

GỢI Ý CÀI ĐẶT

I. Các bước CSDLPT

1. Điều kiện để phân tán được CSDL: (có dịch vụ SQL Server Agent)
Cài đặt SQL server(ít nhất là 3 ser, ko dung bản expre). Sau khi cài xong kiểm tra điều kiện cần để tiến hành phân tán hệ thống dịch vụ agent (SQL Ser Agent-> chạy start->yes->có màu xanh)
2. Phân tán cơ sở dữ liệu có 2 bước b và c ở phần 2. hướng dẫn phân tán dữ liệu trên SQL server), làm tại trung tâm
 - Định nghĩa publication(**tạo ra một tên tại máy trung tâm**, chỉ định những phần mà trên đó chứa. Gồm có tên CSDL phân tán:tên của publication) : 1 container chứa các article (table, view, stored procedure: chứa cc modul xử lý dl, UDF). Trong đó, phải chỉ rõ các server :
 - + Publisher : chứa bản cơ sở dữ liệu gốc (vd my trung tâm của nhà trường)
 - + Distributor : điều phối quá trình phân tán dữ liệu ,trong quá trình phân tán chỉ 1 sev đóng vai trò phân phối, ser đóng vai trò Distributor
 - Định nghĩa subscription (database):
 - o Chính 1 database sau khi phânn mảnh xuống ccs server
 - o 1 container nhận publication. Trong đó, phải chỉ rõ server Subscriber : chứa cơ sở dữ liệu sau khi phân tán
3. Các hình thức phân mảnh : phân mảnh ngang (Horizontal), phân mảnh dọc (Vertical), phân mảnh hỗn hợp.
- Đẩy dữ liệu

Ví dụ: Giả sử ta muốn phân tán cơ sở dữ liệu QLDSV theo khoa thì nếu:
KHOA : MAKH='CNTT' → Khoa là phân mảnh ngang nguyên thủy
LOP : MAKH='CNTT' → LOP là phân mảnh ngang nguyên thủy
LOP.MAKH = KHOA.MAKH → LOP là phân mảnh ngang dẫn xuất

4. Các quy tắc phân mảnh: quá trình phân tán phải thỏa các qui tắc sau: **tính đầy đủ, tính tái thiết**, tính tách biệt

Ví dụ: Giả sử ta muốn phân tán cơ sở dữ liệu QLDSV theo 2 khoa (CNTT, VT) từ Server THU-PC\SERVER2 đến 2 Server THU-PC\SUSU (chứa thông tin khoa CNTT) và Server THU-PC\TINTIN (chứa thông tin khoa VT)

5. Tạo Link Server:

- Mục đích: cho phép truy cập(xem, sửa từ xa) dữ liệu từ server 1 đến Server 2 qua tập lệnh DML
 - Cú pháp : Giả sử ta đang đứng ở Server THU-PC\TINTIN, ta muốn tạo 1 link server tên LINK2 đến Server THU-PC\SUSU
- ```
EXEC sp_addlinkedserver @server='LINK2', @srvproduct='',
@provider='SQLOLEDB', @datasrc=' THU-PC\SUSU'
```
- Ví dụ áp dụng: Sử dụng link server vừa tạo để in ra danh sách lớp của khoa viễn thông đang đặt ở Server THU-PC\SUSU
- ```
Select * from LINK2.QLDSV.DBO.LOP
```

6. Quy tắc để giúp cho các câu lệnh phân tán được “trong suốt”:

- Trong suốt : nghĩa là 1 SP khi ta cho thực thi ở 1 Server phân mảnh bất kỳ thì vẫn thực thi được ở tất cả các Server phân mảnh còn lại mà ta không cần chỉ lại đường dẫn đến SP cần truy xuất.
- Quy tắc :
 - a. Tên các cơ sở dữ liệu ở các Server Subscriber phải giống nhau
 - b. Tên Link Server phải giống nhau

7. Store Procedure

Lệnh Select ... Union : lấy dữ liệu từ các server phân mảnh về

Ví dụ: Viết 1 SP tên DS_LOP sao cho khi chạy SP này trên 1 server bất kỳ , nó đều hiển thị danh sách lớp của trường

```
CREATE PROCEDURE SP_DSLOP AS
SELECT MALOP, TENLOP FROM DBO.LOP
UNION
SELECT MALOP, TENLOP FROM LINK1.QLDSV.DBO.LOP
```

8. Tìm kiếm dữ liệu trên cơ sở dữ liệu phân tán:

Ví dụ: hãy in ra tên lớp và tên khoa, ho, ten của 1 sv có mã số là @X. (SP có tên sp_TimSV)

```
create procedure sp_TimSV
```

```
@X varchar(10)
```

```
AS
```

```
DECLARE @MALOP VARCHAR(10), @HO nvarchar(50), @TEN nvarchar(10)
```

```
if exists(select masv from sinhvien where masv =@X)
```

```
BEGIN
```

```
SELECT @MALOP = MALOP, @HO=HO, @TEN=TEN
```

```
FROM SINHVIEN WHERE MASV=@X
```

```
select tenlop= (SELECT TENLOP FROM LOP WHERE MALOP=@MALOP),
```

```
tenkh = (SELECT TENKH FROM KHOA),
```

```
HO=@HO, TEN=@TEN
```

```
END
```

```
else
```

```
if exists(select masv from link1.qldsv.dbo.sinhvien where masv =@X)
```

```
select tenlop, tenkh , ho, ten
```

```
from link1.qldsv.dbo.khoa KH, link1.qldsv.dbo.lop L, link1.qldsv.dbo.Sinhvien
```

```
SV
```

```
where KH.MAKH=L.MAKH and L.MALOP=SV.MALOP
```

```
and SV.MASV=@X
```

```
else
```

```
if exists(select masv from link2.qldsv.dbo.sinhvien where masv =@X)
```

```
select tenlop, tenkh , ho, ten
```

```
from link2.qldsv.dbo.khoa KH, link2.qldsv.dbo.lop L, link2.qldsv.dbo.Sinhvien
```

```
SV
```

```
where KH.MAKH=L.MAKH and L.MALOP=SV.MALOP
```

```
and SV.MASV=@X
```

TS.Phan Thị Hà-HVCNBCVT: Tài liệu nháp, không được public

ELSE

raiserror ('Ma sinh vien ban tim khong co', 16, 0)

Lệnh kiểm tra SP vừa viết: Ta gọi lệnh sau ở cả 2 Server phân mảnh thì kết quả đều giống nhau, như vậy user sử dụng lệnh ta viết sẽ không cảm nhận được là cơ sở dữ liệu đang phân tán hay tập trung.

exec TimSV 'ABC'

9. Viết 1 SP cho phép sửa họ của 1 sv có mã sinh viên là @X, biết rằng ta chưa rõ sv này thuộc khoa nào ?

create procedure Update_Hosv

@X char(10), @ho varchar(40)

AS

if exists(select masv from sinhvien where masv =@X)

Update qlcsv.dbo.sinhvien

set Ho= @ho

where masv=@X

else

if exists(select masv from link1.qlcsv.dbo.sinhvien where masv =@X)

Update link1.qlcsv.dbo.sinhvien

set Ho= @ho

where masv=@X

else

print 'Ma sinh vien ban tim khong co'

exec update_hosv '96v10002', 'Nguyen van'

- Viết 1 SP để xóa 1 sv có mã số @X

create procedure Delete_Masv

@X char(8)

AS

if exists(select masv from sinhvien where masv =@X)

Delete from sinhvien

where masv=@X

else

if exists(select masv from link1.qlsv.dbo.sinhvien where masv =@X)

Delete from link1.qlsv.dbo.sinhvien

where masv=@X

else

print 'Ma sinh vien ban tim khong co'

GO

- Viết 1 SP để chuyển 1 sv có mã số @X sang lớp có mã @malop

create procedure ChuyenLop

TS.Phan Thị Hà-HVCNBCVT: Tài liệu nháp, không được public

```

    @X nvarchar(8), @ML nvarchar(10)
AS
if exists(select masv from sinhvien where masv=@X)
    if exists(select malop from Lop where malop=@ML)
        Update sinhvien set malop=@ML where masv=@X
    else
        Print 'Ma lop' + @ML + 'khong co trong danh sach lop cua khoa'
else
    if exists(select masv from link1.qlsv.dbo.sinhvien where masv=@X)
        if exists(select malop from link1.qlsv.dbo.Lop where malop=@ML)
            Update link1.qlsv.dbo.sinhvien set malop=@ML where masv=@X
        else
            Print 'Ma lop' + @ML + 'khong co trong danh sach lop cua khoa'
    else
        print 'Error...'

```

```

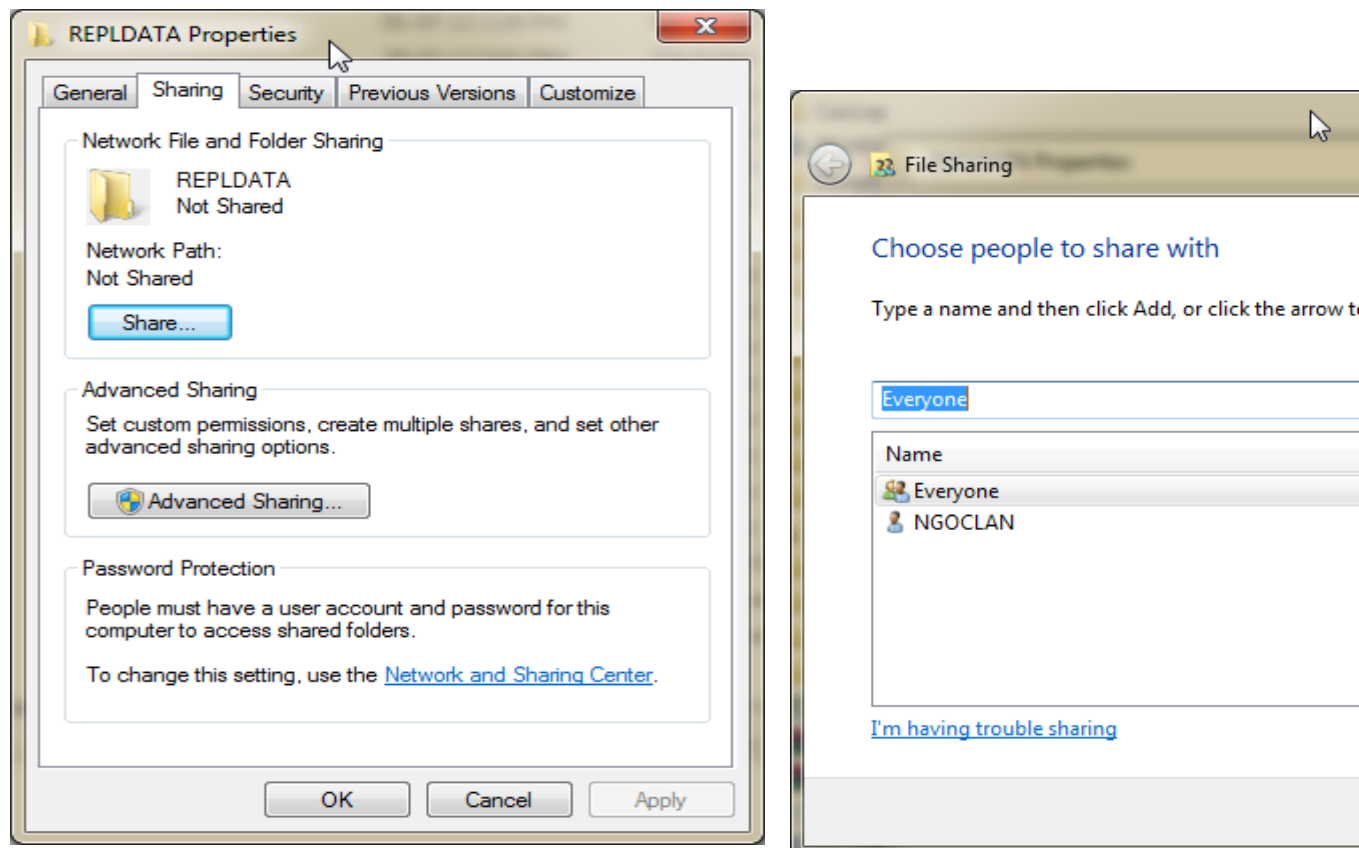
ALTER PROCEDURE TIMPHIEU
@X CHAR (10)
AS
if exists(select PHIEU from link1.QLVT.dbo.PHATSINH where
PHIEU =@X)
    select PS.PHIEU,HOTENKH,NGAY,LOAI,
    TENVT,SOLUONG,DONGIA,TRIGIA=SOLUONG*DONGIA
    from link1.QLVT.dbo.PHATSINH PS ,
    link1.QLVT.dbo.CT_PHATSINH CTPS ,
    link1.QLVT.dbo.VATTU VT
    where PS.PHIEU=CTPS.PHIEU AND
    CTPS.MAVT=VT.MAVT
    and PS.PHIEU=@X
ELSE
if exists(select PHIEU from link2.QLVT.dbo.PHATSINH where
PHIEU =@X)
    select PS.PHIEU,HOTENKH,NGAY,LOAI,
    TENVT,SOLUONG,DONGIA,TRIGIA=SOLUONG*DONGIA
    from link2.QLVT.dbo.PHATSINH PS ,
    link2.QLVT.dbo.CT_PHATSINH CTPS ,
    link2.QLVT.dbo.VATTU VT
    where PS.PHIEU=CTPS.PHIEU AND
    CTPS.MAVT=VT.MAVT
    and PS.PHIEU=@X
ELSE
print 'Ma PHIEU ban tim khong co'

```

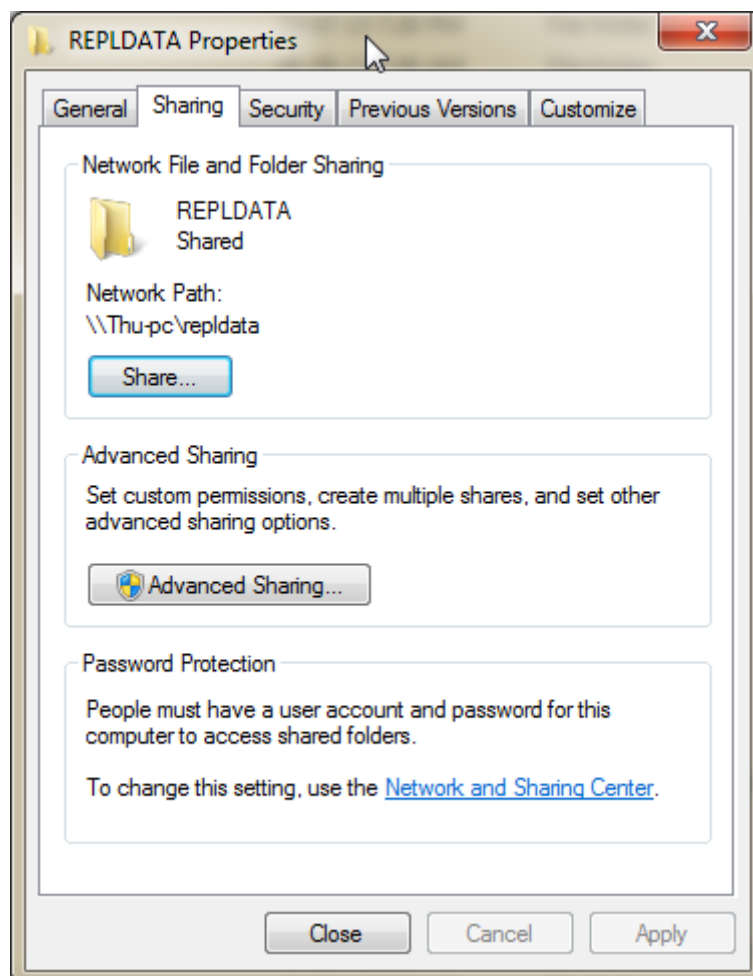
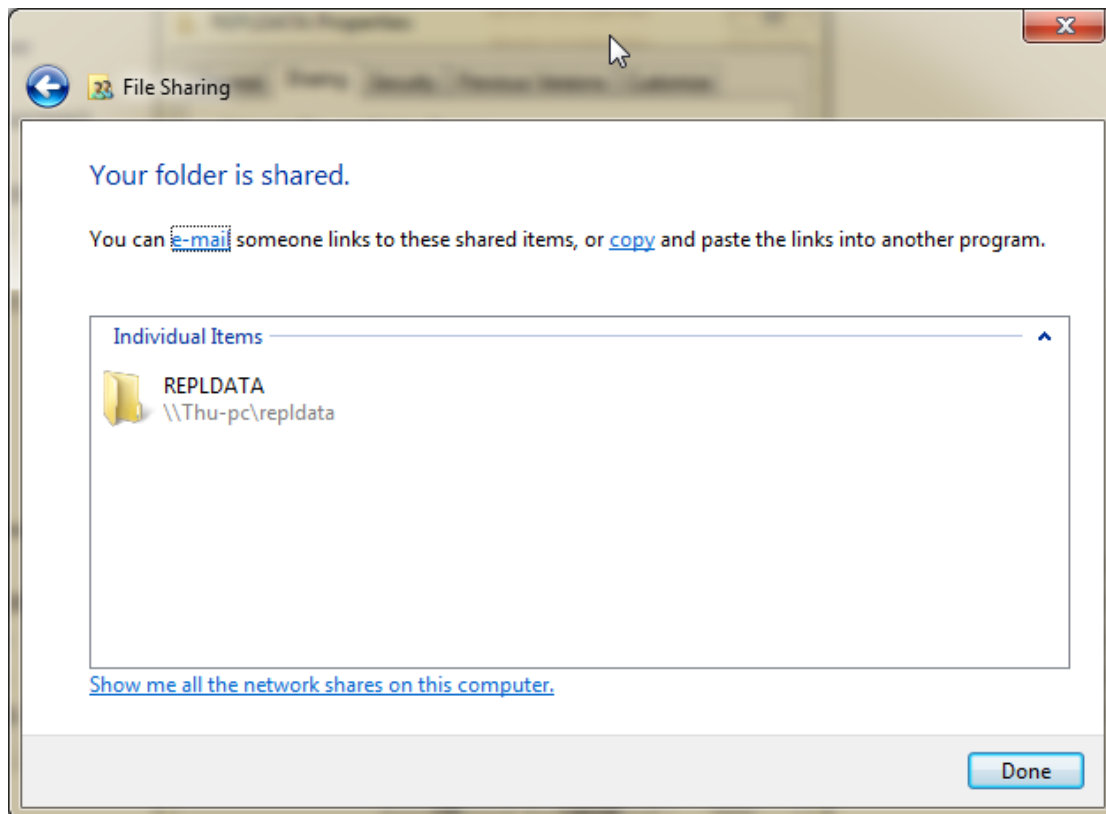
II. HƯỚNG DẪN PHÂN TÁN DỮ LIỆU TRÊN SQL-SERVER

Chuẩn bị :

- Ta tạo 1 folder D:\ReplData để chứa các dữ liệu trao đổi trong quá trình update dữ liệu từ các phân mảnh về cơ sở dữ liệu gốc, và từ cơ sở dữ liệu gốc đến các phân mảnh
 - Tiến hành cho folder này là 1 snapshot folder: thực chất là 1 shared folder trong Windows, cho phép các users được quyền read/write (giả sử shared folder có tên \\THU-PC\REPLDATA)
- Right click trên folder REPLDATA, chọn Properties, chọn tab Sharing – Share

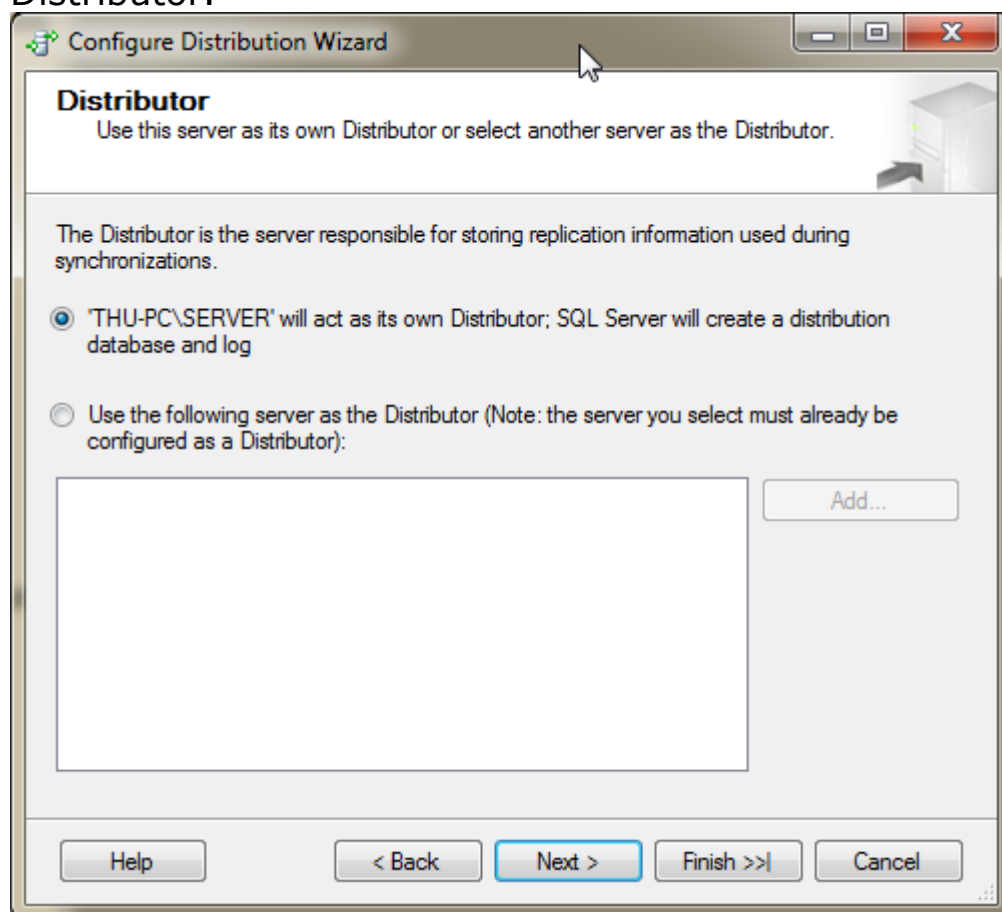


Chọn Everyone, click Add, và chọn quyền Read/Write như trong hình. Cuối cùng, click nút lệnh Share.

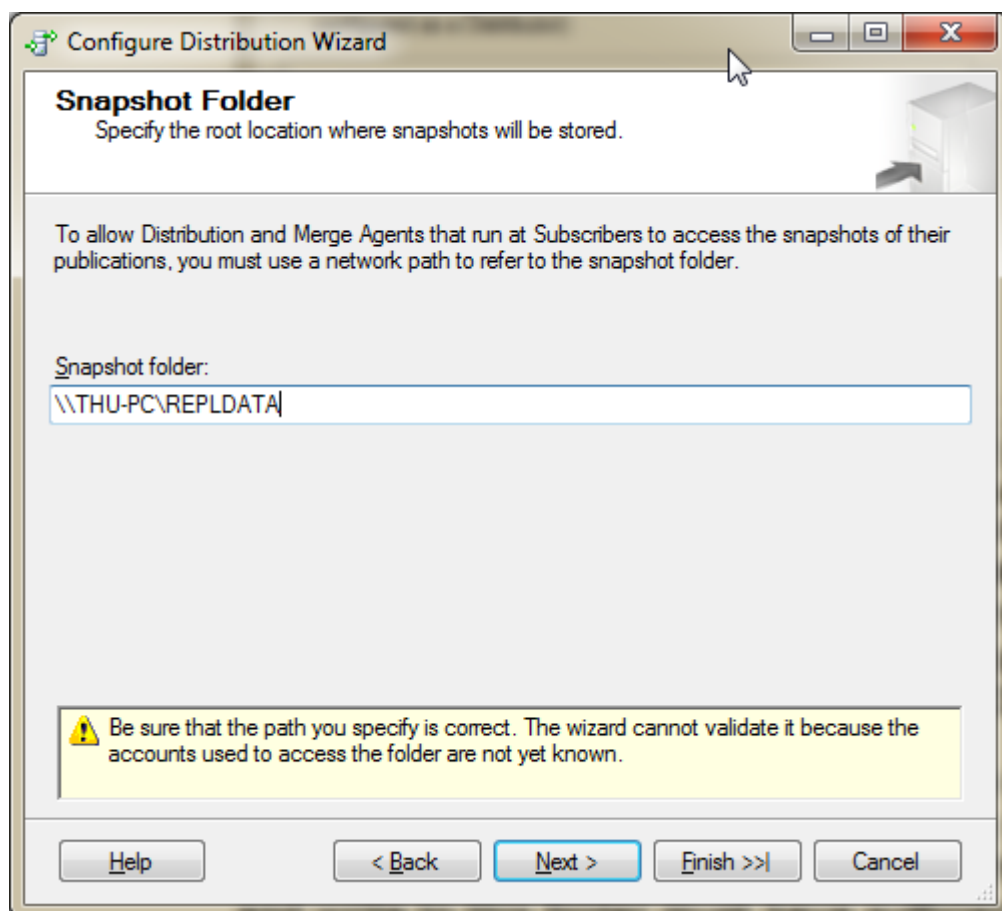


A. To configure distribution

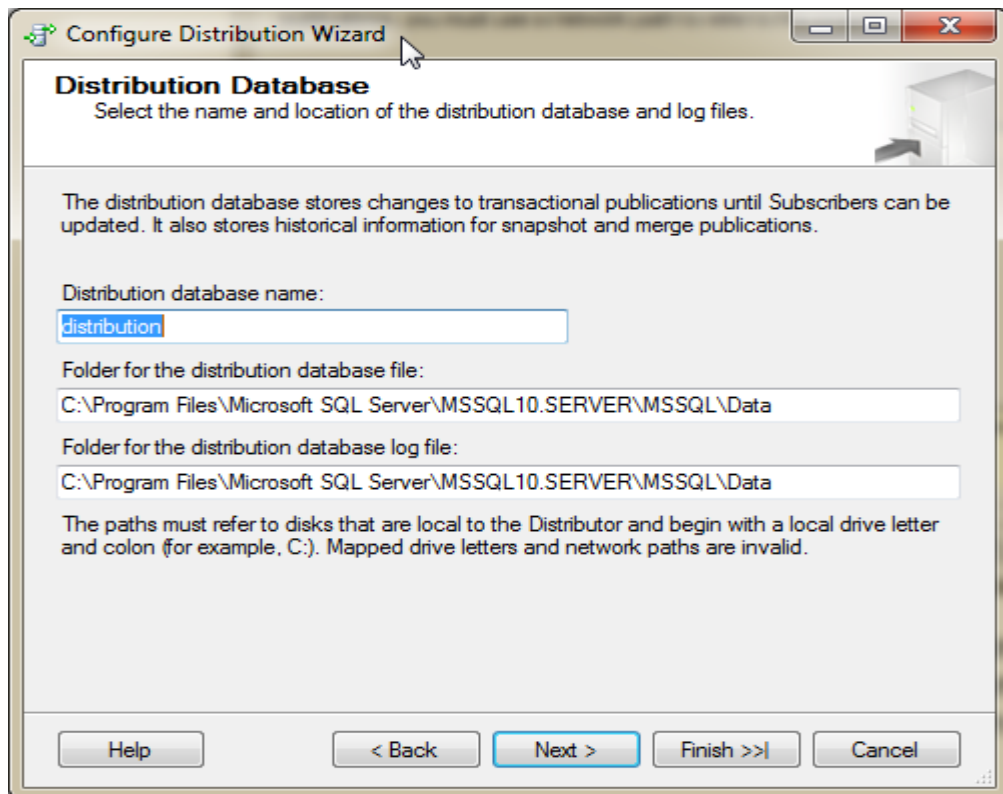
1. In Microsoft SQL Server Management Studio, connect to the server that will be the Distributor (in many cases, the Publisher and Distributor are the same server), and then expand the server node.
2. Right-click the **Replication** folder, and then click **Configure Distribution.**;
3. Follow the Configure Distribution Wizard to:
 - Select a Distributor. To use a local Distributor, select **'<ServerName>' will act as its own Distributor; SQL Server will create a distribution database and log.**
 - To use a remote Distributor, select **Use the âfollowing server as the Distributor**, and then select a server. The server must already be configured as a Distributor, and the Publisher must be enabled to use the Distributor.



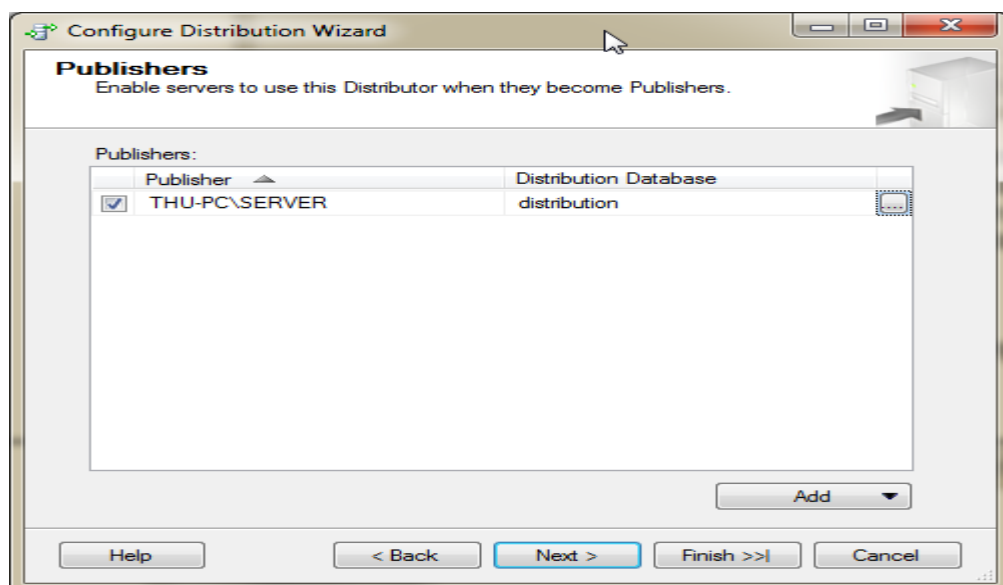
- If you select a remote Distributor, you must enter a password on the **Administrative Password** page for connections made from the Publisher to the Distributor. This password must match the password specified when the Publisher was enabled at the remote Distributor.
- Specify a root snapshot folder (for a local Distributor). The snapshot folder is simply a directory that you have designated as a share; agents that read from and write to this folder must have sufficient permissions to access it. Each Publisher that uses this Distributor creates a folder under the root folder, and each publication creates folders under the Publisher folder in which to store snapshot files.



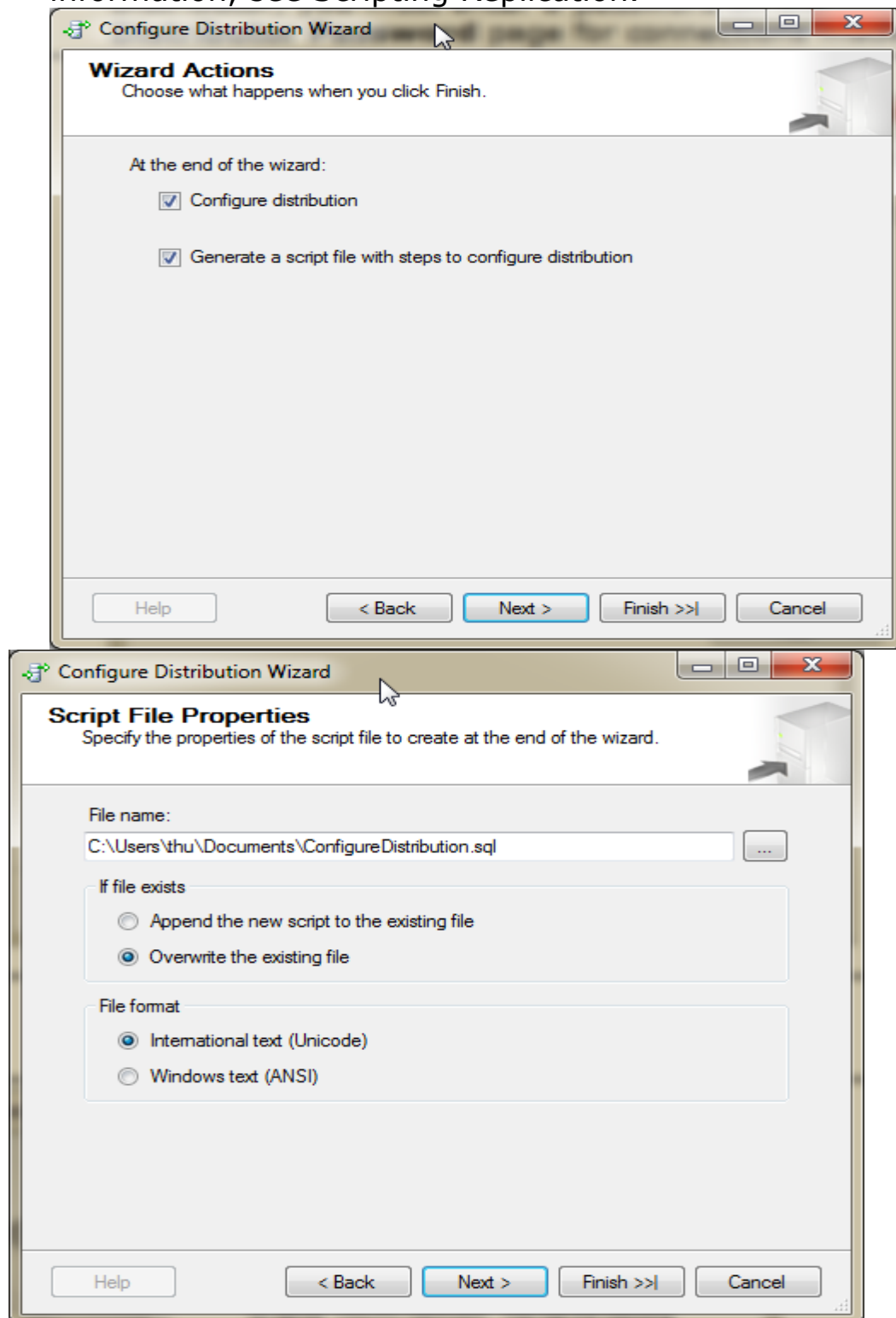
- Specify the distribution database (for a local Distributor). The distribution database stores metadata and history data for all types of replication and transactions for transactional replication.

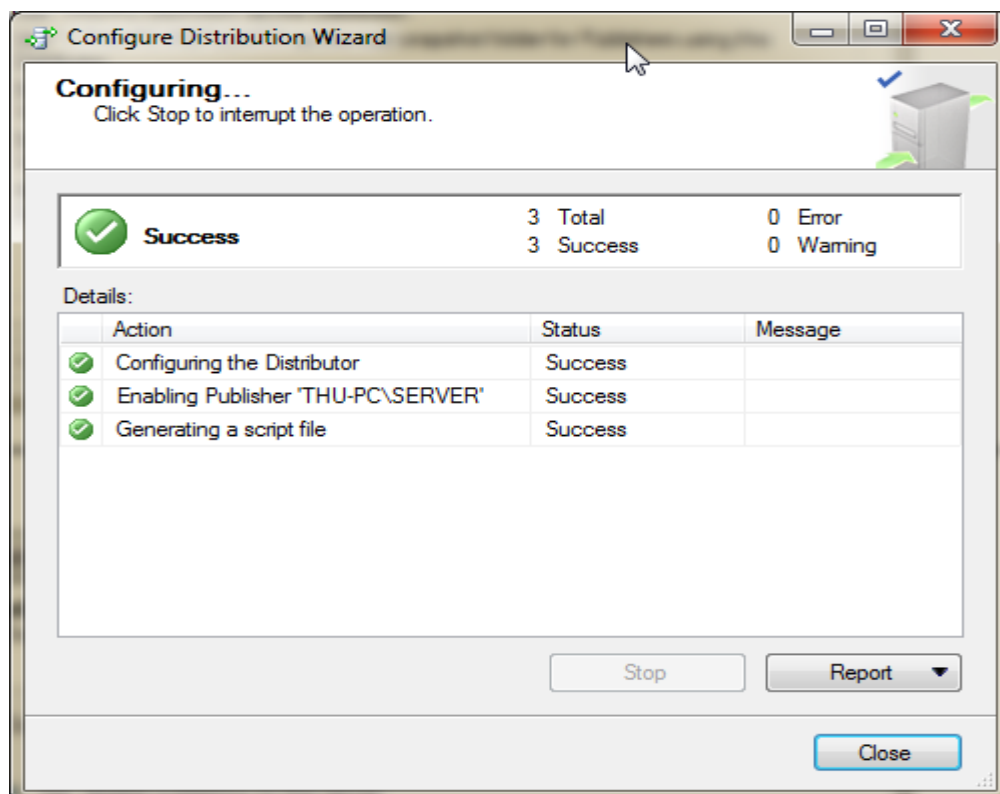
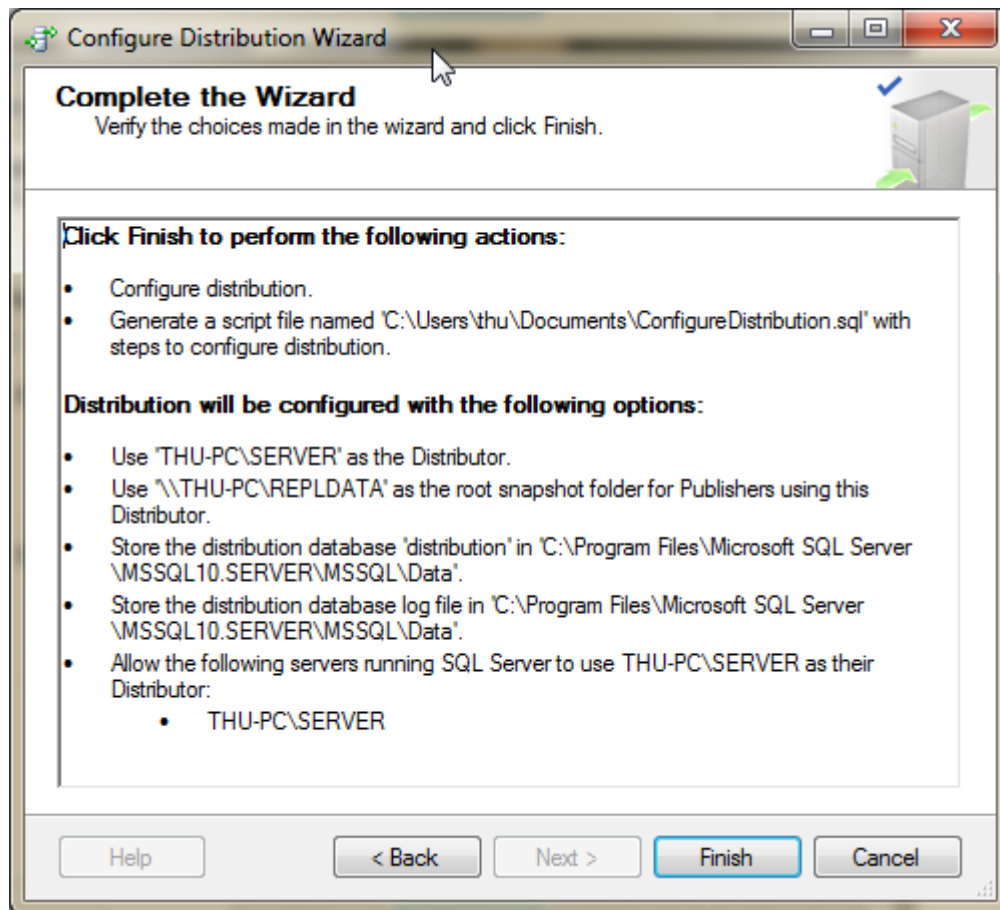


- Optionally enable other Publishers to use the Distributor. If other Publishers are enabled to use the Distributor, you must enter a password on the **Distributor Password** page for connections made from these Publishers to the Distributor.

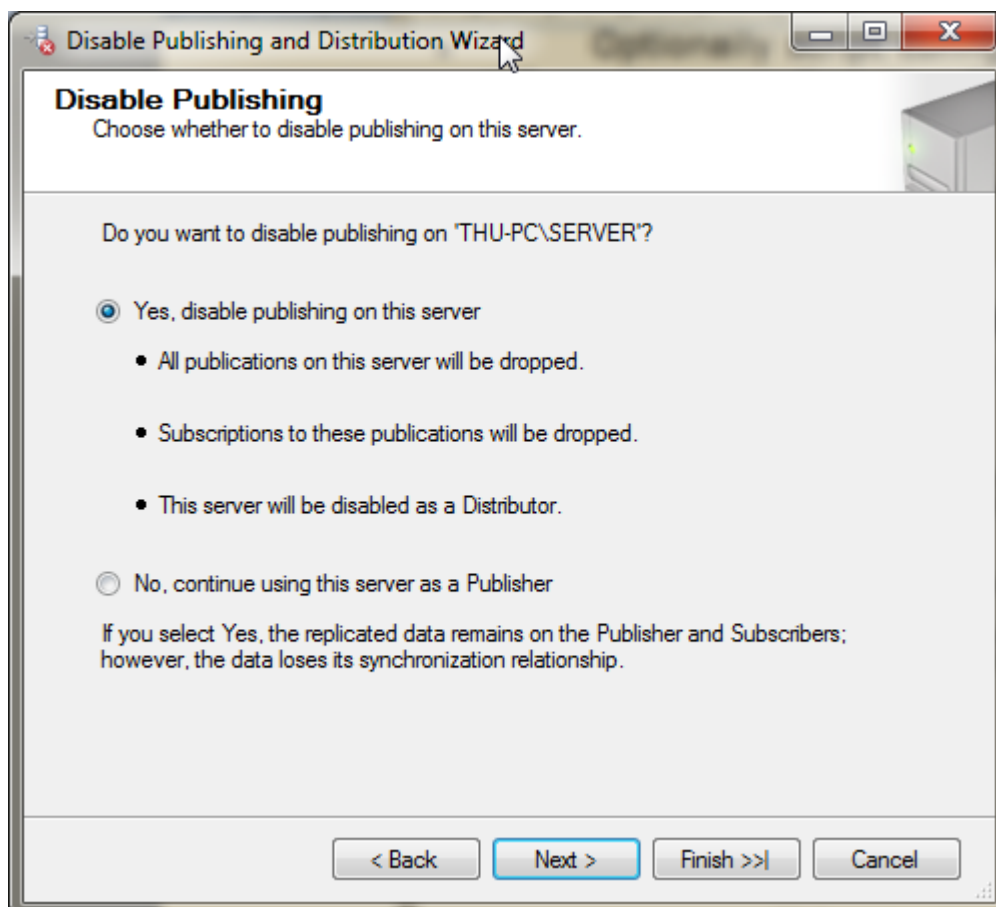
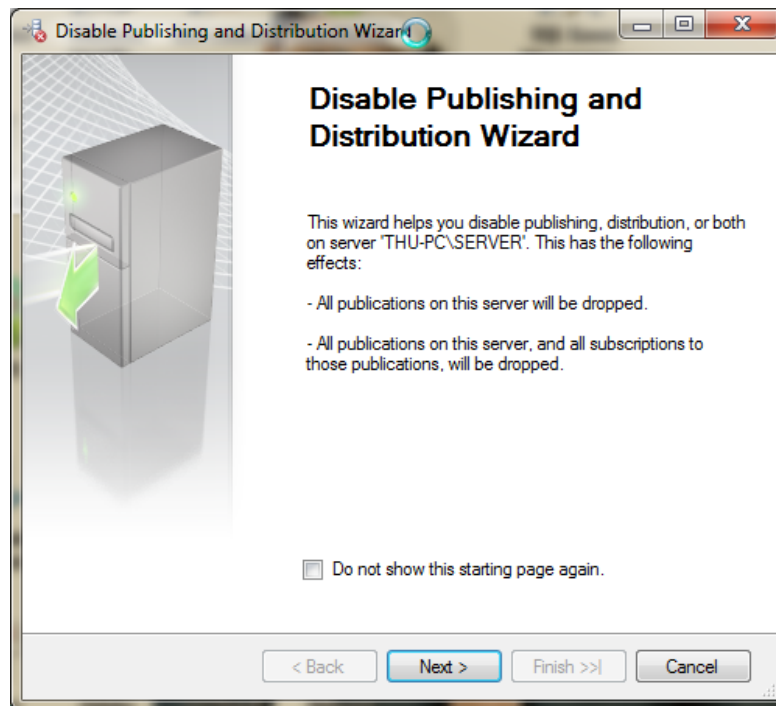


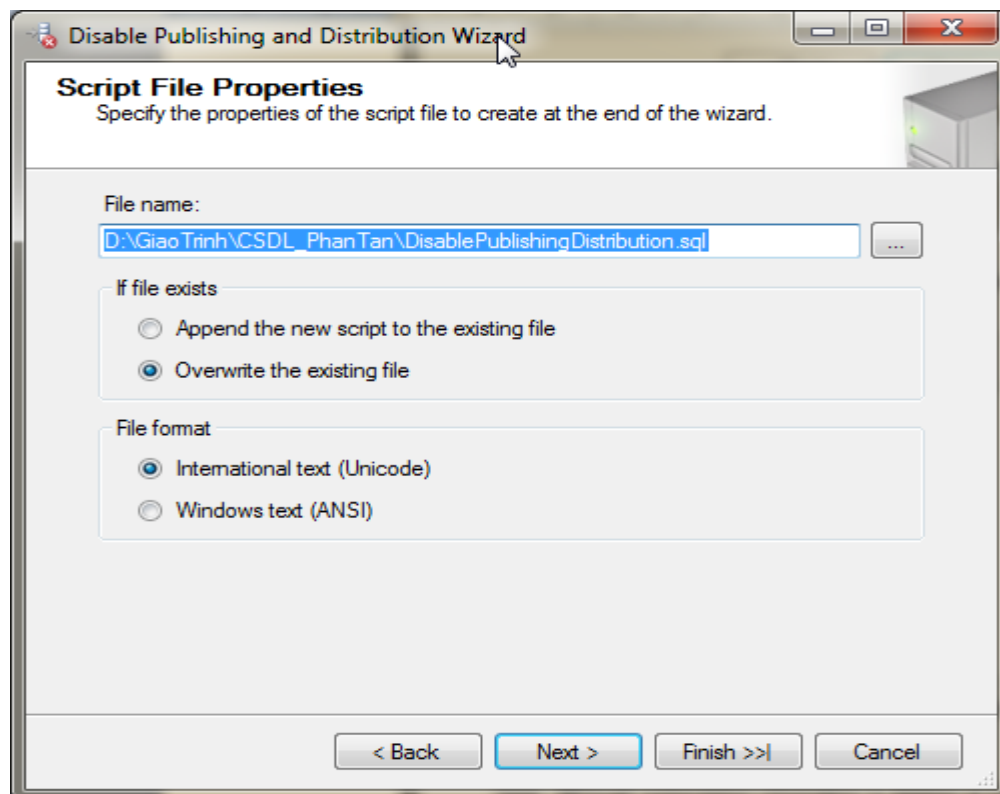
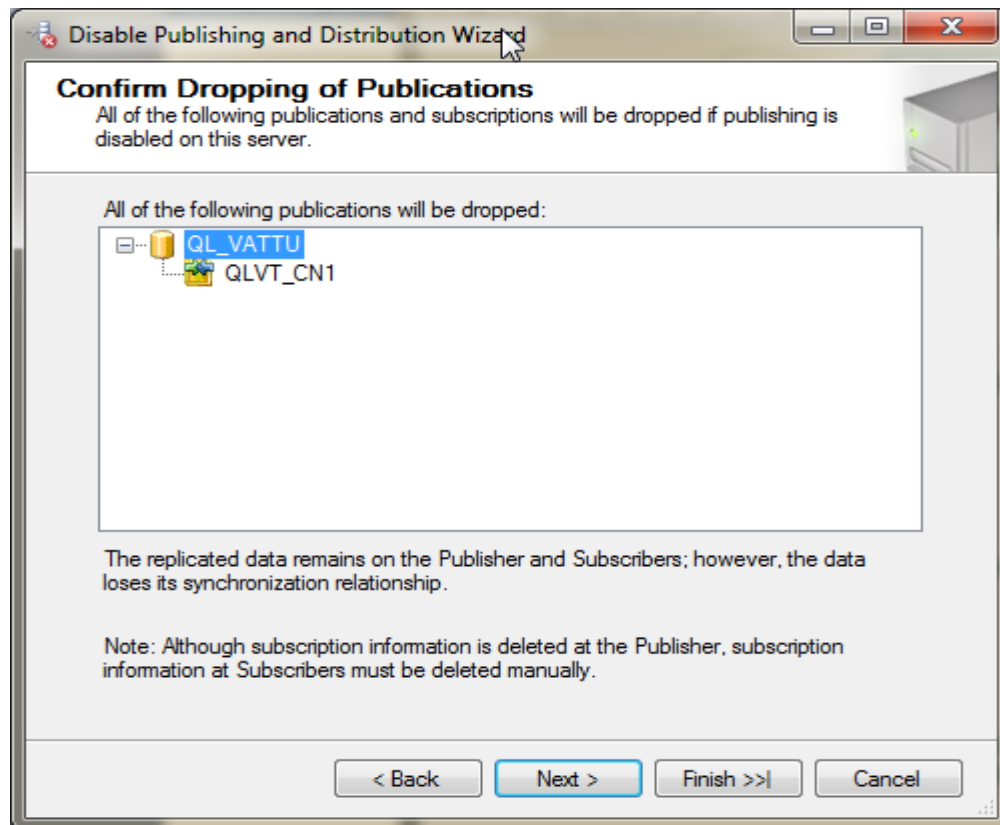
- Optionally script configuration settings. For more information, see Scripting Replication.

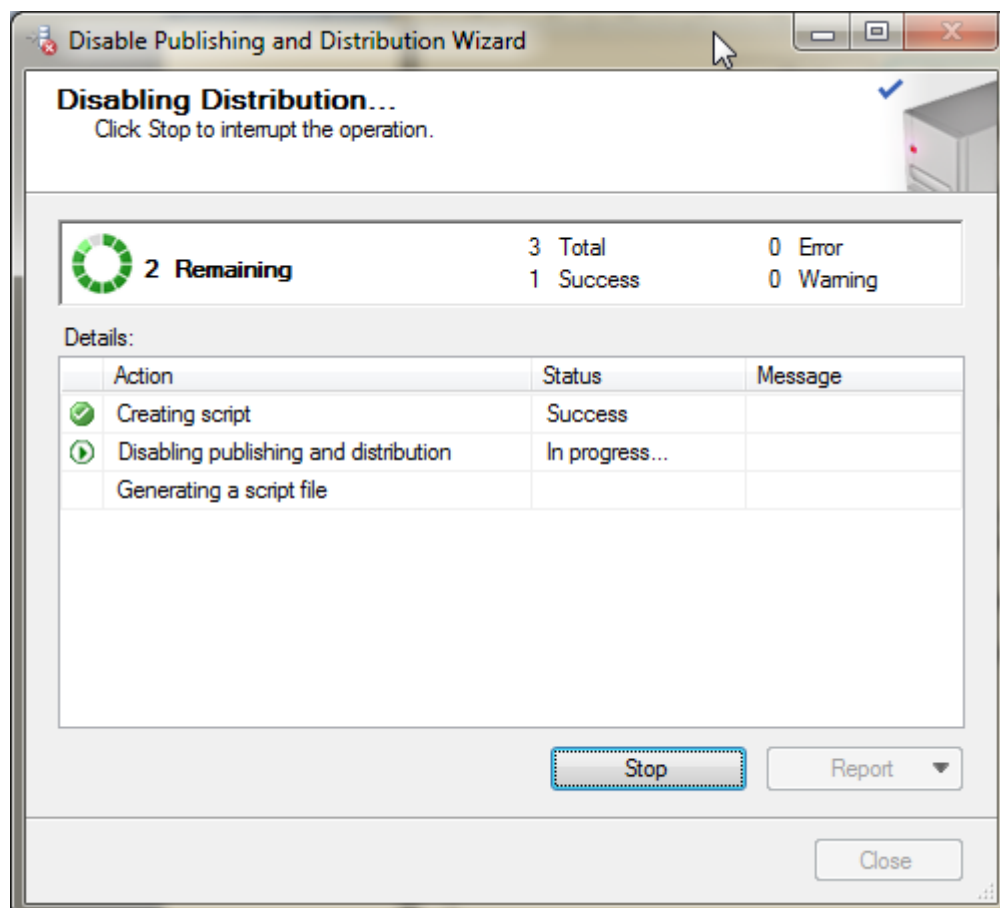
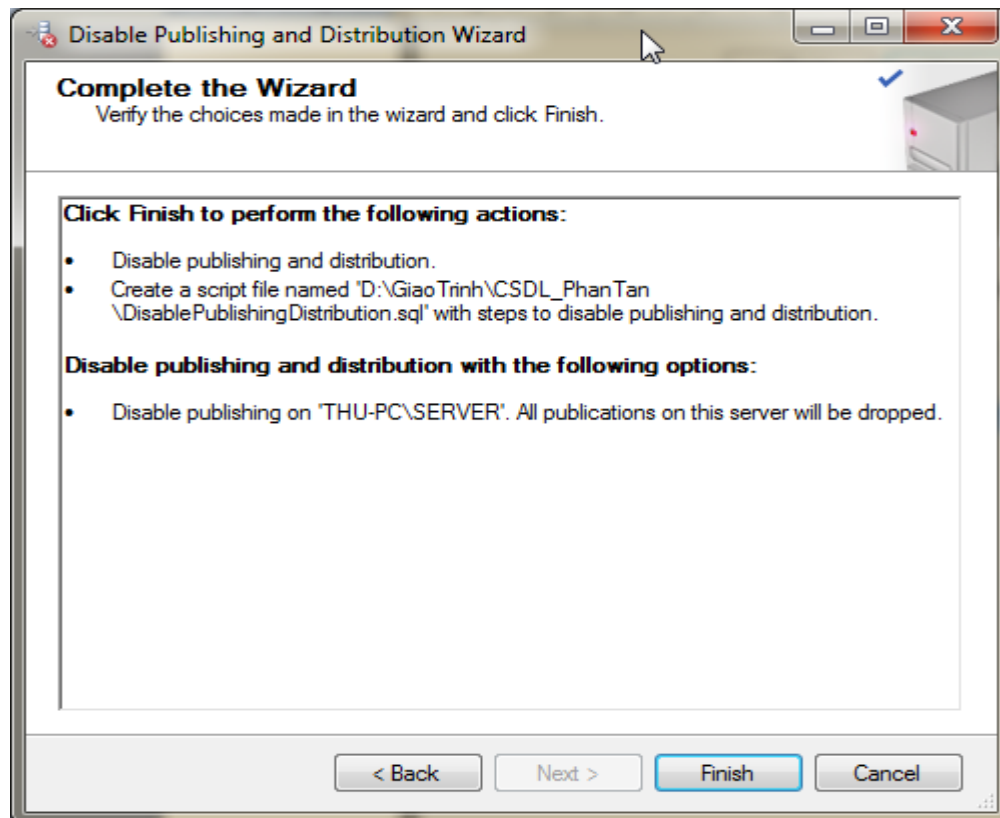


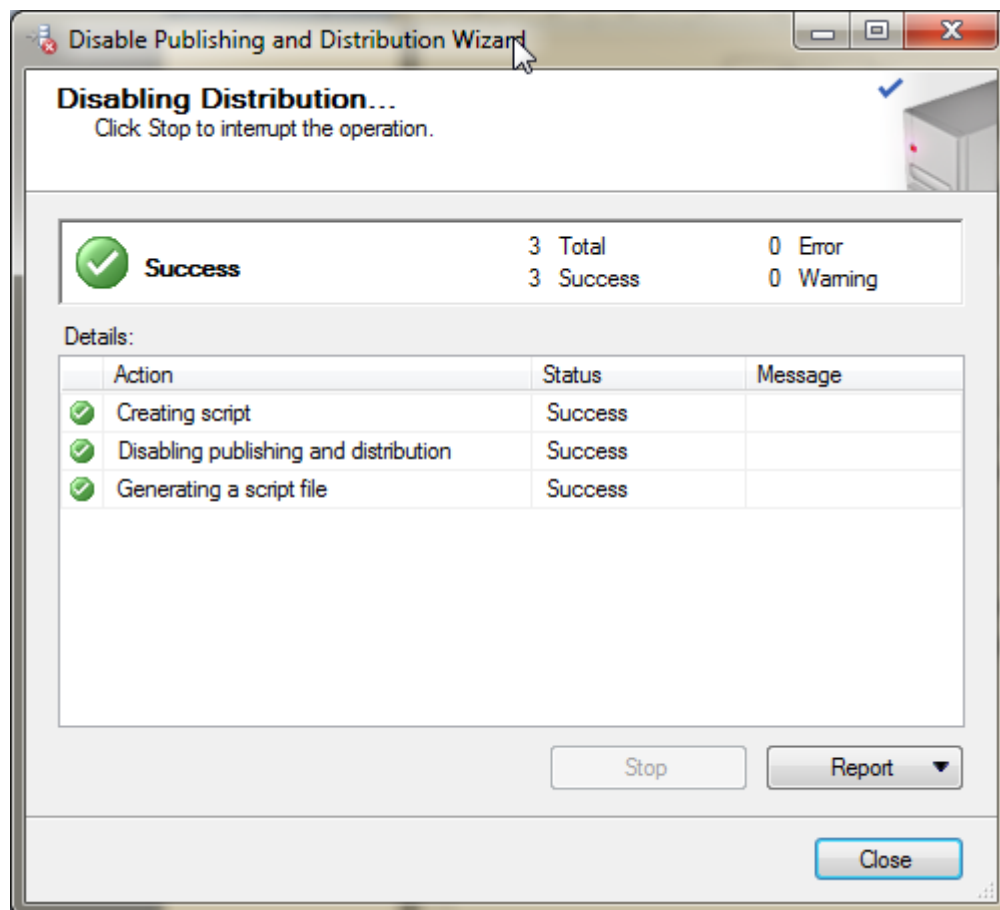


Note : Để DISABLE các Server Publisher và Distributor, ta thực hiện các bước :









B. Create publications

Create publications and define articles with the New Publication Wizard. After a publication is created, view and modify publication properties in the **Publication Properties - <Publication>** dialog box.

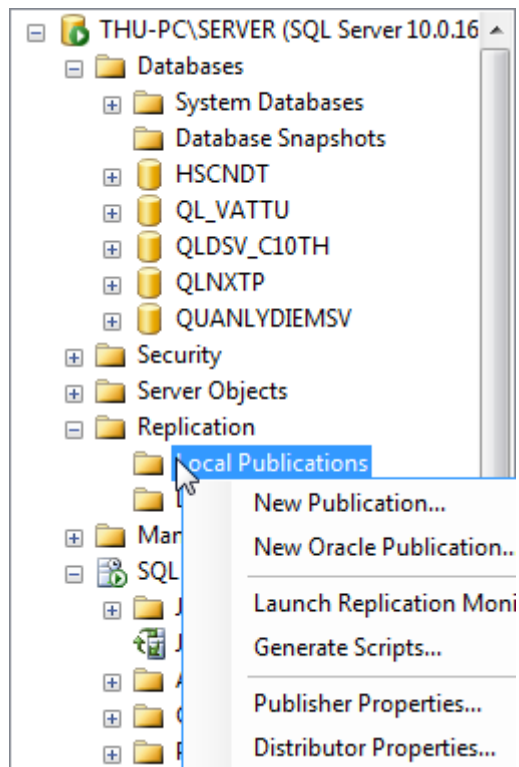
Note:

Publication and article names cannot include any of the following characters: % , * , [,] , | , : , " , ? , ' , \ , / , < , > . If objects in the database include any of these characters and you want to replicate them, you must specify an article name that is different from the object name in the **Article Properties - <Article>** dialog box, which is available from the **Articles** page in the wizard.

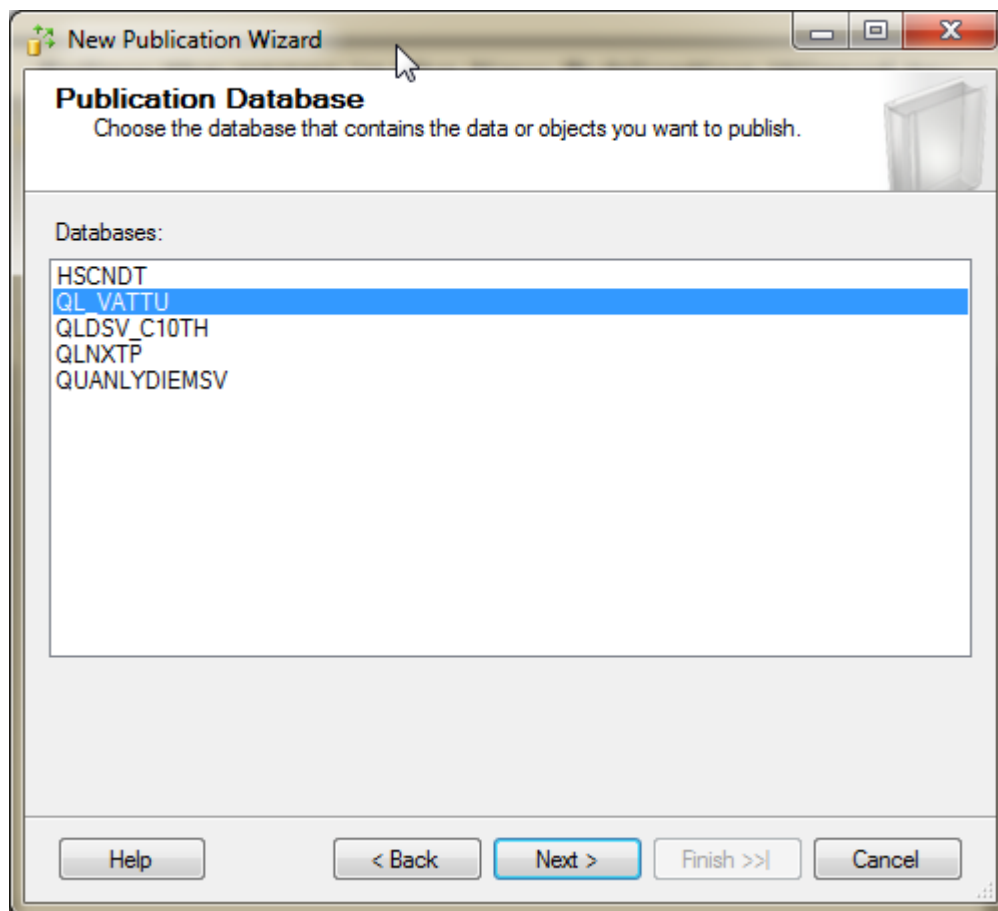
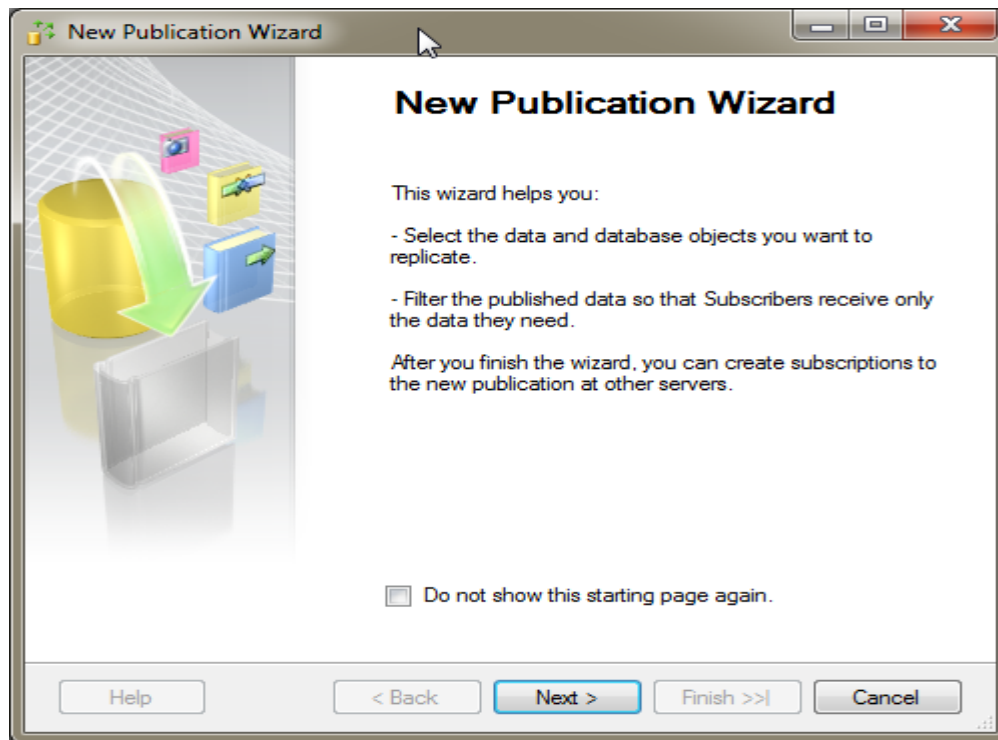


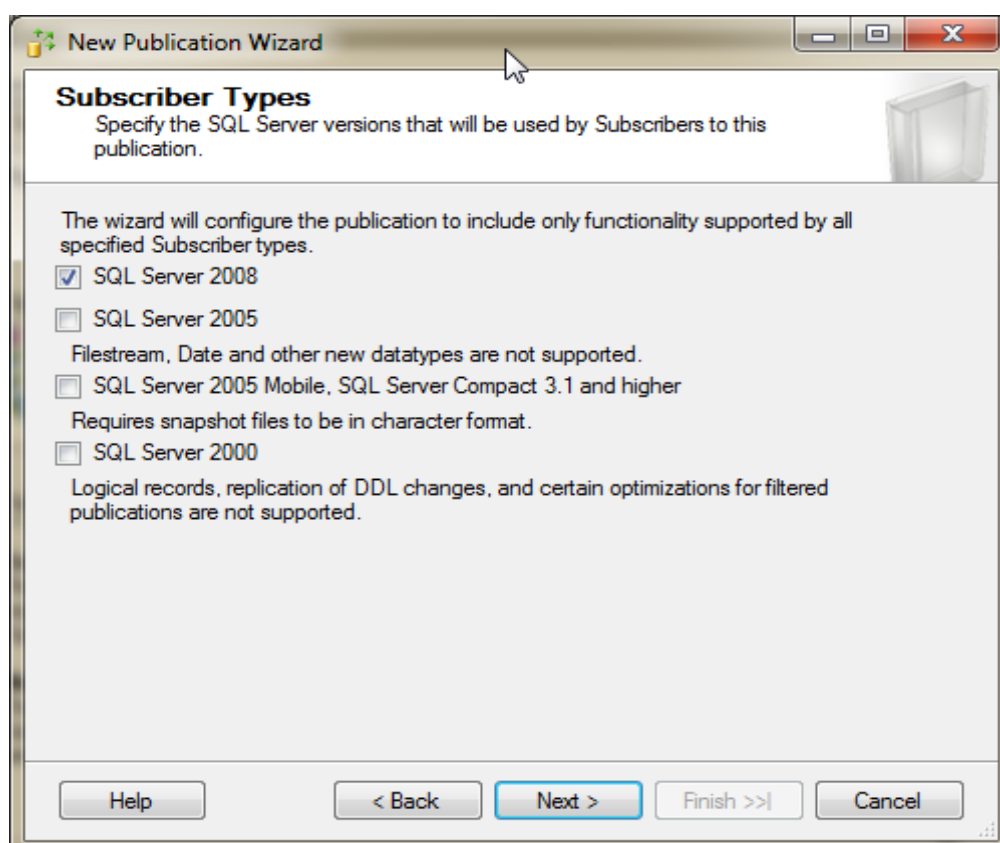
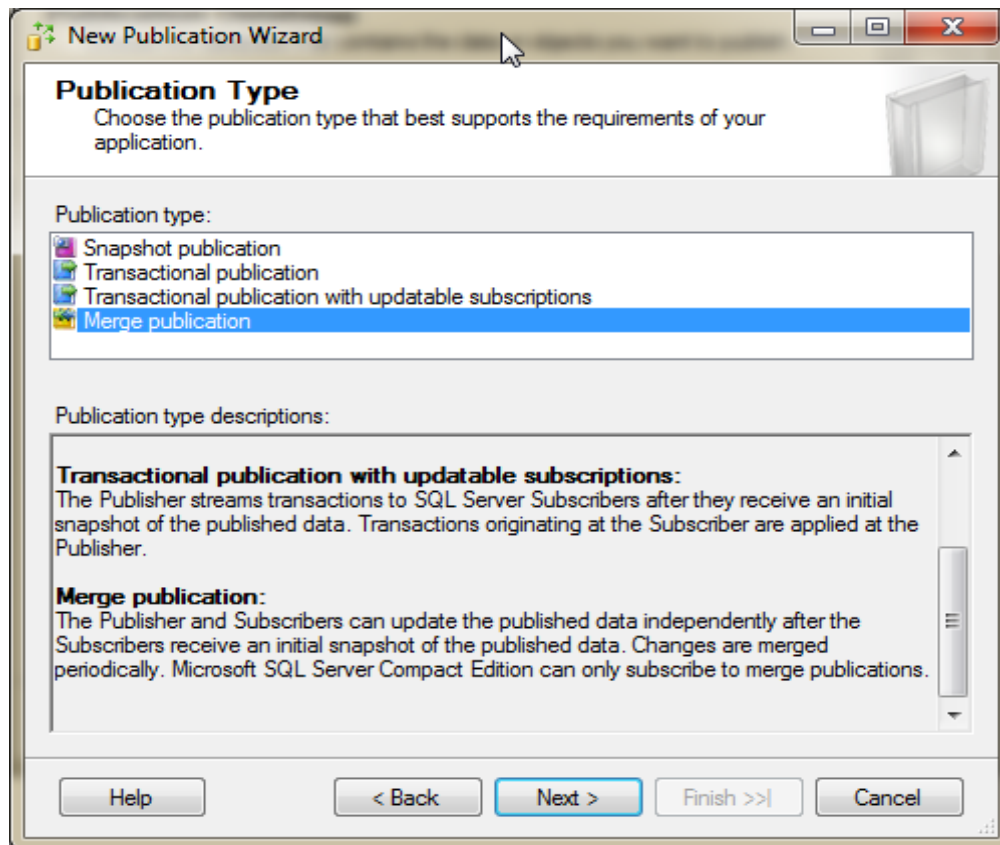
To create a publication and define articles

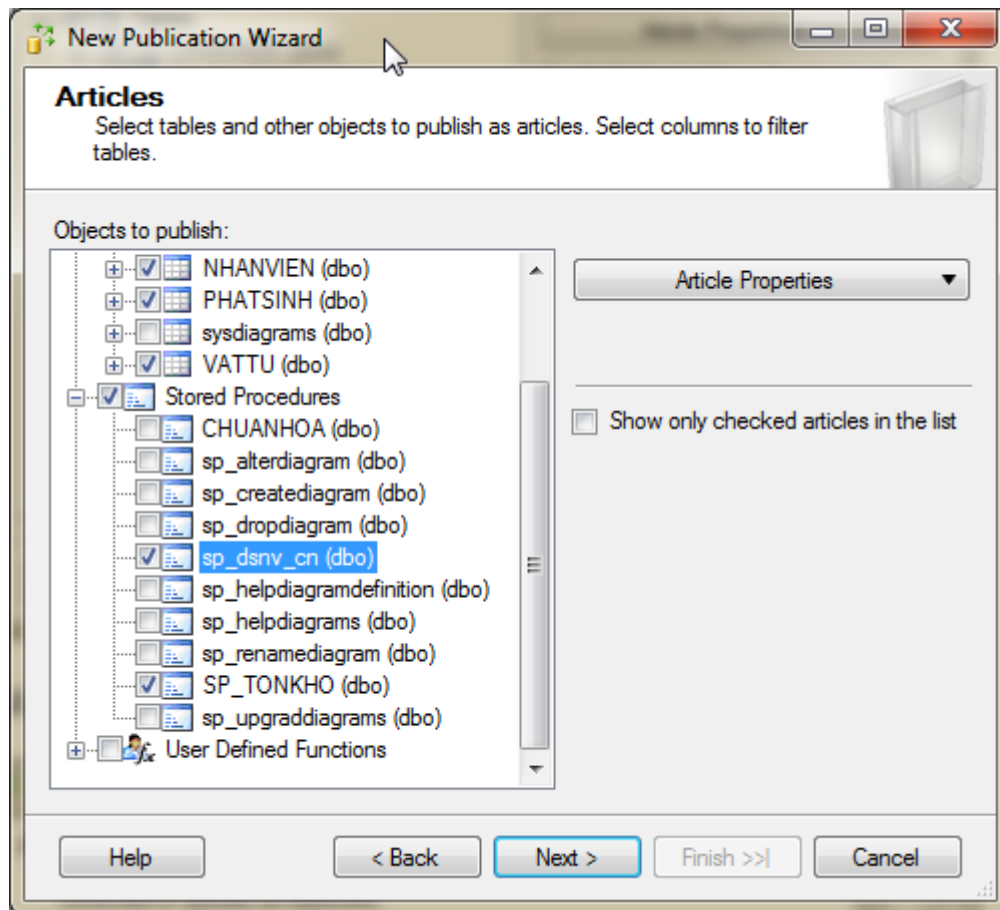
1. Connect to the Publisher in Microsoft SQL Server Management Studio, and then expand the server node.
2. Expand the **Replication** folder, and then right-click the **Local Publications** folder.
3. Click **New Publication**.



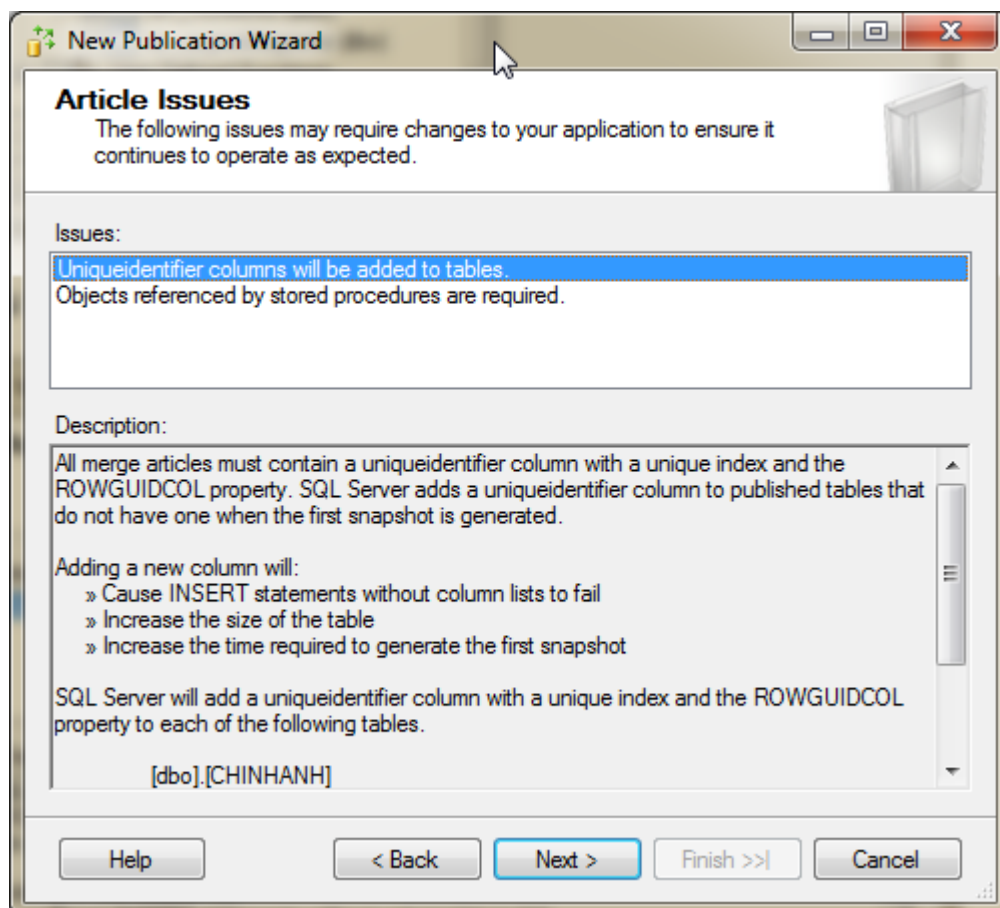
4. Follow the pages in the New Publication Wizard to:

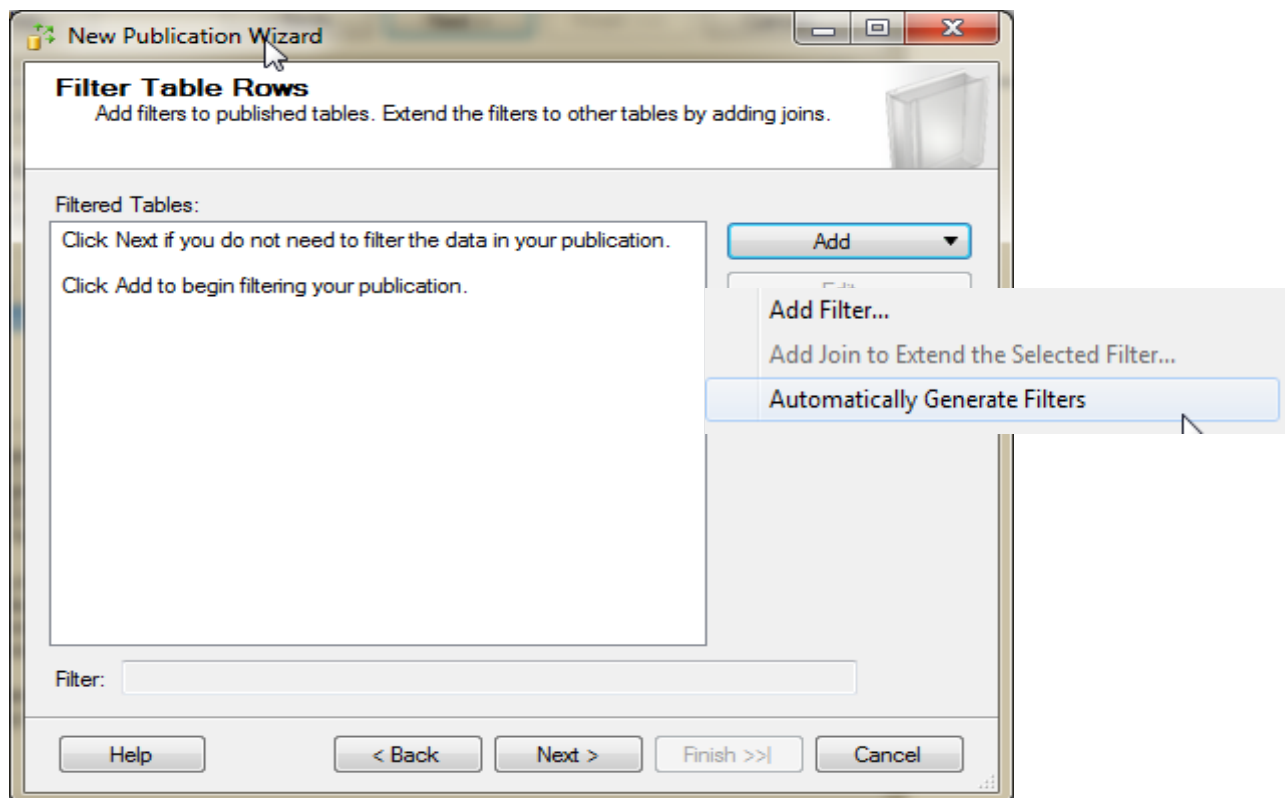




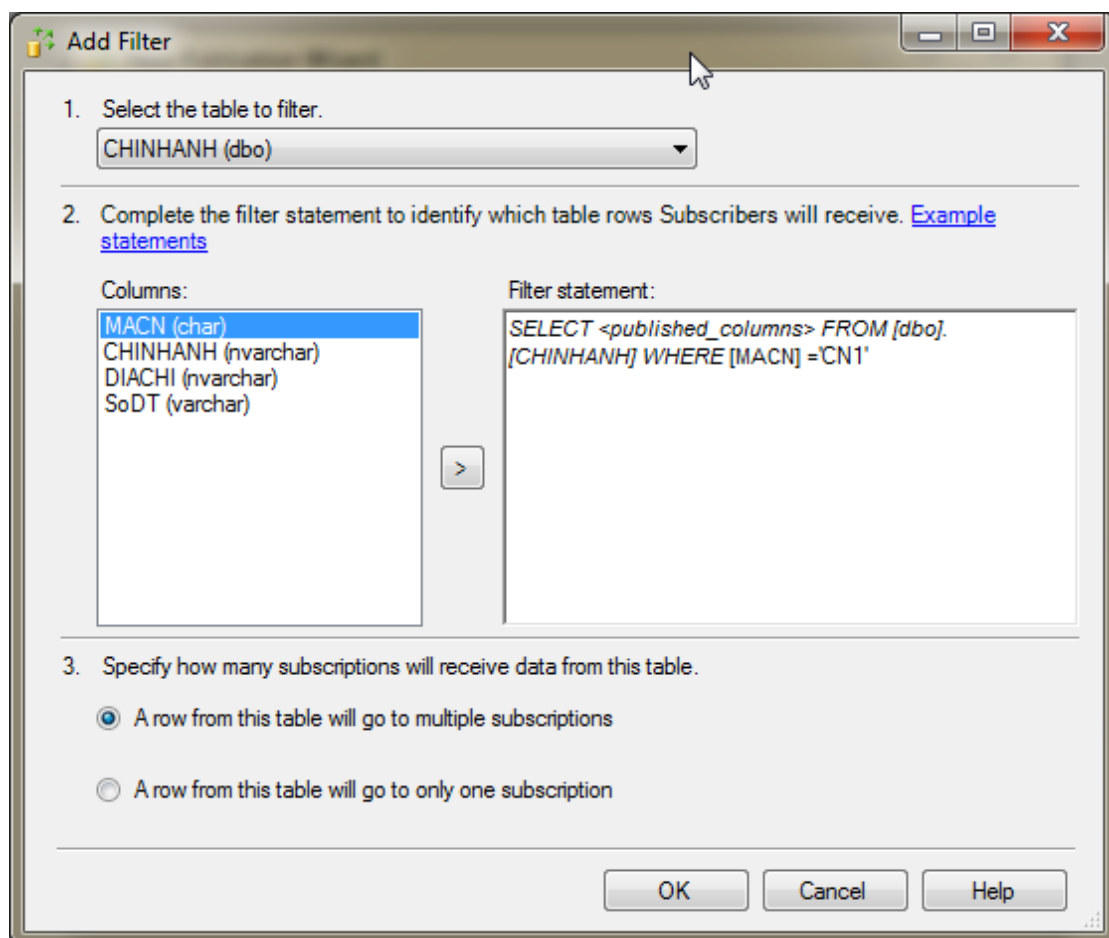


Lưu ý: **bỏ chọn** table sysdiagrams



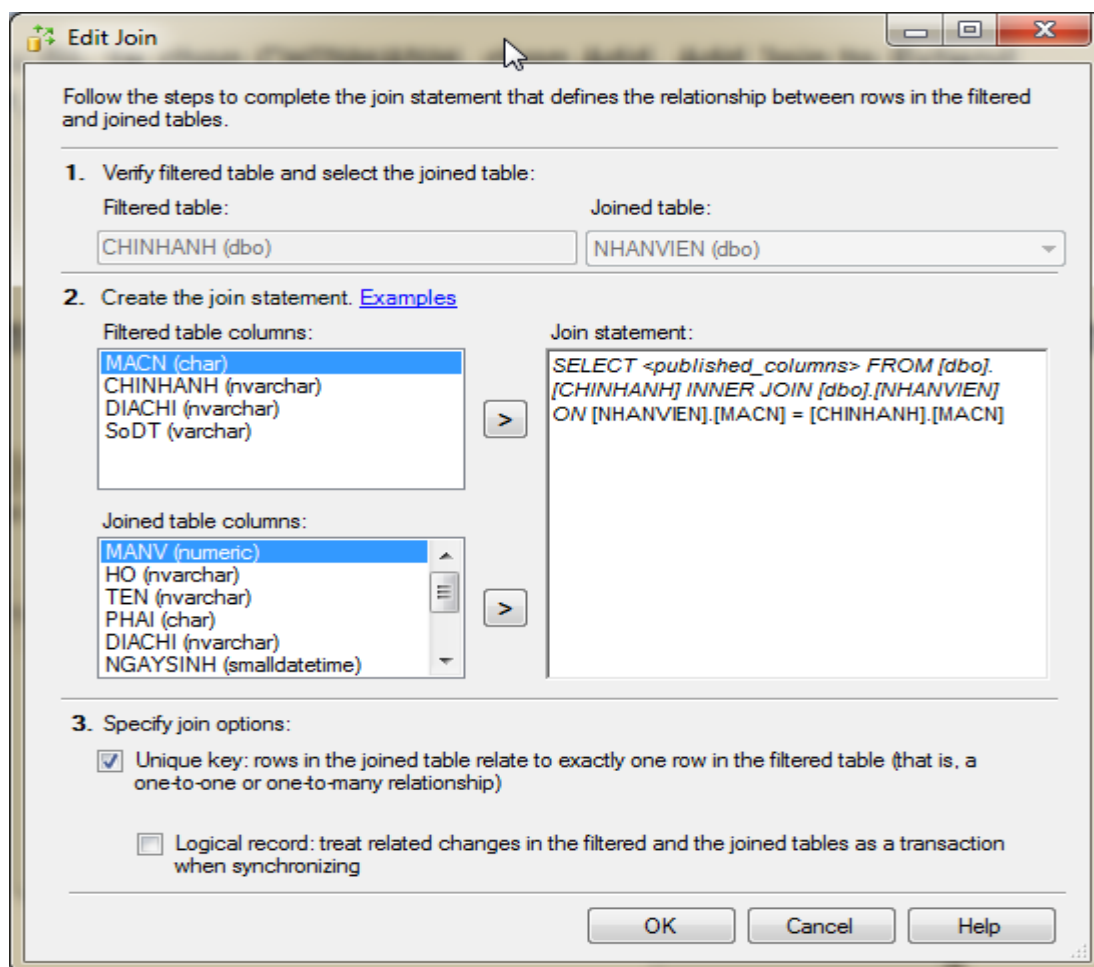


Chọn Add Filter, chọn CHINHANH và đưa vào điều kiện phân tán như hộp thoại dưới đây:

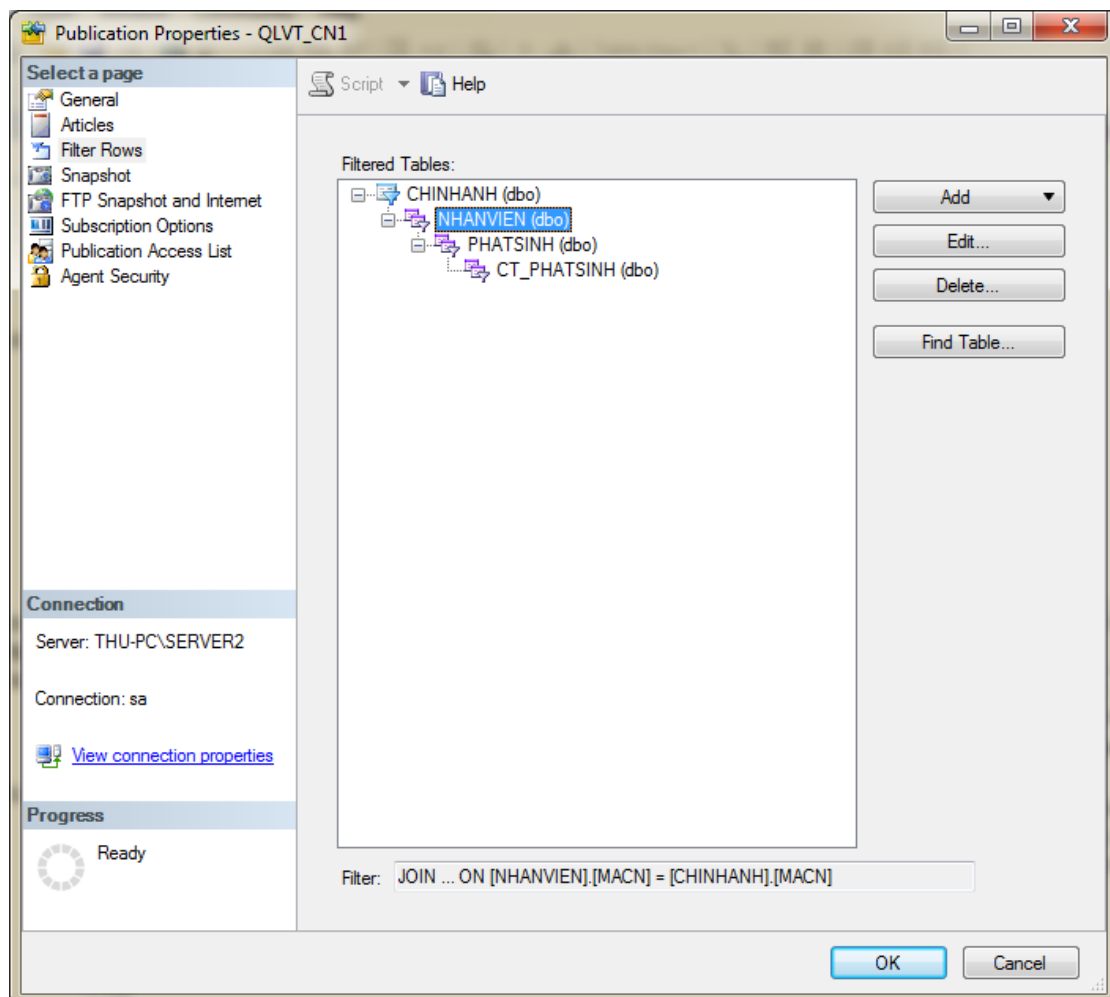


Click OK, để thoát ra ngoài

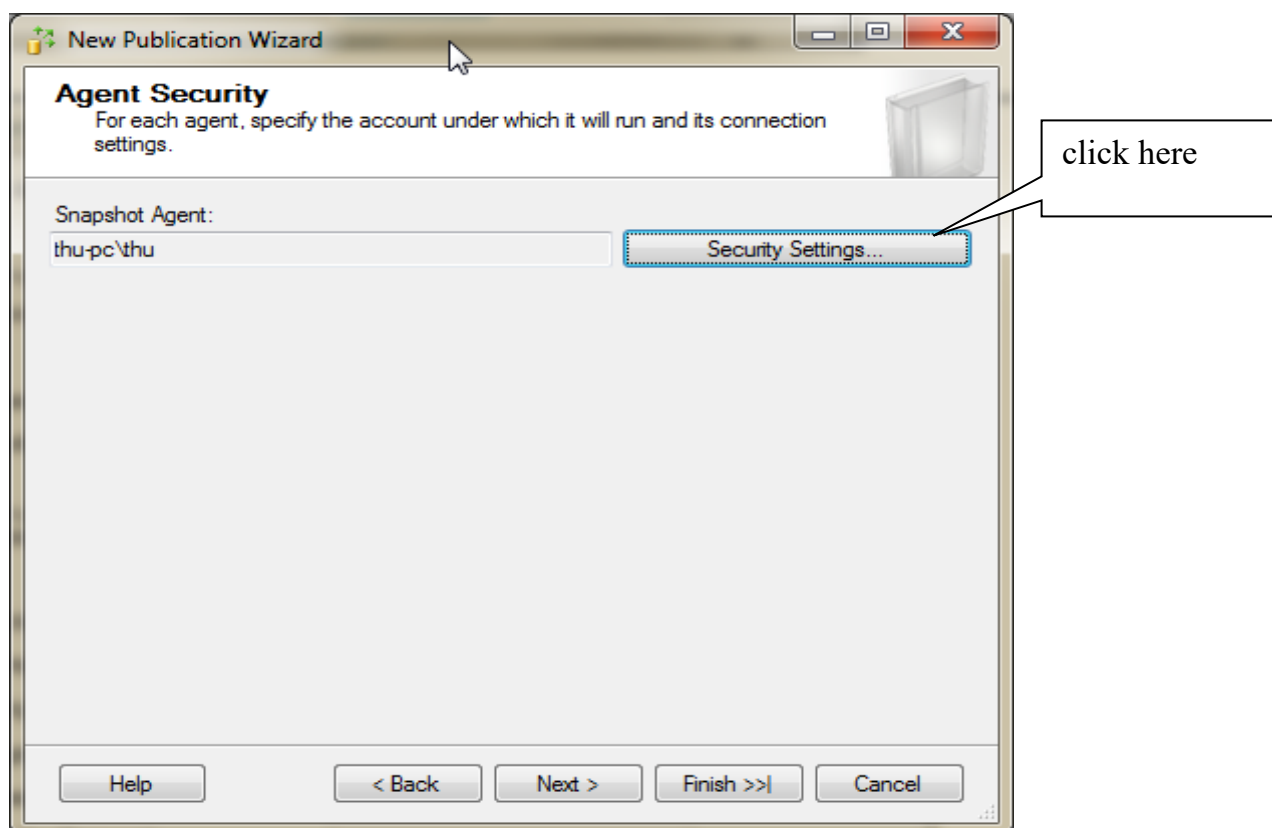
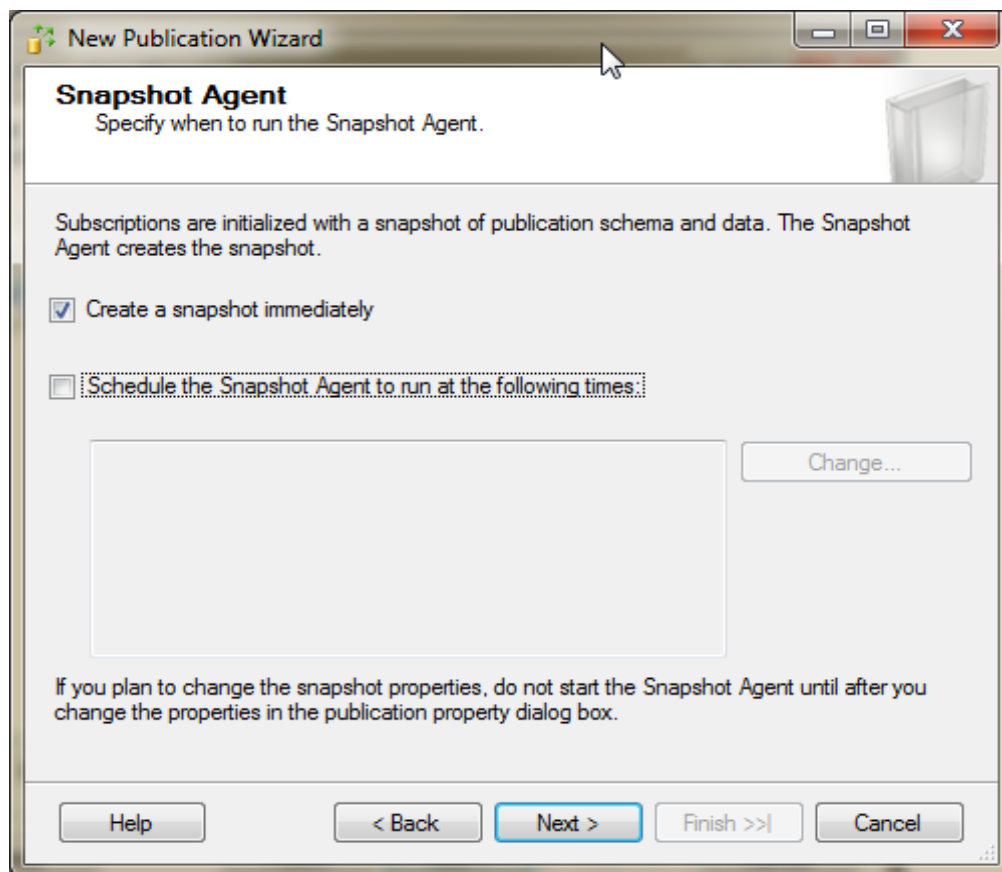
Sau đó, ta chọn CHINHANH, chọn Add, Add Join to Extend the Selected Filter để chọn NHANVIEN là quan hệ dẫn xuất theo CHINHANH như hình sau:



Tương tự, ta tạo các phân mảnh ngang còn lại như trong hình sau:



click OK để qua bước kế:



Snapshot Agent Security

Specify the domain or machine account under which the Snapshot Agent process will run.

☒ Run under the following Windows account:

Process account:
Example: domain\account

Password:
Confirm Password:

☐ Run under the SQL Server Agent service account (This is not a recommended security best practice.)

Connect to the Publisher

☒ By impersonating the process account

☐ Using the following SQL Server login:

Login:
Password:
Confirm Password:

OK Cancel Help

New Publication Wizard

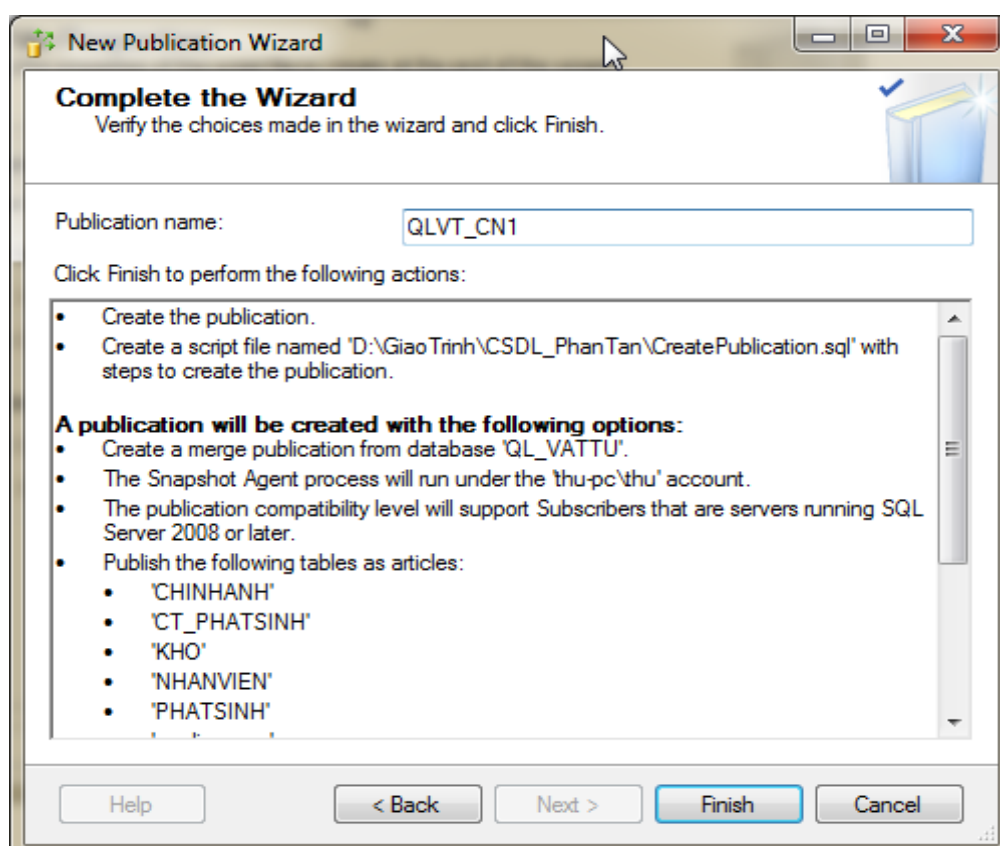
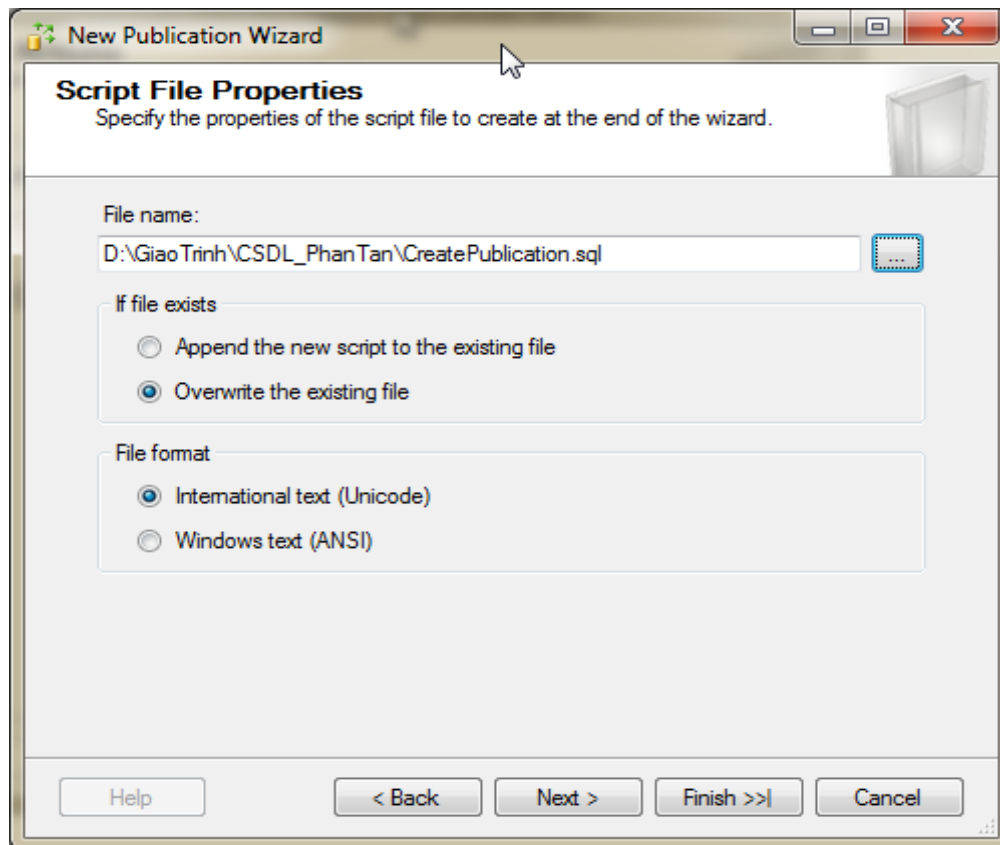
Wizard Actions
Choose what happens when you click Finish.

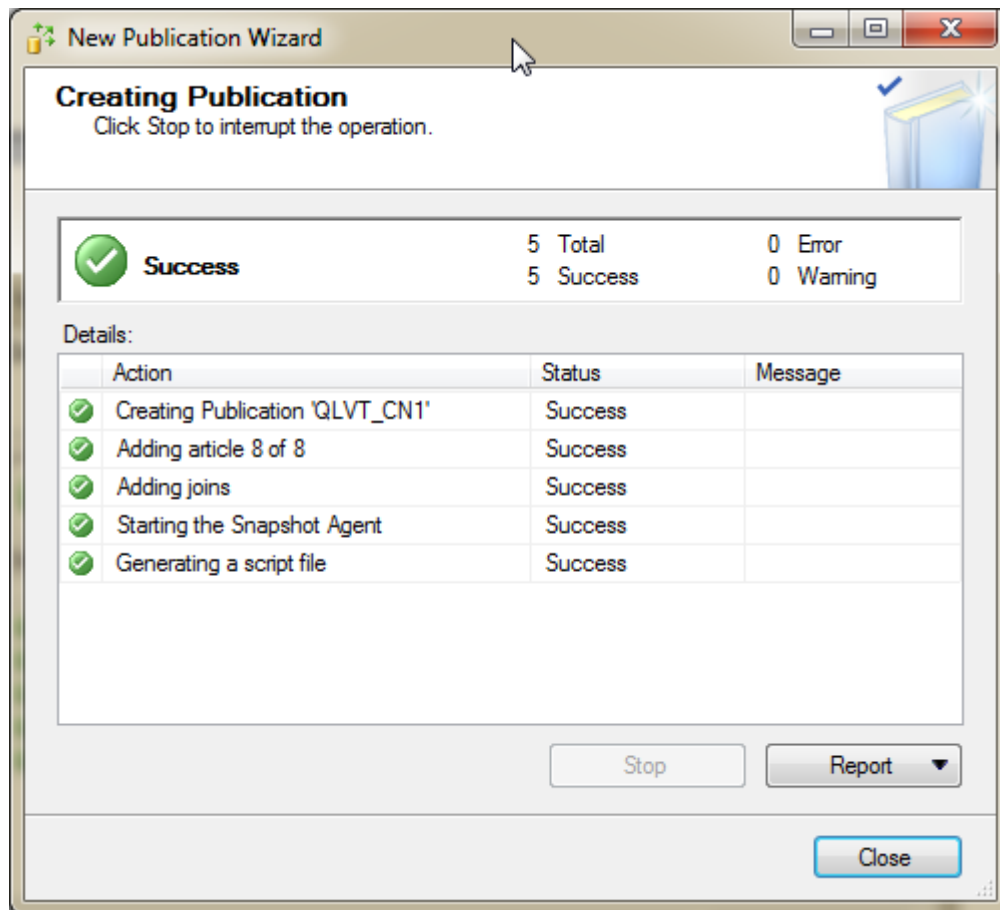
At the end of the wizard:

☒ Create the publication

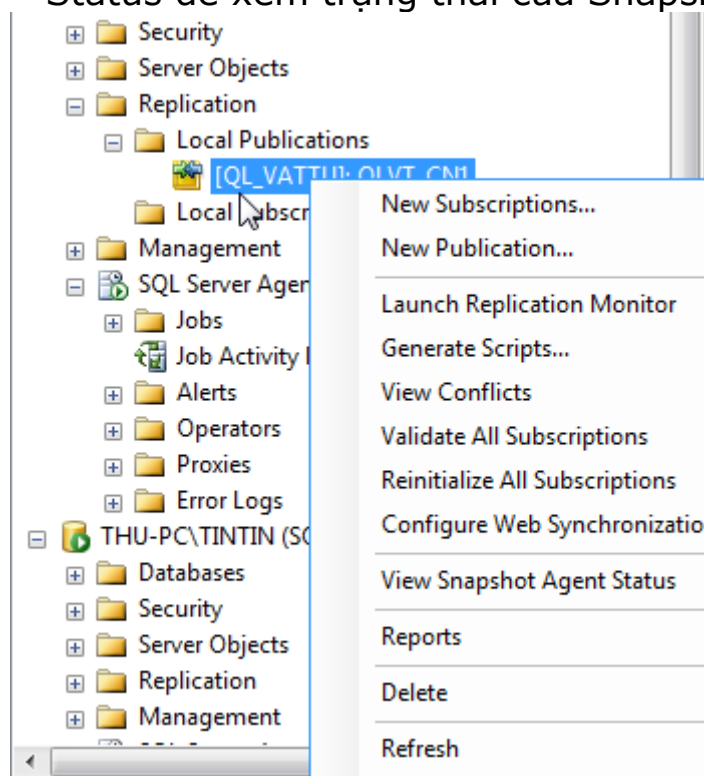
☒ Generate a script file with steps to create the publication

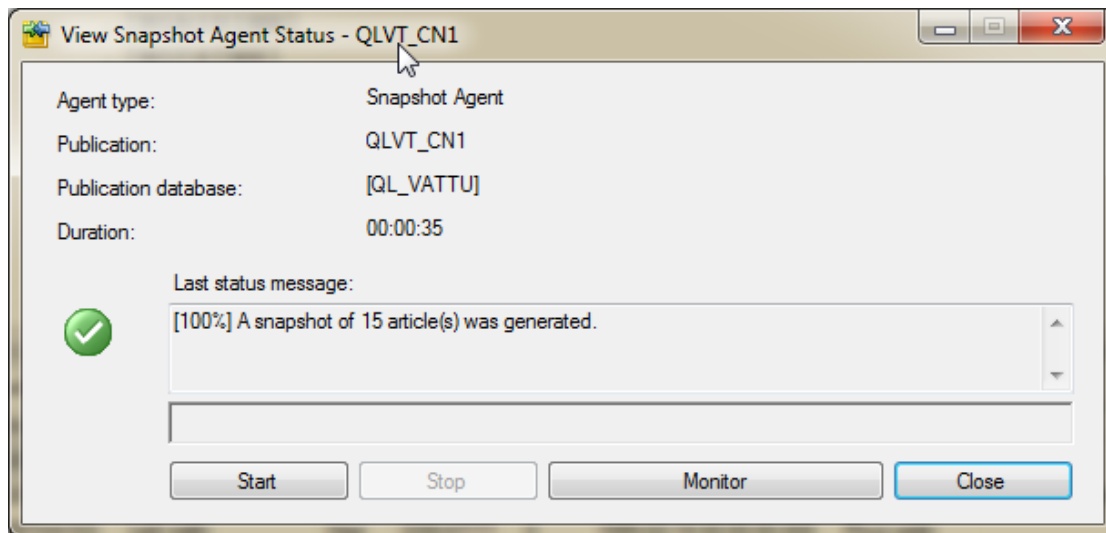
Help < Back Next > Finish >>| Cancel





- Ta đặt tên cho publication. Click chọn View Snapshot Agent Status để xem trạng thái của Snapshot Agent

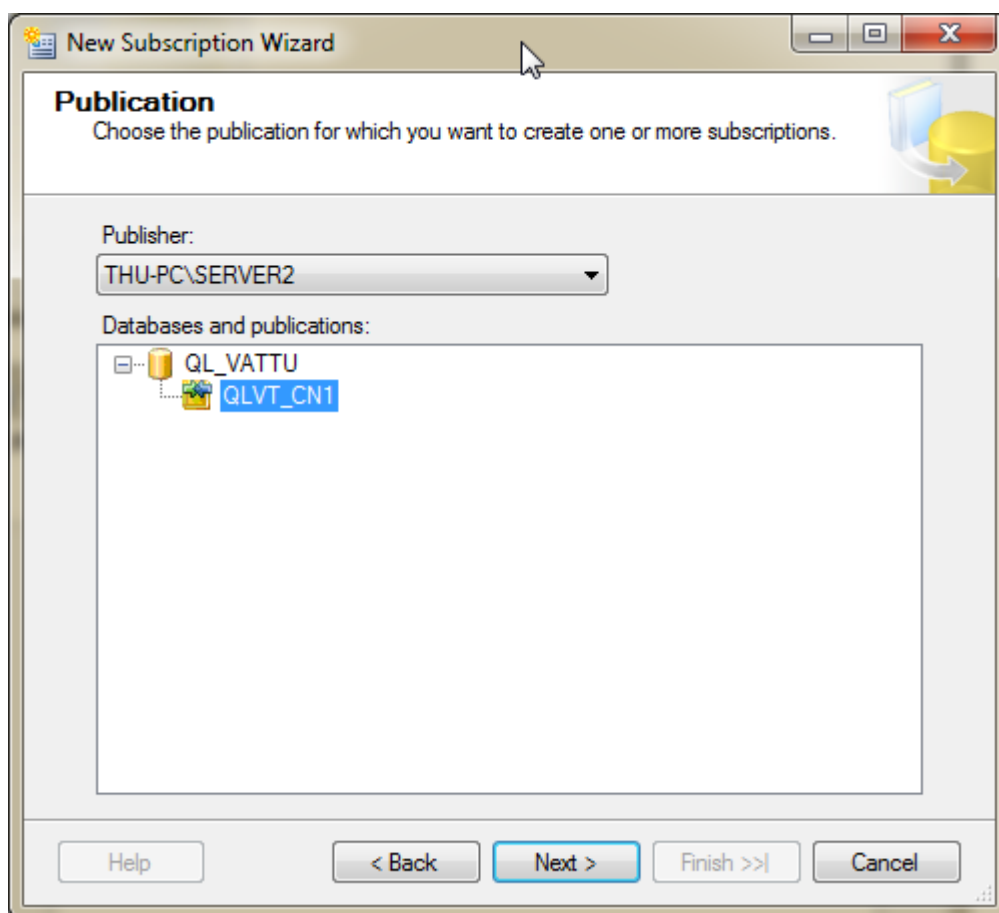
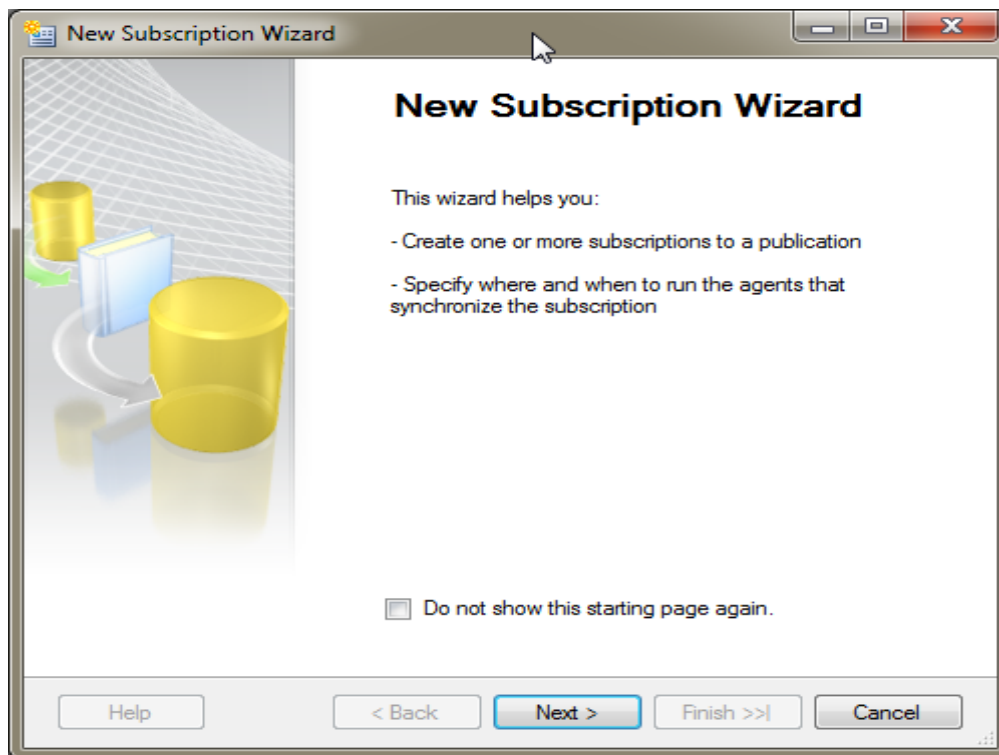


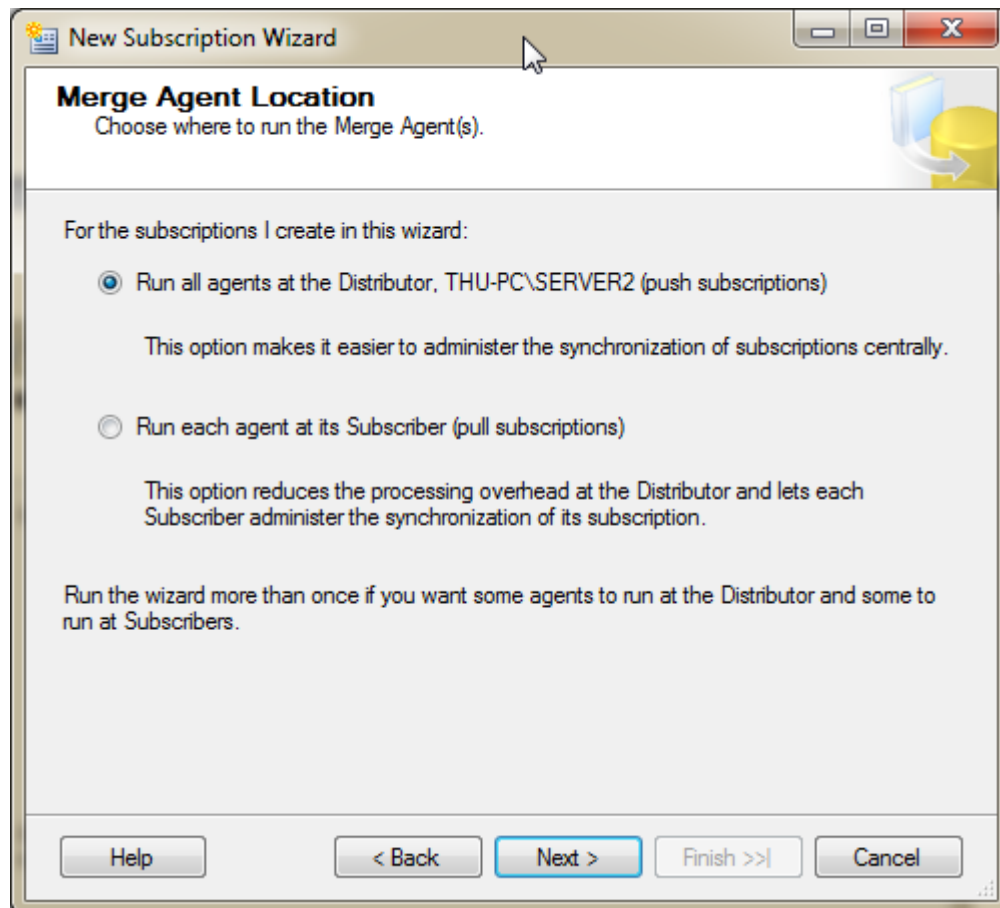


- Start : Cho Snapshot Agent hoạt động
- Monitor : mở window theo dõi quá trình đồng bộ dữ liệu .

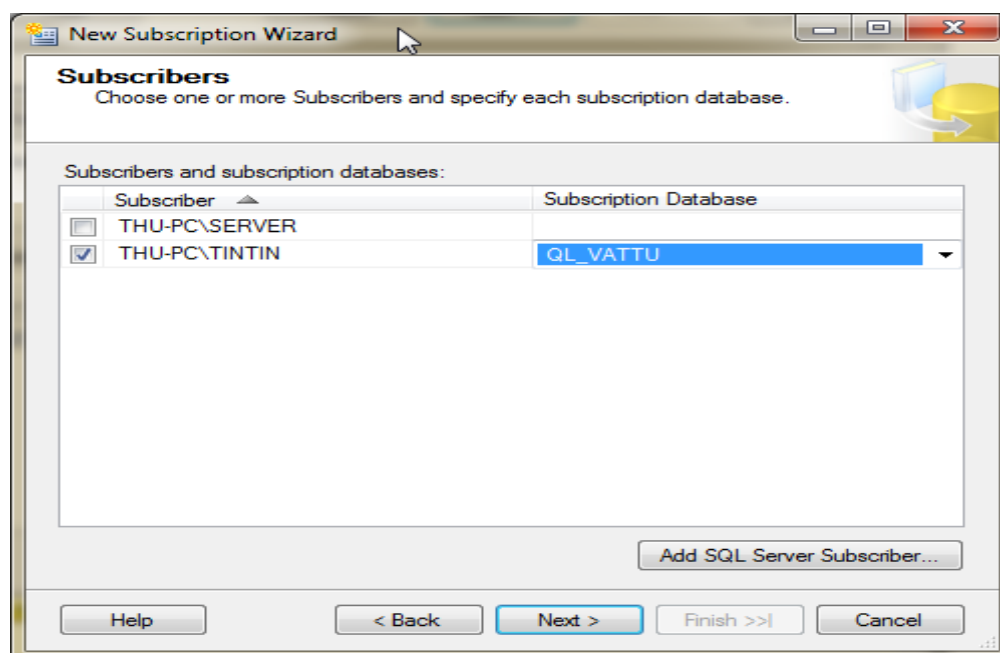
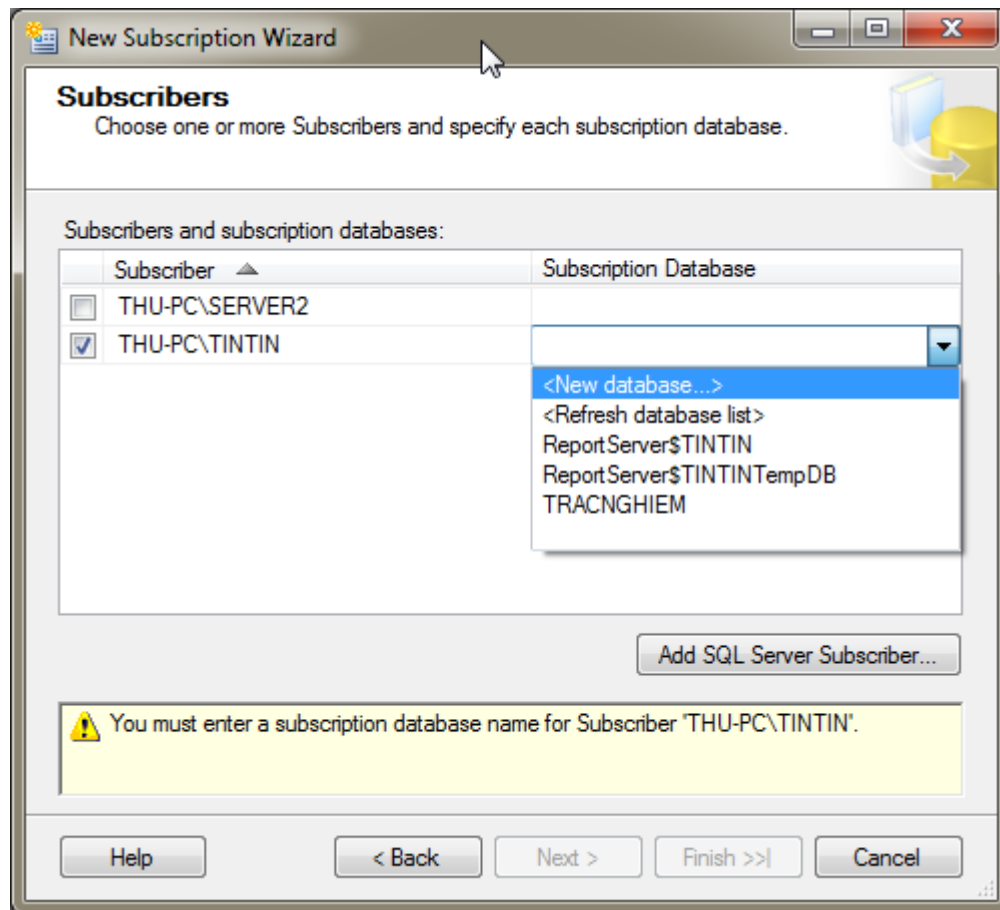
C. Tạo Subscription

Right Click trên 1 publication, chọn New Subscriptions ...





Click chọn nút lệnh Add SQL Server Subscriber để chỉ định 1 Server làm nơi chứa cơ sở dữ liệu phân tán. Sau đó, ta chỉ định tiếp 1 cơ sở dữ liệu làm nơi chứa các Article (nên là 1 cơ sở dữ liệu mới)



Merge Agent Security

Specify the domain or machine account under which the Merge Agent process will run when synchronizing this subscription.

☒ **Run under the following Windows account:**

Process account:
Example: domain\account

Password:

Confirm Password:

☐ **Run under the SQL Server Agent service account (This is not a recommended security best practice.)**

Connect to the Publisher and Distributor

☒ **By impersonating the process account**

☐ **Using the following SQL Server login:**

Login:

Password:

Confirm password:

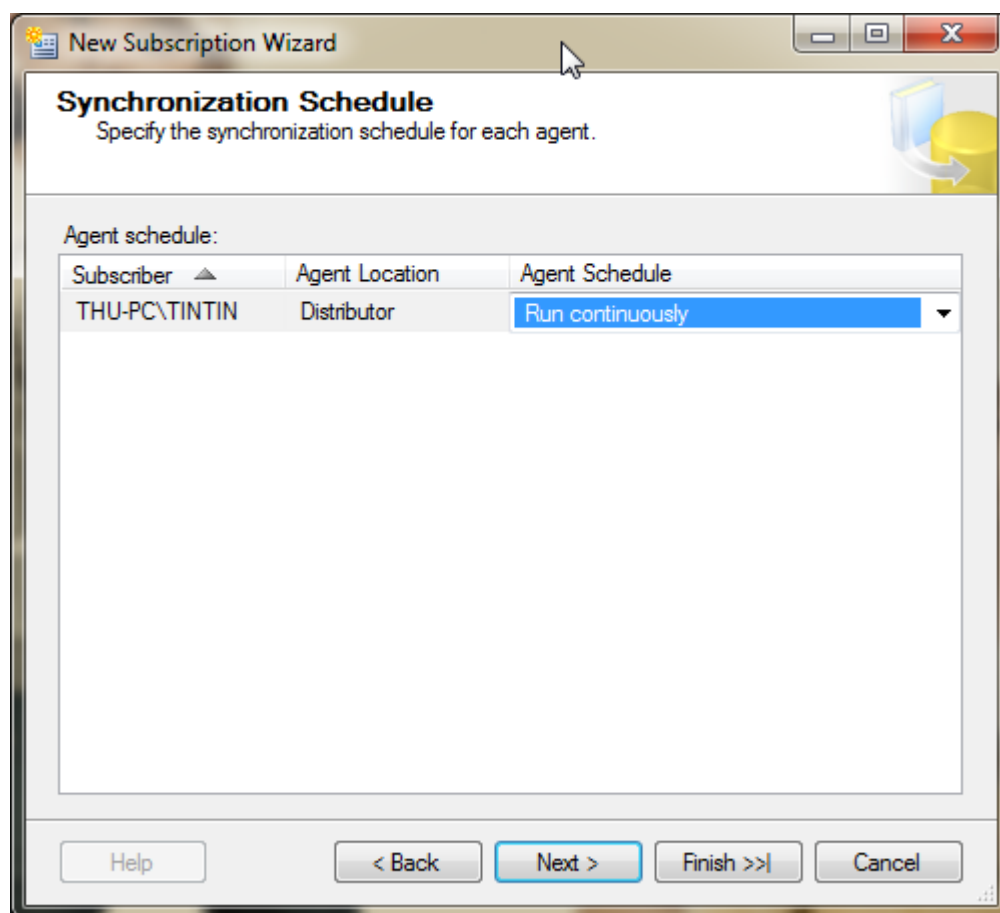
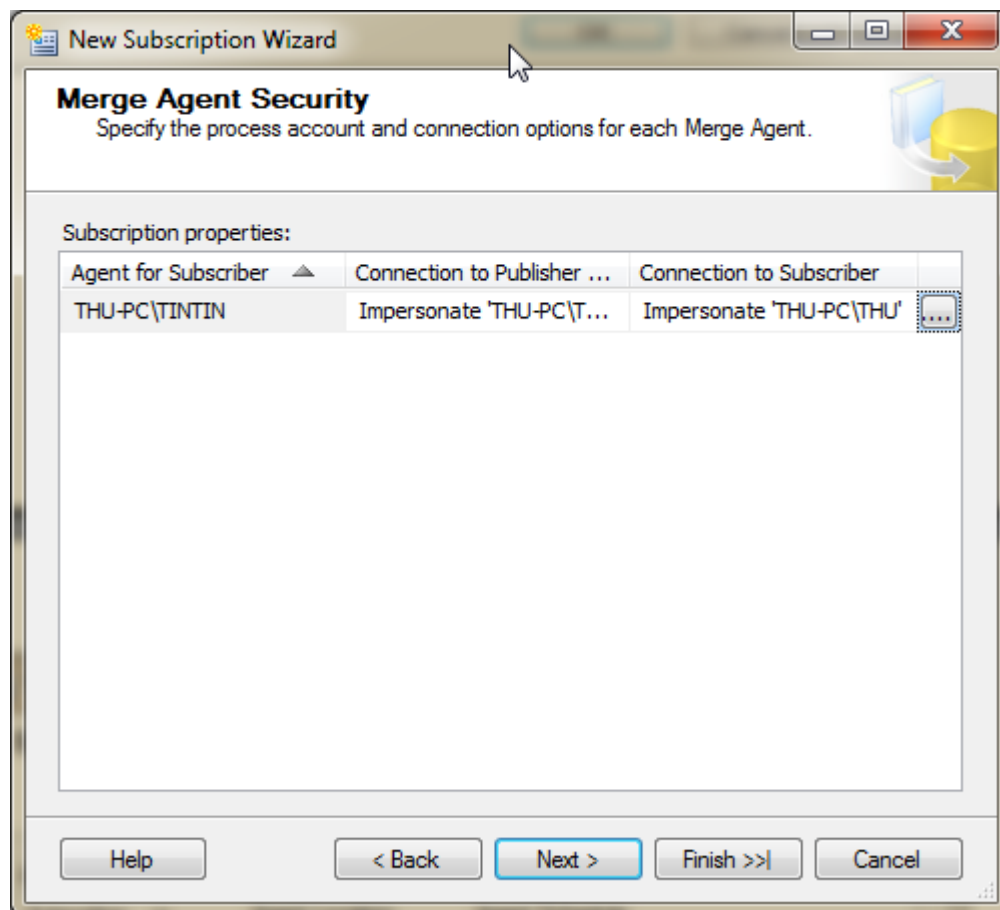
The login used to connect to the Publisher must be a member of the Publication Access List.

Connect to the Subscriber

☒ **By impersonating the process account**

☐ **Using a SQL Server login**

The connection to the server on which the agent runs must impersonate the process account.
The process account must be a database owner of the subscription database.



New Subscription Wizard

Initialize Subscriptions
Specify whether to initialize each subscription with a snapshot of the publication data and schema.

Subscription properties:

Subscriber	Initialize	Initialize When
THU-PC\TINTIN	<input checked="" type="checkbox"/>	Immediately

A subscription database needs to be initialized with a snapshot of the publication data and schema unless it has already been specially prepared for the subscription.

Help < Back Next > Finish >>| Cancel

New Subscription Wizard

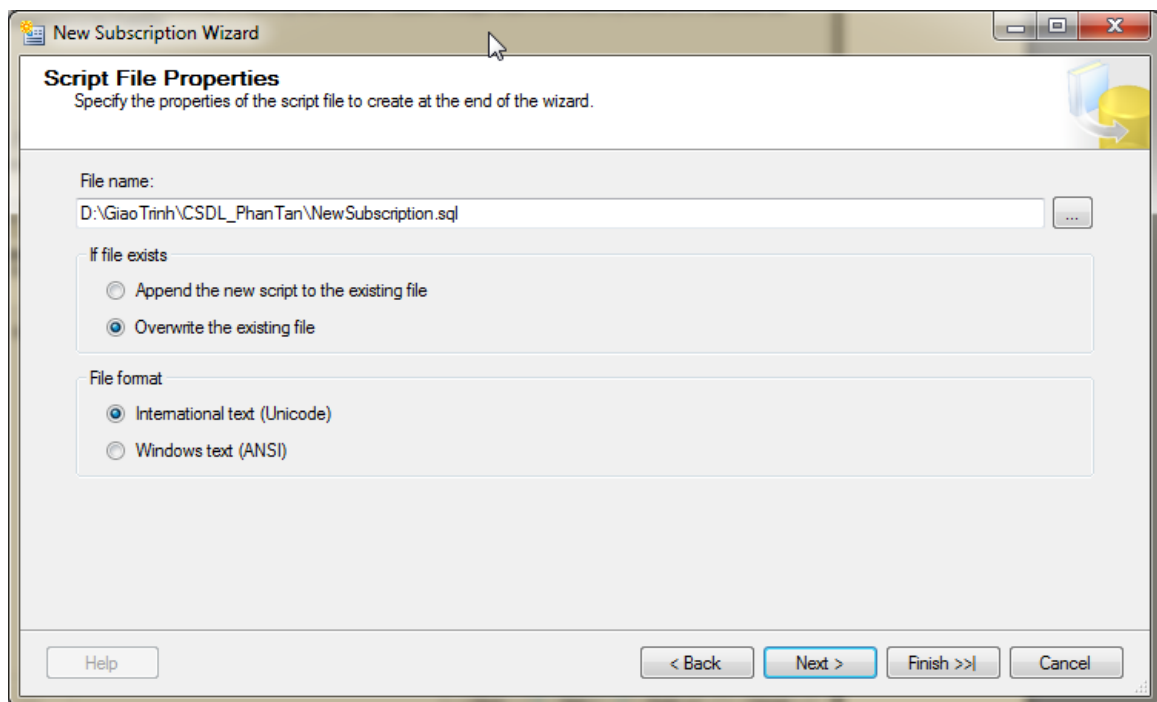
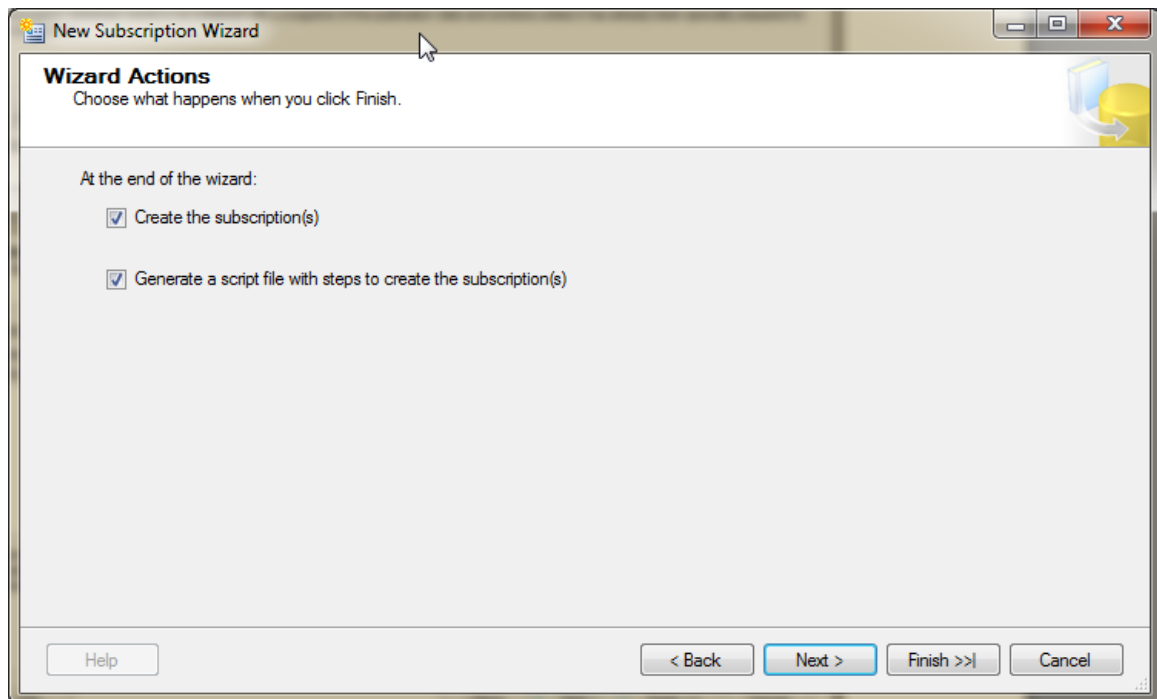
Subscription Type
Specify the type of each subscription and assign a priority for conflict resolution.

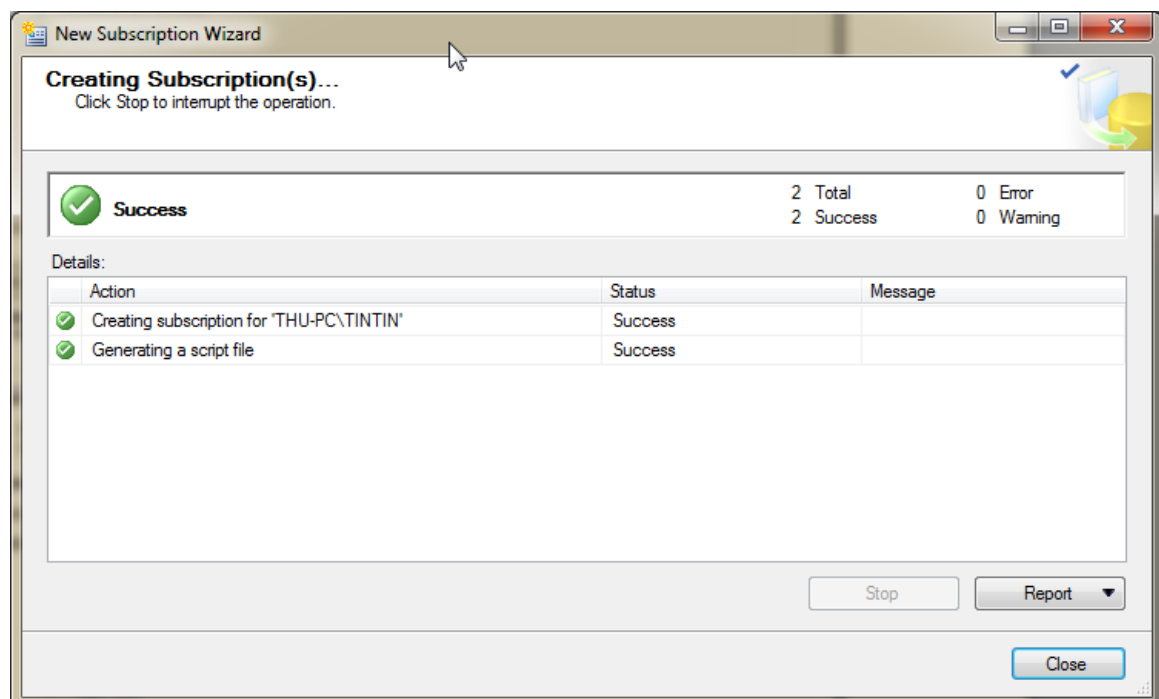
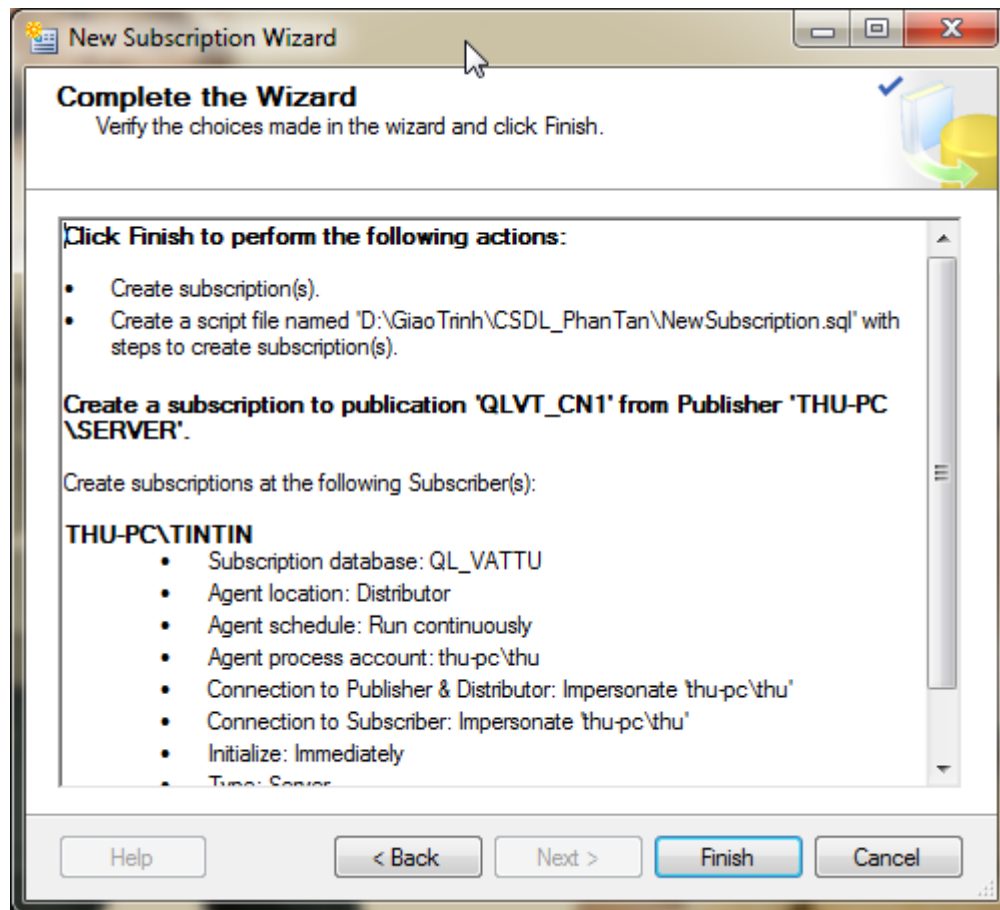
Subscription properties:

Subscriber	Subscription Type	Priority for Conflict Resolution
THU-PC\TINTIN	Server	75.00

A server subscription can republish the data to, and be a synchronization partner with, other Subscribers. It has its own priority, a number between 0 (lowest priority) and 99.99 (highest priority), for resolving data conflicts. In addition, changes made to download-only articles at the Subscriber are replicated back to the Publisher.

Help < Back Next > Finish >>| Cancel





1. Tạo Linked Server:

Sử dụng lệnh

```
sp_addlinkedserver [ @server = ] 'server'  
    [ , [ @srvproduct = ] 'product_name' ]  
    [ , [ @provider = ] 'provider_name' ]  
    [ , [ @datasrc = ] 'data_source' ]  
    [ , [ @location = ] 'location' ]  
    [ , [ @provstr = ] 'provider_string' ]  
    [ , [ @catalog = ] 'catalog' ]
```

Arguments

[@server =] 'server'

Is the local name of the linked server to create. *server* is **sysname**, with no default.

With multiple instances of SQL Server, *server* may be *servername\instancename*. The linked server then may be referenced as the data source for

```
SELECT *FROM      [servername\instancename.]pubs.dbo.authors.
```

If *data_source* is not specified, *server* is the actual name of the instance.

[@srvproduct =] 'product_name'

Is the product name of the OLE DB data source to add as a linked server.

product_name is **nvarchar(128)**, with a default of NULL. If **SQL Server**, *provider_name*, *data_source*, *location*, *provider_string*, and *catalog* do not need to be specified.

[@provider =] 'provider_name'

Is the unique programmatic identifier (PROGID) of the OLE DB provider corresponding to this data source. *provider_name* must be unique for the specified OLE DB provider installed on the current computer. *provider_name* is **nvarchar(128)**, with a default of NULL. The OLE DB provider is expected to be registered with the given PROGID in the registry.

[@datasrc =] 'data_source'

Is the name of the data source as interpreted by the OLE DB provider. *data_source* is **nvarchar(4000)**, with a default of NULL. *data_source* is passed as the DBPROP_INIT_DATASOURCE property to initialize the OLE DB provider.

When the linked server is created against the SQL Server OLE DB provider, *data_source* can be specified in the form of *servername\instancename*, which can be used to connect to a specific instance of SQL Server running on the specified computer. *servername* is the name of the computer on which SQL Server is running, and *instancename* is the name of the specific SQL Server instance to which the user will be connected.

[@location =] 'location'

TS.Phan Thị Hà-HVCNBCVT: Tài liệu nháp, không được public

Is the location of the database as interpreted by the OLE DB provider. *location* is **nvarchar(4000)**, with a default of NULL. *location* is passed as the DBPROP_INIT_LOCATION property to initialize the OLE DB provider.

[**@provstr** =] '*provider_string*'

Is the OLE DB provider-specific connection string that identifies a unique data source. *provider_string* is **nvarchar(4000)**, with a default of NULL. *provstr* is passed as the DBPROP_INIT_PROVIDERSTRING property to initialize the OLE DB provider.

When the linked server is created against the SQL Server OLE DB provider, the instance can be specified using the SERVER keyword as SERVER=*servername**instancename* to specify a specific instance of SQL Server. *servername* is the name of the computer on which SQL Server is running, and *instancename* is the name of the specific SQL Server instance to which the user will be connected.

[**@catalog** =] '*catalog*'

Is the catalog to be used when making a connection to the OLE DB provider. *catalog* is **sysname**, with a default of NULL. *catalog* is passed as the DBPROP_INIT_CATALOG property to initialize the OLE DB provider.

Return Code Values

0 (success) or 1 (failure)

Ví dụ: Giả sử ta đã tạo 1 login tên 'kythu', pass '123' ở THU-PC\SERVER2 với quyền db_datareader trên cơ sở dữ liệu QL_VATTU; Trên Server THU-PC\TINTIN, ta sẽ tạo 1 tạo linked server 'TOSERVER2' đến THU-PC\SERVER2 để 1 user bất kỳ trên phân mảnh thuộc Server THU-PC\TINTIN có thể truy cập đến cơ sở dữ liệu QL_VATTU.

USE master

GO

-- To use named parameters:

use master

EXEC sp_addlinkedserver

@server='TOSERVER2', @srvproduct='',

@provider='SQLOLEDB',

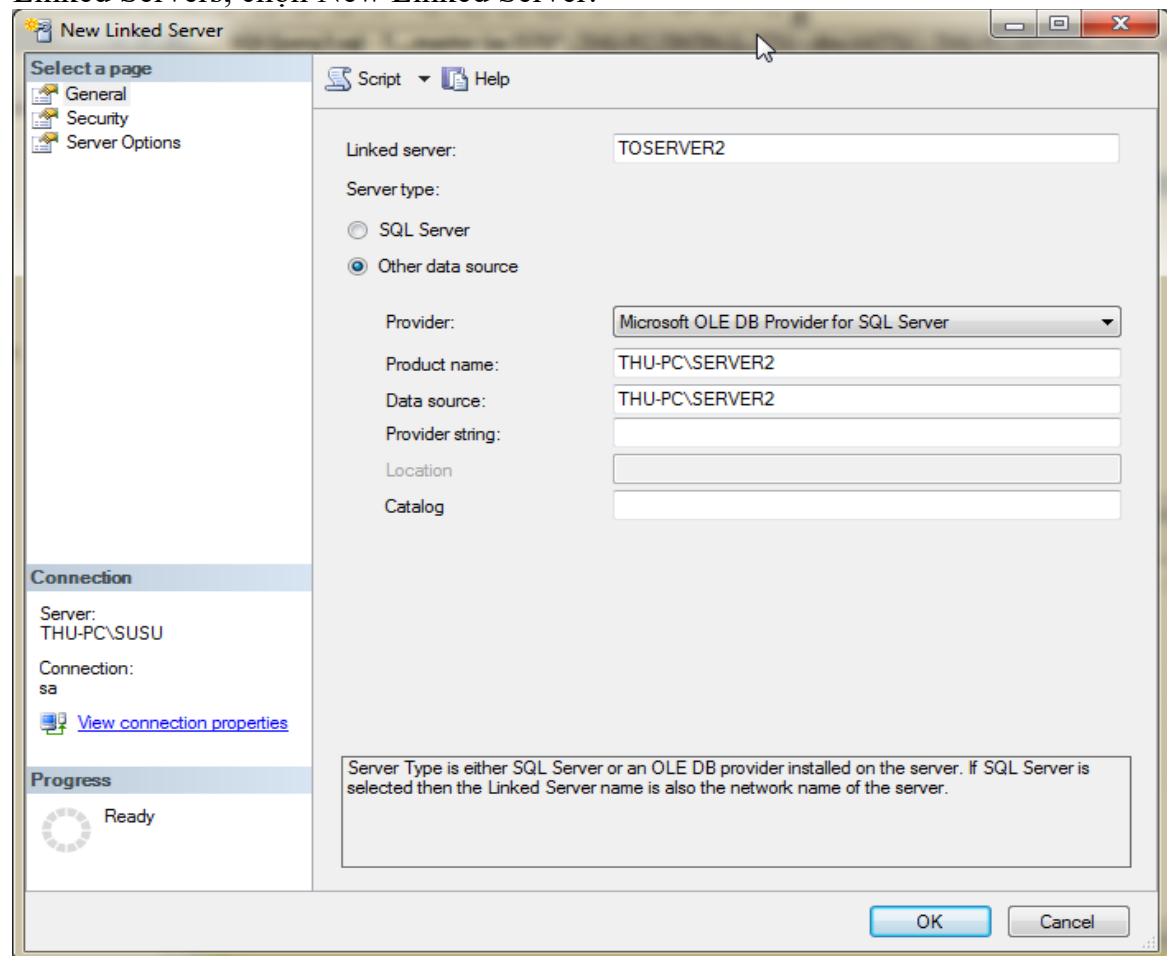
@provstr = 'DRIVER={SQL Server};SERVER=THU-PC\SERVER2;UID=kythu;PWD=123;'

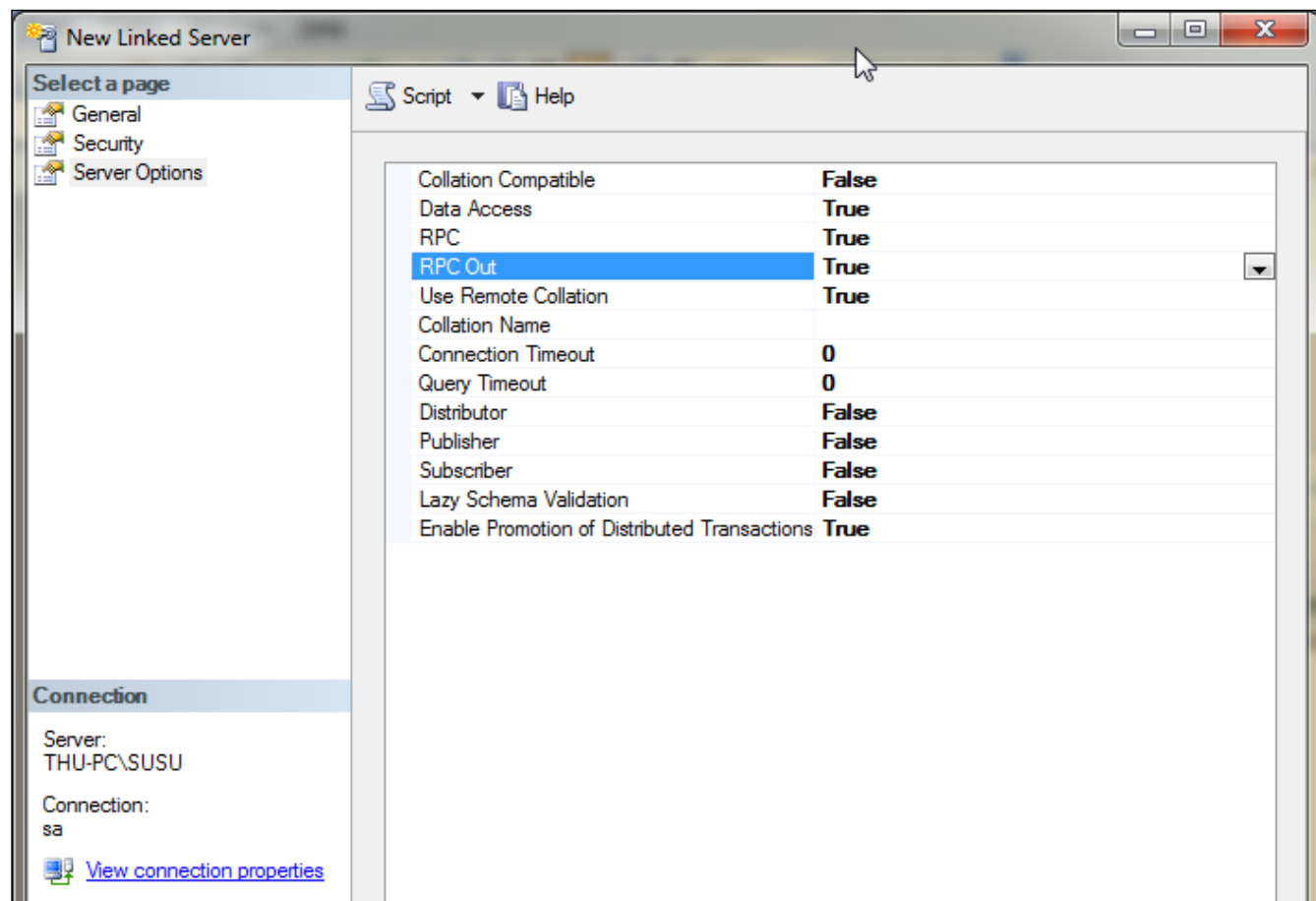
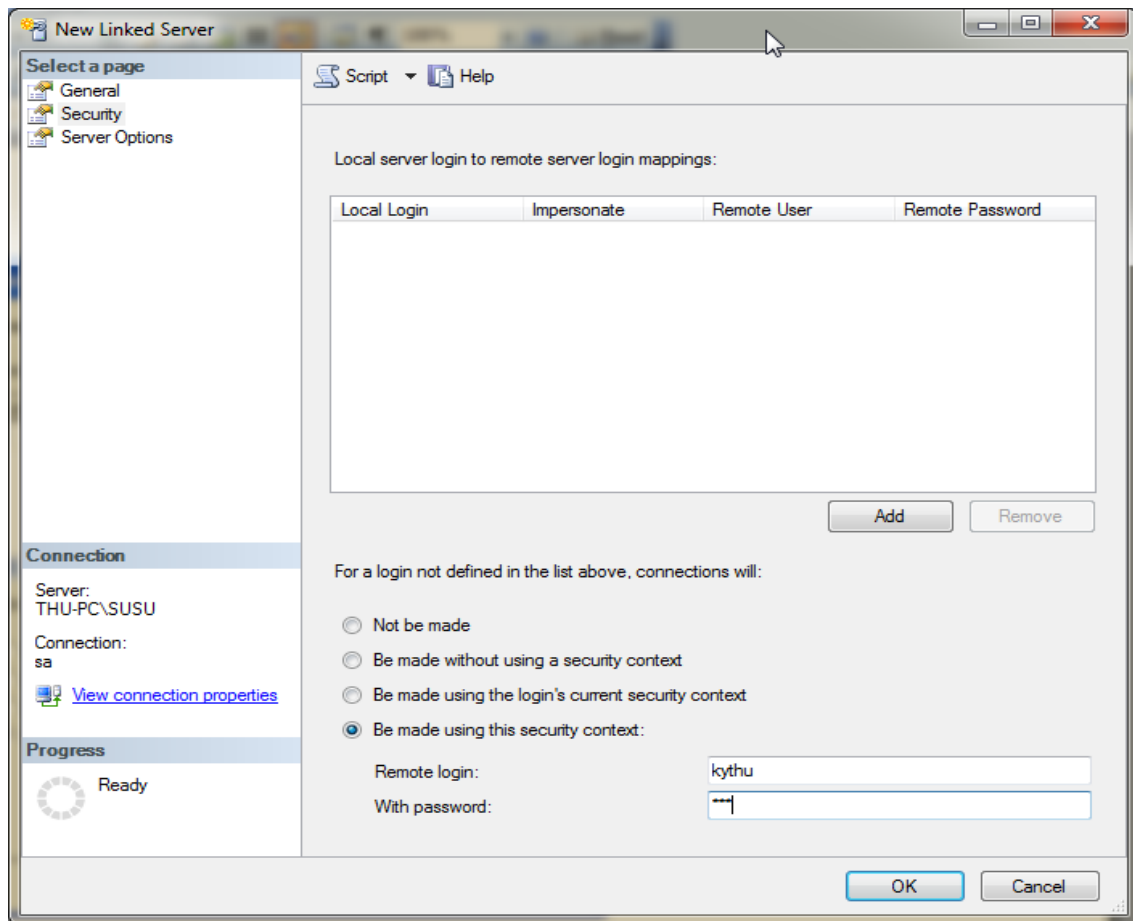
GO

Sử dụng Tool

Trong trường hợp ta muốn tạo linked server qua giao diện (tool) thì :

- Connect vào Server THU-PC\TINTIN , right click Server Object, right click Linked Servers, chọn New Linked Server:





Sau khi tạo xong Linked Server , ta kiểm thử bằng cách đứng ở Server THU-PC\TINTIN ,

```
SELECT * FROM TOSERVER2.QL_VATTU.DBO.NHANVIEN
```

1. a/ Liệt kê các sinh viên thi chõa ãi lần 1 môn 'LTVB'
Liệt kê các ãn ãi hõng do nhõn viõn cõ mã =2 nhõn.

```
select orderid, C.CustomerID, CompanyName  
from Home.Northwind_Customer_Other.dbo.Orders O,  
Home.Northwind_Customer_Other.dbo.Customers C  
where O.CustomerID= C.CustomerID and O.EmployeeID=2
```

Union

```
select orderid, C1.CustomerID, CompanyName  
from duonglan1.Northwind_Customer_UK.dbo.Orders O1,  
duonglan1.Northwind_Customer_UK.dbo.Customers C1  
where O1.CustomerID= C1.CustomerID and O1.EmployeeID=2
```

b. Liệt kê dssv cõ lờp 'TH01'

```
if exists ( select * from Home.Northwind_Customer_Other.dbo.Orders  
where OrderID=10249)
```

```
select * from Home.Northwind_Customer_Other.dbo.Orders O  
where OrderID=10249
```

else

```
if exists ( select * from Duonglan1.Northwind_Customer_UK.dbo.Orders  
where OrderID=10249)
```

```
select * from Duonglan1.Northwind_Customer_UK.dbo.Orders  
where OrderID=10249
```

```
else raiserror ('Khõng cõ ãn ãt hõng bõn chõn',16,10)
```

e/ Thoáng kê số số cõ sinh viõn toã ãn trõõng:

```
select country, count(*) from home.northwind_customer_other.dbo.customers  
group by country
```

union

```
select country, count(*) from  
duonglan1.northwind_customer_uk.dbo.customers  
group by country
```

III. Ví dụ cụ thể cho 1 dự án

Phần 1.Kịch bản

.....

Phần 2. Phân tích và thiết kế

1. Các chức năng chính của hệ thống trong dự án

Các chức năng nhiệm vụ trong hệ thống Quản lý hệ thống bán vé máy bay cho hãng hàng không:

- Quản lý thông tin hành khách (1)
- Quản lý thông tin vé(2)
- Quản lý máy bay(3)
- Quản lý chuyến bay(4)
- Quản lý nhân viên(5)
- Thống kê(6)

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

2.1. Tại đại lý tổng

- Quản lý tổng thể các thông tin về Sân bay, Máy bay, Chuyến bay, Nhân viên, Hành khách, Vé
- Có quyền thêm thông tin của các bảng Sân bay, Máy bay, Chuyến bay rồi tự động cập nhật về chso các đại lý

2.2. Đại lý HN

- Quản lý các thông tin tại ĐL của mình gồm Nhân viên, Hành khách, Vé
 - Thông tin Nhân viên sẽ được cập nhật lên Server định kì theo tháng
 - Thông tin về Vé và Hành khách sẽ được cập nhật lên đại lý tổng định kì vào 23h30 các ngày
- Tại Đlý HN có chỉ có quyền xem thông tin về Sân bay, Máy bay, Chuyến bay nhưng không có quyền sửa các thông tin này

2.3. Đại lý Tp HCM

- Quản lý các thông tin tại site của mình gồm Nhân viên, Hành khách, Vé
 - Thông tin Nhân viên sẽ được cập nhật lên ĐL tổng kì theo tháng

▪ Thông tin về Vé và Hành khách sẽ được cập nhật lên DL tổng định kì vào 23h30 các ngày

- Tại DL Tp HCM có chỉ có quyền xem thông tin về Sân bay, Máy bay, Chuyến bay nhưng không có quyền sửa các thông tin này

3. Chức năng ở máy trạm

3.2.1.1 Chức năng Quản lý hành khách

Giúp các nhân viên có thực hiện các thao tác như thêm, sửa, xóa các thông tin của hành khách. Các thông tin sẽ được lưu trữ trong cơ sở dữ liệu

Các thông tin quản lý bao gồm:

- Mã hành khách được lấy số thứ tự tiếp theo
- Nhập tên hành khách
- Nhập địa chỉ
- Nhập số điện thoại
- Nhập số CMTND

3.2.1.3. Chức năng Quản lý nhân viên

Giúp các nhân viên có thực hiện các thao tác thêm, sửa, xóa thông tin nhân viên. Các thông tin sẽ được lưu trữ trong cơ sở dữ liệu

Các thông tin của nhân viên bao gồm:

- Mã nhân viên được lấy trong bảng hành khách
- Tạo thêm 1 tài khoản đăng nhập cho nhân viên với tên đăng nhập và mật khẩu
- Nhập chức vụ của nhân viên

3.2.1.3. Chức năng Quản lý vé

Giúp các người bán vé thực hiện các thao tác bán vé online hoặc off line, cập nhật thông tin đặt vé online

Thông tin đặt vé của khách hàng: Mã KH, chuyến bay, ưu đãi, giá vé

3.2.1.4. Chức năng, xem và tìm kiếm thông tin liên quan đến sân bay, chuyến bay, máy bay

3.2.1.5. Chức năng thống kê: Hành khách, Tiền, Vé bán theo mỗi chuyến bay,.....từ ngày=> ngày

4. Các Chức Năng ở Máy Chủ

4.1 Có toàn bộ chức năng của các máy trạm

4.1.2 Chức năng Quản lý thông tin chuyến bay

Giúp các người quản lý thực hiện các thao tác như

TS.Phan Thị Hà-HVCNBCVT: Tài liệu nháp, không được public

thêm, sửa, xóa các thông tin chi tiết của chuyến bay. Các thông tin sẽ được lưu trữ trong cơ sở dữ liệu. Dữ liệu của Chuyến bay từ Hà nội sẽ được chuyển đến Máy trạm tại HN quản lý thông tin các chuyến bay từ Hà Nội, tương tự đối với HCM.

Các thông tin quản lý bao gồm:

- Mã chuyến bay
- Thông tin máy bay
- Đường bay
- Thông tin của các sân bay xuất phát, kết thúc
- Time xuất phát, kết thúc

4.1.3 Chức Năng Báo Cáo Thống Kê

Giúp cho người quản lý biết được những thông tin cần thiết của công việc quản lý bán vé máy bay như:

- Báo cáo doanh thu trong tháng
- Báo cáo số vé bán ra
- Báo cáo tình trạng số lượng máy bay
- Báo cáo về nhân viên
- Báo cáo tình trạng các chuyến bay

.....

4.1.4 Chức năng Phân quyền

Giúp cho người quản lý thực hiện phân quyền cho các tài khoản.

Ngoài ra còn có các chức năng khác như: chức năng tìm kiếm, chức năng In Các hóa đơn , danh sách, báo cáo, đổi mật khẩu... Để quản lý việc bán vé được tốt hơn

5. Phân quyền cho các nhóm đối tượng thực hiện dự án

1. Nhân viên

- Có quyền xem thông tin của các hành khách và các vé được đặt thuộc ĐL của mình
- Có thể thêm và sửa lại thông tin của hành khách cũng như thông tin về vé khi có sự yêu cầu của hành khách

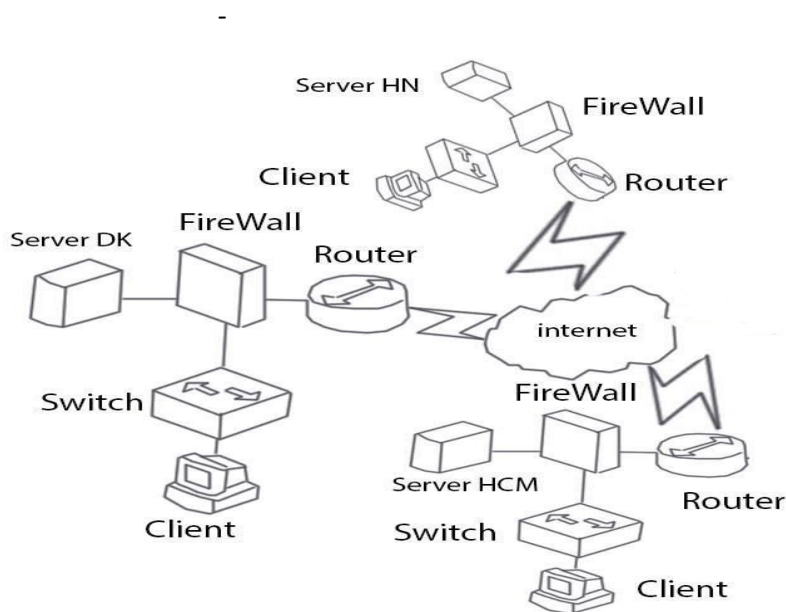
2. Hành khách

- Có quyền đăng kí tài khoản của mình tại một trong các đại lý cơ sở thông qua web site
- Xem và chỉnh sửa thông tin các nhân
- Có thể thực hiện đặt vé online thông qua website vào đại lý mà khách hàng đã đăng kí

3. Người Quản lí

- Quản lí có tất cả các quyền của nhân viên tại mỗi đại lý trung tâm
- Ngoài ra quản lí còn có quyền tạo báo cáo và thống kê

6. Thiết kế hệ thống mạng tổng quan



- **Đại lý tổngr:** Đặt máy Server DK thực hiện các chức năng và lưu CSDL của toàn hệ thống
- **Đại lý Hà Nội:** Đặt máy Server trạm HN và các client tương ứng với các đại lý nhỏ, thực hiện các chức năng và lưu CSDL nằm ở HN
- **DL HCM:** Đặt máy Server trạm HCM và các client tương ứng với các đại lý nhỏ thực hiện chức năng và lưu toàn bộ CSDL thuộc TPHCM

7. Thiết kế CSDLPT

7.1 Phân tích thực thể

1. Sân bay

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
1	Ma	nvarchar(100)	Not null	PK	Mã sân bay

2	Ten	nvarchar(100)	Not null		Tên sân bay
3	diachi	nvarchar(100)	Not null		Địa chỉ của sân bay
4	hang	nvarchar(100)	Not null		Hạng của sân bay

2. Ghế ngồi

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
1	Ma ghế			PK	Mã chỗ ngồi
2	Loại ghế	nvarchar(100)	Not null		
3	Giá	float	Not null		

3. Máy bay

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
1	Ma	nvarchar(100)	Not null	PK	Mã máy bay
2	Ten	nvarchar(100)	Not null		Tên máy bay
3	hang	nvarchar(100)	Not null		Hãng sản xuất
4	kichthuoc	nvarchar(100)	Not null		Kích thước
5	gheloi1	int	Not null		Số lượng ghế loại 1
6	gheloi2	int	Not null		Số lượng ghế loại 2

4. Chuyến bay

STT	Tên thuộc tính	Kiểu	Ràng	Kh	Mô tả
-----	----------------	------	------	----	-------

			buộc	óa	
1	Ma	nvarchar(100)	Not null	PK	Mã chuyến bay
2	maybay	nvarchar(100)	Not null	FK	Mã máy bay
4	diemxuatphat	nvarchar(100)	Not null		Điểm xuất phát
5	diemden	nvarchar(100)	Not null		Điểm đến
6	thoigianbatdau	Datetime	Not null		Thời gian máy bay cất cánh
7	Gía vé				
7	Loại ghế	
8	thoigianketthuc	Datetime	Not null		Thời gian máy bay hạ cánh

5. Nhân viên

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
1	Ma	nvarchar(100)	Not null	PK	Mã nhân viên
2	Chi nhánh	nvarchar(100)			
3	Ten	nvarchar(100)	Not null		Tên nhân viên
4	ngaysinh	Date	Not null		Ngày sinh
5	diachi	nvarchar(100)	Not null		Địa chỉ
6	vitri	nvarchar(100)	Not null		Vị trí
7	Sdt	nvarchar(100)	Not null		Số điện thoại
8	bacluong	int	Not null		Bậc lương

6. Khách hàng

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
-----	----------------	------	-----------	------	-------

1	Ma	nvarchar(100)	Not null	PK	Mã khách hàng
2	Ten	nvarchar(100)	Not null		Tên
3	ngaysinh	nvarchar(100)	Not null		Ngày sinh
4	diachi	nvarchar(100)	Not null		Địa chỉ
5	mahochieu	nvarchar(100)	Not null		Mã hộ chiếu
6	Sdt	nvarchar(100)	Not null		Số điện thoại
7	tendangnhap	nvarchar(100)			Tên đăng nhập
8	matkhau	nvarchar(100)			Mật khẩu

7. Vé

STT	Tên thuộc tính	Kiểu	Ràng buộc	Khóa	Mô tả
1	Ma	nvarchar(100)	Not null	PK	Mã vé
3	khachhang	nvarchar(100)	Not null	FK	Mã khách hàng
4	chuyenbay	nvarchar(100)	Not null	FK	Mã chuyến bay
5	Giave	float	Not null		Gia vé
6	Uudai	float	Not null		Ưu đãi giảm giá

7.2Lược đồ Liên kết thực thể

.....Học sinh tự vẽ

7.3Lược đồ quan hệ liên kết

.....Học sinh tự vẽ

7.4 Quan hệ giữa các bảng

..... Học sinh tự vẽ

7.5 Thiết kế phân mảnh ngang, nhân bản

- Bao gồm các bảng Sân bay, Máy bay, Chuyến bay, ghế ngồi sẽ được sử dụng chung cho các Site, nhưng chỉ được cập nhập, sửa, xóa tại máy chủ tổng
- Bao gồm các bảng Nhân viên, Hành khách, Vé sẽ được dùng riêng tại mỗi site. Site nào có quyền quản lí Nhân viên, Hành khách và Vé của site đó.

a. Thiết kế hệ thống thông tin về CSDL cho phân mảnh ngang

Máy trạm HN sẽ chỉ được lưu trữ các bản ghi dữ liệu về hành khách, nhân viên, vé được đăng ký tại site mình và vé được bán tại site mình

Các dữ liệu về chuyến bay, máy bay, sân bay, vé được lưu và thay đổi tại server và tại các trạm có thể có nhân bản

.....

b. Phân mảnh dữ liệu:

- **Sinh viên tự** Phân mảnh ngang nguyên thủy và phân mảnh ngang dẫn xuất để chia quan hệ tổng thể thành 3 mảnh đặt tại 3 vị trí như sau:

Vị trí 1: Máy chủ

Vị trí 2: Máy trạm đặt tại Hà Nội

Vị trí 3: Máy trạm đặt tại Thành phố Hồ Chí Minh

Để phân mảnh, chúng tôi chọn quan hệ tổng thể CHI_NHANH làm tiêu chí để phân mảnh, phân thành 3 mảnh đặt tại 3 vị trí để quản lý các chi nhánh ở các vùng lân cận, sau đó dựa vào các mảnh của chi nhánh sẽ dẫn xuất đến sự phân mảnh của các quan hệ tổng thể còn lại.

- **Sinh viên tự** vẽ lược đồ phân mảnh ngang nguyên thủy, phân mảnh ngang dẫn xuất

.....

.....

.....

3.5. Thiết kế định vị và vẽ sơ đồ định vị

- Thực thể	Trụ sở chính	Chi nhánh
CHI NHANH	H.RWED	H.R
HANH KHACH	H.RWED	H.RWED
VE	L.R	H.RWED
CHUYEN BAY	H.RWED	H.R
SAN BAY	H.RWED	H.R
NHANVIEN	H.R	H.RWED
MAYBAY	H.RWED	H.R
.....	.	

Trong đó:

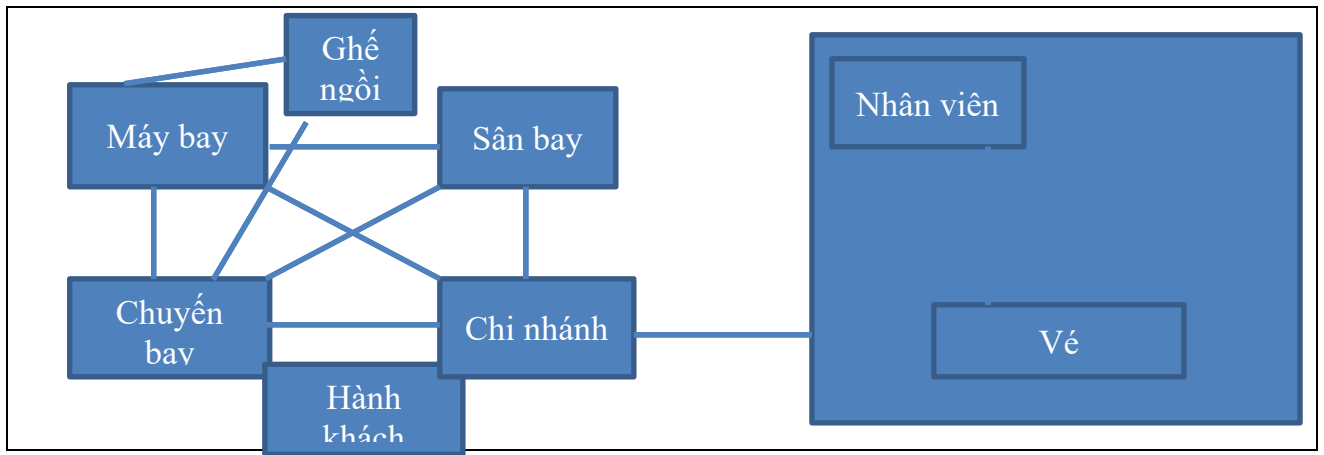
- W: tạo mới và ghi
- E: sửa
- D: xóa
- R: đọc
- H: tần suất cao
- L: tần suất thấp

- Phân tán Cơ sở dữ liệu Quản lí bán vé máy bay (QLBVMB) ra làm 3 mảnh đặt tại 3 nơi khác nhau:

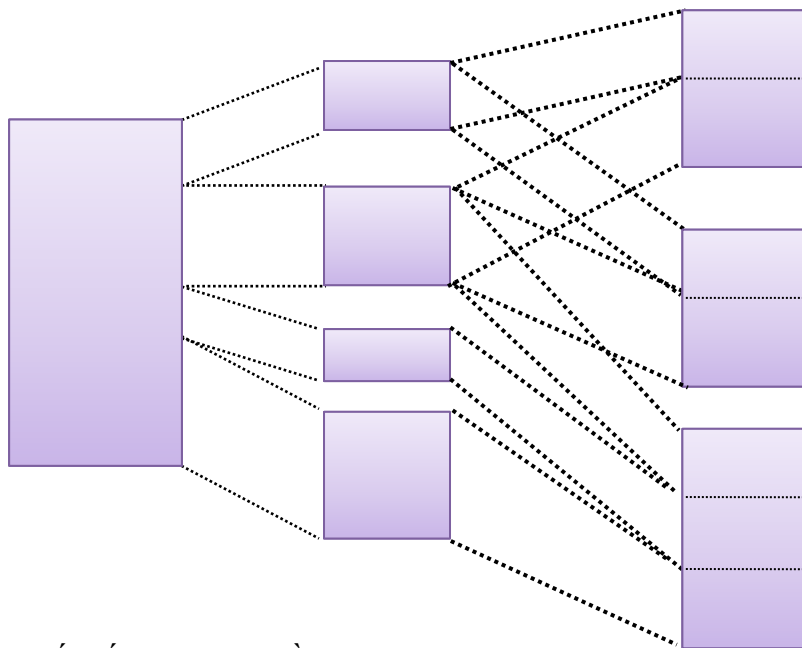
- Server chứa thông tin về Sân bay, Máy bay, Chuyến bay, ghế ngồi
- QLBVMB được đặt trên site 1: chứa thông tin bán vé máy bay cho các **chuyến bay của các hành khách đăng ký tại đây**
- QLBVMB được đặt trên site 2: chứa thông tin bán vé máy bay cho các **chuyến bay của các hành khách đăng ký tại đây**

Vẽ sơ đồ định vị.

Chung	riêng
-------	-------



-Sinh viên tự vẽ ánh xạ của các quan hệ sau khi đã được nhân bản và định vị theo lược đồ tổng quát sau



3.6Thiết kế nhân bản, đồng bộ hóa

- Thông tin tại máy chủ : Sân bay, Máy bay, Chuyến bay, ghế ngồi tại máy chủ sẽ được nhân bản tại máy trạm (đồng bộ hóa về máy trạm mỗi khi có sự thay đổi từ máy chủ);
- Nếu tại máy chủ có sự cập nhập thông tin nhân viên, khách hàng, vé thì sẽ được đồng bộ hóa xuống chi nhánh mà nó thuộc về
- Các thông tin tại máy trạm: nhân viên, khách hàng, vé đc cập nhập sẽ được đồng bộ hóa về máy chủ mỗi khi có thông tin;

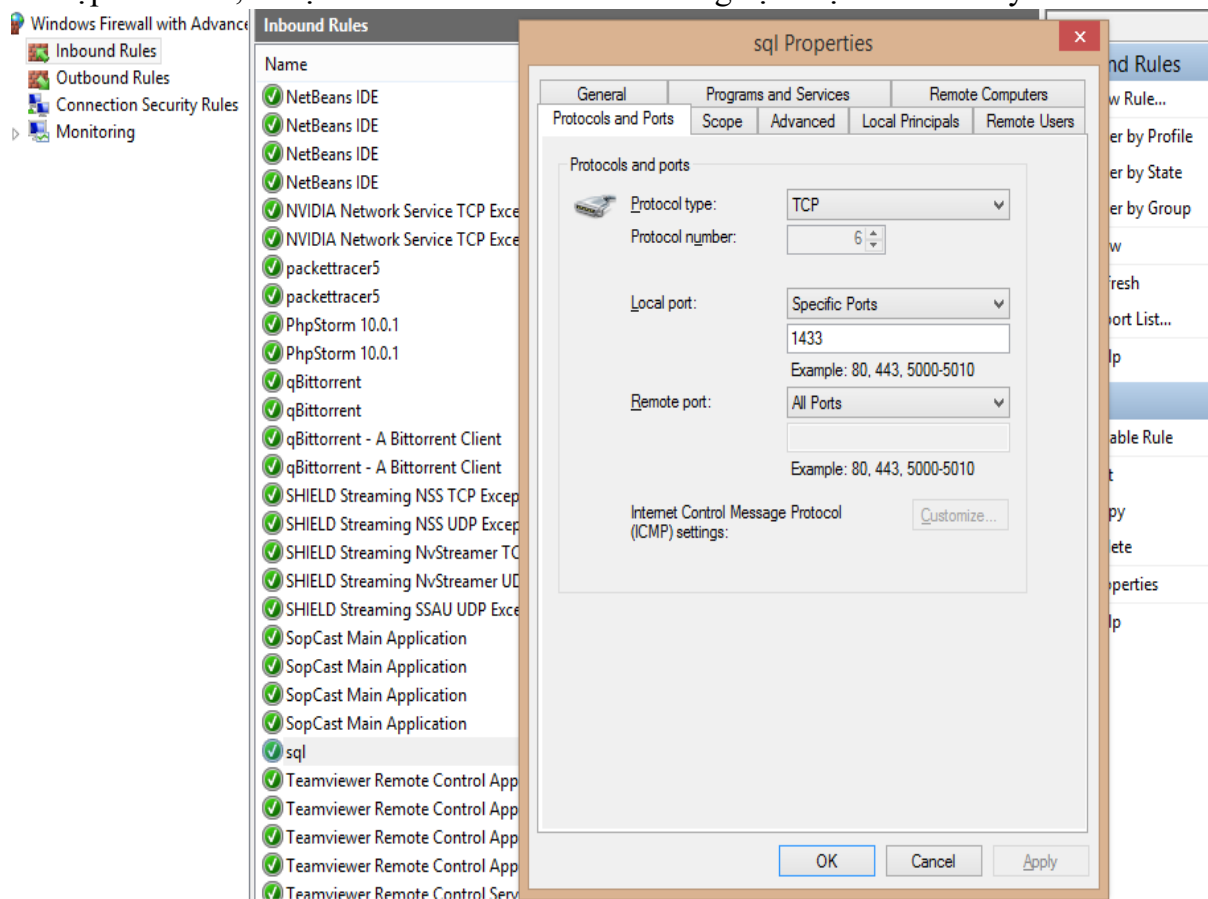
.....

3.7 Thiết kế vật lí các bảng tại các trạm,

3.8cài đặt CSDL, tạo liên kết giwux các bảng,

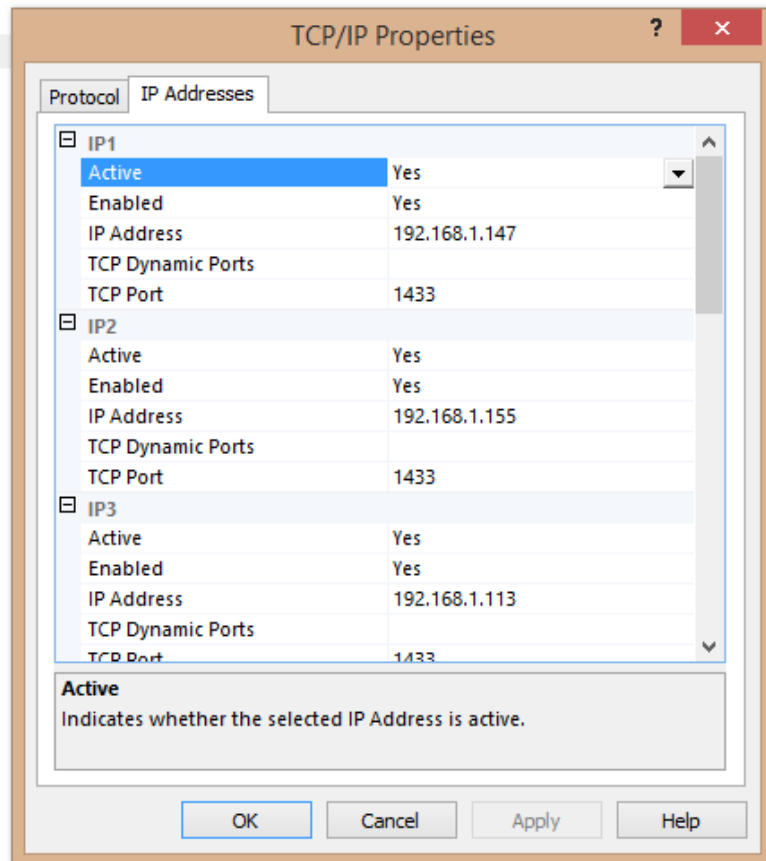
5.2. Quá trình cài đặt hệ thống với SQL Server

Ở tất cả các máy ta sẽ vô hiệu hóa Firewall để tránh bị chặn khi các máy truy cập lẫn nhau, và tạo 1 Inbound Rule để mở cổng dịch vụ cho các máy



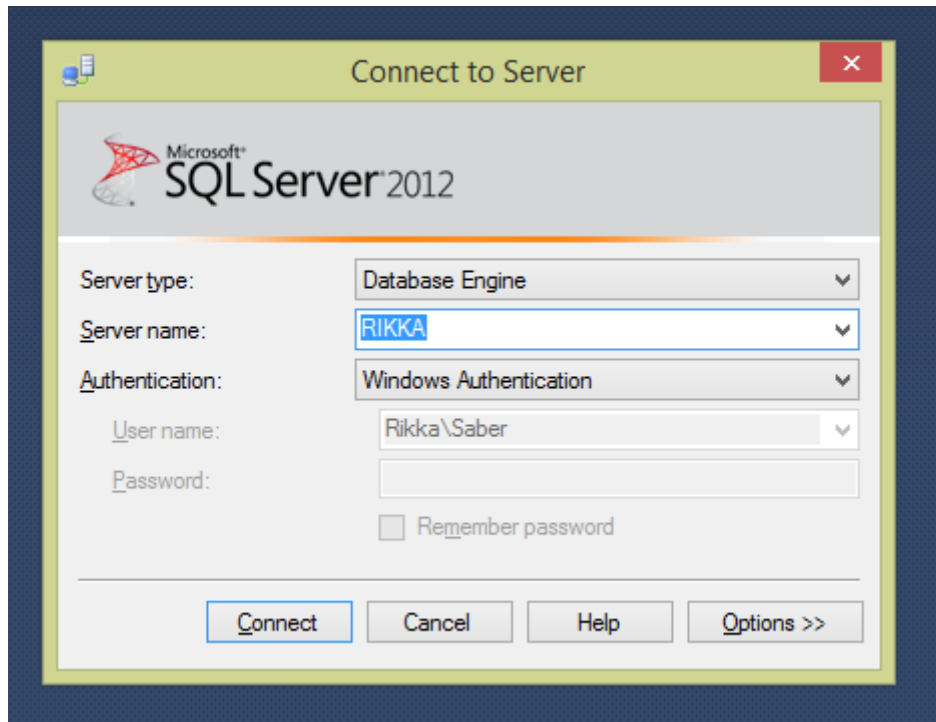
Sau đó mở Sql Server Configuration Manager để cấu hình IP cho từng máy

name	Status
Memory	Enabled
Pipes	Enabled
	Enabled

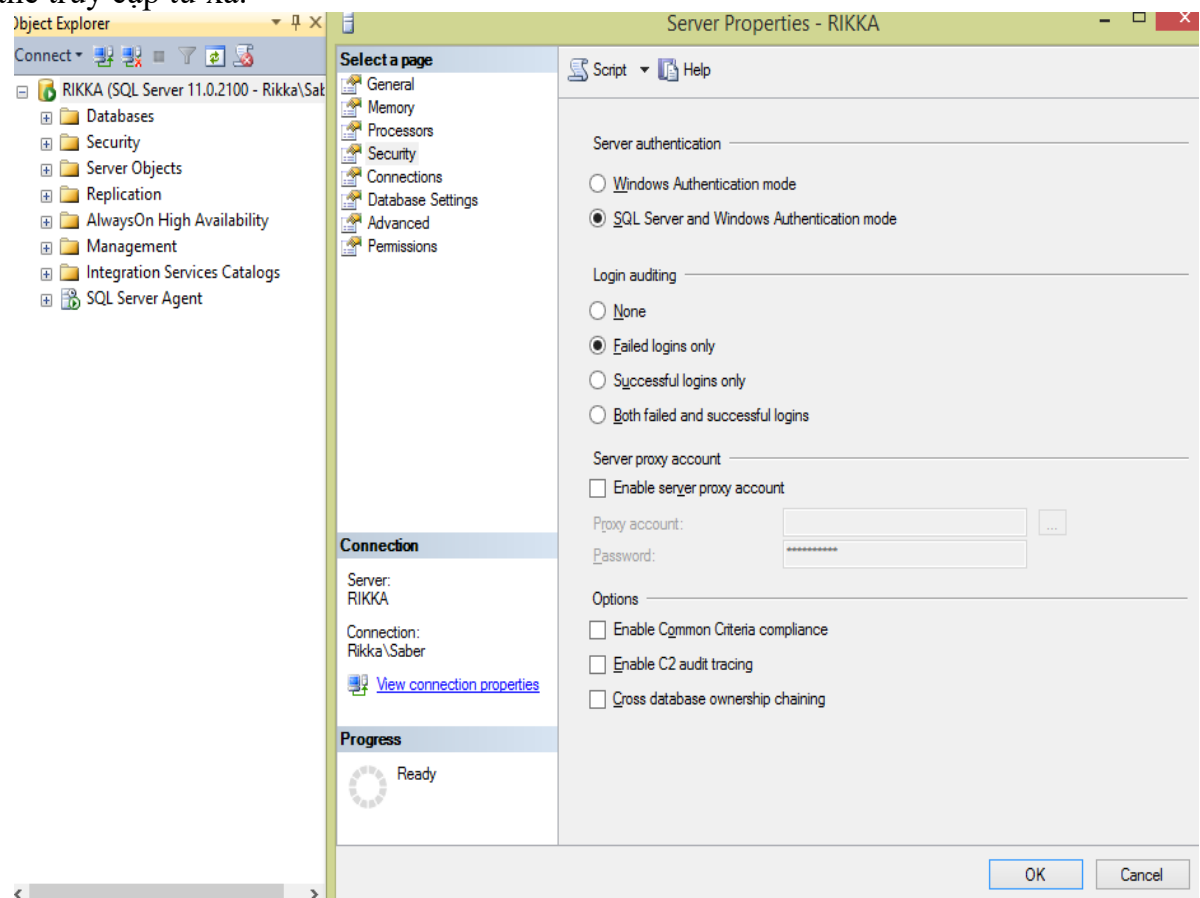


Ở đây, máy chủ sẽ có IP là 192.168.1.147 và 2 máy trạm có IP lần lượt là 192.168.1.155 và 192.168.1.113.

Sau khi đã cấu hình xong TCP/IP, ta mở SQL Server Management Studio và connect vào local database.

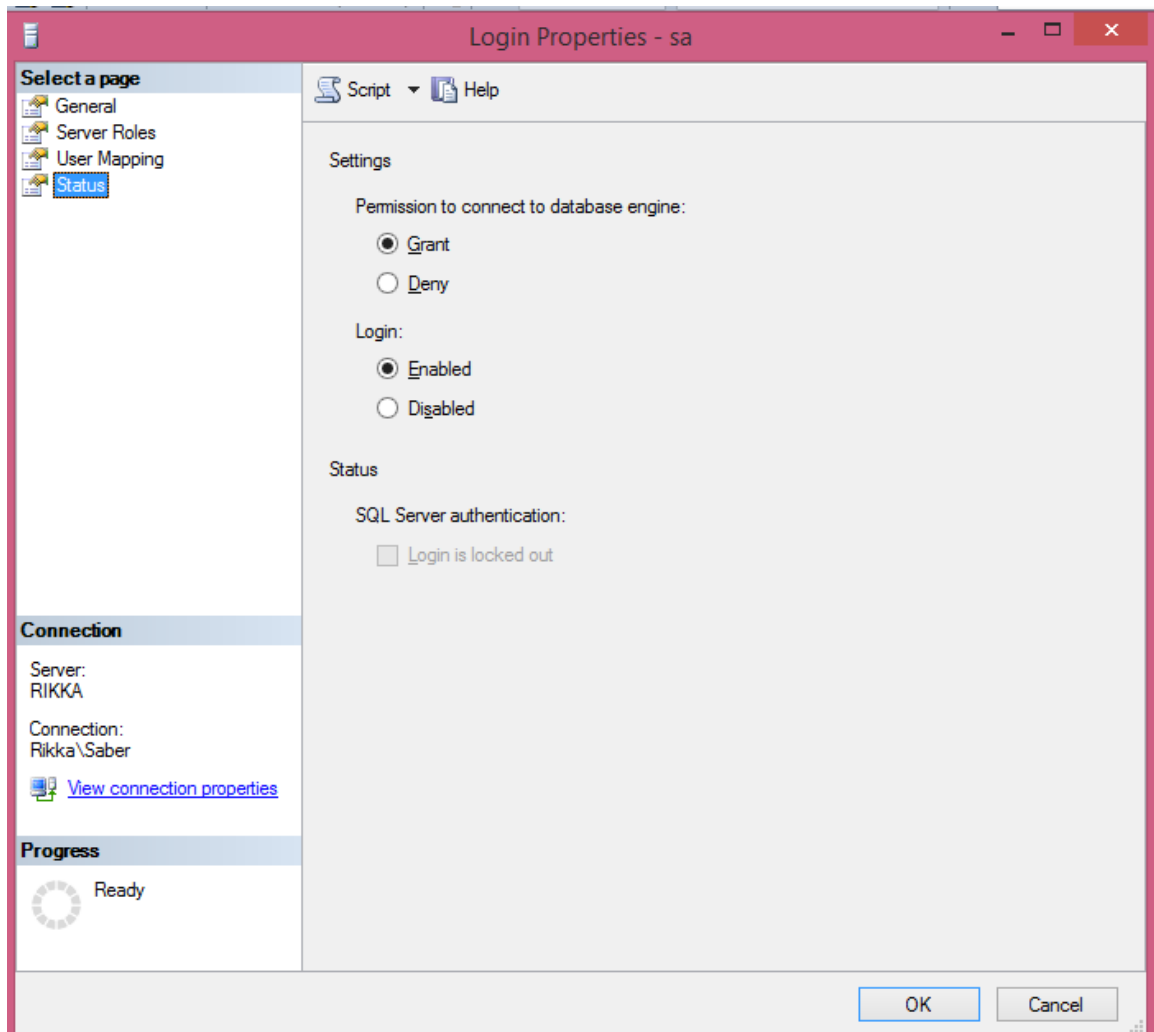


Thay đổi thuộc tính Security trong Properties của DB từ Window Authentication Mode sang Sql Server and Window Authentication Mode để có thể truy cập từ xa.



Ta sẽ dùng tài khoản sa của sql server để đăng nhập kết nối giữa các máy nên phải phân quyền cho tài khoản này.

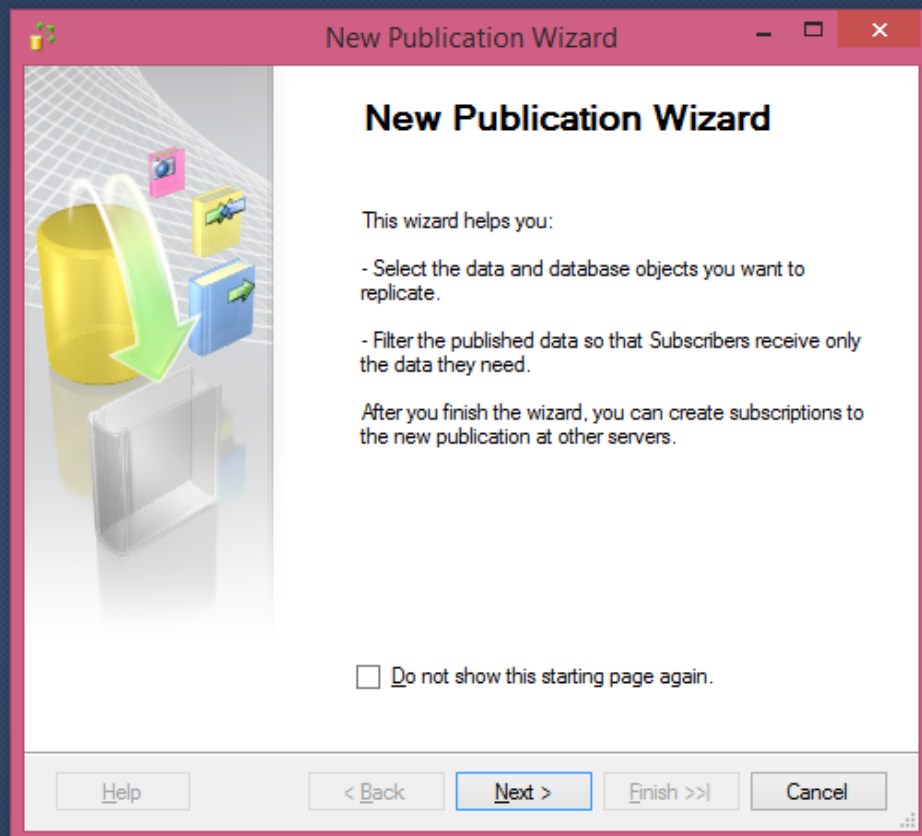
TS.Phan Thị Hà-HVCNBCVT: Tài liệu nháp, không được public

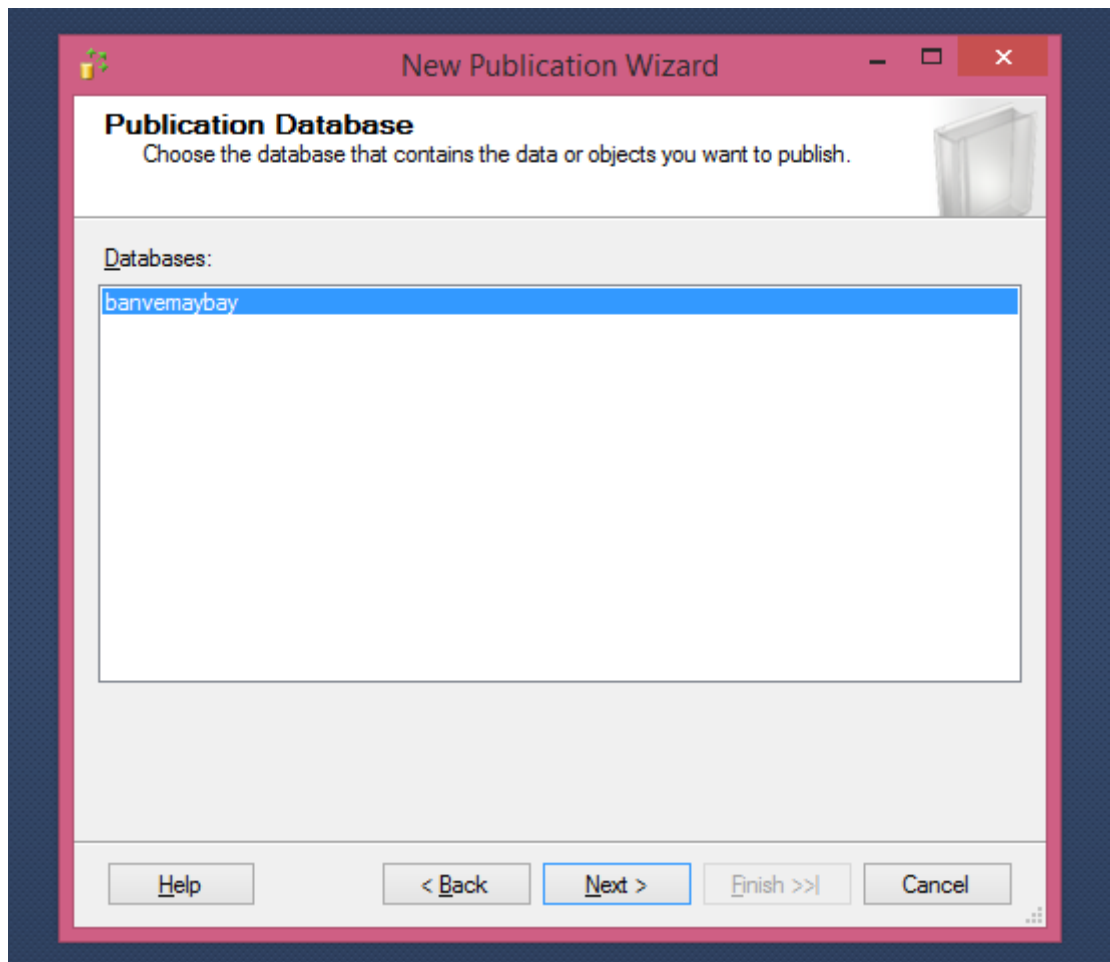


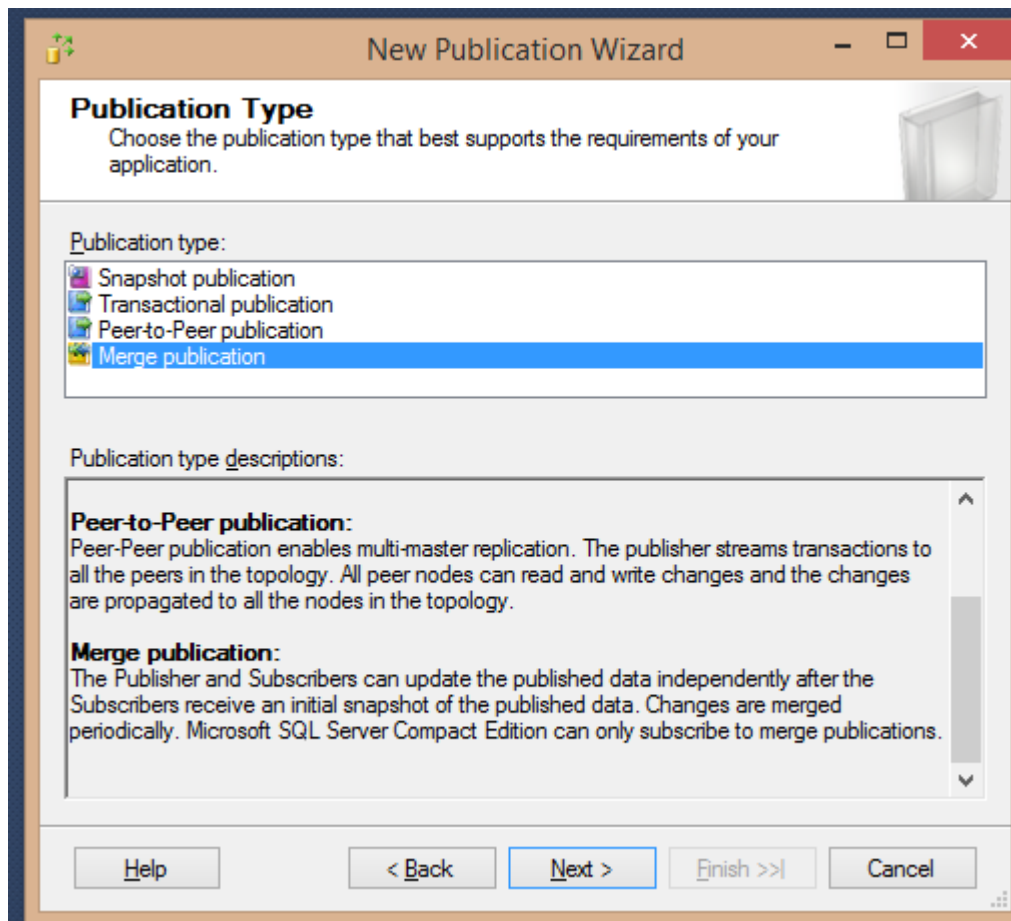
Mọi thao tác trên đều làm trên cả máy chủ và máy trạm.

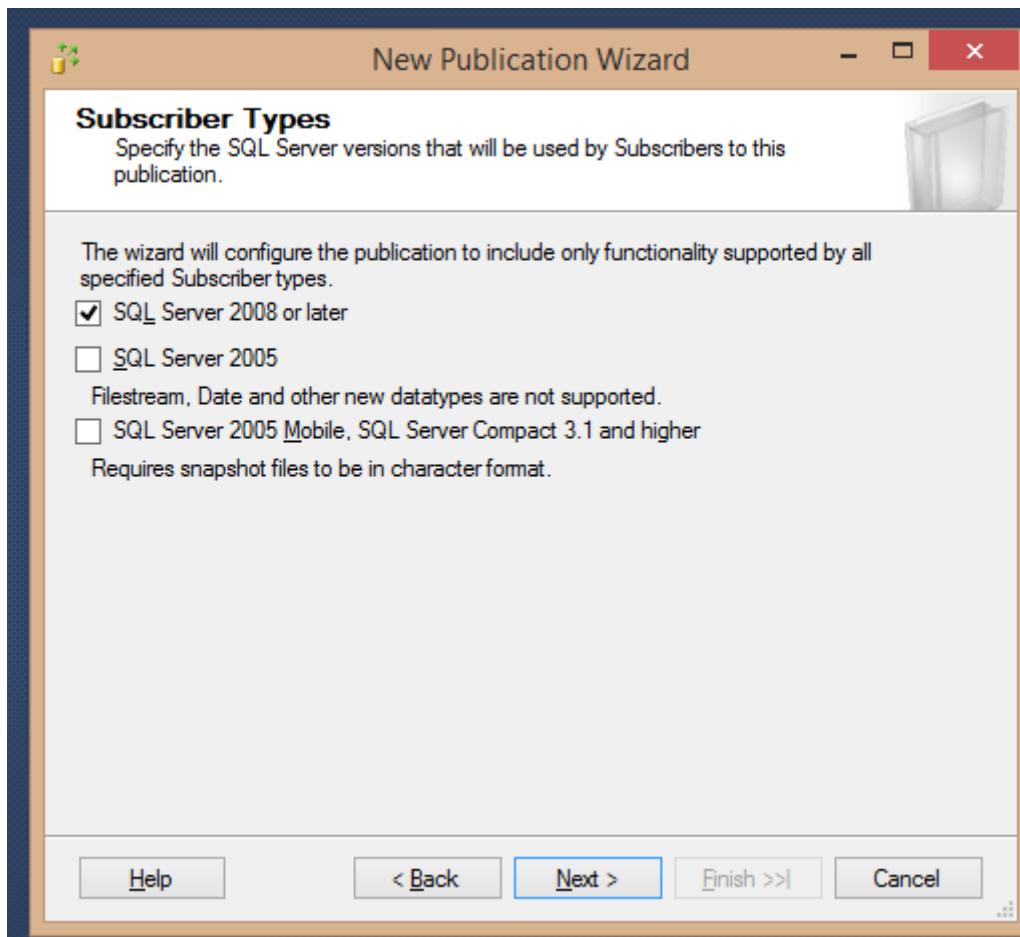
Tiếp đến tiến hành nhân bản DB ở máy chủ sang các máy trạm

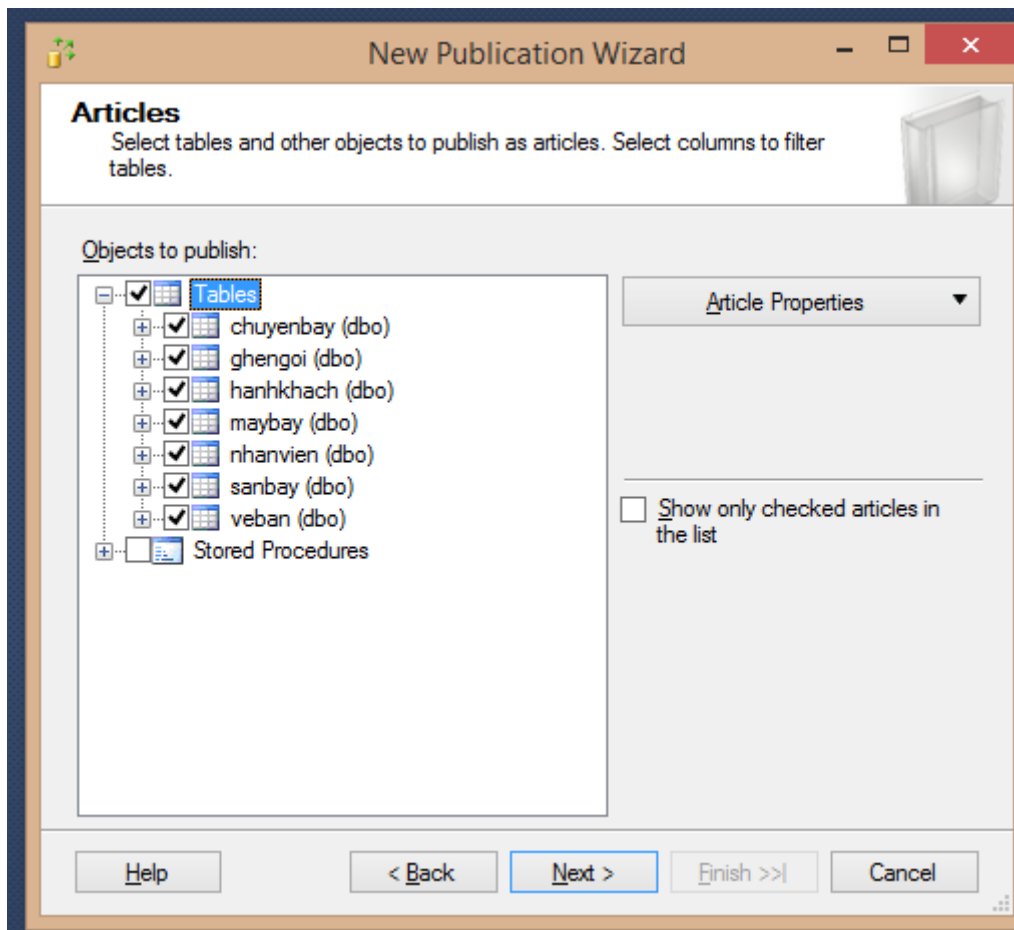
Trước hết tạo 1 Publication mới (chọn DB muốn nhân bản)

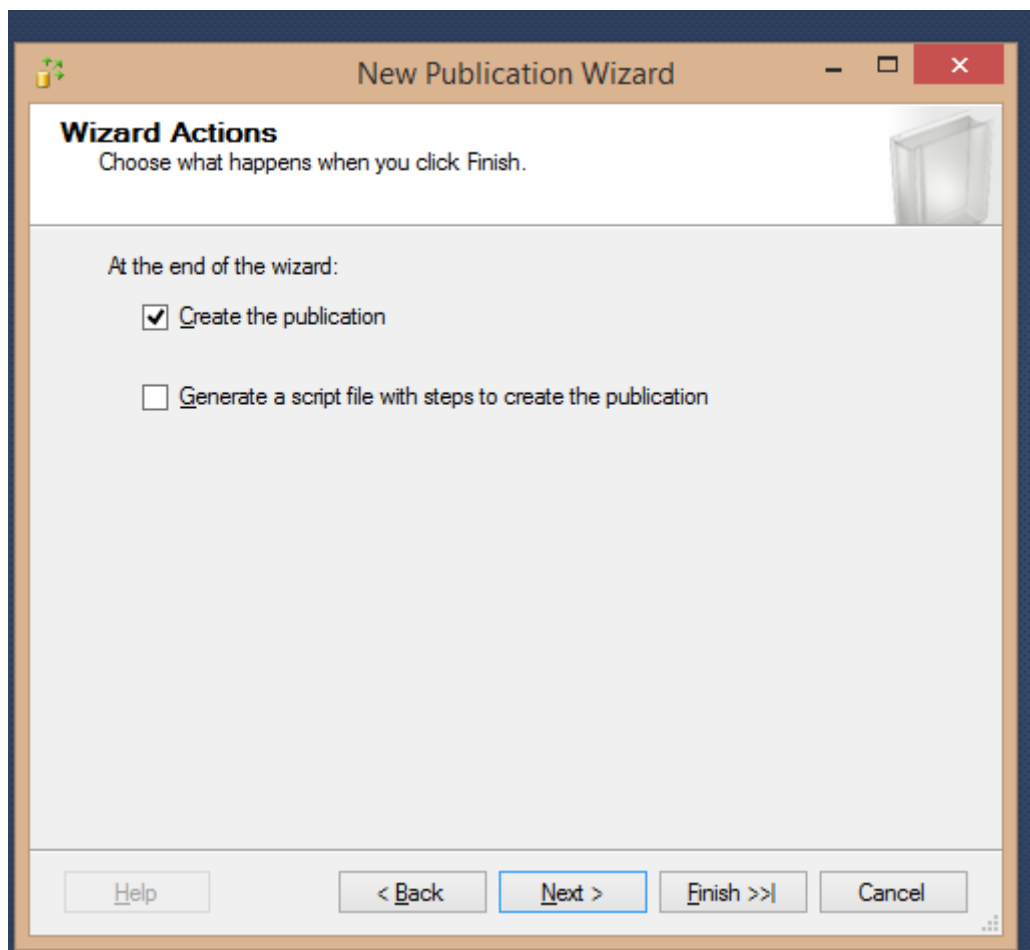


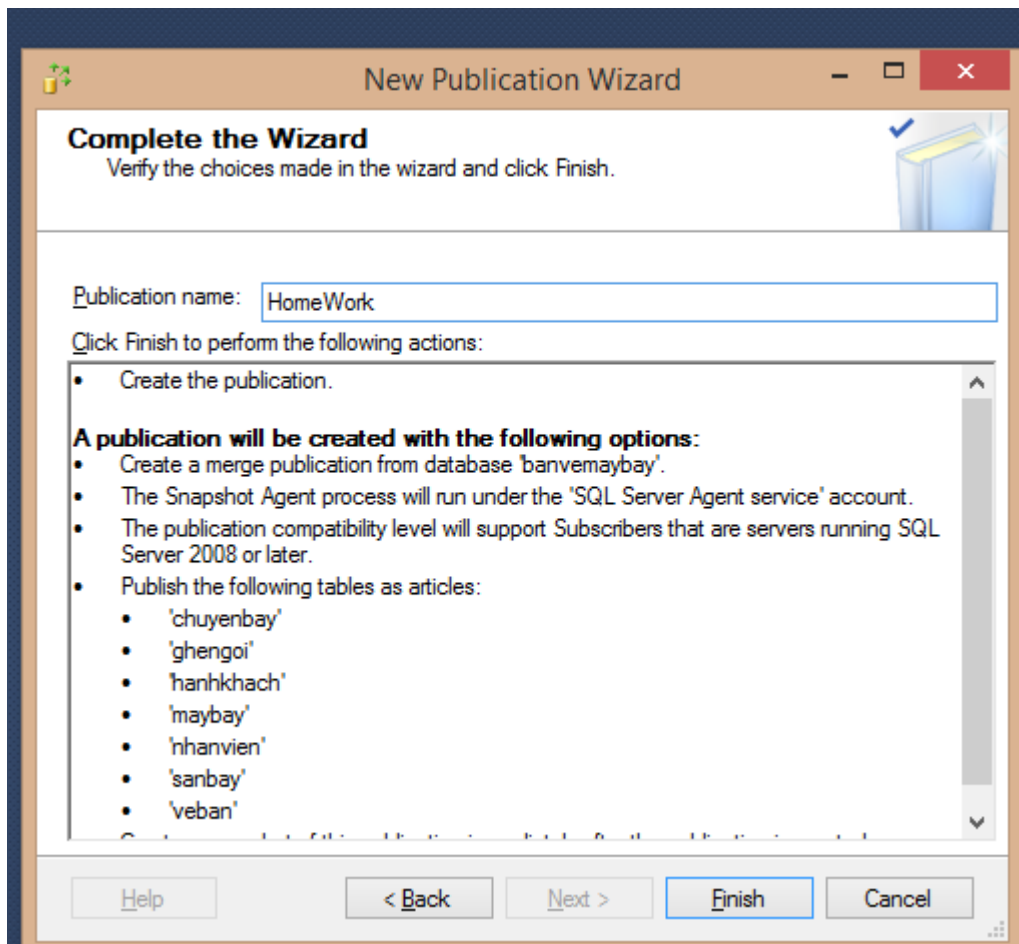




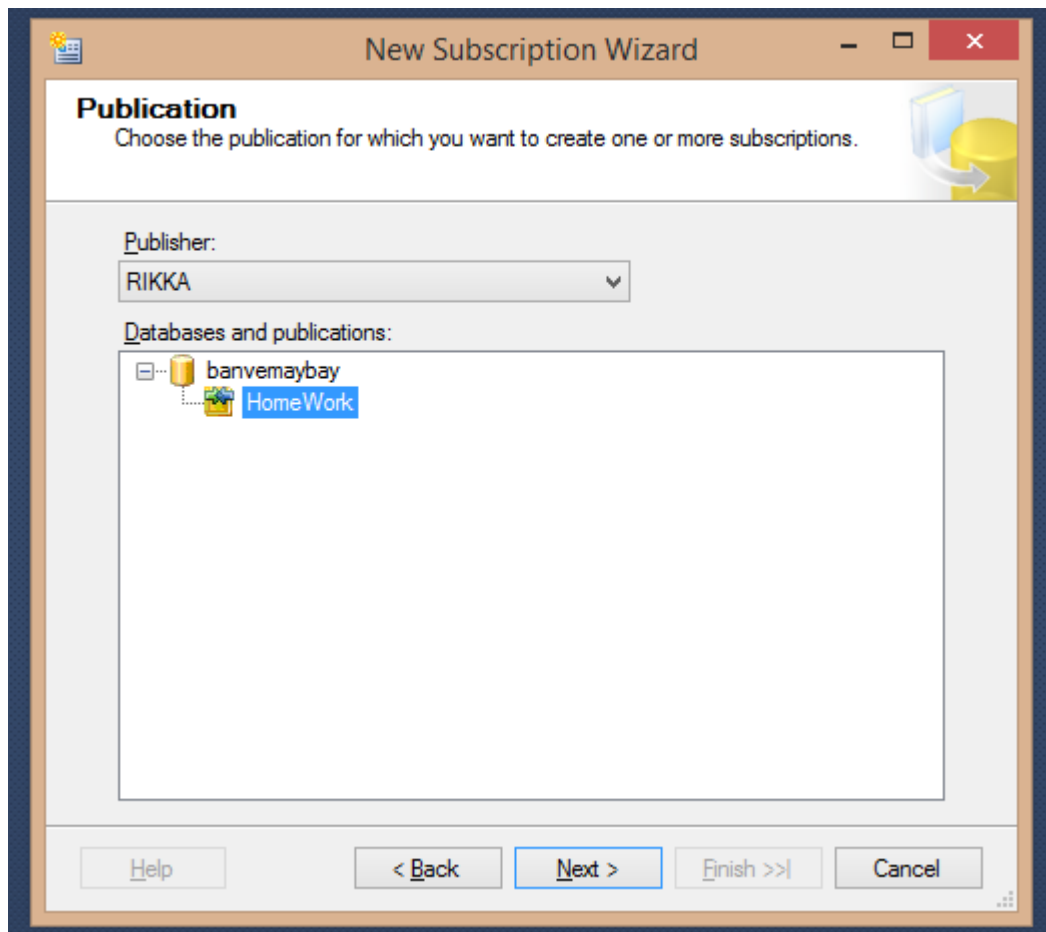




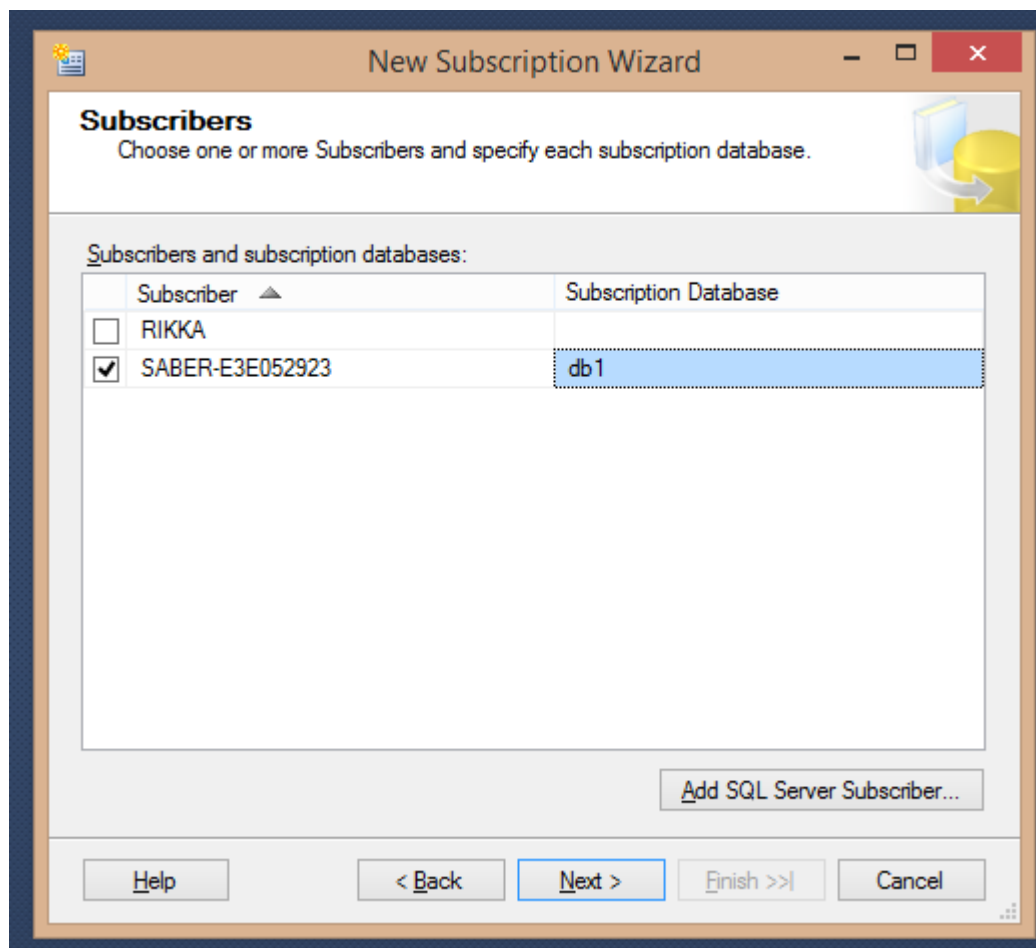


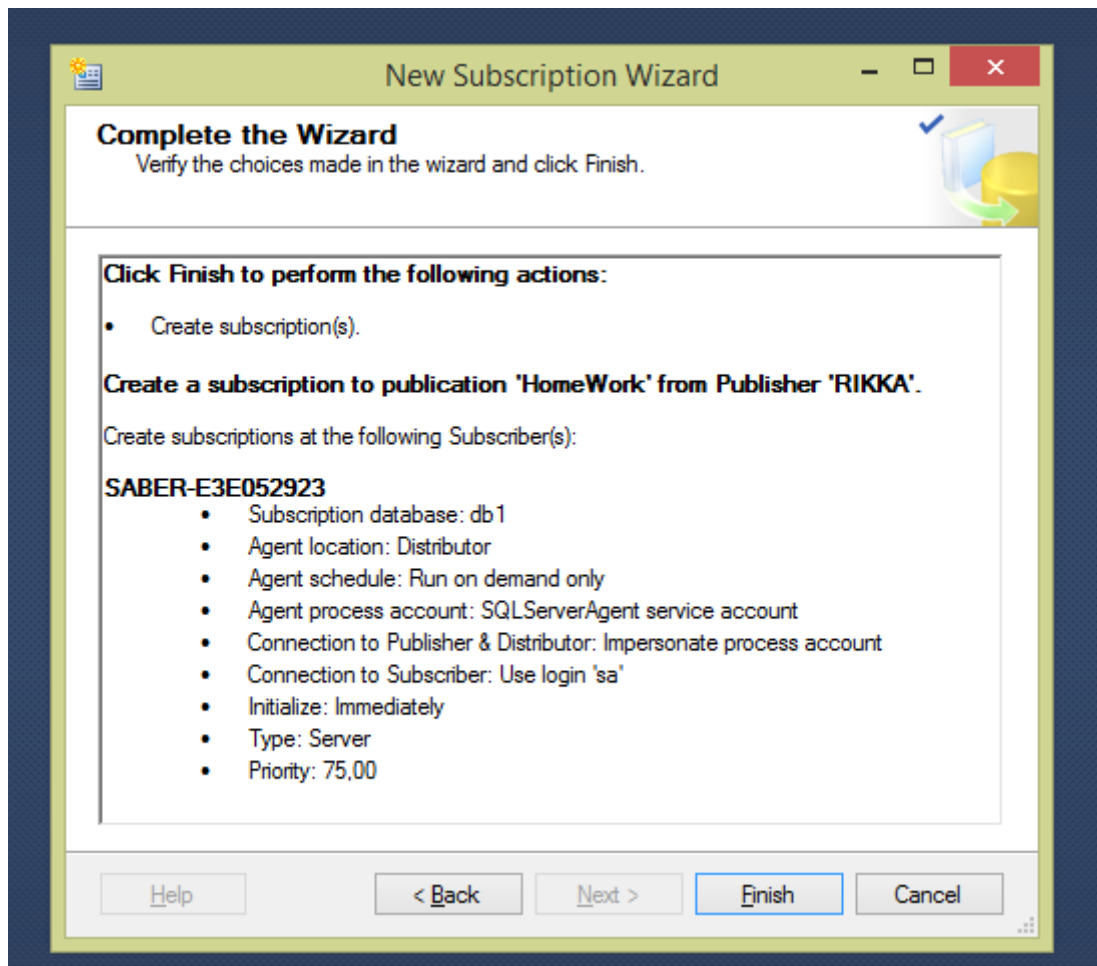


Sau khi click Finish ta sẽ có 1 Publication với tên là HomeWork. Tiếp đến ta tạo 2 Subscription tương đương với 2 máy trạm

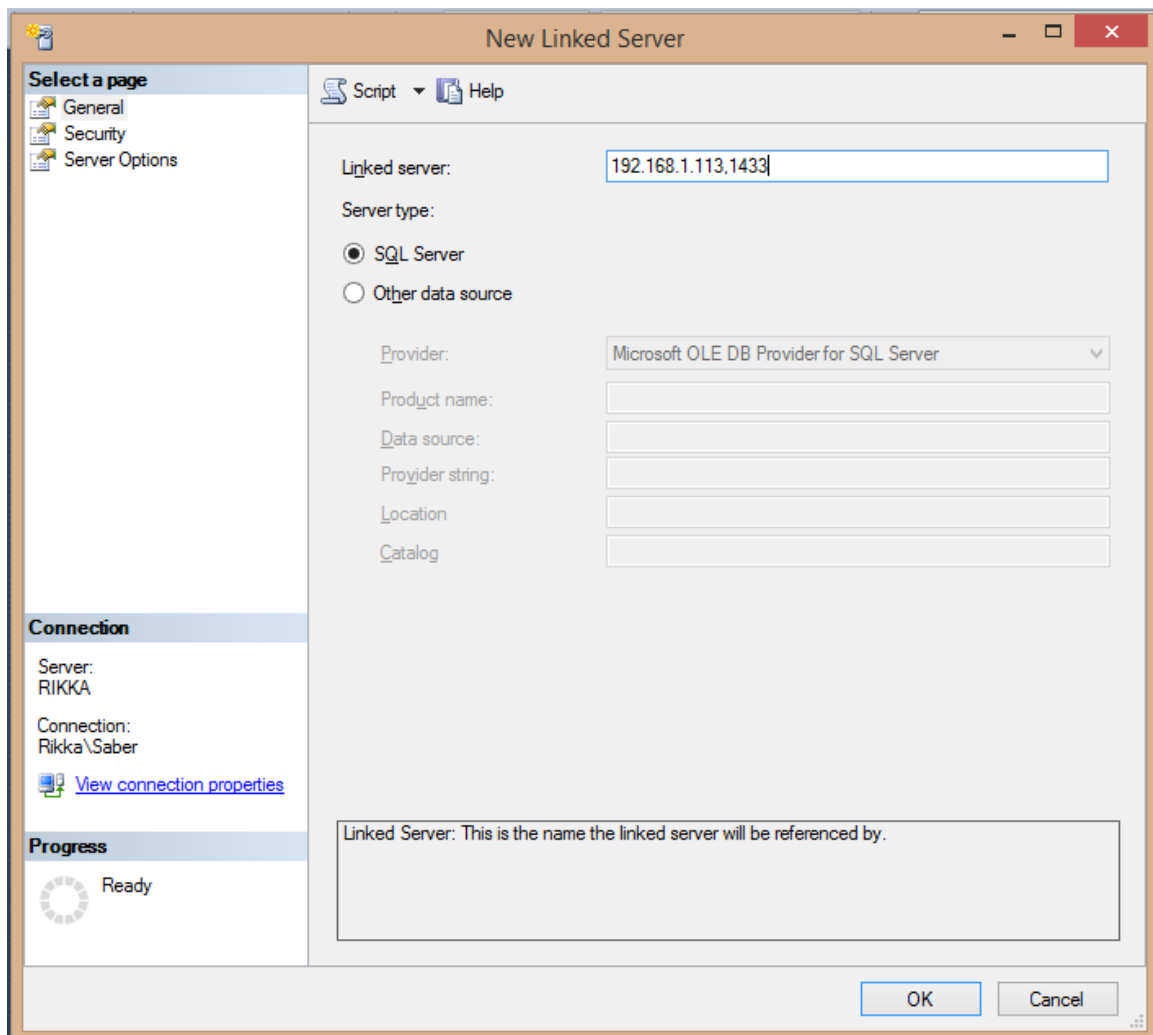


Tiếp đến ta chọn máy trạm và tạo 1 DB mới là để chứa dữ liệu nhân bản



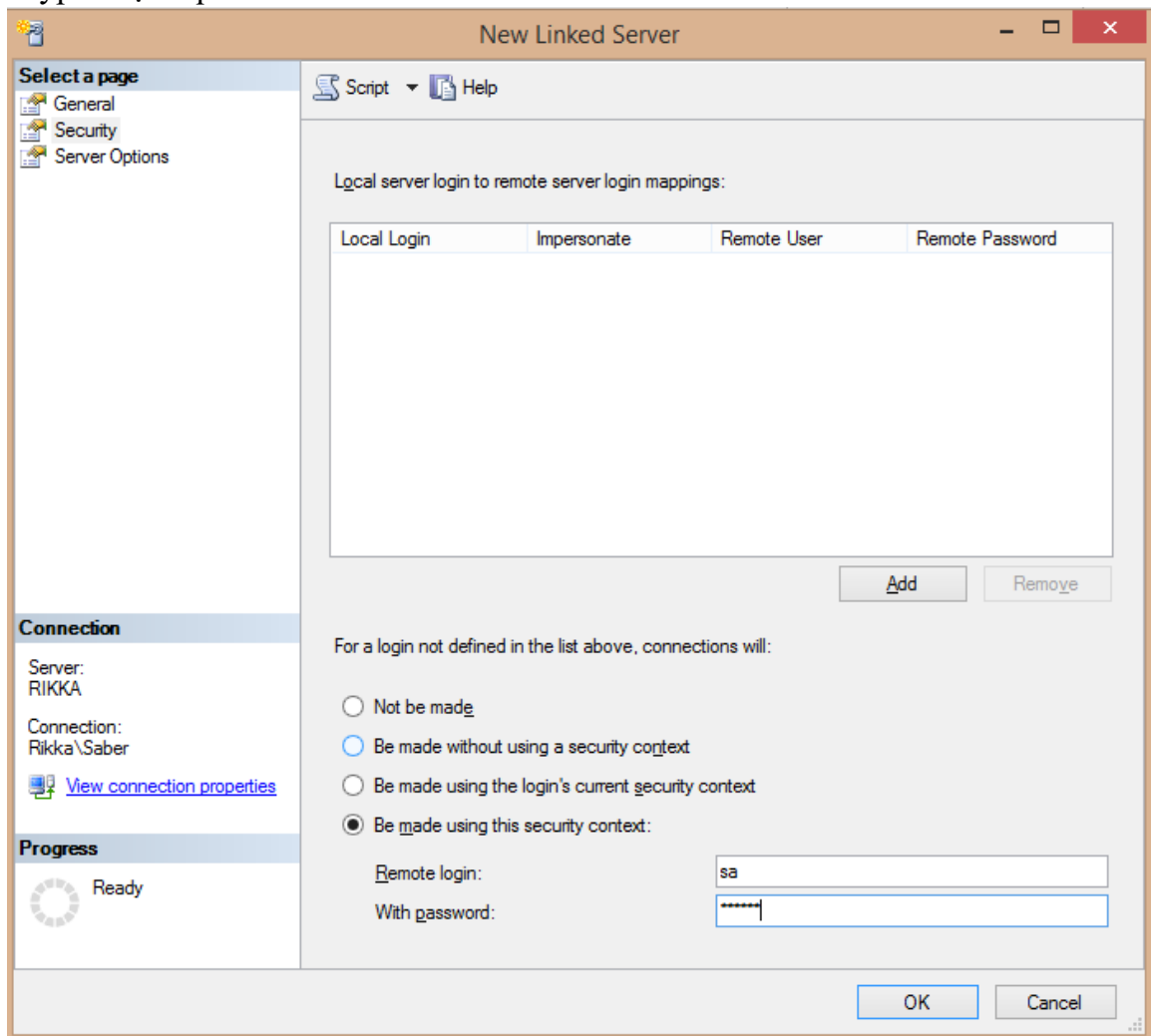


-Nếu muốn Link Server(trong kịch bản này ko cần)



Tiếp đến ta link các server lại để có thể truy vấn đến nhau.

Ở mục linked server có thể điền tên của server hoặc địa chỉ và cổng. Server type chọn sql server



ở tab security, chọn **be made using this security context** và điền thông tin đăng nhập cho tài khoản sa đã được cấu hình ở bước trên.

Ta sẽ link máy chủ tới 2 máy trạm và từ máy trạm tới máy chủ

Kết thúc phần cài đặt cho hệ thống.

4. Trigger

- Để phân chia dữ liệu một cách có chọn lọc và tự động update dữ liệu về các trạm khi có sự thay đổi từ phía Server, trạm Hà Nội chỉ có quyền xem thông tin nhân viên, khách hàng, vé tại Hà Nội, trạm Tp HCM chỉ có quyền xem thông tin nhân viên, khách hàng, vé tại Tp HCM nhóm sử dụng Trigger để xử lý vấn đề này

- Trigger là một thủ tục nội tại được thực thi ở phía Server khi có một sự kiện xảy ra như Insert, Delete hay Update. Khi một trong các hành động cập nhật trên thì thủ tục trigger sẽ tự động được thực hiện

- Một số Trigger được sử dụng để phân mảnh dữ liệu trên mỗi trạm

○ Trigger bảng nhanvien tại site Tp HCM

```
ALTER TRIGGER [dbo].[nhanvien_hcm]
ON [dbo].[tblNhanvien]
AFTER INSERT,UPDATE
AS
declare @chinhanh varchar(5)
set @chinhanh =(select ma from inserted)
if(@chinhanh like '%HC%')
begin
print (N'Cập nhật thành công')
end
else
BEGIN
raiserror(N'Lỗi',16,1)
rollback tran
END
```

○ Trigger bảng ve tại site Tp HCM

```
ALTER TRIGGER [dbo].[ve_hcm]
ON [dbo].[tblVe]
AFTER INSERT,UPDATE
AS
declare @nv varchar(5),
        @kh varchar(5)
set @nv =(select nhanvien from inserted)
set @kh = (select khachhang from inserted)
if(@nv like '%HC%' and @kh like '%HC%' )
begin
print (N'Cập nhật thành công')
end
else
BEGIN
raiserror(N'Lỗi',16,1)
rollback tran
END
```

○ Trigger bảng ve tại site Hà Nội

```

ALTER TRIGGER [dbo].[ve_hn]
ON [dbo].[tblVe]
AFTER INSERT,UPDATE
AS
declare @nv varchar(5),
        @kh varchar(5)
set @nv =(select nhanvien from inserted)
set @kh = (select khachhang from inserted)
if(@nv like '%HN%' and @kh like '%HN%' )
begin
print (N'Cập nhật thành công')
end
else
BEGIN
raiserror(N'Lỗi',16,1)
rollback tran
END

```

- Trigger bảng nhanvien tại site Hà Nội

```

ALTER TRIGGER [dbo].[nhanvien_hn]
ON [dbo].[tblNhanvien]
AFTER INSERT,UPDATE
AS
declare @chinhanh varchar(5)
set @chinhanh =(select ma from inserted)
if(@chinhanh like '%HN%')
begin
print ('Cập nhật thành công')
end
else
BEGIN
raiserror('Lỗi',16,1)
rollback tran
END

```

- Trigger bảng khachhang tại site Hà Nội

```

ALTER TRIGGER [dbo].[khachhang_hn]
ON [dbo].[tblKhachhang]
AFTER INSERT,UPDATE
AS
declare @chinhanh varchar(5)
set @chinhanh =(select ma from inserted)
if(@chinhanh like '%HN%')
begin
print (N'Cập Nhật Thành Công')
end
else
BEGIN
raiserror(N'Lỗi',16,1)
rollback tran
END

```

- Trigger bảng kháchhang tại site Tp HCM

```

ALTER TRIGGER [dbo].[khachhang_hcm]
ON [dbo].[tblKhachhang]
AFTER INSERT, UPDATE
AS
declare @chinhanh varchar(5)
set @chinhanh =(select ma from inserted)
if(@chinhanh like '%HC%')
begin
print (N'Cập nhật thành công')
end
else
BEGIN
raiserror(N'Lỗi',16,1)
rollback tran
END

```

-Một số trigger để kiểm soát việc thêm sửa dữ liệu trên máy chủ

- Trigger kiểm tra update and delete từ bảng máy bay. Nếu máy bay đang có sử dụng trong chuyến bay thì không được xóa hoặc sửa

```

ALTER TRIGGER [dbo].[kiemtra]
ON [dbo].[tblMaybay]
AFTER DELETE
AS
declare @c int, @ma varchar(5)
set @ma = (select ma from deleted)
set @c = (select count(*) from tblChuyenbay where maybay = @ma)
if(@c = 0)
BEGIN
print(N'Cập nhật thành công')
END
else
begin
rollback tran
raiserror(N'Lỗi',16,1)
end

```

```

ALTER TRIGGER [dbo].[kiemtra2]
ON [dbo].[tblMaybay]
AFTER UPDATE
AS declare @c int, @ma varchar(5)

set @ma = (select ma from inserted)
set @c = (select count(*) from tblChuyenbay where maybay = @ma)
if(@c = 0)
BEGIN
print(N'Cập nhật thành công')
END
else
begin
rollback tran
raiserror(N'Lỗi',16,1)
END

```

- Trigger kiểm tra update and delete từ bảng máy bay. Nếu máy bay đang có sử dụng trong chuyến bay thì không được xóa hoặc sửa

```

ALTER TRIGGER [dbo].[kiemtra3]
ON [dbo].[tblSanbay]
AFTER Delete
AS
declare @c int, @ma varchar(5)
set @ma = (select ma from deleted)
set @c = (select count(*) from tblChuyenbay where diemden= @ma or diemxuatphat=@ma)
if(@c = 0)
BEGIN
print(N'Cập nhật thành công')
END
else
begin
rollback tran
raiserror(N'Lỗi',16,1)
End

```

```

ALTER TRIGGER [dbo].[kiemtra4]
ON [dbo].[tblSanbay]
AFTER Update
AS
declare @c int, @ma varchar(5)

set @ma = (select ma from inserted)
set @c = (select count(*) from tblChuyenbay where diemden= @ma or diemxuatphat=@ma)
if(@c = 0)
BEGIN
print(N'Cập nhật thành công')
END
else
begin
rollback tran
raiserror(N'Lỗi',16,1)
End

```

- Trigger kiểm tra xem vé cho chuyến bay đã hết hay chưa

```

ALTER TRIGGER [dbo].[check_ve]
ON [dbo].[tblVe]
After INSERT,DELETE
AS declare @ghe1_booked int, @ghe2_booked int,@ghe1 int, @ghe2 int,
@ma_cb varchar(5),@ma_mb varchar(5),@giave1 float, @giave2 float
set @ma_cb = (select chuyenbay from inserted)
set @ma_mb = (select maybay from tblChuyenbay where ma = @ma_cb)
set @ghe1 = (select gheloai1 from tblMaybay where ma = @ma_mb)
set @ghe2 = (select gheloai2 from tblMaybay where ma = @ma_mb)
set @giave2 =(select dongia from tblChuyenbay where ma = @ma_cb)
set @giave1 =@giave2 + @giave2/10
set @ghe1_booked = (select count(giave) from tblVe where giave = @giave1 and chuyenbay=@ma_cb)
set @ghe2_booked = (select count(giave) from tblVe where giave = @giave2 and chuyenbay =@ma_cb)

if(@ghe1_booked > @ghe1 or @ghe2_booked > @ghe2)
BEGIN
raiserror('xit',16,1)
rollback tran
END
else
begin
print('Oke')
end

```

5. Giao diện hệ thống cho chức năng đặt vé

6.1. Giao diện đăng nhập

Đăng NhậpTrạm Hà Nội

Tên Đăng Nhập

Mật Khẩu

Đăng nhập

6.2. Giao diện tìm

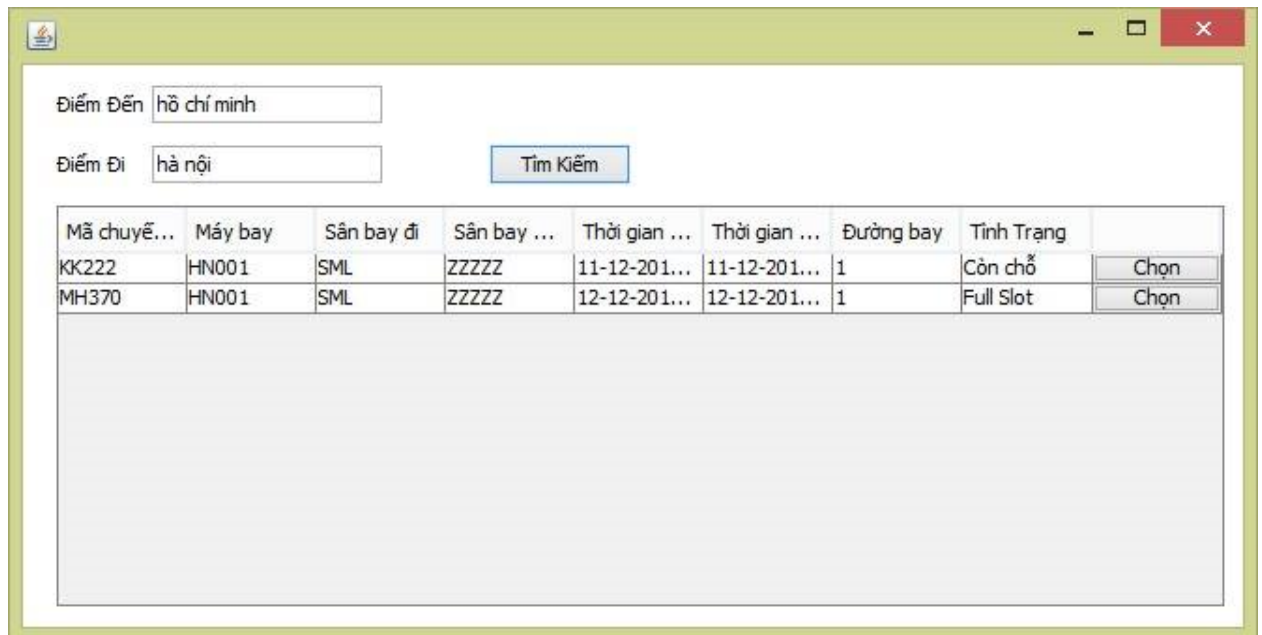
kiếm

Tên Khách Hàng Tuyết Tìm Thêm Đăng xuất

Họ Tên	Địa Chỉ	Ngày Sinh	Hộ Chiếu	Số Điện Thoại	
Tuyết	Nam Định	25-12-1995	4513	45152	Chọn

khách hàng để đặt vé

6.3. Giao diện tìm kiếm chuyến bay



Mã chuyến...	Máy bay	Sân bay đi	Sân bay ...	Thời gian ...	Thời gian ...	Đường bay	Tình Trạng	
KK222	HN001	SML	ZZZZZ	11-12-201...	11-12-201...	1	Còn chỗ	Chọn
MH370	HN001	SML	ZZZZZ	12-12-201...	12-12-201...	1	Full Slot	Chọn

6.4. Giao diện chọn loại vé



Chọn loại vé: Loại 1 Tình Trạng: 2

Giá: 110.0 Đặt Mua

vé

6.5. Giao diện thông tin



Thông Tin Vé

Mã vé: 8

Mã chuyến bay: KK222

Đi: Sân bay SML Lúc 11-12-2016 10:30:00

Đến: Sân bay ZZZZZ Lúc 11-12-2016 01:00:00

Tên khách hàng: Tuyết

Nhân viên bán vé: Đạt Đz