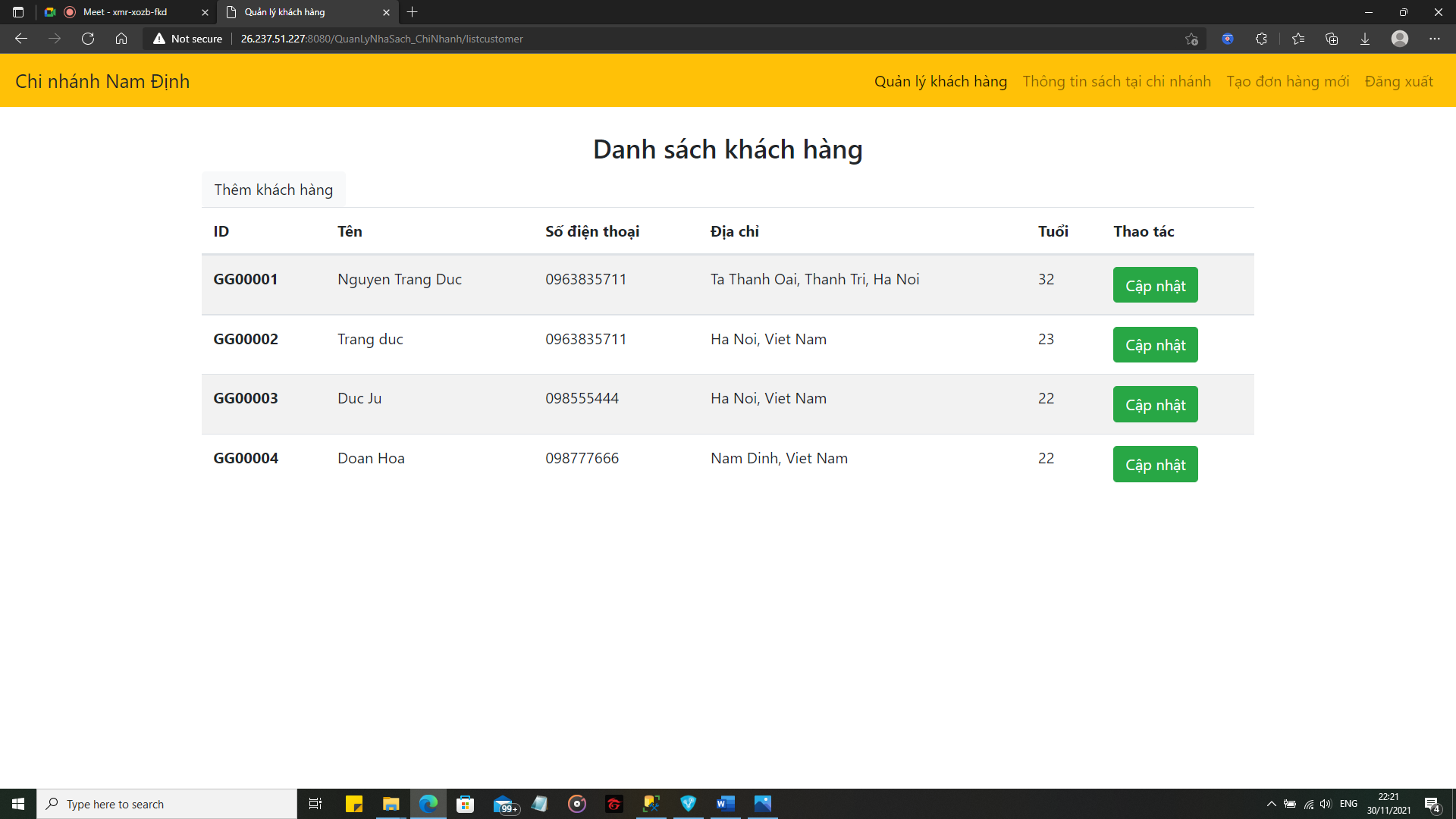
**Giao diện quản lý khách hàng** 

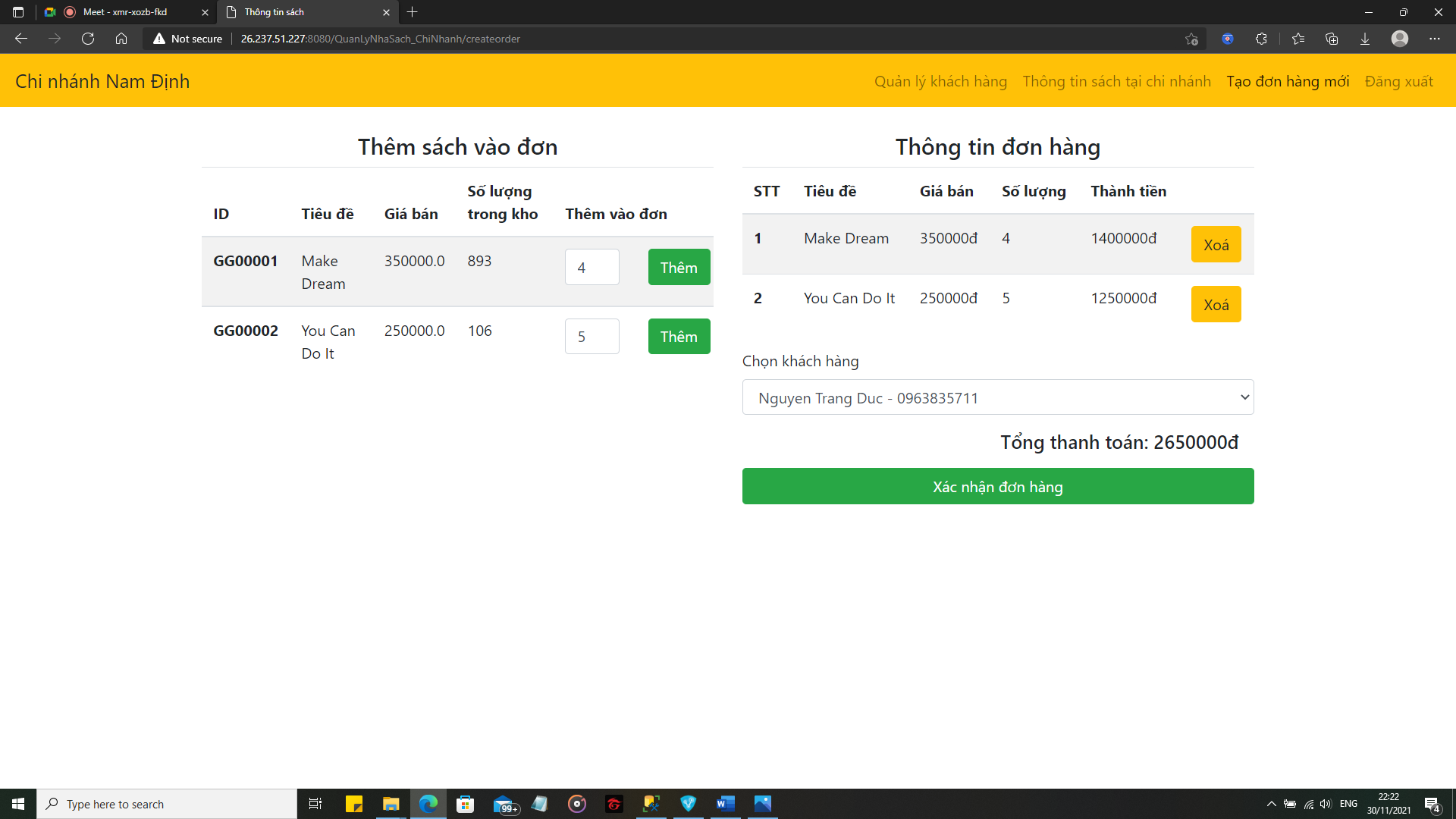
**Giao diện quản lý nhân viên** 

**Giao diện tạo đơn hàng** 

### Web ở chi nhánh cho nhân viên bán hàng

**Giao diện quản lý sách** 

**Giao diện quản lý khách hàng** 

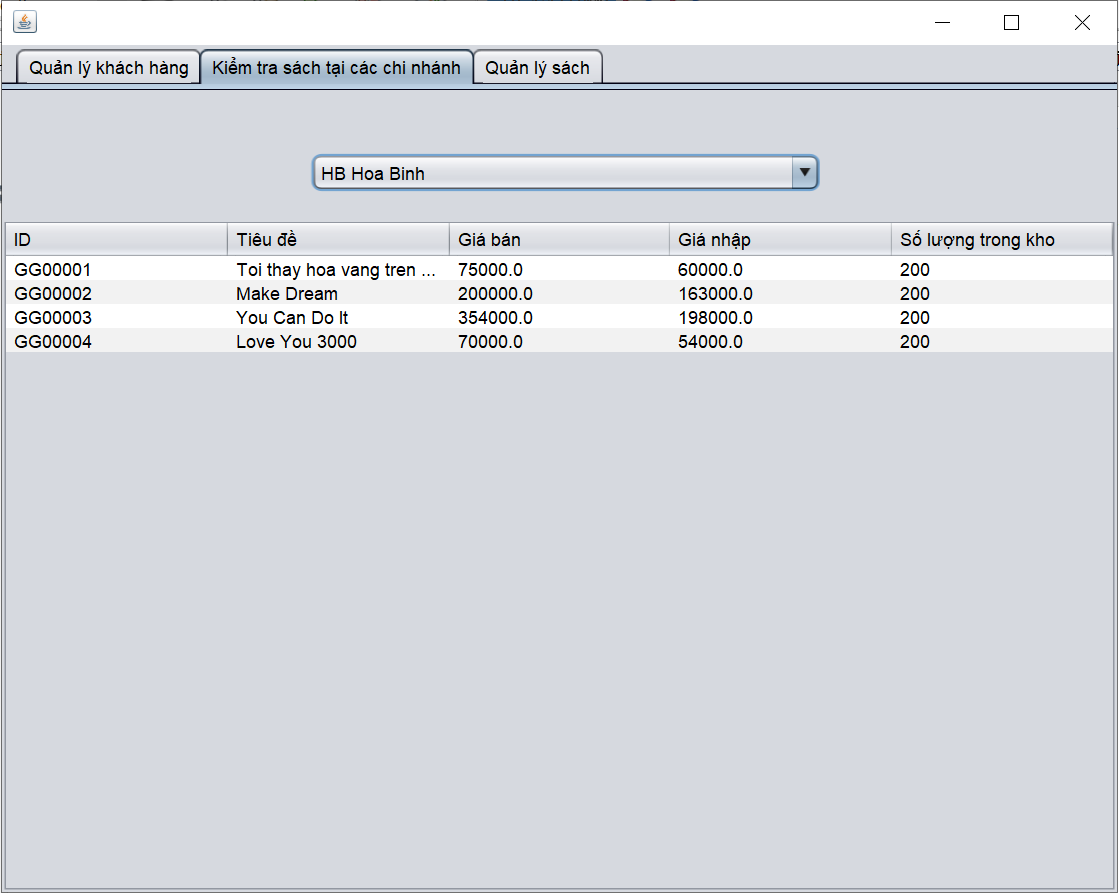
**Giao diện tạo đơn hàng** 

## App

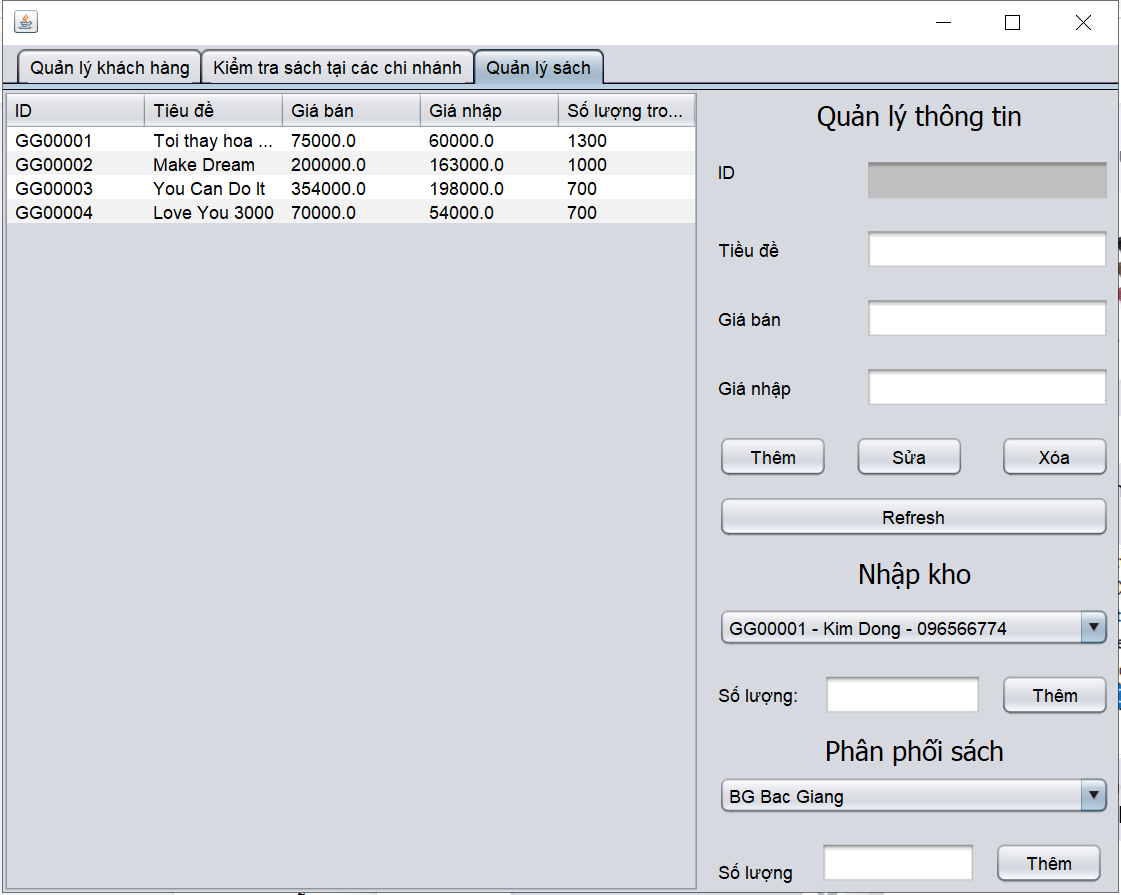
### App ở phía máy chủ

**Giao diện quản lý khách hàng**

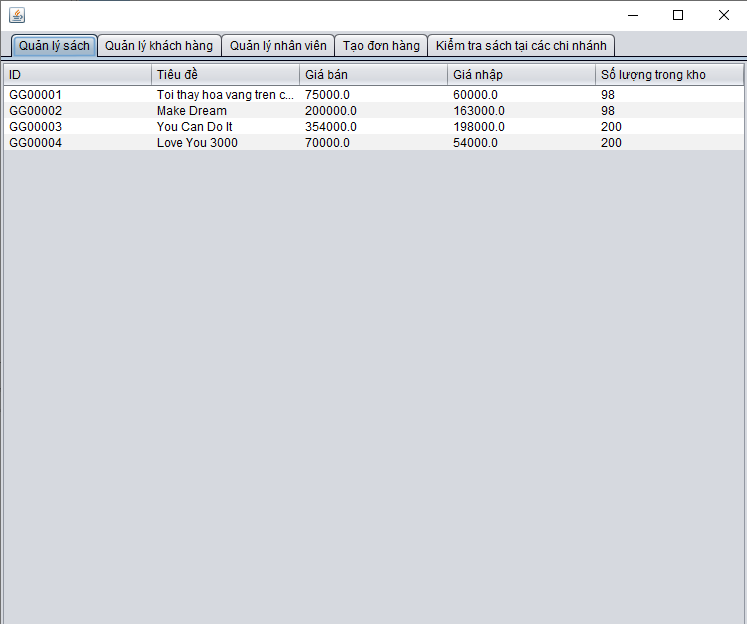


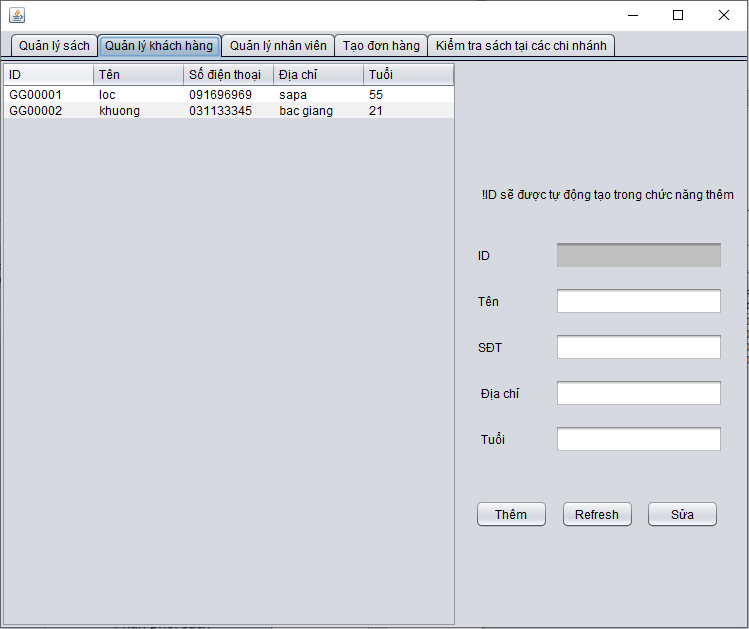
**Giao diện kiểm tra sách tại chi nhánh**

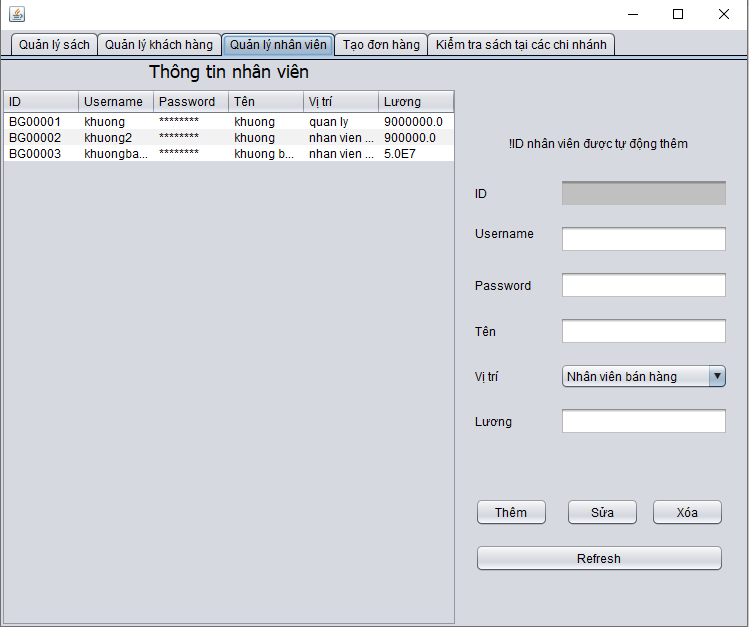
**Giao diện quản lý sách**

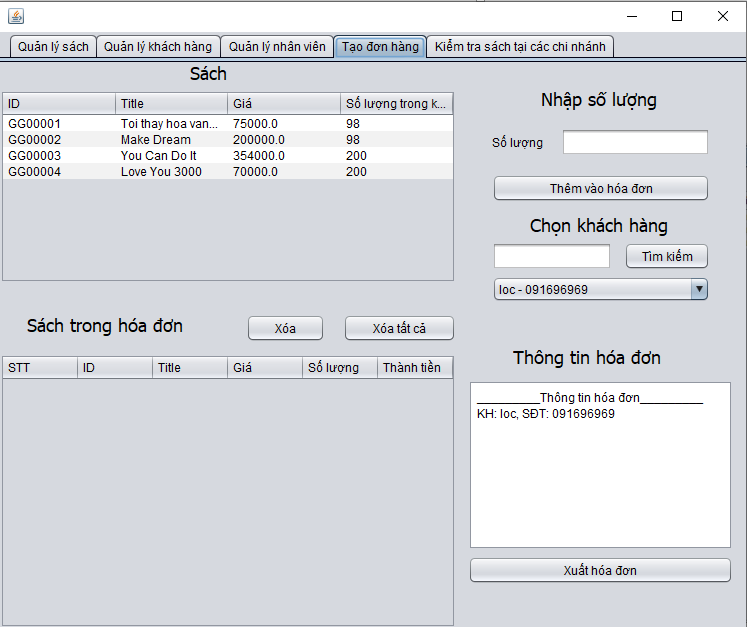


### App tại phía chi nhánh cho nhân viên quản lý

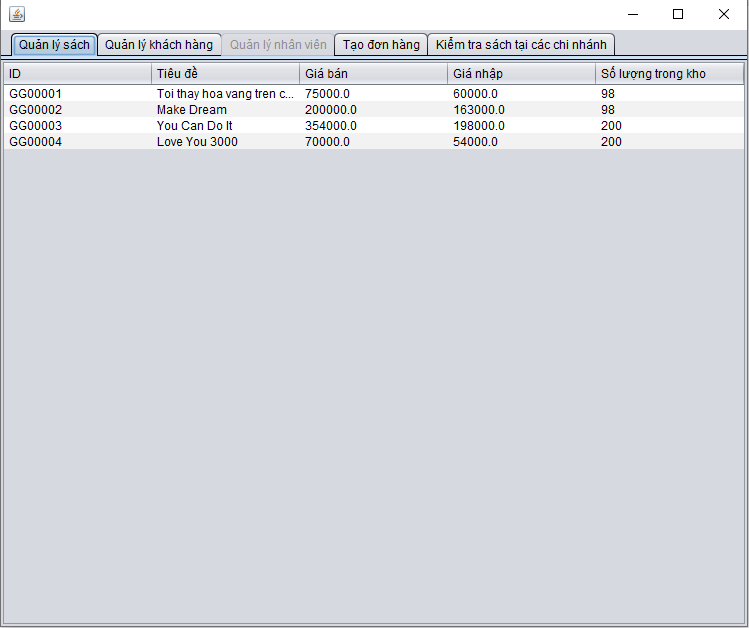
**Giao diện quản lý sách tại chi nhánh**

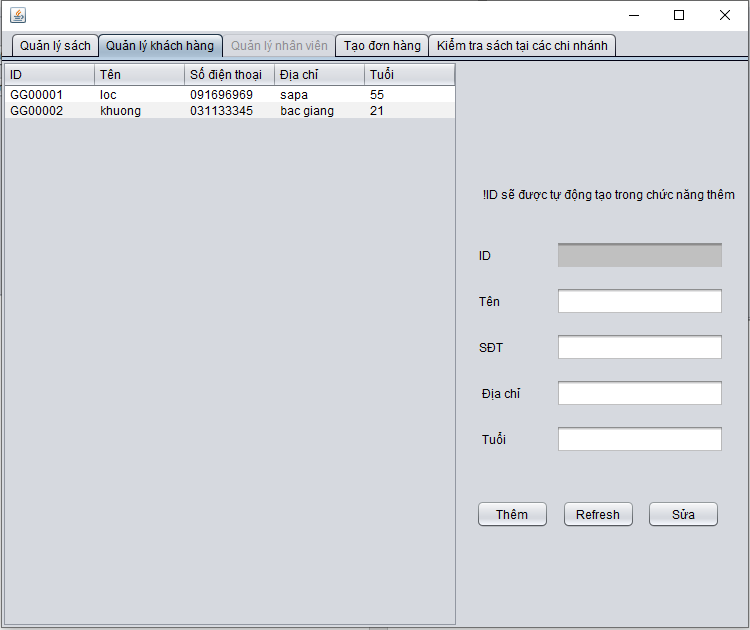
**Giao diện quản lý khách hàng**

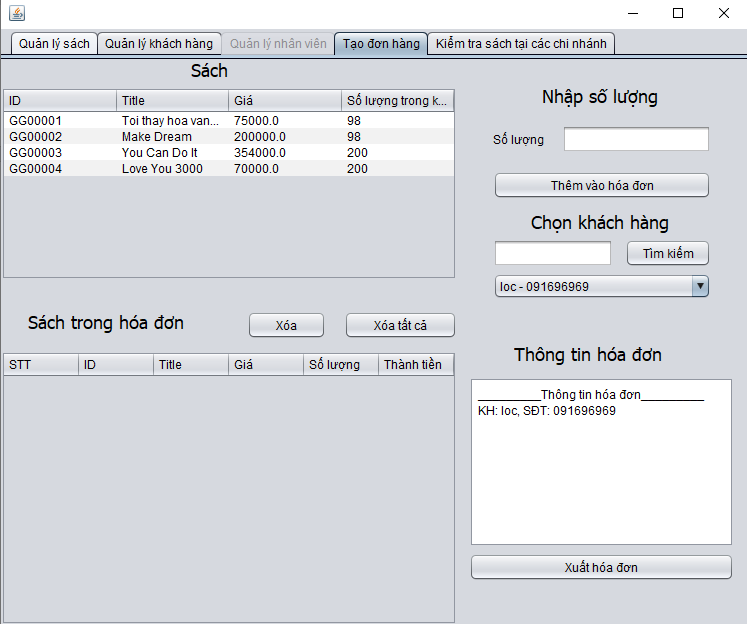
**Giao diện quản lý nhân viên**

**Giao diện tạo đơn hàng**

### App ở chi nhánh cho nhân viên bán hàng

**Giao diện quản lý sách**

**Giao diện quản lý khách hàng**

**Giao diện tạo đơn hàng**

## Procedures

**AddBook**

CREATE PROC usp\_ThemSach\_TruSo

@title varchar(255),

@price float,

@importPrice float

AS

BEGIN

DECLARE @MaSach varchar(50) = 'GG00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Book where ID = @MaSach)

BEGIN

SET @Idx = @Idx +1

SET @MaSach = 'GG' + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

INSERT into dbo.Book(ID,title,price,importPrice)

VALUES(@MaSach,@title,@price,@importPrice)

END

**addCustomer**

CREATE PROC usp\_ThemKhachHang

@name varchar(255),

@phoneNumber varchar(255),

@address varchar(255),

@age int

AS

BEGIN

DECLARE @MaKH varchar(50) = 'GG00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Customer where ID = @MaKH)

BEGIN

SET @Idx = @Idx +1

SET @MaKH = 'GG' + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaKH

INSERT into dbo.Customer(ID,name,phoneNumber,address,age)

VALUES(@MaKH,@name,@phoneNumber,@address,@age)

END

**addDistribute**

CREATE PROC usp\_ThemLuotPhanPhat

@quantity int,

@bookID varchar(255),

@bookStoreID varchar(255)

AS

BEGIN

DECLARE @supplyed int = (SELECT SUM(dbo.Supply.quantity) FROM dbo.Supply WHERE BookID = @bookID)

DECLARE @distributed int = (SELECT SUM(dbo.Distribute.quantity) FROM Distribute WHERE BookID = @bookID)

DECLARE @avaiable int

if(@supplyed <= 0 OR @quantity = 0)

BEGIN

RAISERROR('Sach chua duoc cung cap hoac so luong =0', 16,1)

END

else

BEGIN

if(@distributed>0)

BEGIN

SET @avaiable = @supplyed-@distributed

END

else

BEGIN

SET @avaiable = @supplyed

END

if(@avaiable-@quantity>=0)

BEGIN

DECLARE @date date = (SELECT GETDATE())

DECLARE @MaPP varchar(50) = 'GG00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Distribute where ID = @MaPP)

BEGIN

SET @Idx = @Idx +1

SET @MaPP = 'GG' + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaPP

INSERT into dbo.Distribute(ID,quantity,date,BookID,BookStoreID)

VALUES(@MaPP,@quantity,@date,@bookID,@bookStoreID)

END

else

BEGIN

RAISERROR('So luong phan phat lon hon so luong kho', 16,1)

END

END

END

**addOrder**

CREATE PROC usp\_ThemDonDatHang

@StaffID varchar(255),

@CustomerID varchar(255)

AS

BEGIN

DECLARE @date date = (SELECT GETDATE())

declare @bID varchar(50)

if(@@SERVERNAME = 'DESKTOP-N8KAOLQ\MSSQLSERVER1')

begin

SET @bID = 'ND'

end

if(@@SERVERNAME = 'DESKTOP-67JPP3J\DUONGTB')

begin

SET @bID = 'HD'

end

if(@@SERVERNAME = 'LOCNX14')

begin

SET @bID = 'HY'

end

if(@@SERVERNAME = 'RANDOM251')

begin

SET @bID = 'BG'

end

if(@@SERVERNAME = 'DESKTOP-61JKQ7T')

begin

SET @bID = 'HB'

end

if(@@SERVERNAME = 'TRAN\_QUYET')

begin

SET @bID = 'BK'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'TN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'QN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'HP'

end

DECLARE @MaHD varchar(50) = @bID+'00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.[Order] where ID = @MaHD)

BEGIN

SET @Idx = @Idx +1

SET @MaHD = @bID + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaHD

SELECT @MaHD

INSERT into dbo.[Order](ID,date,StaffID,CustomerID)

VALUES(@MaHD,@date,@StaffID,@CustomerID)

END

**addOrderDetail**

CREATE PROC usp\_ThemHoaDonChiTiet

@quantity int,

@OrderID varchar(50),

@bookID varchar(255)

AS

BEGIN

DECLARE @buyed int = (SELECT SUM(dbo.OrderDetail.quantity) FROM OrderDetail WHERE BookID = @bookID)

DECLARE @distributed int = (SELECT SUM(dbo.Distribute.quantity) FROM Distribute WHERE BookID = @bookID)

DECLARE @avaiable int

if(@buyed > 0)

begin

SET @avaiable = @distributed-@buyed;

end

else

begin

SET @avaiable = @distributed

end

DECLARE @check int = @avaiable - @quantity

if(@check < 0)

BEGIN

RAISERROR('So luong mua vuot qua so luong con trong kho', 16,1)

END

else

BEGIN

INSERT INTO OrderDetail(quantity,OrderID,BookID) VALUES(@quantity,@OrderID,@bookID)

END

END

**addStaff**

CREATE PROC usp\_ThemNhanVien

@username varchar(255),

@password varchar(255),

@name varchar(255),

@position varchar(255),

@salary float

AS

BEGIN

declare @bID varchar(50)

if(@@SERVERNAME = 'DESKTOP-N8KAOLQ\MSSQLSERVER1')

begin

SET @bID = 'ND'

end

if(@@SERVERNAME = 'DESKTOP-67JPP3J\DUONGTB')

begin

SET @bID = 'HD'

end

if(@@SERVERNAME = 'LOCNX14')

begin

SET @bID = 'HY'

end

if(@@SERVERNAME = 'RANDOM251')

begin

SET @bID = 'BG'

end

if(@@SERVERNAME = 'DESKTOP-61JKQ7T')

begin

SET @bID = 'HB'

end

if(@@SERVERNAME = 'TRAN\_QUYET')

begin

SET @bID = 'BK'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'TN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'QN'

end

if(@@SERVERNAME = 'DESKTOP-THOM77P')

begin

SET @bID = 'HP'

end

DECLARE @MaNV varchar(50) = @bID+'00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Staff where ID = @MaNV)

BEGIN

SET @Idx = @Idx +1

SET @MaNV = @bID + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaNV

INSERT into dbo.Staff(ID,username,password,name,position,salary,BookStoreID)

VALUES(@MaNV,@username,@password,@name,@position,@salary,@bID)

END

**addSupplier**

CREATE PROC usp\_ThemNhaCungCap

@name varchar(255),

@phoneNumber varchar(255)

AS

BEGIN

DECLARE @MaNCC varchar(50) = 'GG00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Supplier where ID = @MaNCC)

BEGIN

SET @Idx = @Idx +1

SET @MaNCC = 'GG' + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaNCC

INSERT into dbo.Supplier(ID,name,phoneNumber)

VALUES(@MaNCC,@name,@phoneNumber)

END

**addSupply**

CREATE PROC usp\_ThemLuotCungCap

@quantity int,

@bookID varchar(255),

@supplierID varchar(255)

AS

BEGIN

DECLARE @date date = (SELECT GETDATE())

DECLARE @MaCC varchar(50) = 'GG00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Supply where ID = @MaCC)

BEGIN

SET @Idx = @Idx +1

SET @MaCC = 'GG' + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @MaCC

INSERT into dbo.Supply(ID,quantity,date,BookID,SupplierID)

VALUES(@MaCC,@quantity,@date,@bookID,@supplierID)

END

**getAvaiableQuantity**

CREATE PROC usp\_TinhTonKho

@ID varchar(255)

AS

BEGIN

DECLARE @buyed int = (SELECT SUM(dbo.OrderDetail.quantity)

FROM dbo.OrderDetail

WHERE dbo.OrderDetail.BookID = @ID)

DECLARE @avaiable int

if(@buyed > 0)

begin

SELECT SUM(dbo.Distribute.quantity) - @buyed

FROM dbo.Distribute

WHERE dbo.Distribute.BookID = @ID

end

else

begin

SELECT SUM(dbo.Distribute.quantity)

FROM dbo.Distribute

WHERE dbo.Distribute.BookID = @ID

end

return @avaiable

END

**getAvaiableQuantity\_Root**

CREATE PROC usp\_TinhTonKho\_TruSo

@ID varchar(255)

AS

BEGIN

DECLARE @distributed int = (SELECT SUM(dbo.Distribute.quantity)

FROM dbo.Distribute

WHERE dbo.Distribute.BookID = @ID)

DECLARE @supplyed int = (SELECT SUM(dbo.Supply.quantity)

FROM dbo.Supply

WHERE dbo.Supply.BookID = @ID)

if(@supplyed is not null AND @distributed is not null)

BEGIN

SELECT @supplyed-@distributed

END

if(@supplyed is null AND @distributed is null)

BEGIN

SELECT 0

END

if(@supplyed is not null AND @distributed is null)

BEGIN

SELECT @supplyed

END

END

**getAvaiableQuantityBranch\_Root**

CREATE PROC usp\_TinhTonKhoChiNhanh\_TruSo

@ID varchar(255),

@BookStoreID varchar(255)

AS

BEGIN

DECLARE @buyed int = (SELECT SUM(dbo.OrderDetail.quantity)

FROM dbo.OrderDetail, dbo.[Order],dbo.Staff

WHERE dbo.OrderDetail.BookID = @ID

AND dbo.[Order].ID = dbo.OrderDetail.OrderID

AND dbo.[Order].StaffID = dbo.Staff.ID

AND dbo.Staff.BookStoreID = @BookStoreID)

DECLARE @avaiable int

if(@buyed > 0)

begin

SELECT SUM(dbo.Distribute.quantity) - @buyed

FROM dbo.Distribute

WHERE dbo.Distribute.BookID = @ID

AND dbo.Distribute.BookStoreID = @BookStoreID

end

else

begin

SELECT SUM(dbo.Distribute.quantity)

FROM dbo.Distribute

WHERE dbo.Distribute.BookID = @ID

AND dbo.Distribute.BookStoreID = @BookStoreID

end

return @avaiable

END

**nextID**

CREATE PROC usp\_TimMaNVTiepTheo

AS

BEGIN

DECLARE @MaCN varchar(50) = (select ID from dbo.BookStore)

DECLARE @MaNV varchar(50) = @MaCN+ '00001'

DECLARE @Idx int

Set @Idx = 1

WHILE EXISTS (select ID from dbo.Staff where ID = @MaNV)

BEGIN

SET @Idx = @Idx +1

SET @MaNV = @MaCN + REPLICATE('0',5-LEN(Cast(@Idx AS varchar))) + Cast(@Idx AS varchar)

END

print @maNV

END

EXEC usp\_TimMaNVTiepTheo