

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM ĐÀ NẴNG**

**KHOA TIN HỌC**

----🕮----



**BÁO CÁO MÔN CƠ SỞ DỮ LIỆU NÂNG CAO**

**ĐỀ TÀI:**

**HỆ PHÂN TÁN CƠ SƠ DỮ LIỆU**

**Giảng viên :** Phạm Dương Thu Hằng

**Nhóm sinh viên thực hiện :** Trần Bá Đăng

Trần Gia Quyền

Trần Quốc Trung

Đỗ Thành Quang

Phan Hữu Thiện

**Đà Nẵng, ngày 20 tháng 5 năm 2020**

**MỤC LỤC**

[**CHƯƠNG I : CƠ SỞ DỮ LIỆU PHÂN TÁN 3**](#_Toc74601450)

[**1.1 Cơ sở dữ liệu phân tán 3**](#_Toc74601451)

[**1.2 Đặc điểm của cơ sở dữ liệu phân tán 3**](#_Toc74601452)

[**1.2.1 Điều khiển tập trung 3**](#_Toc74601453)

[**1.2.2 Độc lập dữ liệu 3**](#_Toc74601454)

[**1.2.3 Giảm dư thừa dữ liệu 4**](#_Toc74601455)

[**1.3 Ưu và nhược điểm của cơ sở dữ liệu phân tán 4**](#_Toc74601456)

[**1.3.1 Ưu điểm 4**](#_Toc74601457)

[**1.3.2 Nhược điểm 4**](#_Toc74601458)

[**1.4 Kiến trúc một hệ CSDL phân tán 5**](#_Toc74601459)

[**1.5 Hệ quản trị cơ sở dữ liệu phân tán và phân loại 6**](#_Toc74601460)

[**1.6 Kĩ thuật phân đoạn, sao bản và định vị dữ liệu 8**](#_Toc74601461)

[**1.6.1 Các điều kiện ràng buộc trong thiết kế phân đoạn 8**](#_Toc74601462)

[**1.6.2 Các phương pháp phân đoạn 8**](#_Toc74601463)

[**1.6.3 Sao lặp dữ liệu 11**](#_Toc74601464)

[**1.7 Tính trong suốt của Hệ quản trị CSDLPT 11**](#_Toc74601465)

[**1.7.1 Trong suốt phân đoạn (Fragmentation Transparency) 11**](#_Toc74601466)

[**1.7.2 Tính trong suốt về vị trí (Location Transparency) 12**](#_Toc74601467)

[**1.7.3 Trong suốt ánh xạ địa phương (Local Mapping Transparency) 12**](#_Toc74601468)

[**1.7.4 Trong suốt nhân bản (Replication Transparency) 13**](#_Toc74601469)

[**1.8 Xử lý truy vấn trong CSDL phân tán 13**](#_Toc74601470)

[**CHƯƠNG II: CÀI ĐẶT, TRIỂN KHAI PHÂN MẢNH, ĐỒNG BỘ 14**](#_Toc74601471)

[**2.1. Chuẩn bị 14**](#_Toc74601472)

[**2.2 Tiến hành phân mảnh, đồng bộ 16**](#_Toc74601473)

# **CHƯƠNG I : CƠ SỞ DỮ LIỆU PHÂN TÁN**

# **1.1 Cơ sở dữ liệu phân tán**

CSDL phân tán là một tập hợp nhiều CSDL có liên quan logic và được phân bố trên một mạng máy tính

Hệ quản trị CSDL phân tán là hệ thống phần mềm cho phép quản lí CSDL phân tán và đảm bảo tính trong suốt về sự phân tán đối với người dùng

**Ví dụ**: ATM và google phân tán theo cách tự nhận biết,một yêu cầu gần server nào thì server đó xử lí,ATM phân tán rộng khắp,Google ở đâu cũng có.Tùy theo người lập trình mà cách xử lí CSDL được phát tán cho hợp lí

# **1.2 Đặc điểm của cơ sở dữ liệu phân tán**

## **1.2.1 Điều khiển tập trung**

Điều khiển tập trung là một đặc điểm của cơ sở dữ liệu tập trung, toàn bộ dữ liệu tập trung lại nhằm để tránh sự dư thừa dữ liệu,đảm bảo được tính độc lập của dữ liệu. Dữ liệu được quản lý tập trung bởi những người quản trị cơ sở dữ liệu. Chức năng  cơ bản của người quản trị cơ sở dữ liệu (DBA - Database Administrator) là đảm bảo sự an toàn dữ liệu .Trong cơ sở dữ liệu phân tán , sự điểu khiển được thực hiện theo cấu trúc điều khiển phân cấp gồm 2 loại người quản trị cơ sở dữ liệu:

* Người quản trị cơ sở dữ liệu toàn cục( Global Database Administrator) là người có trách nhiệm chính về toàn bộ cơ sở dữ liệu phân tán.
* Người quản trị sở dữ liệu cục bộ(Local Database Administrator) là người có trách nhiệm về cơ sở dữ liệu cục bộ của họ

## **1.2.2 Độc lập dữ liệu**

Độc lập dữ liệu là một đặc điểm của cơ sở dữ liệu , tổ chức lưu trữ dữ liệu là trong suốt đối với người lập trình ứng dụng

* Với trong suốt phân tán chúng ta hiểu rằng các chương trình ứng dụng có thể sử dụng cơ sở dữ liệu như là nó không được tổ chức phân tán.
* Sự chính xác của chương trình không bị ảnh hưởng bởi việc dịch chuyển dữ liệu từ trạm này đến trạm khác nhưng tốc độ thực hiện có thể bị ảnh hưởng.

Về ưu điểm của độc lập dữ liệu thì các chương trình không bị ảnh hưởng bởi những thay đổi về tổ chức lưu trữ vật lý của dữ liệu.

## **1.2.3 Giảm dư thừa dữ liệu**

Ở cơ sở dữ liệu tập trung ,sự dư thừa dữ liệu được giảm thiểu, vì tránh sự không nhất quán giữa nhiều bản sao bằng cách chỉ có 1 bản sao và tiết kiểm vùng bộ nhớ.

Ở cơ sở dữ liệu phân tán , sự dư thừa là 1 đặc điểm cần thiết vì:

* Làm tăng tính cục bộ của các ứng dụng nếu dữ liệu được nhân bản tại các vị trí mà ứng dụng cần dữ liệu này ,ứng dụng được thực hiện nhanh hơn vì không cần truy xuất dữ liệu từ xa.
* Làm tăng tính sẵn sàng của hệ thống ứng dụng, vì 1 vị trí có sự cố sẽ không làm ngưng sự thực hiện  của các ứng dụng ở những vị trí khác nếu dữ liệu tại vị trí bị hỏng được nhân bản tại các vị trí khác.

# **1.3 Ưu và nhược điểm của cơ sở dữ liệu phân tán**

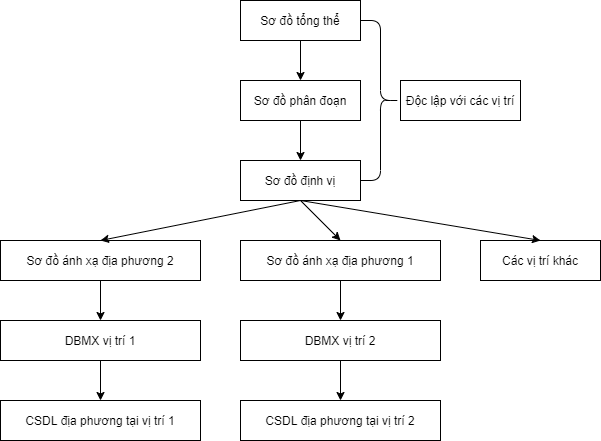
## **1.3.1 Ưu điểm**

* Đáp ứng nhanh hầu hết các ứng dụng sử dụng dữ liệu tại các trạm
* Tăng cường các đơn thể ứng dụng và CSDL mà không làm cản trở người sử dụng hiện tại
* Kiểm soát dữ liệu địa phương theo hướng hoàn thiện sự tích hợp và quản trị dữ liệu từ xa
* Tăng cường khả năng của hệ thống liên quan đến sự dư thừa dữ liệu

## **1.3.2 Nhược điểm**

* Phần mềm đắt và phức tạp.
* Phải xử lý các thay đổi thông báo trong mọi địa điểm.
* Khó kiểm soát tính toàn vẹn dữ liệu với nhiều bản sao dữ liệu được phân bố khắp mọi nơi.
* Đáp ứng chậm nhu cầu của các trạm trong trường hợp các phần mềm ứng dụng không được phân bố phù hợp với nhiều việc sử dụng chung.

# **1.4 Kiến trúc một hệ CSDL phân tán**



**Sơ đồ tổng thể**

* Sơ đồ này xác định tất cả các dữ liệu được lưu trữ trong CSDL phân tán.
* Sơ đồ hình thành từ mô hình quan hệ, sử dụng mô hình này sơ đồ tổng thể bao gồm định nghĩa của 1 tập các quan hệ tổng thể

**Sơ đồ phân đoạn**

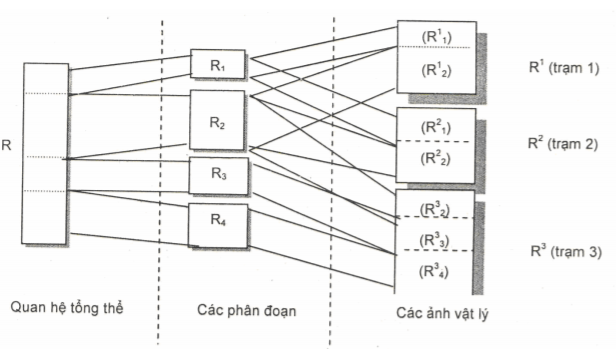
* Mỗi quan hệ tổng thể được chia thành 1 vài phần nhỏ hơn không giao nhau được gọi là đoạn
* Sơ đồ tổng thể mô tả các ánh xạ giữa các quan hệ tổng thế và các đoạn được định nghĩa trong sơ đồ phân đoạn

**Sơ đồ định vị**

* Các đoạn là các phần logic của một quan hệ tổng thể được định vị trên một hoặc nhiều vị trí vật lý trên mạng. Sơ đồ định vị xác định đoạn nào ở các trạm nào

**Ví Dụ** : **ký hiệu R chỉ tới đoạn thứ i của quan hệ tổng thể R**

**Rj là ảnh vật lý của quan hệ tổng thể R tại trạm j**



**Sơ đồ ánh xạ địa phương**

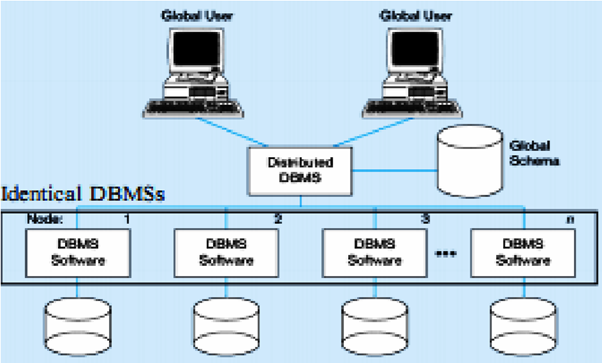
* Ánh xạ các ảnh vật lý tới các đối tượng được các hệ quản trị CSDL địa phương thao tác tại các trạm.
* Ánh xạ này phụ thuộc vào các hệ quản trị CSDL địa phương
* Trong 1 hệ thống không đồng nhất phải có các kiẻu ánh xạ địa phương khác nhau tại các trạm khác nhau

# **1.5 Hệ quản trị cơ sở dữ liệu phân tán và phân loại**

Hệ quản trị CSDL phân tán cung cấp công cụ như tạo lập và quản lý CSDL phân tán. HQTCSDLPT có chức năng hỗ trợ việc tạo và bảo trì CSDLPT , chúng có các thành phần tương tự như 1 HQTCSDL tập trung và các thành phần hỗ trợ trong chuyển tải dữ liệu đến các trạm và ngược lại.

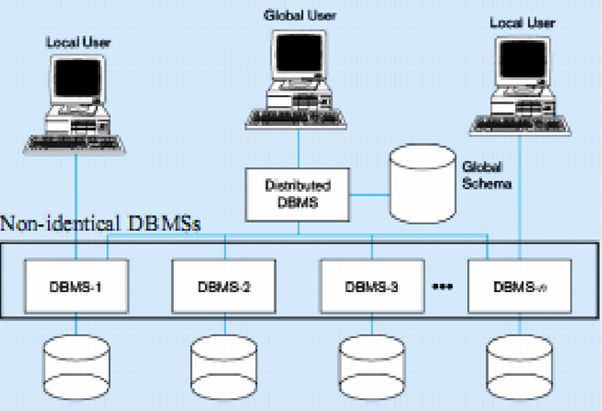
**CSDL phân tán thuần nhất**

* Tất cả các nút cùng sử dụng 1 loại HQTCSDL
* Thuận lợi cho việc tăng trưởng và cho phép nâng cao hiệu năng.
* Có được bằng cách chia 1 CSDL thành 1 cặp CSDL cục bộ



**CSDL phân tán hỗn tạp**

* Các nút có thể thực hiện trên các hệ quản trị CSDL khác nhau.
* Xảy ra khi các nút đã cài đặt CSDL riêng.
* Có được bằng cách tích hợp các CSDL cục bộ đã có.



# **1.6 Kĩ thuật phân đoạn, sao bản và định vị dữ liệu**

Phân hoạch cơ sở dữ liệu thành các đoạn (fragments): sự phân đoạn cho phép một đối tượng riêng lẻ thành hai hay nhiều mảnh. Thông tin lưu trữ được lưu trong catalog dữ liệu phân tán. Phần mềm xử lý giao tác sẽ truy nhập thông tin ở đây để xử lý các yêu cầu của người dùng.

Việc chia quan hệ tổng thể thành các đoạn có thể thực hiện bằng cách ứng dụng các kiểu phân đoạn sau:

* Phân đoạn ngang.
* Phân đoạn dọc.
* Phân đoạn hỗn hợp: là sự kết hợp giữa phân đoạn ngang và phân đoạn dọc.

## **1.6.1 Các điều kiện ràng buộc trong thiết kế phân đoạn**

Một phương pháp thiết kế phân đoạn đúng đắn phải thỏa mãn ba ràng buộc sau:

* *Tính đầy đủ*: Toàn bộ dữ liệu của quan hệ tổng thể phải được ánh xạ vào các đoạn quan hệ và ngược lại. Điều này có nghĩa là, không tồn tại một mục dữ liệu nào thuộc vào quan hệ tổng thể mà không thuộc vào bất kỳ một đoạn nào.
* *Xây dựng lại*: Quan hệ tổng thể có thể xây dựng là từ các đoạn mà nó đã tách ra. Điều kiện này là hiển nhiên, bởi vì trong thực tế chỉ có các đoạn được lưu trữ trong CSDL phân tán, và quan hệ tổng thể phải được xây dựng lại thông qua các đoạn khi cần thiết.
* *Tính rời nhau*: Các đoạn được tách ra từ quan hệ tổng thể phải là rời nhau. Vì vậy, việc tạo các bản sao phải rõ ràng với các đoạn được chia. Tuy nhiên, điều này chỉ áp dụng vào việc phân đoạn ngang, trong khi việc phân đoạn dọc nhiều khi vẫn được phép vi phạm điều kiện này.

## **1.6.2 Các phương pháp phân đoạn**

1. **Phân đoạn ngang**

Phân đoạn ngang(horizontal fragmentation partitioning) là tách quan hệ tổng thể R thành các tập con R1, R2, …, Rn. Mỗi tập con chứa một số n\_ bộ của R, điều này rất hữu ích trong CSDL phân tán, nơi mỗi tập con bao gồm các dữ liệu có các thuộc tính địa lý chung. Mỗi n\_bộ thuộc vào một trong các thành phần để có thể khôi phục được quan hệ tổng thể R khi cần thiết. Việc khôi phục quan hệ R được thực hiện bằng phép hợp các quan hệ: R= R1∪ R2 ∪ …  ∪ Rn

Ta có thể sử dụng các phép toán chọn lọc dựa trên quan hệ tổng thể để tách quan hệ tổng thể thành các đoạn.

**Ví dụ:** Ta có quan hệ sinh viên: SV(ID, Ten, NgaySinh, QueQuan)

Ta có thể phân thành hai phân đoạn ngang như sau:

SV1 = SELECT QueQuan= “Đà Nẵng” SV

SV2 = SELECT QueQuan = “Quảng Nam” SV

Điều kiện xây dựng lại được đảm bảo: SV = SV1 ∪ SV2

Dễ dàng thấy rằng, các phân đoạn trên thỏa mãn điều kiện tách rời nhau và đầy đủ. Trong thực tế, nhiều khi cần có các đoạn mà n \_bộ của nó thuộc vào nhiều đoạn khác nhau.

**Ví dụ**: Ta có quan hệ tổng thể của giáo viên sau: GV(ID, Ten, GioiTinh, Quequan)

Do mỗi giáo viên thuộc vào một trường, các trường khác nhau có cách quản lý nhân viên của mình khác nhau, nên việc phân chia các nhân viên thuộc các trường khác nhau vào các quan hệ khác nhau là điều cần thiết. Tuy nhiên, trường không phải là một thuộc tính của quan hệ GV, nó là thuộc tính của quan hệ:

GV1 = GV JOINID = ID NV1

GV2 = GV JOINID = ID NV2

Ở trên đã sử dụng phép toán nửa nối giữa các quan hệ GV và NV1, NV2.

Ta có thể mô tả điều kiện tham chiếu đầy đủ của hai đoạn trên là:

q1: GV.ID = NV.ID AND NV.Truong = “Sư phạm”

q2: GV.ID = NV.ID AND NV.Truong = “Bách khoa”

Điều kiện xây dựng lại của quan hệ tổng thể GIÁO VIÊN có thể thực hiện được thông qua phép toán hợp như đã nêu ra ở quan hệ sinh viên ở trên.

1. **Phân đoạn dọc**

Phân đoạn dọc (vertical fragmentation partitioning) của một quan hệ tổng thể là việc chia các thuộc tính của nó thành các nhóm. Điều này là hữu ích trong các CSDL phân tán, khi mà mỗi nhóm các thuộc tính có thể chứa dữ liệu có chung những thuộc tính địa lý. Việc phân đoạn là đúng đắn nếu như mỗi thuộc tính của quan hệ tổng thể được ánh xạ thành thuộc tính của ít nhất một đoạn con, và phải đảm bảo điều kiện là quan hệ tổng thể có thể được xây dưng lại từ các phân đoạn mà nó chia ra bằng các phép kết nối.

* Phân rã theo chiều dọc quan hệ tổng thể R thành các quan hệ Ri:

Ri= Chiếu thuộc tính Ri (R)

* Quan hệ ban đầu được khôi phục nhờ các phép kết nối tự nhiên:

R =R1 |X| R2 |X| … |X| Rn

* Việc đưa các khóa của quan hệ tổng thể vào trong mỗi đoạn được tách ra là cách dễ nhất để có thể xây dựng lại quan hệ tổng thể bằng các phép toán kết nối.

**Ví dụ,** cho quan hệ tổng thể: NV(ID\_NV, Ten\_NV, Luong, Thue\_TN, Phong, Bophan)

Một phân đoạn dọc của quan hệ trên có thể được định nghĩa như sau:

NV1 = PROJECTIONID\_NV, Ten\_NV, Phong, Bophan (NV)

NV2 = PROJECTIONID\_NV, Luong, Thue\_TN (NV)

Việc xây dựng lại quan hệ tổng thể nhân viên có thể được thể hiện như sau:

NV = NV1 JOINID\_NV = ID\_NV (NV2)

Quan hệ tổng thể NV được xây dựng lại nhờ việc kết nối hai phân đoạn NV1 và NV2 dựa vào khóa chính ID\_NV.

Trong thực tế, các phân đoạn dọc nhiều khi chứa cùng một số các thuộc tính (không phải thuộc tính khóa) như nhau trong các phân đoạn khác nhau, ta có thể loại bỏ các thuộc tính này khi xây dựng lại quan hệ tổng thể.

**Ví dụ**, với quan hệ tổng thể nhân viên ở trên, ta có thể phân thành hai phân đoạn dọc sau:

NV1 = PROJECTIONID\_NV, Ten\_NV, Phong, Bophan (NV)

NV2 = PROJECTIONID\_NV, Ten\_NV, Luong, Thue\_TN (NV)

Thuộc tính tên\_NV được lặp lại trong cả hai phân đoạn. Việc xây dựng lại quan hệ tổng thể NHÂN VIÊN được thể hiện như sau:

NV = NV1 JOINID\_NV= ID\_NV OJECTIONID\_NV, Luong, Thuế\_TN (NV2)

1. **Phân đoạn hỗn hợp**

**Ví dụ** : Cho quan hệ tổng thể :NV(ID\_NV, Ten\_NV, Luong, Thue\_TN, Phong, Bophan)

Sau đây là một phân đoạn hỗn hợp (hybrid fragmentation partitioning), thu được bằng áp dụng các phân đoạn ngang dựa trên các phân đoạn dọc ở ví dụ trước.

NV1 = SELECTID\_NV<=10 PROJECTIONID\_NV, Ten\_NV, Phong, Bopphan(NV)

NV2 = SELECTID\_NV<=20 PROJECTIONID\_NV, Ten\_NV, Phong, Bopphan(NV)

NV1 = SELECTID\_NV>20 PROJECTIONID\_NV, Ten\_NV, Phong, Bopphan(NV)

NV4 = PROJECTION ID\_NV ,Ten\_NV, Luong, Thue\_NV (NV)

## **1.6.3 Sao lặp dữ liệu**

CSDL được sao thành nhiều bản sao từng phần, hay đầy đủ và được đặt ở hai hay nhiều vị trí trên mạng. Nếu bản sao của CSDL được lưu trữ tại mọi trạm, ta có 14 trường hợp sao lặp đầy đủ. Phương thức này làm cực đại việc truy nhập tới dữ liệu ở mọi địa phương. Tuy nhiên, phương thức này nảy sinh nhiều vấn đề khi cập nhật (khi có thay đổi dữ liệu ở một cơ sở thì cần được xử lý lại và đồng bộ hóa dữ liệu cho tất cả các vị trí khác). Một kỹ thuật mới hơn cho phép tạo các bản sao không đầy đủ phù hợp với yêu cầu dữ liệu mỗi trạm lưu trữ và một bản sao đầy đủ ở máy dịch vụ. Sau mỗi thời gian, các bản sao được làm đồng bộ với bản chinh ở máy dịch vụ bằng một công cụ phần mềm nào đó.

Ví dụ: Phần mềm Brieafcase là một công cụ của Microsoft cho phép thực hiện dịch vụ này trên CDSL Acess.

**1.6.4 Định vị dữ liệu**

Định vị dữ liệu liên quan đến các công việc của người sử dụng và người lập trình ứng dụng trên các đoạn dữ liệu được định vị tại các trạm. Thông qua tính trong suốt định vị (location transparency) người lập trình sẽ biết được vị trí của các đoạn trên các trạm.

# **1.7 Tính trong suốt của Hệ quản trị CSDLPT**

Tính trong suốt của một hệ phân tán được hiểu như là việc che khuất đi các thành phần riêng biệt của hệ đối với người sử dụng và những người lập trình ứng dụng.

Các loại trong suốt trong hệ phân tán:

* Trong suốt phân đoạn (fragmentation transparency)
* Trong suốt về vị trí (location transparency)
* Trong suốt ánh xạ địa phương (local mapping transparency)
* Trong suốt nhân bản (replication transparency)
* Không trong suốt (no transparency)

## **1.7.1** **Trong suốt phân đoạn (Fragmentation Transparency)**

Trong suốt phân đoạn: là cấp độ cao nhất của mức độ trong suốt, người sử dụng hoặc chương trình ứng dụng chỉ làm việc trên các quan hệ của cơ sở dữ liệu.

Khi dữ liệu đã được phân đoạn thì việc truy cập vào CSDL được thực hiện bình thường như là chưa bị phân tán và không ảnh hưởng tới người sử dụng.

**VD:** Xét quan hệ tổng thể TSV (Id,Tên,Tuổi,Địa chỉ) và các phân đoạn được tách ra

TSV1 (Id, Tên, Tuổi,Địa chỉ)

TSV2 (Id, Tên, Tuổi,Địa chỉ)

TSV3 (Id, Tên, Tuổi,Địa chỉ)

Giả sử,hệ thống cơ sở dữ liệu phân tán cung cấp tính trong suốt về phân đoạn,ta có thể thấy tính trong suốt này được thể hiện như sau:

Khi muốn tìm một người có Id=”Id1“ thì chỉ cần tìm trên quan hệ tổng thể NCC mà không cần biết quan hệ TSV có phân tán hay không.

**SELECT \* FROM NCC WHERE Id=”Id1”**

## **1.7.2** **Tính trong suốt về vị trí (Location Transparency)**

Người sử dụng không cần biết về vị trí vật lý của dữ liệu mà có quyền truy cập đến cơ sở dữ liệu tại bất cứ vị trí nào.

Các thao tác để lấy hoặc cập nhật một dữ liệu từ xa được tự động thực hiện bởi hệ thống tại điểm đưa ra yêu cầu.

Tính trong suốt về vị trí rất hữu ích, nó cho phép người sử dụng bỏ qua các bản sao dữ liệu đã tồn tại ở mỗi vị trí. Do đó có thể di chuyển một bản sao dữ liệu từ một vị trí này đến một vị trí khác và cho phép tạo các bản sao mới mà không ảnh hưởng đến các ứng dụng.

**Ví dụ**: Với quan hệ tổng thể  và các phân đoạn như đã nói ở trên nhưng giả sử rằng hệ thống  CSDL phân tán cung cấp trong suốt về vị trí nhưng không cung cấp trong suốt về phân đoạn. Xét câu truy vấn tìm người có Id=”Id1”

**SELECT \*** **FROM TSV1** **WHERE Id=”Id1”**

**IF NOT #FOUND THEN**

**SELECT \*** **FROM TSV2** **WHERE Id=”Id1”**

Đầu tiên hệ thống sẽ thực hiện tìm kiếm ở phân đoạn TSV1 và nếu hệ quản lý CSDL  trả về biến điều khiển #FOUND thì một câu lệnh truy vấn tương tự được thực hiện trên phân đoạn TSV2

Ở đây quan hệ TSV2 được sao làm hai bản trên hai vị trí2 và vị trí3, ta chỉ cần tìm thông tin trên quan hệ TSV2 mà không cần quan tâm nó ở vị trí nào.

## **1.7.3 Trong suốt ánh xạ địa phương (Local Mapping Transparency)**

Là một đặc tính quan trọng trong một hệ thống DBMS không đồng nhất

Ứng dụng tham chiếu đến các đối tượng có các tên độc lập từ các hệ thống cục bộ địa phương.

Ứng dụng được cài đặt trên một hệ thống không đồng nhất nhưng được sử dụng như một hệ thống đồng nhất.

## **1.7.4** **Trong suốt nhân bản (Replication Transparency)**

Mức trong suốt bản sao liên quan chặt chẽ tới mức trong suốt định vị. Mức trong suốt bản sao có nghĩa là người sử dụng không biết bản sao của đoạn đặt ở vị trí nào. Mức trong suốt bản sao tương đương mức trong suốt định vị. Tuy nhiên, trong những trường hợp thực tế người sử dụng không có mức trong suốt định vị nhưng lại có mức trong suốt bản sao.

# **1.8 Xử lý truy vấn trong CSDL phân tán**

VD: Cho lượt đồ cơ sở dữ liệu:

SV(MSV,TenSV,QUE,LOP)

DT(MDT,MSV,TenDT,CN,KINHPHI,)

Liệt kê tên các sinh viên hiện đang tham gia đề tài số 1

* Câu truy vấn

**SELECT TenSV FROM SV,DT WHERE SV.MSV=DT.MSV AND TenDT= “đề tài 1”**

* Trình bày dưới dạng đại số quan hệ từ câu truy vấn 2 cách :

**C1: π(TenNV)( σ(TênDT="đề tài 1")ʌ(SV.MSV=DT.MSV)(SV><DT))**

**C2:π(TenNV)( SV|><|MSV(σ(TênDT="đề tài 1"(DT)))**

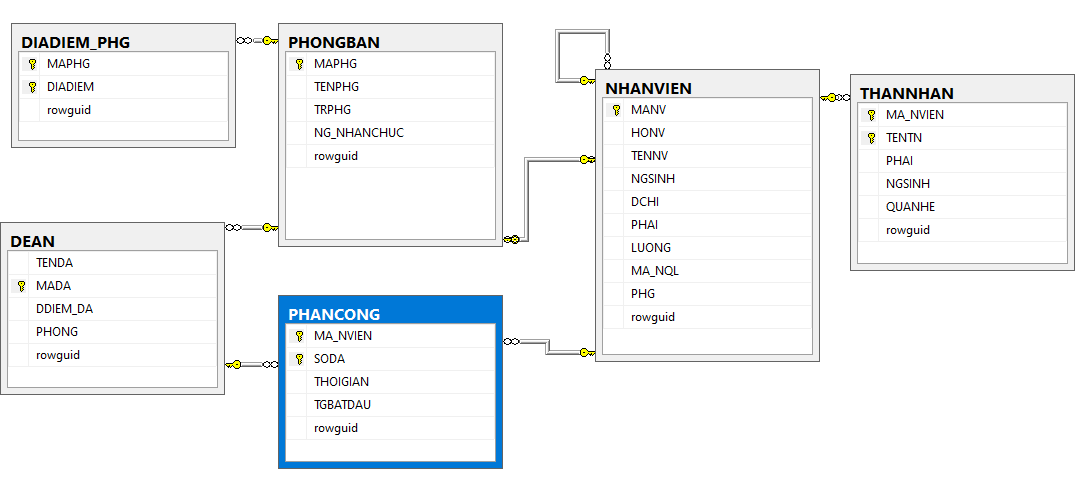
Qua câu truy vấn c2 ta đã sử dụng tích Descartes nên tiêu dùng ít tài nguyên hơn c1 vì thế nên được giữ lại.Vì trong hệ phân tán thì đại số quan hệ không đủ để diễn tả các chiến lược thực thi , nó phải được cung cấp thêm các phép toán trao đổi dữ liệu giữa các vị trí

(còn bổ sung thêm)

# **CHƯƠNG II: CÀI ĐẶT, TRIỂN KHAI PHÂN MẢNH, ĐỒNG BỘ**

# **2.1. Chuẩn bị**

* Cài đặt SQL Server 2012 Evalution
* Cài đặt 3 Server gồm: 1 Server gốc và 2 Server con
* Tạo sẵn 1 database về quản lí



* Lược đồ phân mảnh

🡪 PHONGBAN

🡪 NHANVIEN

🡪 THANNHAN

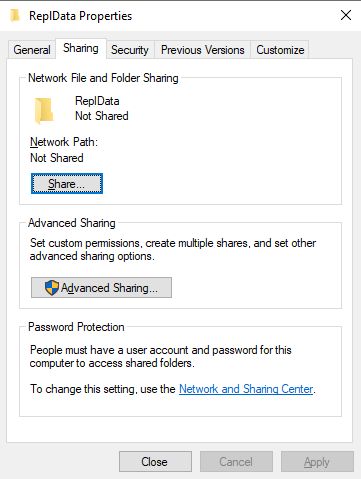
🡪 PHANCONG

🡪 DIADIEM\_PHG

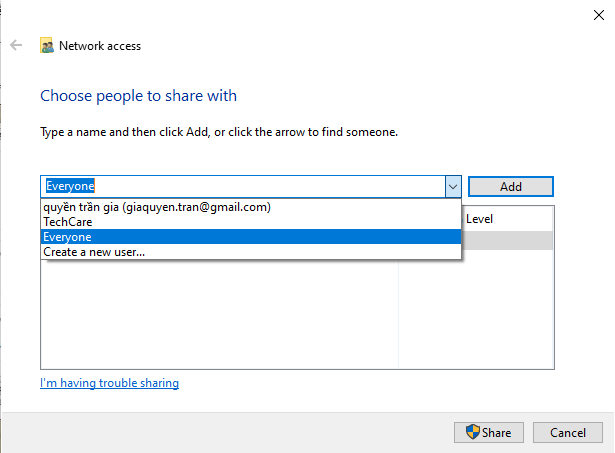
🡪 DEAN

🡪 PHANCONG

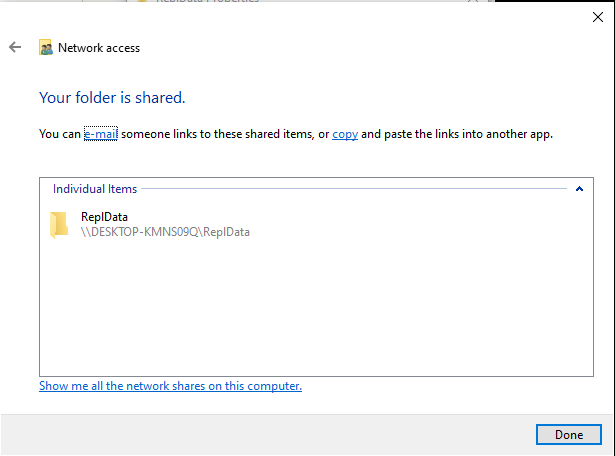
* Tạo một folder ReplData để chứa các dữ liệu trao đổi trong quá trình update dữ liệu từ các phân mảnh về cơ sở dữ liệu gốc, và từ cơ sở dữ liệu gốc đến các phân mảnh
* Tiến hành cho folder ReplData cấp phép các Users được quyền Read/Write



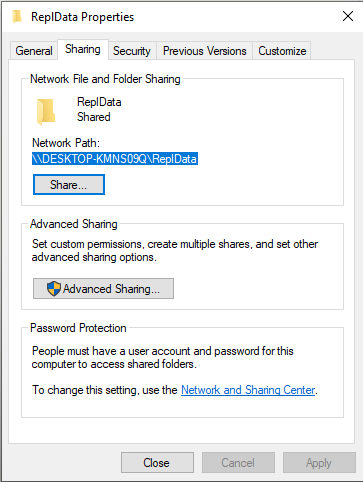
Right click folder ReplData 🡪 Properties 🡪 Sharing 🡪 Share



Click Add 🡪 Everyone



Done

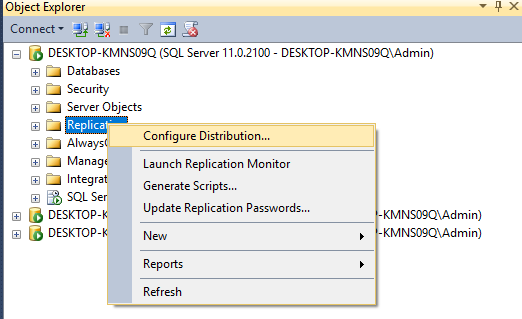


**2.2 Tiến hành phân mảnh, đồng bộ**

**Bước 1**: Cấu hình dístribution

Ở bước này ta cấu hình distribution để trao đổi dữ liệu trong quá trình đồng bộ hóa dữ liệu các Server con - Server gốc và ngược lại

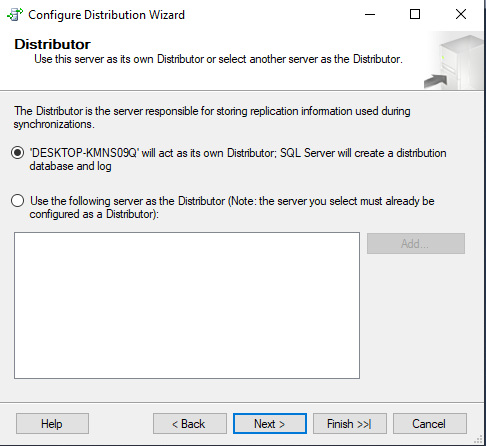
**Lưu ý** : bước này chỉ cần cấu hình ở Server gốc



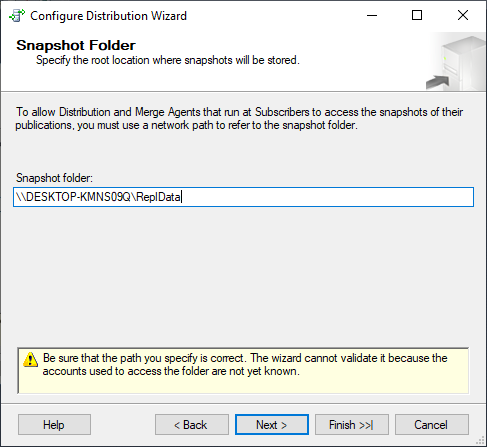
Chuột phải Replication 🡪 Configure Distribution



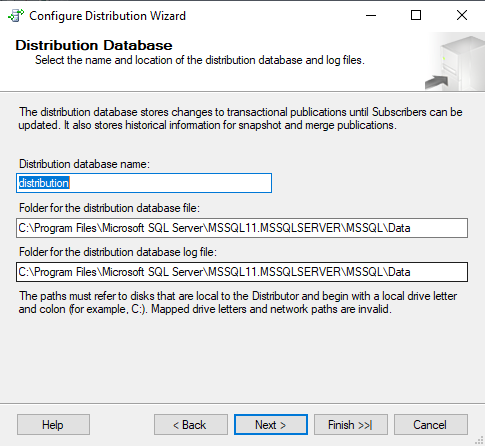
Click Next



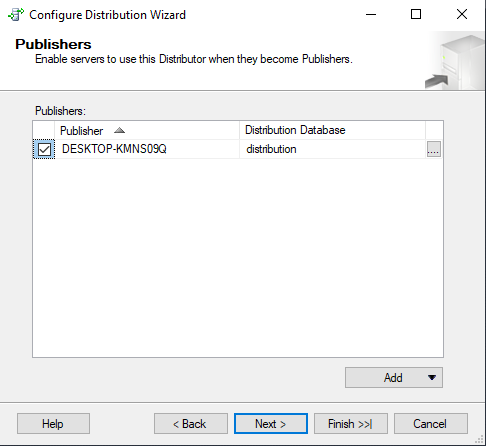
Click Next



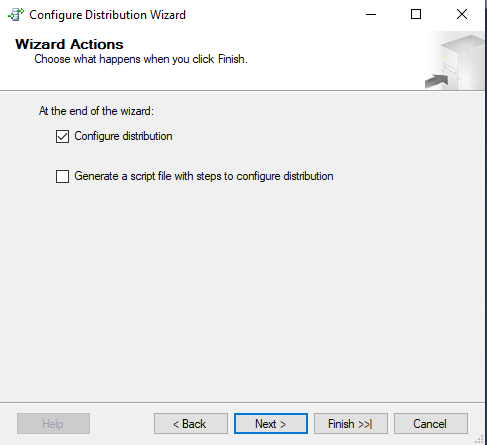
Sao chép đường dẫn khi Network Path bỏ vào đường dẫn Snapshot Folder



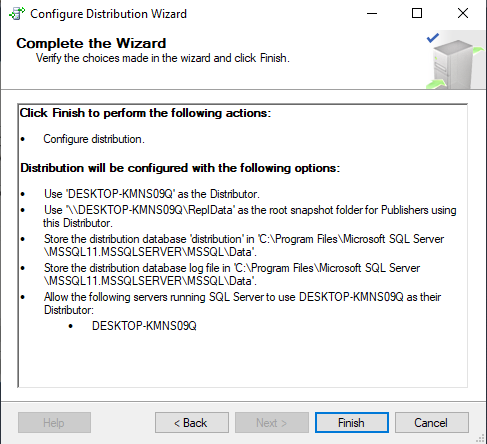
Click Next



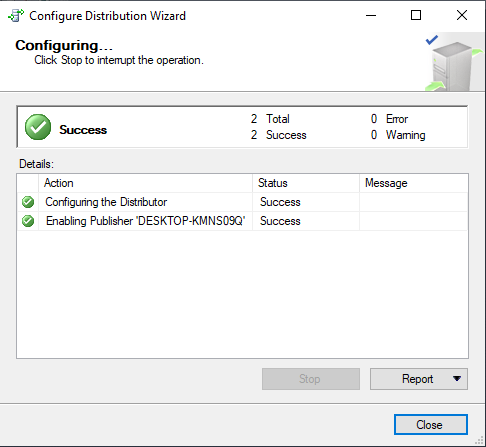
Chọn vào tên Server Gốc và Next



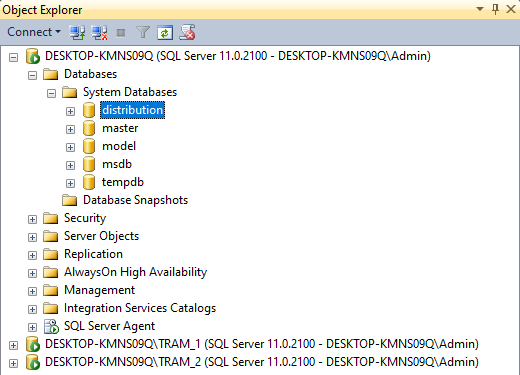
Click Next



Click Finish



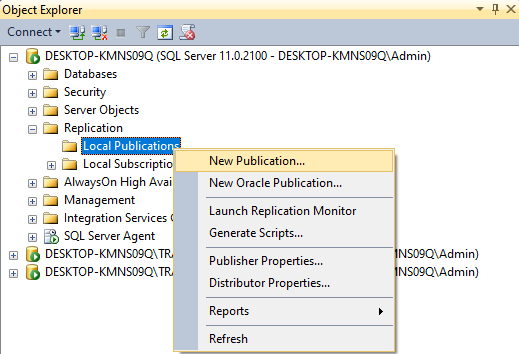
Chờ Server thiết lập và Close



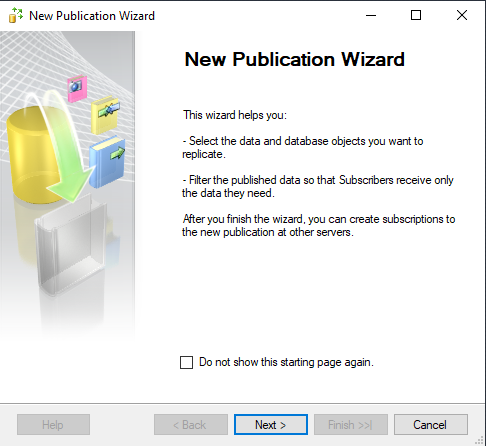
Kết quả của bước 1

Bước 2: Tạo publications

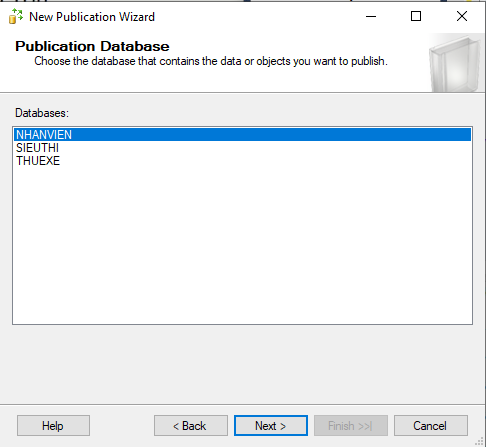
Ở bước tiếp theo, ta tạo publications để chứa “Lược đồ phân mảnh” nhằm thể hiện mối quan hệ dữ liệu giữa các bảng khi chúng ta phân tán dữ liệu



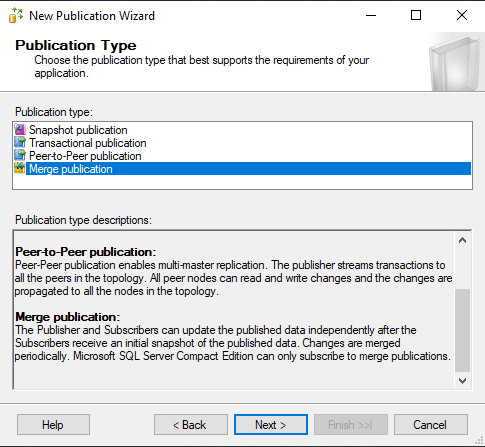
Chuột phải Local Publications 🡪 New Publication



Click Next

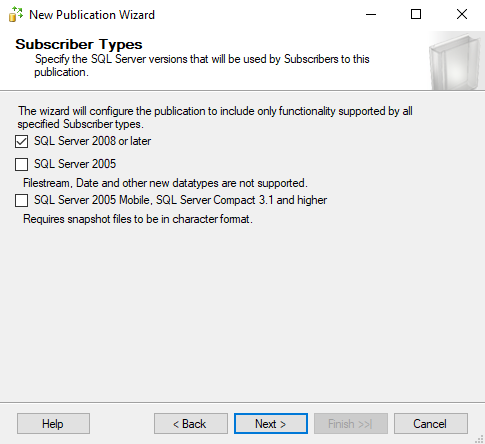


Lựa chọn database cần phân tán

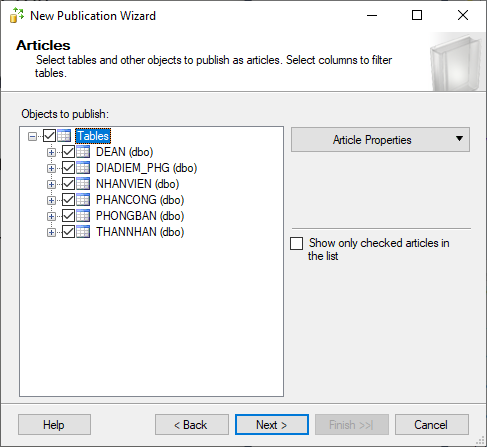


Click Merge publication

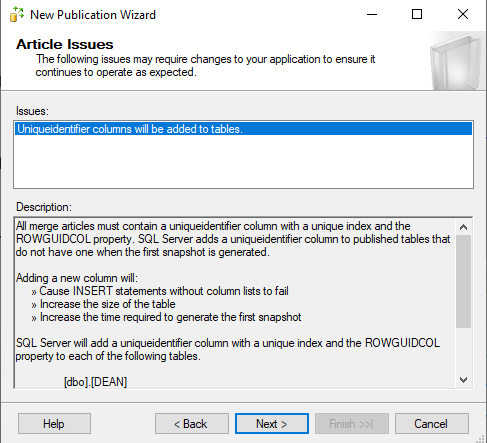
Merge replication cho phép nhiều server làm việc độc lập (online hay ofline) sau đó hợp nhất dữ liệu đã thay đổi lại dựa vào độ ưu tiên, thời điểm chỉnh sửa hoặc do người dùng tự quy định. Subscriber sẽ đồng bộ với publisher khi được kết nối vào mạng và sẽ chuyển giao tất cả các row đã thực hiện thay đổi giữa publisher và subscriber kể từ lần đồng bộ cuối cùng.



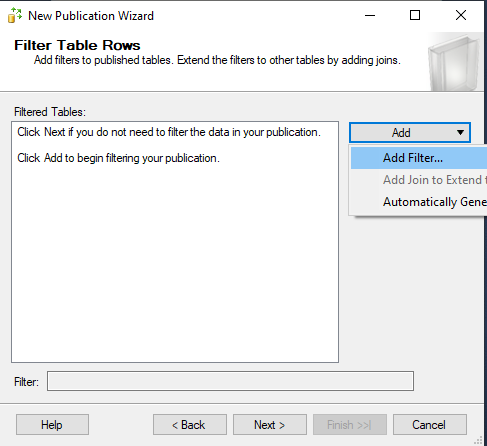
Lựa chọn phiên bản SQL Server đang sử dụng và Next



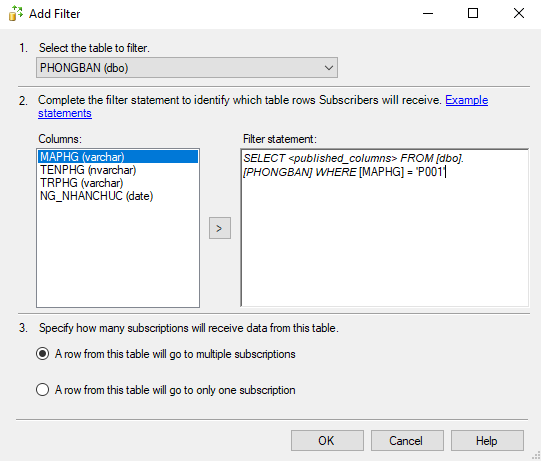
Chọn vào các bảng thực hiện nhân bảng



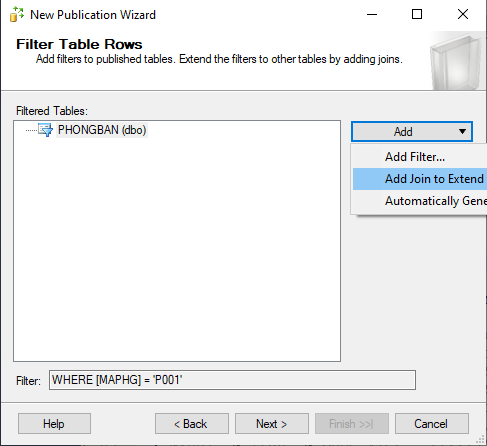
Click Next



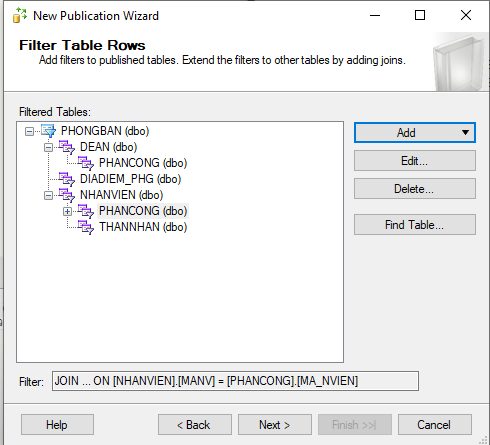
Click Add 🡪 Add Filter



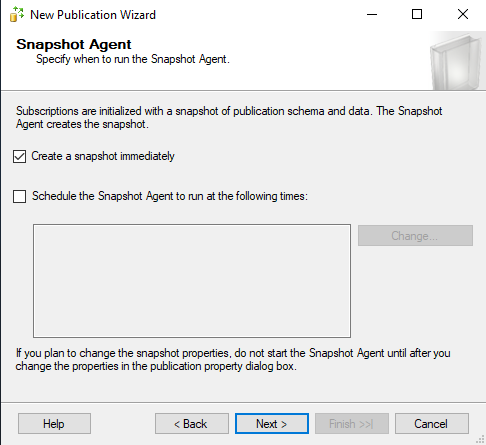
Thiết lập gốc cho lược đồ phân mảnh



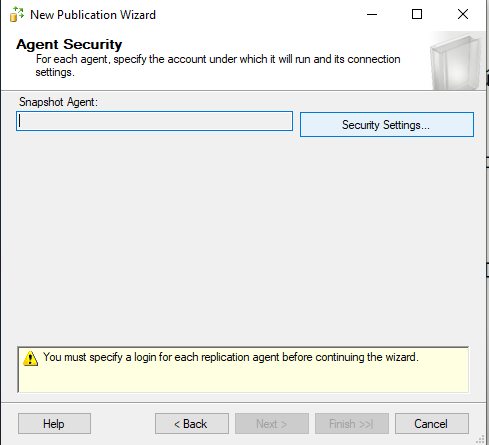
Click vào table cần thiết lập khóa phụ và Add 🡪 Add Join to Extend



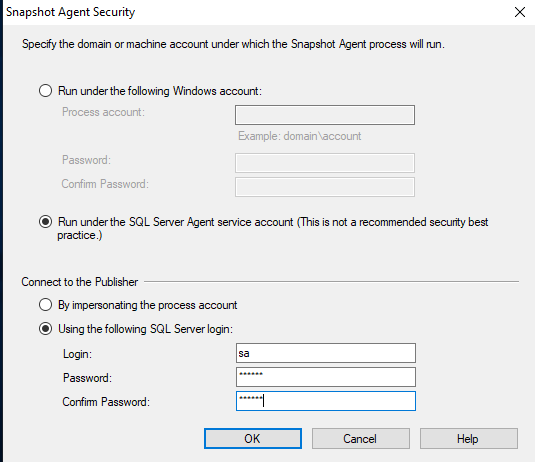
Sau khi thiết lập xong các liên kết thì Next



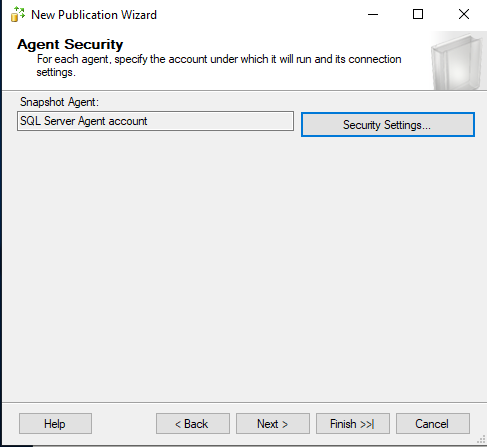
Click Next



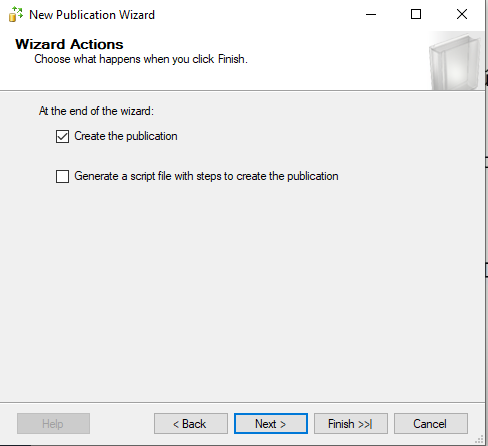
Ấn Security Settings



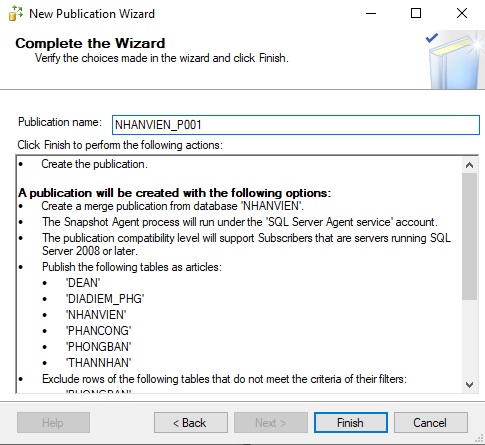
Nhập tài khoản Login, Pass và click OK



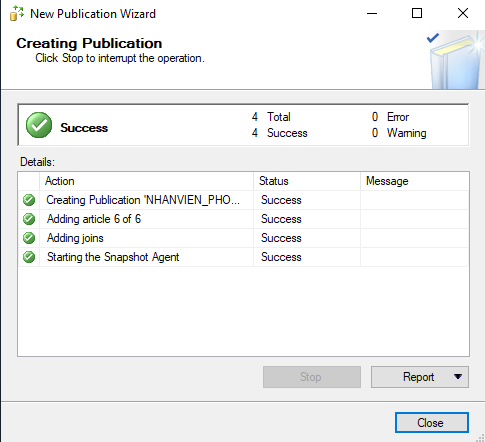
Click Next



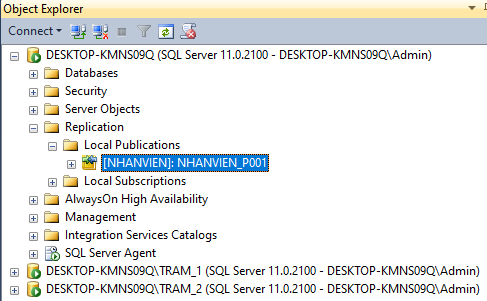
Click Next



Click Finish



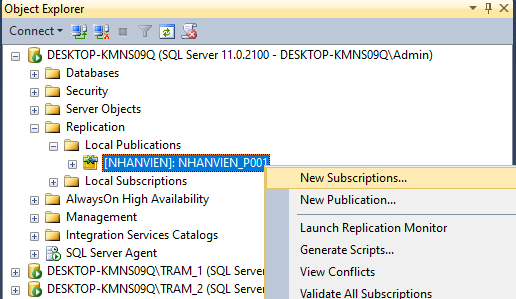
Chờ thiết lập và Close



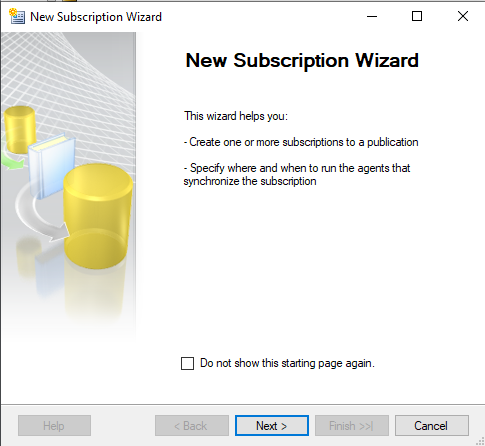
Kết quả khi thiết lập xong 1 Local Publications

**Bước 2**: Tạo Subscription

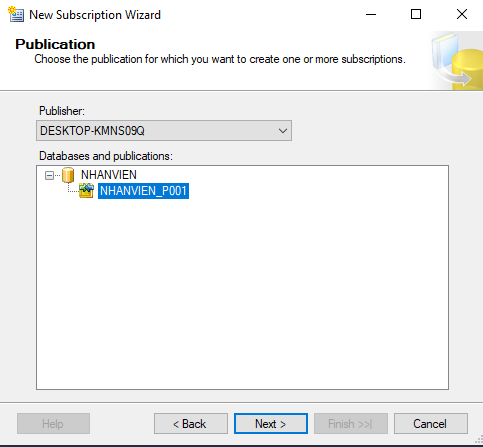
Mục đích của việc tạo subscription là chỉ định server con nào sẽ chứa publication đã tạo ở bước 1 – chứa dữ liệu phân tán theo lược đồ phân mảnh



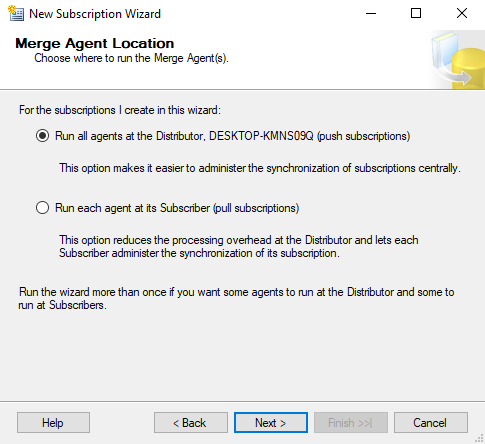
Chuột phải vào Local Publications cần tạo Sub 🡪 New Subscriptions



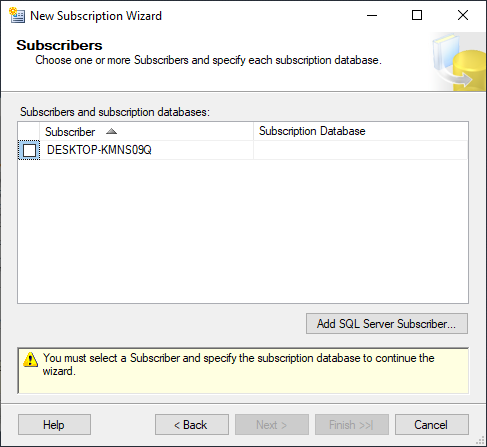
Click Next



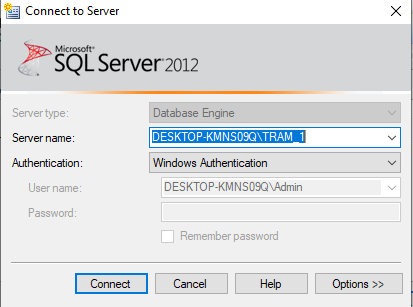
Chọn vào Pulications cần tạo Subscription



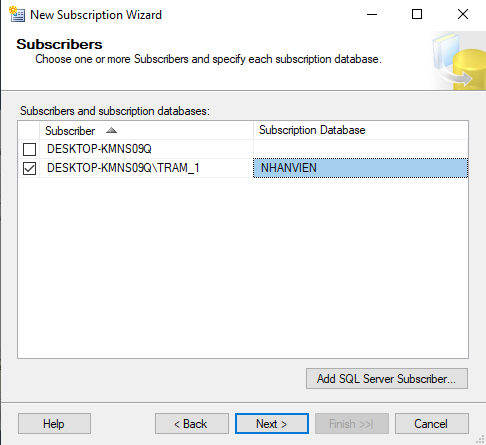
Click Next



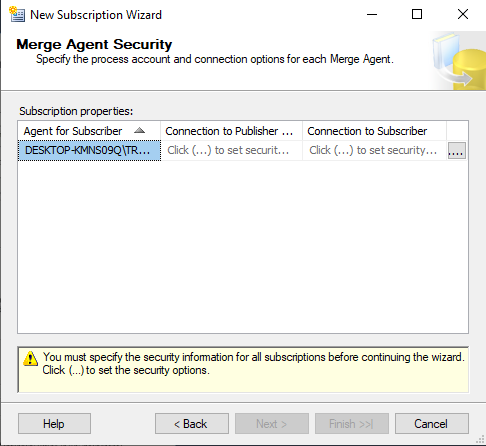
Click Add SQL Server và chọn server con cần thực hiện



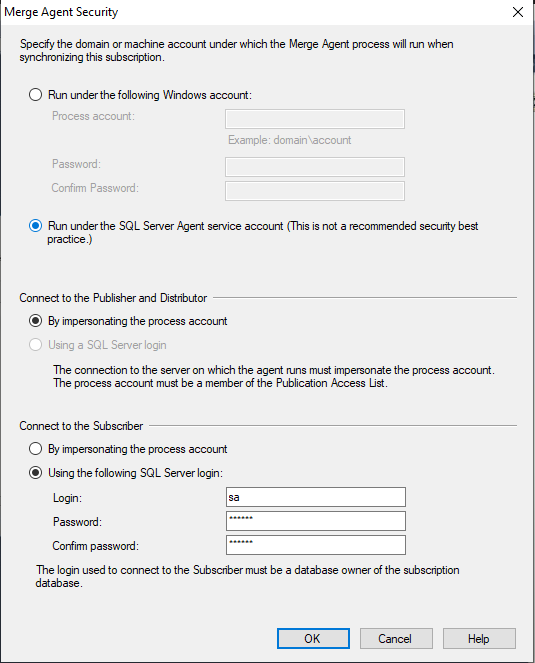
Đăng nhập vào Server con



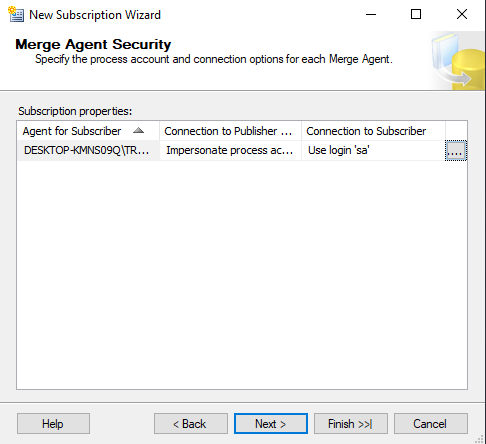
Chọn vào server con và Next



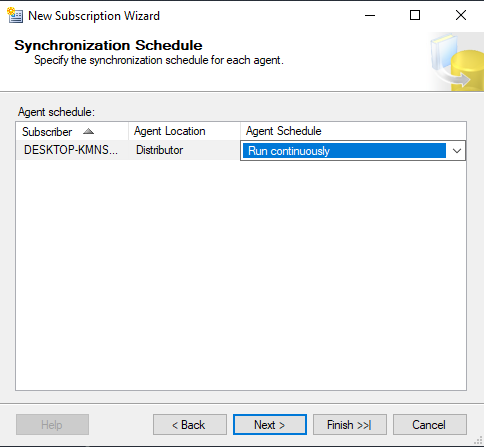
Click vào ….



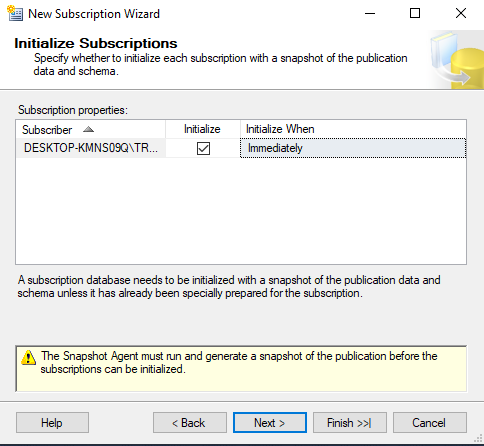
Nhập Login và Pass cho server con và Ok



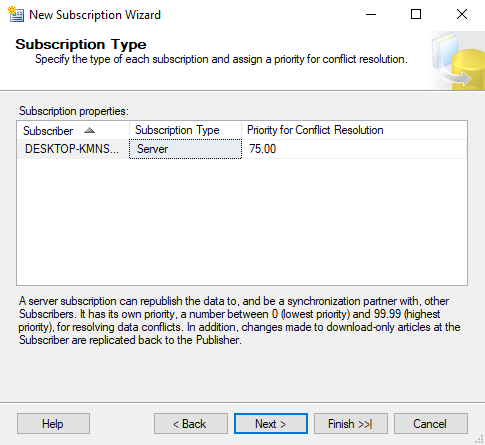
Click Next



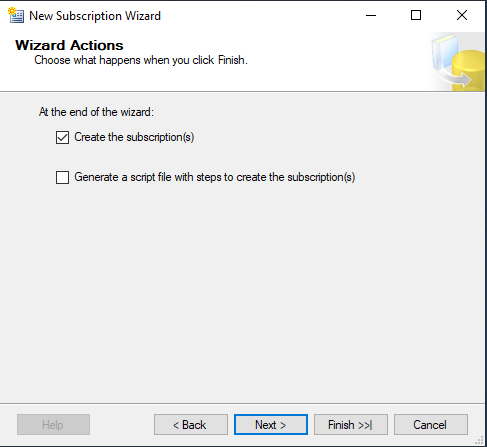
Lựa chọn Run continuously



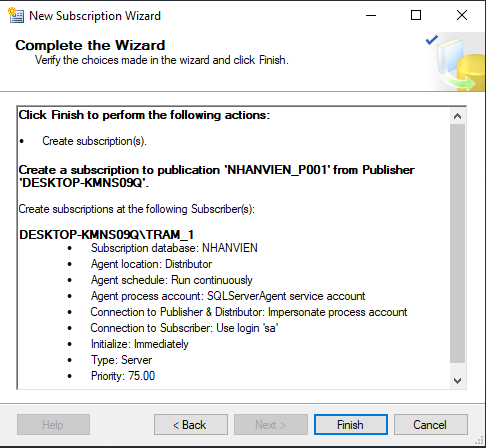
Click Next



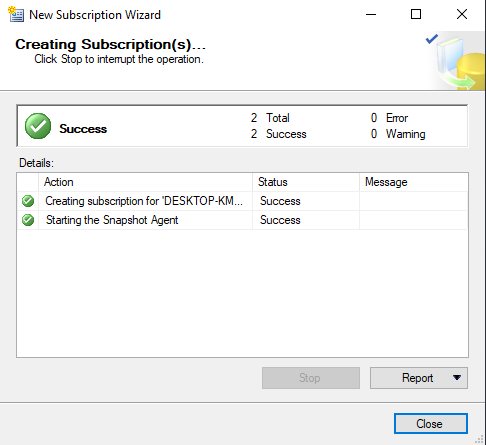
Click Next



Click Next



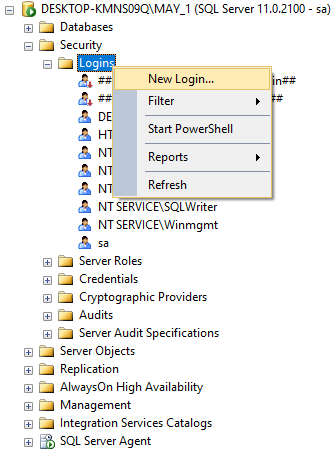
Click Finish



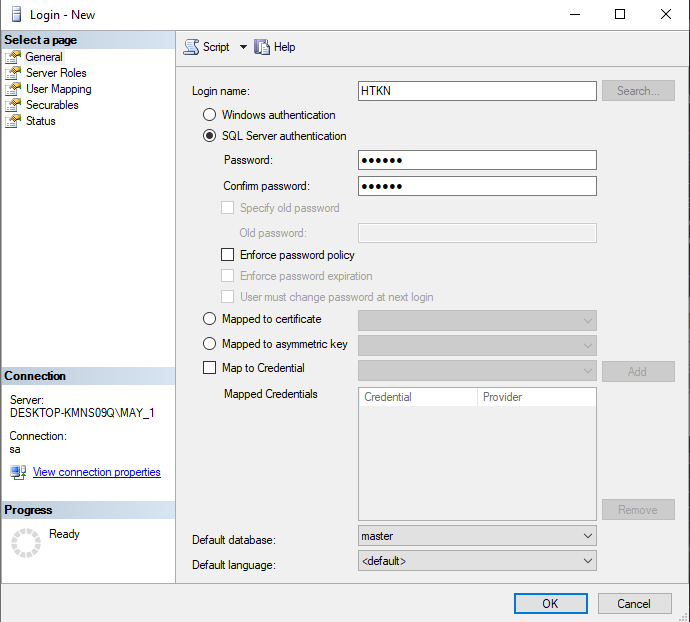
Chờ thiết lập hoàn thành và Close

**Bước 3** : Liên kết 2 Server được phân mảnh dựa vào Link Server

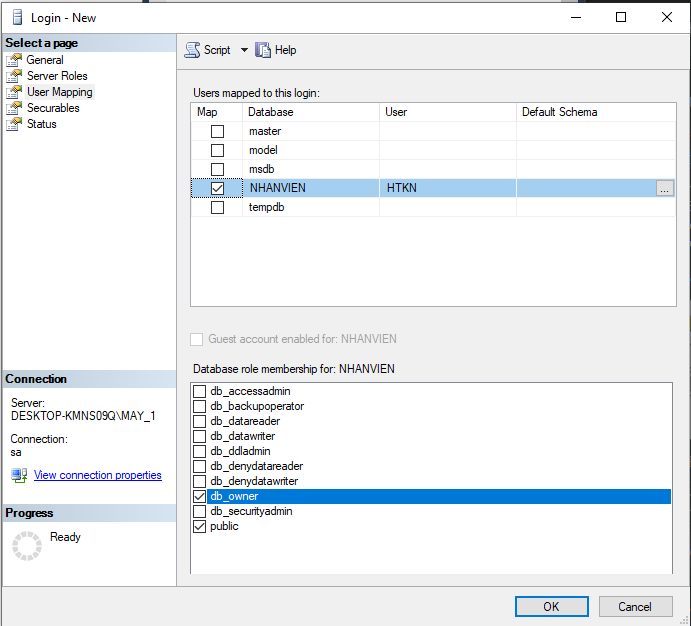
* Tạo 1 Remote Login (HTKN) ở MAY\_1
* Tạo Link Server (LINK) từ MAY\_2 đến MAY\_1
* Và ngược lại



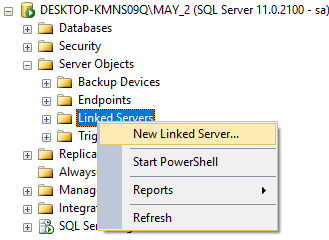
Security 🡪  Logins 🡪 Chuột phải 🡪 New Login



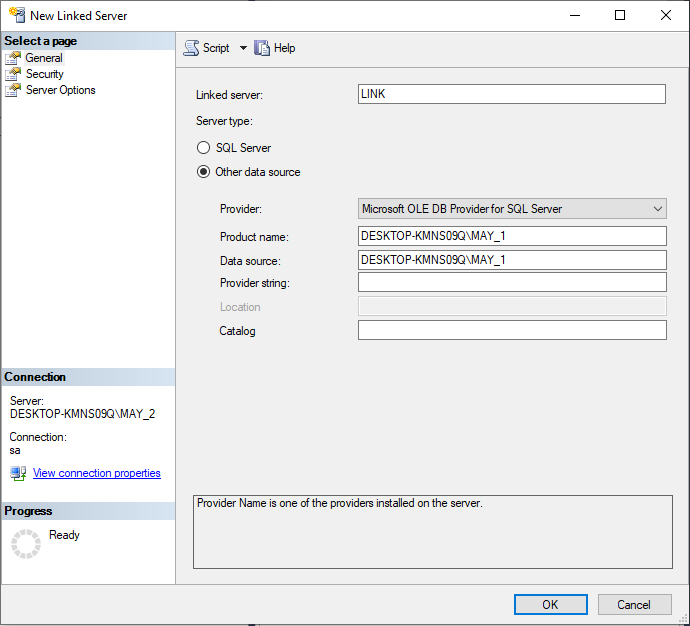
Điền Login name và Password cho MAY\_1



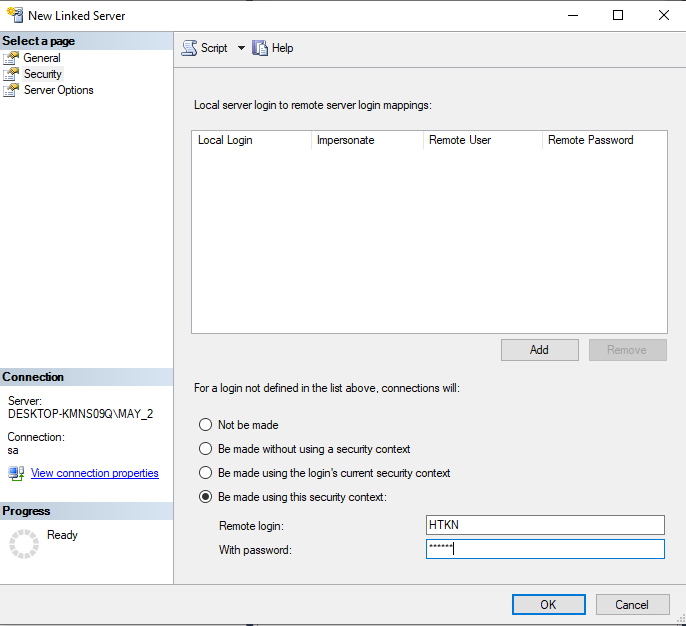
Phân quyền cho database được dùng để đồng bộ 🡪 OK



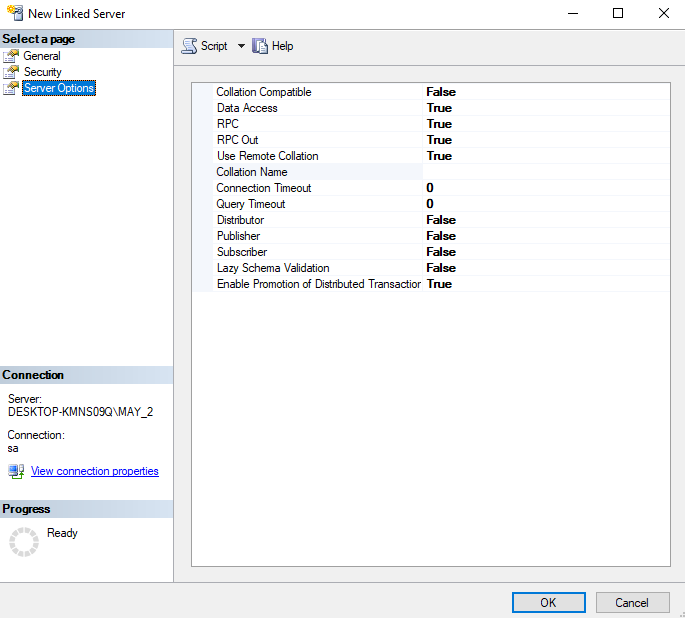
Server Objects 🡪 Linked Servers 🡪 New Linked Server



General : Nhập Linked Server và Product name – Data source của MAY\_1



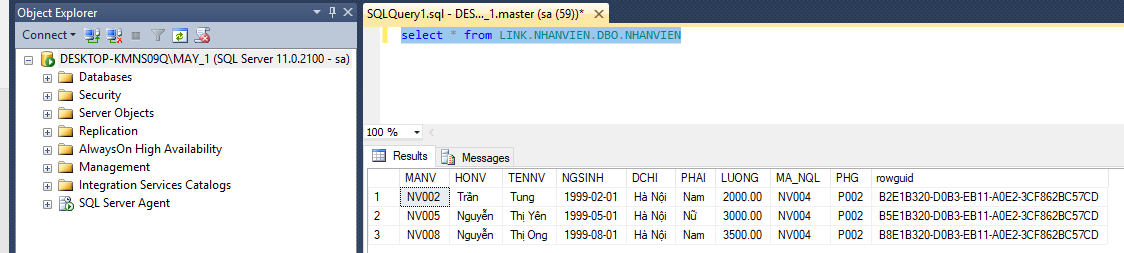
Security : Nhập Remote login và With password



Server Options : Chọn True – False như hình 🡪 OK

**Bước 4** : Kiểm tra đồng bộ giữa 2 server con

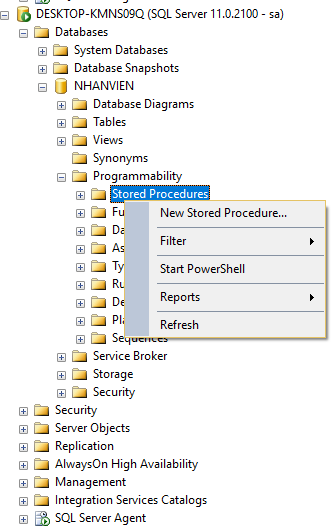
Bằng cách truy vấn nhân viên ở server MAY\_2 từ server MAY\_1



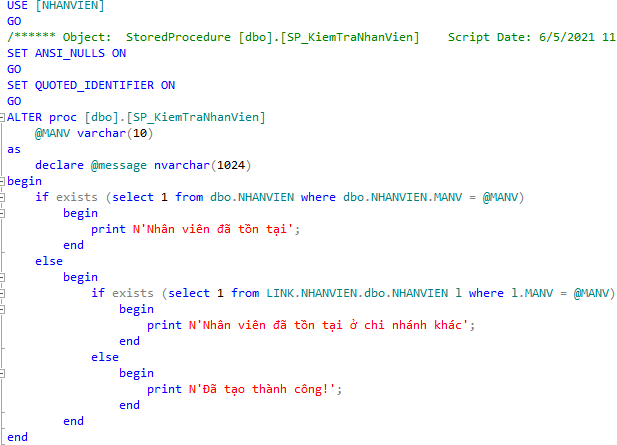
Như vậy bước đầu ta đã thành công khi liên kết các phân mảnh lại với nhau .

**\*Vì lý do tránh trùng lặp và thực hiện thành công khi đồng bộ dữ liệu từ các server mảnh về server gốc .Vì vậy ta phải viết câu lệnh kiểm tra tồn tại nhân viên hay chưa,vì câu lệnh kiểm tra sẽ được xử lý liên tục trong quá trình trình thực thi câu lệnh ,ta sẽ viết nó vào trong STORE PROCEDURE để có thể tái sử dụng nó một cách tối ưu nhất**

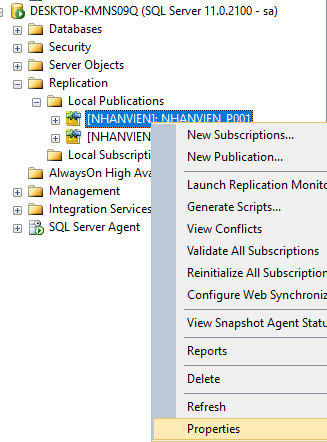
**Bước 5** : Tạo Store Procedure :



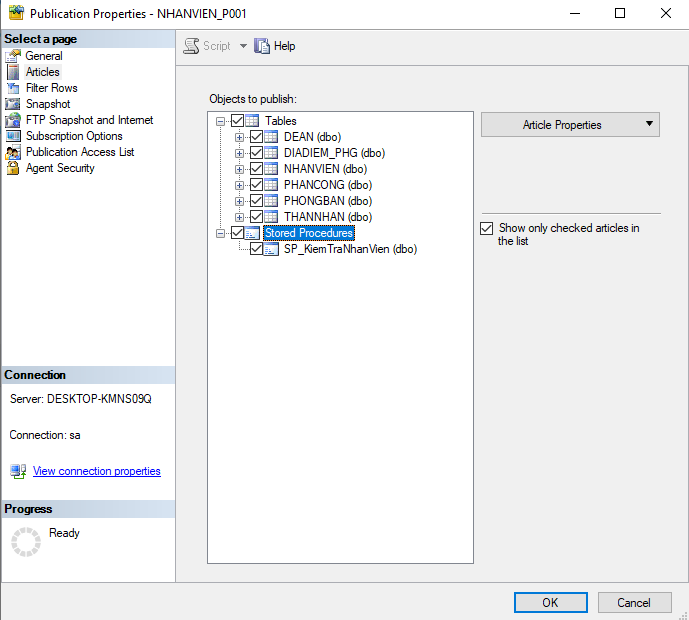
Database 🡪 ‘Name DB’ (NHANVIEN) 🡪 Programmability 🡪 Stored Procedures 🡪 New



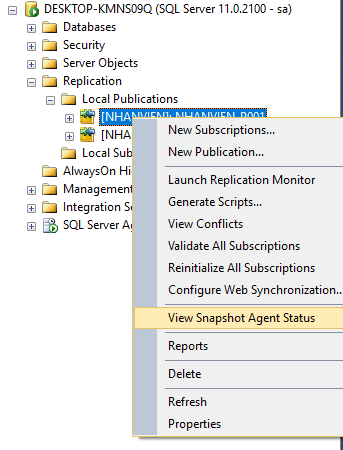
Khởi tạo Store Procedure để kiểm tra tồn tại MANV



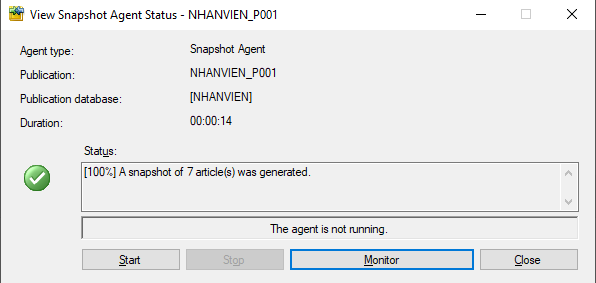
Local Publications 🡪 ‘Name Local Publications’ (NHANVIEN\_P001) 🡪 Properties



Articles 🡪 lựa chọn SP cần được đồng bộ cho server con 🡪 OK



Local Publications 🡪 ‘Name Local Publications’ (NHANVIEN\_P001) 🡪 View Snapshot



Start và đợi sau khi chạy xong kết quả.