

Kiến trúc nhân bản(Replication)

Nhân bản là một kĩ thuật quan trọng và hữu hiệu trong việc phân bố cơ sở dữ liệu (CSDL) và thực thi các Stored procedure. Kĩ thuật nhân bản trong SQL Server cho phép bạn tạo ra những bản sao dữ liệu giống hệt nhau, di chuyển các bản sao này đến những vùng khác nhau và đồng bộ hoá dữ liệu một cách tự động để tất cả các bản sao có cùng giá trị dữ liệu. Nhân bản có thể thực thi giữa những CSDL trên cùng một server hay những server khác nhau được kết nối bởi mạng LANs, WANs hay Internet.

SQL Server đã đưa ra nhiều cơ chế nhân bản để đáp ứng các yêu cầu khác nhau của ứng dụng. Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập "Site" và sự nhất quán các giao dịch.

Mục tiêu chính của nhân bản

SQL Server đã đưa ra nhiều cơ chế nhân bản để đáp ứng các yêu cầu khác nhau của ứng dụng. Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập "Site" và sự nhất quán dữ liệu.

Nhất quán dữ liệu (Data consistency)

Có 2 cách để đạt được tính nhất quán dữ liệu:

- Nhất quán giao dịch(Transaction Consistency)
- Hội tụ dữ liệu (Data Convergence)

Nhất quán giao dịch

- Bảo đảm tất cả dữ liệu giống nhau tại mọi site ở bất kì thời điểm.
- Tất cả giao dịch thực hiện tại một site duy nhất.

Có 2 loại :

Nhất quán lập tức (Immediate Transactional Consistency hay Tight Consistency):

Ở kiểu này, tất cả các site được bảo đảm là luôn thấy cùng giá trị dữ liệu tại cùng một thời điểm. Cách duy nhất để đạt được nhất quán giao dịch (transactional consistency) trong môi trường cập nhật phân tán (distributed update environment) là sử dụng 2-phase commit protocol giữa tất cả site tham gia (participating site). Mỗi site phải commit đồng thời mọi thay đổi hoặc không site nào commit những thay đổi. Giải pháp này rõ ràng không khả thi khi số lượng site quá lớn.

Nhất quán ngầm (Latent Transactional Consistency hay Loose Consistency) :

Có một sự nhất quán ngầm giữa các site tham gia do có một sự trì hoãn trong việc phản ánh các giá trị dữ liệu đến các site tham gia và vào lúc này các site không bảo đảm có cùng giá trị dữ liệu. Việc sửa đổi các giá trị dữ liệu có thể bị trì hoãn đủ lâu để tất cả các site cùng cập nhật, sau đó tất cả các site sẽ có cùng giá trị dữ liệu. Ngoài ra các giá trị dữ liệu này cũng phải giống với những giá trị đạt được khi thực hiện các công việc tại một site. Sự khác nhau duy nhất giữa nhất quán giao dịch lập tức và nhất quán giao dịch ngầm là dữ liệu có nhất quán tại cùng một lúc hay không.

Hội tụ dữ liệu

Với sự hội tụ dữ liệu, tất cả các site có thể quy về cùng một giá trị dữ liệu nhưng không nhất thiết là giá trị dữ liệu này bị gây ra bởi những tác vụ được làm trên một site duy nhất. User có thể tự do thao

tác trên các site theo các cách khác nhau. Khi các nút (node) đồng bộ, tất cả các site sẽ hội tụ về cùng một giá trị.

Nếu độ đồng bộ gây ra bởi sự sửa đổi cùng một dữ liệu tại những site khác nhau thì những sửa đổi này sẽ được giải quyết một cách tự động (chọn site có độ ưu tiên cao hơn hay site đưa sửa đổi đến trước...).

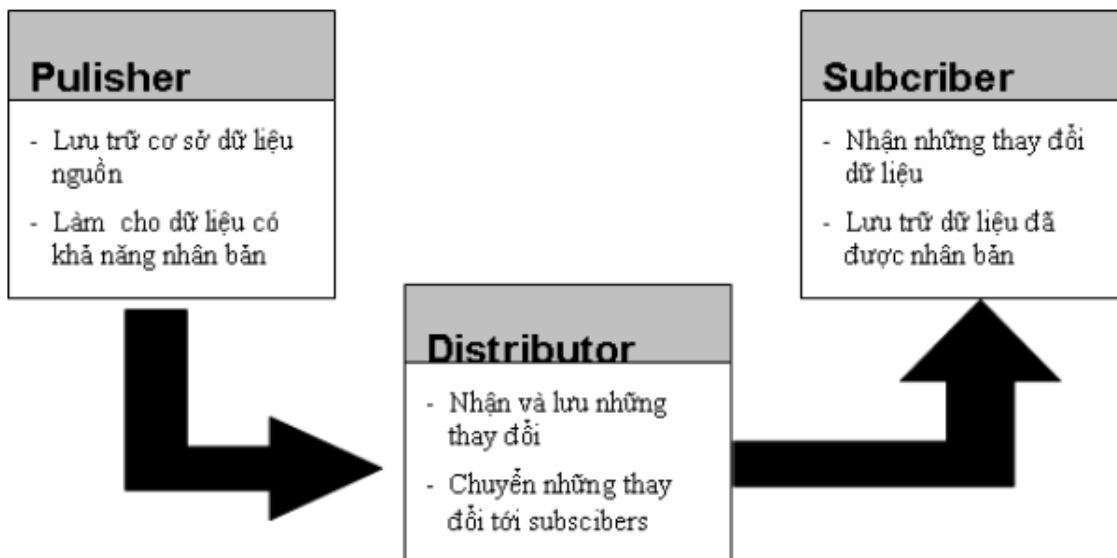
Độc lập site (site autonomy)

Độc lập site xét đến ảnh hưởng của những thao tác trên một site đến các site khác. Thường độc lập site càng tăng thì tính nhất quán dữ liệu giảm. Nhân bản kết hợp (Merge replication) có mức độc lập site cao nhất, tạo ra sự hội tụ nhưng lại không đảm bảo tính nhất quán dữ liệu. 2PC (two phase commit) có tính nhất quán dữ liệu cao nhưng lại không có tính độc lập site. Những giải pháp khác thì thường ở giữa hai tính này.

Kiến trúc nhân bản

Các thành phần chính của nhân bản:

- Publisher: Là một server tạo dữ liệu để nhân bản đến các server khác. Nó xác định dữ liệu nào được nhân bản, dữ liệu nào thay đổi và duy trì những thông tin về các công bố tại site đó.
- Subscriber: Là một server lưu giữ nhân bản và nhận các tác vụ cập nhật. SQL Server 2000 cho phép Subscriber cập nhật dữ liệu nhưng quá trình cập nhật ở Subscriber không giống như ở Publisher. Một Subscriber có thể là một Publisher của các Subscriber khác.
- Distributor: Là một server mà chứa CSDL phân tán (distribution database) và lưu trữ metadata, history data và transaction. SQL Server sử dụng CSDL phân tán để lưu và chuyển (store_and_forward) dữ liệu nhân bản từ Publisher đến các Subscriber. Có 2 loại Distributor : Local Distributor và remote Distributor.
- Publication: Đơn giản là một tập hợp các mẫu dữ liệu (article). Một mẫu là một nhóm dữ liệu được nhân bản. Một mẫu có thể bao gồm một table hay chỉ là một vài hàng (horizontal fragment) hay cột (vertical fragment). Một Publication thường gồm nhiều mẫu



Chiều di chuyển dữ liệu

Có 2 kiểu di chuyển dữ liệu:

Push subscription

- Publisher đẩy (push) những thay đổi đến Subscriber mà không quan tâm Subscriber có cập nhật hay không.
- Push subscription được sử dụng trong những ứng dụng mà yêu cầu gửi những thay đổi đến Subscriber ngay khi những thay đổi này xảy ra ở Publisher.
- Push Subscription giúp việc quản lý các Subscriber đơn giản và tập trung hơn, đồng thời giúp bảo mật tốt hơn vì quá trình khởi động (initialization process) sẽ được quản lý tại một chỗ . Nhưng vì thế, Distributor có thể phải đảm nhận nhiều quá trình phân bổ subscription đến các Subscriber cùng một lúc . Điều này dễ dẫn đến hiện tượng tắc cổ chai (bottleneck) .
- Mô hình này không thích hợp khi số lượng các Subscriber trở nên quá lớn.
- Push subscription gây ra 1 phí xử lý cao hơn tại Publisher. Để tránh hiện tượng này, những thay đổi có thể được đẩy đến Subscriber theo một lịch định kì.

Pull subscription

- Subscriber kéo (pull) những thay đổi tại Publisher về theo một khoảng thời gian định kì.
- Tốt cho những user độc lập thay đổi bởi vì chúng cho phép user xác định khi nào thì những thay đổi dữ liệu được đồng bộ
- Ngược với push subscription ,pull subscription bảo mật thấp nhưng cho phép số lượng Subscriber cao hơn .
- Một publication có thể sử dụng cả hai push và pull subscription.

Tác nhân (Agent)

Việc thiết kế các nhân bản có thể tạo ra 1 hay nhiều agent.

Snapshot agent:

- Chuẩn bị lược đồ, data file, stored procedure
- Lưu snapshot lên Distributor và ghi lại những thông tin về trạng thái đồng bộ vào CSDL phân bổ (distribution database) .
- Mỗi publication có 1 snapshot agent riêng chạy trên Distributor và liên kết với Publisher.

Log Reader agent:

- Di chuyển những transaction cần nhân bản từ transaction log trên Publisher đến CSDL phân bổ .
- Mỗi publication dùng nhân bản transaction có một log reader agent, chạy trên Distributor và liên kết (connect) đến Publisher.

Distribution Agent:

- Di chuyển transaction và những tác vụ sao chép giữ trong CSDL phân bổ đến Subscriber.
- TH: Nhân bản transaction hay snapshot mà đồng bộ lập tức (immediate synchronization): khi 1 push subscription được tạo, mỗi publication có 1 distribution agent riêng, chạy trên Distributor và liên kết với Subscriber.
- TH: Nhân bản transaction và snapshot không đồng bộ lập tức : Publisher và Subscriber sẽ dùng chung distribution agent , chạy trên Distributor và liên kết với Subscriber.

- TH: pull subscription đến snapshot publication hay transactional publication: có distribution agent, chạy trên Subscriber
- Nhân bản kết hợp (merge replication) không có distribution agent.

Merge agent:

Di chuyển và điều hòa những thay đổi dữ liệu xảy ra sau khi 1 snapshot khởi động (initial snapshot) được tạo. Mỗi merge publication có một merge agent, liên kết và cập nhật được với cả hai Publisher và Subscriber.

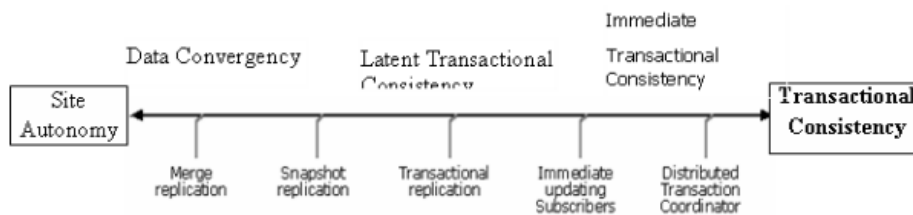
Các loại nhân bản

Trong thực tế khó có thể có được một loại nhân bản phù hợp mọi yêu cầu. Công việc kinh doanh thường đòi hỏi nhiều ứng dụng khác nhau vì thế SQL Server đã đưa ra nhiều cách thức nhân bản để đáp ứng các yêu cầu đó.

SQL Server đưa ra 3 loại nhân bản để sử dụng khi thiết kế ứng dụng:

- Nhân bản snapshot
- Nhân bản transaction

Immediate Transactional Consistency-Nhân bản kết hợp



Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập site và sự nhất quán dữ liệu.

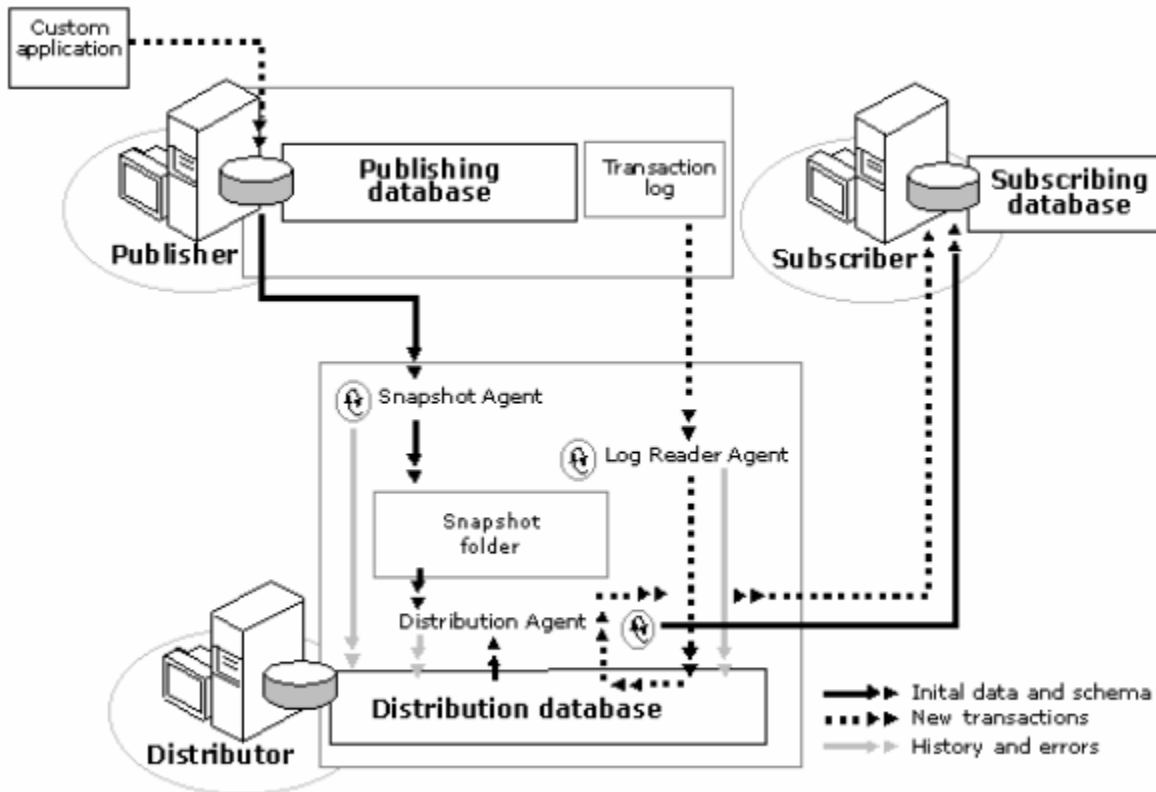
Mỗi loại cung cấp các khả năng và thuộc tính khác nhau nhằm đạt đến mục tiêu của tính độc lập site và sự nhất quán dữ liệu.

Nhân bản snapshot(Snapshot replication)

Giới thiệu

Nhân bản snapshot là loại nhân bản đơn giản nhất, nhân bản snapshot sao chép toàn bộ dữ liệu cần nhân bản (còn gọi là quá trình làm tươi dữ liệu) từ Publisher đến các Subscriber. Nó đảm bảo sự nhất quán tiềm ẩn (Latent Transactional Consistency) giữa Publisher và Subscriber. Nhân bản snapshot được đánh giá cao trong các ứng dụng chỉ đọc như tìm kiếm hay các hệ thống không yêu cầu dữ liệu mới nhất và dung lượng dữ liệu không lớn.

Nhân bản Snapshot gửi tất cả dữ liệu đến cho Subscriber thay vì chỉ gửi những thay đổi. Nếu mẫu dữ liệu rất lớn nó phải cần đến hệ thống mạng đủ mạnh để truyền dữ liệu. Khi sử dụng nhân bản snapshot cần phải tính đến tỉ lệ giữa kích cỡ của toàn bộ dữ liệu và những thay đổi của nó.



Nhân bản snapshot(Snapshot replication)

Nhân bản snapshot(Snapshot replication)

Tác nhân (agent)

Cập nhật snapshot được thực hiện bởi snapshot agent và distribution agent. Snapshot agent chuẩn bị những snapshot file (*snapshot file là file sao chép lược đồ và dữ liệu của những table phân bố*) chứa lược đồ và dữ liệu của những table phân bố, lưu những file này vào snapshot folder trên Distributor và ghi lại những công việc đồng bộ trong CSDL phân bố (distribution database). Distribution agent gửi những snapshot job (tác vụ sao chép dữ liệu) giữ trong bảng dữ liệu phân bố đến Subscriber.

CSDL phân bố (distribution database) chỉ được sử dụng trong nhân bản, không chứa user table.

Snapshot agent

Snapshot agent thực hiện theo các bước sau:

- Thiết lập một **share-lock** lên tất cả table (article) trong publication. Share-lock ngăn không cho các user khác cập nhật lên table đó.
- Sao chép lược đồ dữ liệu của mỗi article (*.sch file*) và các index, các ràng buộc (*.idx file*) lên Distributor. Các file này được lưu vào 1 thư mục con trong thư mục làm việc của Distributor.
- Nếu tất cả các Subscriber đều là MS SQL Server thì bản sao của dữ liệu được lưu thành **.bcp file**. Nếu các Subscriber không đồng nhất (các Subscriber chứa nhiều loại CSDL khác nhau , ví dụ: Access, Oracle...) thì bản sao của dữ liệu được lưu thành **.txt file**.

- Cuối cùng agent gỡ bỏ **share-lock** trên mỗi table phân bố và hoàn tất việc viết vào 1 log history file (*log history file ghi lại quá trình làm việc của các agent*).

Distribution agent

Tác nhân áp dụng những lược đồ và những dữ liệu vào CSDL của Subscriber. Nếu Subscriber không phải là SQL Server, distribution agent sẽ chuyển đổi kiểu dữ liệu trước khi những dữ liệu này được áp dụng vào Subscriber.

Publisher sử dụng SQL Server, Subscriber sử dụng Oracle. Trước khi những dữ liệu được áp dụng lên Subscriber, nó sẽ được chuyển đổi kiểu từ SQL Server sang Oracle.

Snapshot có thể được áp dụng khi subscription được tạo hay theo 1 khoảng thời gian nhất định.

Nhân bản giao dịch (transactional replication)

Giới thiệu

Sử dụng nhân bản giao dịch để nhân bản hai kiểu đối tượng khác nhau: table và stored procedure. Bạn có thể chọn tất cả hay một phần của một table được nhân bản như là một article trong publication. Tương tự, bạn cũng có thể chọn một hay nhiều stored procedure được nhân bản như là một article trong cùng hay khác publication.

Nhân bản giao dịch sử dụng transaction log để giữ những thay đổi được làm trên dữ liệu trong một article. SQL Server giám sát những lệnh insert, update, delete hay những sửa đổi trên dữ liệu và lưu những thay đổi đó lên CSDL phân bố (distribution database). Những thay đổi đó sẽ được gửi đến Subscriber và tuân theo một trật tự nhất định.

Với nhân bản giao dịch, bất cứ yếu tố dữ liệu nào cũng có một publication. Những thay đổi được làm tại Publisher tiếp tục chảy đến một hay nhiều các Subscriber hay theo những khoảng thời gian định trước.

Tác nhân (agent)

Nhân bản giao dịch được thực hiện bởi Snapshot agent, Log Reader agent và Distribution agent. Log Reader agent giám sát transaction log của mỗi CSDL được thiết lập để nhân bản và sao chép những transaction cần nhân bản từ transaction log vào CSDL phân bố (distribution database) . Distribution agent di chuyển những transaction và những tác vụ khởi tạo snapshot được giữ trong table của CSDL phân bố.

Snapshot agent

Trước khi một Subscriber mới có thể nhận những thay đổi từ Publisher, nó phải chứa những table có cùng lược đồ và dữ liệu với những table tại Publisher. Quá trình copy toàn bộ publication từ Publisher qua Subscriber được gọi là **initial snapshot**. Việc nhân bản những dữ liệu thay đổi chỉ xảy ra sau khi nhân bản giao dịch chắc chắn rằng Subscriber có snapshot (bản sao của những lược đồ và dữ liệu). Khi những snapshot đó được phân bố và áp dụng lên các Subscriber thì chỉ những Subscriber chờ để khởi tạo snapshot mới bị ảnh hưởng. Những Subscriber khác ứng với publication đó mà nhận insert, delete, update hay những thay đổi dữ liệu rồi thì không bị ảnh hưởng. Những hàm mà Snapshot agent thực thi để khởi tạo snapshot trong nhân bản giao dịch tương tự như các hàm được sử dụng trong nhân bản Snapshot.

Log Reader agent

Log reader agent chạy tiếp tục hay theo một khoảng thời gian xác định mà bạn thiết lập vào lúc publication được tạo. Khi thực thi, đầu tiên Log reader agent đọc transaction log của publication và xác định lệnh (insert, delete, update) hay những sửa đổi làm trên dữ liệu được đánh dấu nhân bản. Kế tiếp agent sao chép những transaction đó vào CSDL phân bố tại Distributor. CSDL phân bố (distribution database) trở thành hàng lưu và đẩy (store-and-forward queue) những thay đổi dữ liệu đến Subscriber. Chỉ có commit transaction mới được gửi đến CSDL phân bố.

Có sự tương ứng 1-1 giữa transaction trên Publisher và transaction được nhân bản trong CSDL phân bố. Một transaction có thể bao gồm nhiều lệnh. Sau khi toàn bộ transaction được viết vào CSDL phân bố một cách thành công, nó sẽ được commit. Sau đó những transaction này sẽ được đẩy đến các Subscriber. Cuối cùng, agent đánh dấu những hàng (row) đã được công bố đến Subscriber trong transaction log để sẵn sàng loại bỏ. Điều này đảm bảo những hàng (row) còn chờ để nhân bản sẽ không bị loại bỏ. Vì thế, transaction log trên Publisher có thể được đổ xuống mà không cản trở việc nhân bản bởi vì chỉ những transaction bị đánh dấu mới bị loại bỏ.

Log read agent thực thi trên Distributor.

Distribution agent

Những transaction được lưu trong CSDL phân bố cho đến khi distribution agent “đẩy” chúng đến tất cả các Subscriber (hoặc một distribution agent tại Subscriber “kéo” những thay đổi về). CSDL phân bố chỉ được sử dụng bởi nhân bản và không chứa bất cứ user table. Trong bất kì trường hợp nào bạn cũng không nên tạo những object khác vào trong CSDL phân bố. Những tác vụ làm thay đổi dữ liệu tại Publisher sẽ chảy đến Subscriber và Subscriber sẽ thay đổi dữ liệu theo cùng cách chúng được thay đổi tại Publisher. Điều này đảm bảo rằng các Subscriber sẽ nhận những transaction theo một trật tự như là chúng được làm tại Publisher .

Những hàm mà distribution agent sử dụng để di chuyển những lệnh đến Subscriber cũng tương tự như những hàm được sử dụng trong nhân bản snapshot.

Thu dọn trong nhân bản transaction

Tương tự cho nhân bản snapshot.

Khi CSDL phân tán(distribution database) được tạo, SQL Server sẽ tự động thêm vào 3 tác vụ tại Distributor:

- Agent cleanup
- Transaction cleanup
- History cleanup

Những tác vụ này giúp cho việc nhân bản hiệu quả hơn trong môi trường long_running. Tác vụ cleanup giữ lại những transaction của mỗi publication trong một giai đoạn xác định sau khi tất cả Subscriber đã nhận chúng. Trong suốt giai đoạn này những transaction sẽ được giữ trong CSDL phân bố. Thiết lập một giai đoạn giữ lại kết hợp với lịch backup có thể được sử dụng khôi phục CSDL đích một cách tự động khi xảy ra sự cố.

Các dạng nhân bản giao dịch

Có hai dạng:

Cập nhật Subscriber lập tức(Immediate_Updating Subscriber)

Trong trường hợp đơn giản nhất, cả hai nhân bản snapshot và giao dịch làm việc dựa trên mô hình nhân bản một chiều (dữ liệu chỉ được nhân bản từ Publisher đến Subscriber). Tuy nhiên MS SQL

Server cung cấp thêm một mô hình mới cho phép Subscriber sửa đổi dữ liệu nhân bản, tùy chọn Immediate Updating Subscriber sẽ cung cấp sự nhất quán giao dịch ngầm (latent transactional consistency) giữa các Subscriber (những sửa đổi sẽ xảy ra lập tức giữa Subscriber thực hiện tác vụ cập nhật và Publisher) mà không yêu cầu cập nhật chỉ được làm tại Publisher site. Tùy chọn này được thiết lập vào lúc publication được tạo và cho phép Subscriber cập nhật bản sao dữ liệu cục bộ của nó và những thay đổi đó sẽ được phản ánh lập tức đến Publisher bằng cách sử dụng two-phase commit protocol (2PC). 2PC được quản lý bởi Microsoft Distributed Transaction Coordinator (MS DTC). Nếu cập nhật có thể được thực hiện bằng 2PC thì Publisher sẽ phổ biến những thay đổi đến tất cả các Subscriber khác theo lịch làm việc của distribution agent (hay là theo lần làm tươi snapshot kế tiếp). Bởi vì Subscriber gốc đã cập nhật những thay đổi dữ liệu rồi, nên user có thể tiếp tục làm việc với những dữ liệu đó và đảm bảo những thay đổi đó sẽ được cập nhật tại Publisher. Mô hình này đảm bảo tính toàn vẹn giao dịch.

Với mô hình này, tính độc lập site sẽ giảm, nhưng vẫn cao hơn khi tất cả các Subscriber cập nhật lập tức.

Tùy chọn Immediate-updating Subscribers được hỗ trợ sử dụng những công cụ :

- Triggers
- Stored procedures
- Microsoft Distributed Transaction Coordinator (MS DTC)
- Phát hiện tranh chấp
- Phát hiện Loopback

Triggers:

Triggers tại Subscriber theo dõi các transaction và báo về cho các publication bằng cách sử dụng những stored procedure gọi từ xa (remote stored procedure call) trong 2-phase commit protocol (2PC) mà được điều khiển bởi MS DTC. Trigger sẽ thực hiện các công việc:

- Trích giá trị từ những bảng insert hay delete
- Gọi lệnh BEGIN DISTRIBUTED TRANSACTION
- Thực thi một remote procedure để gọi stored procedure thích hợp tại Publisher, thông qua những giá trị từ những bảng insert hay delete.
- Điều khiển cập nhật giá trị timestamp & identity mới tại Subscriber
- Nếu gọi những stored procedure từ xa thành công, thì commit transaction phản ánh chính xác cùng những thay đổi tại cả hai Publisher và Subscriber. Sau đó, Publisher bảo đảm rằng những thay đổi được phổ biến đến tất cả các Subscriber khác. Ngược lại nếu không được Subscriber sẽ hủy bỏ (rollback) giao dịch và trả về những lỗi cho user.

Stored procedures:

Stored procedure tại Publisher thực thi những giao dịch khi giao dịch đó không tranh chấp với những thay đổi được làm tại Publisher. Nếu một tranh chấp được phát hiện, giao dịch bị hủy bỏ (rollback) ở cả hai site. Mỗi article có ba hàm insert, delete, update. Mỗi hàm tại Publisher sẽ thực hiện các công việc sau:

- Insert procedure : Cố gắng insert hàng (row). Sau đó kiểm tra giá trị @@ROWCOUNT/@@ERROR và trả về tín hiệu thành công hay sự cố cho lời gọi trigger đó của Subscriber.
- Delete procedure : Cố gắng xóa hàng. Sau đó kiểm tra giá trị @@ROWCOUNT/@@ERROR và trả về tín hiệu thành công hay sự cố cho lời gọi trigger đó của Subscriber.

- Update procedure : cố gắng cập nhật hàng có cùng giá trị khoá và timestamp với hàng trong bảng delete. Kiểm tra @@ROWCOUNT / @@ERROR. Nếu thành công, trả về một giá trị timestamp mới.

SQL Server tổ chức 2 bảng (table) : delete và insert để lưu những dữ liệu thay đổi được làm trên một bảng (table) mà chưa được commit. Thực tế lệnh update bao gồm 1 hàng trong bảng insert và 1 hàng trong bảng delete.

Một giao dịch mà ảnh hưởng lên nhiều hàng thì giao dịch đó chỉ được commit khi tất cả các hàng đều được cập nhật lên cả 2 site .

MS DTC (Microsoft Distributed Transaction Coordinator):

MS DTC quản lý những tác vụ 2-phase commit giữa một Subscriber và Publisher trong một lệnh gọi từ xa (BEGIN DISTRIBUTED TRANSACTION trong Transact-SQL).

Phát hiện tranh chấp:

Những stored procedure của Publisher sử dụng cột timestamp để phát hiện một hàng có thay đổi hay không sau khi nó được nhân bản đến Subscriber. Khi Subscriber yêu cầu một giao dịch cập nhật lập tức (immediate-update transaction), nó gửi giá trị timestamp đến Publisher với tất cả những cột khác trong hàng. Khi đó stored procedure của Publisher so sánh giá trị này với giá trị timestamp hiện tại của hàng. Nếu giá trị này giống nhau thì hàng không được sửa đổi sau khi nó được nhân bản đến Subscriber và vì thế giao dịch này được chấp nhận.

***Timestamp** là một giá trị tự động tăng và duy nhất trong một cơ sở dữ liệu.*

Phát hiện loopback :

Nếu một transaction được thực thi thành công lên một Subscriber và Publisher, không cần thiết phải biến những thay đổi trở về cho Subscriber gốc (Subscriber đưa những thay đổi đến Publisher). SQL Server có một cơ chế gọi là phát hiện loopback (loopback detection) để xử lý tình huống này.

Những thông tin sử dụng để phát hiện loopback được lưu thành một transaction bởi cơ sở transaction. Những bảng mà định cư trong những cơ sở dữ liệu khác nhau tại immediate updating Subscriber hay những bảng mà định cư trong những cơ sở dữ liệu khác nhau ở phía bên kia của immediate-updating Subscriber không nên được update trong cùng 1 transaction.

Nhân bản những thực thi của Stored procedure

SQL server không những cho phép bạn nhân bản dữ liệu trong bảng mà còn hỗ trợ bạn nhân bản stored procedure một trong hai cách. Nếu bạn có một hay nhiều stored procedure như là những article trong một snapshot publication, SQL server sao chép toàn bộ store procedure từ Publisher đến Subscriber. Nếu bạn gồm một hay nhiều stored procedure như là article trong một nhân bản giao dịch, SQL Server sẽ nhân bản thực thi của stored procedure hơn là những dữ liệu thay đổi gây ra bởi sự thực thi những stored procedure đó. Cách làm này đặc biệt hữu ích trong nhân bản mà kết quả của những stored procedure có thể ảnh hưởng một số lượng lớn dữ liệu. Nhân bản những thay đổi như thực thi một lệnh đơn làm tăng hiệu quả ứng dụng của bạn.

Có 2 loại:

- Procedure execution
- Serializable Procedure execution

Procedure execution:

Nhân bản procedure execution đến tất cả Subscriber. Điều này xảy ra bất chấp những lệnh đơn trong stored procedure có thành công hay không. Bởi vì những thay đổi dữ liệu được làm bởi stored procedure có thể xảy ra trong nhiều giao dịch, nên dữ liệu tại Subscriber không thể bảo đảm là sẽ nhất quán với dữ liệu tại Publisher.

Serializable Procedure execution:

Chỉ thực hiện nhân bản procedure execution khi procedure được thực thi trong một chuỗi giao dịch tuần tự. Cách này đảm bảo dữ liệu tại Subscriber nhất quán với dữ liệu tại Publisher.

Nhân bản kết hợp (Merge replication)

Giới thiệu

Nhân bản kết hợp có tính độc lập site (site autonomy) cao nhất. Publisher và Subscriber có thể làm việc hoàn toàn độc lập và sẽ kết nối với nhau theo những khoảng thời gian để hội tụ các kết quả lại. Nếu xung đột gây ra bởi các site cùng sửa đổi trên cùng một phần tử dữ liệu thì những xung đột này sẽ được giải quyết một cách tự động. Khi xung đột xảy ra, bộ giải quyết xung đột sẽ chọn site có độ ưu tiên cao hơn hay site sửa đổi dữ liệu đó trước. Các xung đột này có thể được phát hiện và giải quyết theo cấp độ hàng hay cột của bảng dữ liệu.

Nhân bản kết hợp nhận biết những thay đổi trong một CSDL nguồn và đồng bộ những giá trị giữa Publisher và Subscriber. Cả hai Publisher và Subscriber đều có thể cập nhật dữ liệu. Trong nhân bản kết hợp, Publisher là server tạo publication. Mặc dù Publisher tạo publication nhưng nó không tự động "thắng" 1 tranh chấp với 1 Subscriber. "Người thắng cuộc" được xác định bởi tiêu chuẩn do bạn thiết lập và những thay đổi dữ liệu tại CSDL đích sẽ được phổ biến đến CSDL nguồn.

Tác nhân (agent)

Nhân bản kết hợp được thực hiện bởi snapshot agent và merge agent. Snapshot agent chuẩn bị những snapshot file chứa lược đồ và dữ liệu của những table phân bố, lưu những file này vào snapshot folder trên Distributor và ghi lại những công việc đồng bộ trong publication. Merge agent thực hiện những công việc khởi tạo snapshot được tổ chức trong bảng (table) của publication đến Subscriber. Nó cũng kết hợp những dữ liệu thay đổi xảy ra tại Publisher sau khi initial snapshot được tạo và giải quyết tranh chấp theo những luật mà bạn đặt ra hay sử dụng bộ giải quyết tranh chấp (conflict resolver).

Snapshot agent:

Tương tự như Snapshot agent của nhân bản transaction.

Merge agent:

Khi một hàng được cập nhật trong một mẫu (article) một trigger tạo cột generation cho hàng đó và gán nó bằng 0. Khi merge agent được thực thi nó sẽ thu thập tất cả những hàng có generation bằng 0 thành một hay nhiều nhóm và gán cho generation một giá trị lớn hơn tất cả những generation trước đó. Merge agent tại mỗi site sẽ theo dõi generation cao nhất mà nó gửi đến các site khác và các generation cao nhất mà các site khác đã gửi đến nó. Những generation này được lưu trong hàng (row) có thể khác nhau giữa các site bởi vì generation tại một site phản ánh thứ tự những thay đổi được xử lý tại site đó.

Vào lúc đồng bộ, merge agent gửi tất cả những dữ liệu thay đổi đến site khác. Tại CSDL nguồn, những giá trị đến được kết hợp với những giá trị đã tồn tại. Merge agent ước lượng cả hai giá trị dữ liệu đến và hiện có và bất cứ tranh chấp nào giữa hai giá trị cũ và mới cũng được giải quyết một cách tự động dựa theo độ ưu tiên hay user thay đổi dữ liệu trước hay kết hợp giữa hai cách trên (dùng với nhóm site có độ ưu tiên tương đương nhau). Bạn cũng có thể thực thi những chiến lược giải quyết tranh chấp thông qua COM hay bộ giải quyết stored procedure (stored procedure resolver). Những dữ liệu thay đổi được nhân bản đến những site khác chỉ khi một đồng bộ xảy ra và việc đồng bộ này có thể mất vài phút, vài ngày hay thậm chí vài tuần.

Nhân bản kết hợp có tính độc lập site rất cao. Tất cả site đều có thể thực hiện update, delete, insert trên dữ liệu phân bố trên site của nó và độc lập với những thay đổi được làm trên những site khác.

Tuy nhiên nhân bản kết hợp không đảm bảo tính toàn vẹn giao dịch. Thay vì vậy nó đẩy mạnh sự hội tụ dữ liệu. Tất cả những thay đổi được làm tại tất cả các site sẽ hội tụ về cùng một giá trị, mặc dù giá trị đó không đảm bảo là giống nhau như là tất cả những thay đổi đó được áp dụng lên một site. Vì vậy kiểu này không thích hợp cho những ứng dụng yêu cầu toàn vẹn giao dịch.

Merger agent là một công cụ của SQL Server Agent và có thể được quản lý trực tiếp bằng cách sử dụng SQL Server Enterprise Manager. Snapshot agent thực thi trên Distribution. Merge agent thực thi trên Distribution khi dùng push subscription hay thực thi trên Subscriber khi dùng pull subscription.

Giải quyết tranh chấp trong nhân bản kết hợp

MA phát hiện tranh chấp thông qua một cột hệ thống gọi là lineage trong bảng MSmergecontents, đại diện cho quá trình thay đổi trong một hàng. Agent cập nhật cột lineage trong MS mergecontents một cách tự động khi một user cập nhật hàng. Mỗi cột chứa một mục (entry) cho mỗi site mà cập nhật lên hàng. Mỗi entry kết hợp giữa chỉ số (id) của site và version cuối cùng của hàng được tạo bởi site đó. Khi MA kết hợp những thay đổi, và nó đụng phải một hàng mới thay đổi, nó sẽ xem xét cột lineage của hàng trên mỗi site để xác định có tranh chấp hay không? Khi tranh chấp xảy ra, agent khởi động một bộ hoà giải tự động. “Người thắng” tranh chấp có thể dựa theo độ ưu tiên hoặc giải pháp chọn người đến trước nhất hay phương pháp truyền thống sử dụng bộ giải quyết tranh chấp COM hay store procedure.

Tranh chấp dữ liệu trong table có thể được nhận biết ở cấp độ cột hoặc cấp độ hàng. Chọn lựa (option) mặc định là cột (column-tracked articles). Chọn lựa này cho phép những thay đổi được làm trên từng cột riêng biệt nhau, chỉ những thay đổi trên cùng một cột bị đánh dấu như là một tranh chấp. Tuy nhiên trong một vài ứng dụng, những luật của ứng dụng của bạn có thể đối xử đồng thời những thay đổi đến toàn bộ hàng như là một tranh chấp. Trong trường hợp này, cấp độ hàng là một chọn lựa.

Giải quyết tranh chấp

Khi tranh chấp xảy ra, một bộ giải quyết tranh chấp (conflict resolver) sẽ xác định tranh chấp được giải quyết như thế nào. Resolver áp dụng một tập luật qui định lên dữ liệu tranh chấp và chọn tác vụ thích hợp. Conflict resolver tạo ra những bảng conflict-usertablename để lưu những thông tin về tranh chấp. Bảng này được giữ tại Publisher cho những ứng dụng sử dụng centralized conflict logging và tại Subscriber đối với những ứng dụng sử dụng decentralized conflict logging. Bảng này có cùng cấu trúc với bảng gốc, và conflict resolver sao chép phiên bản (version) gần nhất của hàng vào bảng. Version “thắng cuộc” của hàng định cư trong user table thật sự. Những cột trong bảng hệ thống sysmergearticles giữ những thông tin về những bảng có liên quan đến bảng tranh chấp, và tên của những bảng đó. Xóa bỏ những tranh chấp sẽ lần theo bảng MSmerge-delete-conflicts. Với một vài tranh chấp bạn không thể phổ biến những thay đổi từ một site đến các site khác. Ví dụ hai site cùng insert một hàng có cùng một khoá dẫn đến xảy ra tranh chấp. Nếu mỗi lệnh insert đều thành công, thì những ràng buộc vi phạm sẽ không được biết cho đến khi quá trình đồng bộ site xảy ra. Lúc này MS SQL Server tự động xóa một trong hai hàng có cùng khóa chính đó. Những thông tin lỗi này cũng được lưu trong bảng tranh chấp. Những vấn đề khác như là vi phạm ràng buộc duy nhất yêu cầu một vài tác vụ user để khôi phục sự hội tụ. Vi phạm tính toàn vẹn hay gặp nhất là insert một hàng với một khoá ngoại trong khi site khác thì đang xóa hàng với khoá chính tương ứng. SQL Server cũng tự xóa những hàng vi phạm các ràng buộc này.

SQL Server nhận ra sự cần thiết cho những ứng dụng để có một lược đồ giải quyết tranh chấp mà xảy ra trong suốt quá trình kết hợp. Khi xây dựng ứng dụng, bạn có 3 thay đổi theo cơ chế giải quyết tranh chấp:

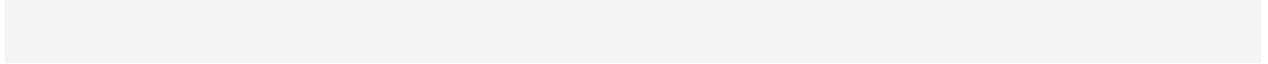
- Giải quyết tranh chấp dựa trên độ ưu tiên là mặc định khi bạn tạo ra ứng dụng.

Bản nháp_TS. Phan Thị Hà

- Giải quyết theo những store procedure mà bạn xây dựng theo những luật hay theo những dữ liệu xác định của bạn.

- Bộ giải quyết COM.

SQL server cho phép bạn xây dựng một bộ giải quyết tranh chấp (conflict resolver). Tuy nhiên, sử dụng bộ giải quyết tranh chấp COM thì phức tạp hơn là thực thi một custom store procedure resolver. Bạn nên sử dụng store procedure conflict resolver bất cứ khi nào có thể.



Mạng xã hội facebook Hay thì share :)

[SQLSERVER] Hướng dẫn cấu hình tạo bản sao đồng bộ dữ liệu giữa 2 server sử dụng Replication trong Sqlserver

Tạo bản sao dữ liệu (replication) là công việc rất quan trọng đối với người quản trị dữ liệu, cho phép chia sẻ, phân tán, đồng bộ dữ liệu giữa các Database server. Bài viết này giới thiệu cách tạo bản sao và đồng bộ dữ liệu trong **SQL Server** thông qua công cụ **Replication**.

[SQLSERVER] Hướng dẫn cấu hình tạo bản sao đồng bộ dữ liệu giữa 2 server sử dụng Replication trong Sqlserver

Mô hình Replication trong SQL Server

Nhắc lại Một số thuật ngữ:

Publisher (nhà xuất bản): Quản lý dữ liệu nguồn, quyết định những dữ liệu nào sẽ được nhân bản (phát hành)

Distributor (nhà phân phối): Thu thập và phân phối các dữ liệu được nhân bản đến người dùng thông qua các đại lý phân phối (Distribution Agent)

Subscriber (người dùng): Người có nhu cầu dữ liệu, đăng ký sử dụng.

Kịch bản triển khai Replication trong SQL Server

02 Database Server (gọi là Server 1 và Server 2)

Trên Server 1 chứa Database để tạo bản sao, trong ví dụ này Database có tên là DB

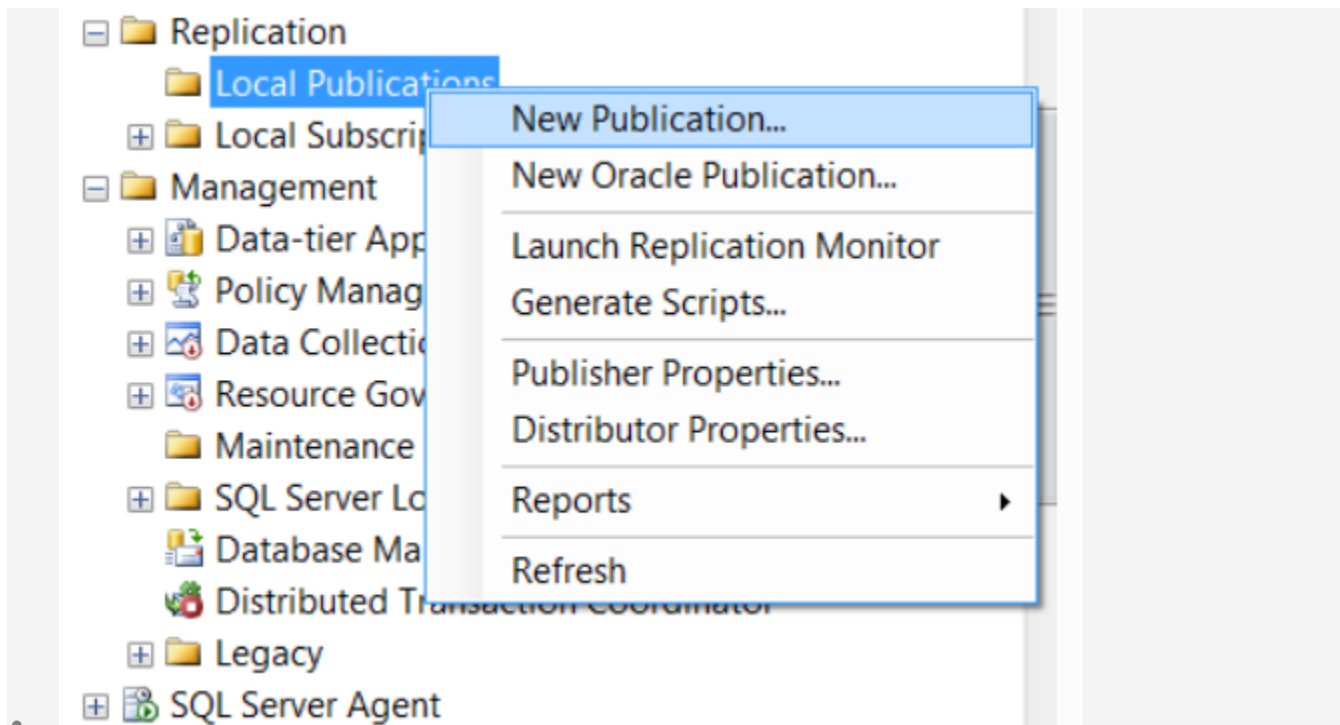
Trên Server 2 tạo Database có tên DB_REP, dùng để đồng bộ với Database DB trên Server 1

Chú ý: Để thực hiện replication, bạn phải có 2 SQL Database Server, để đơn giản bạn có thể cài SQL Server trên cùng một máy với 2 Instances khác nhau.

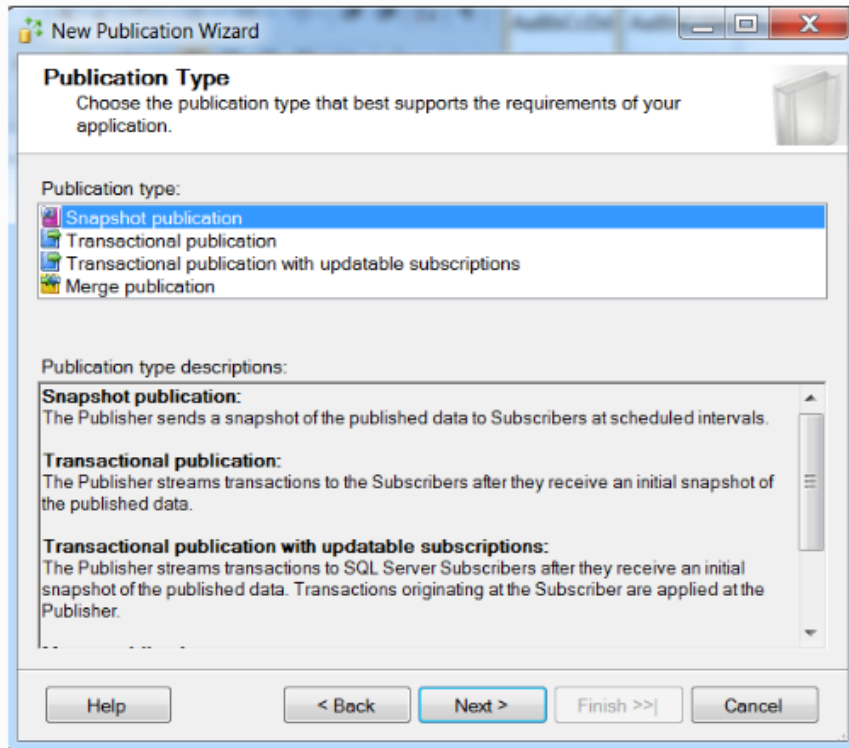
Các bước thực hiện như sau:

Bước 1. Trên Server1 tạo Database có tên DB, và tạo một số bảng để minh họa. Trong ví dụ này Database DB có 2 bảng là Employees và Departments. Trên Server 2 tạo Database có tên DB_REP (không cần tạo bảng dữ liệu) để nhận bản sao dữ liệu từ Server1.

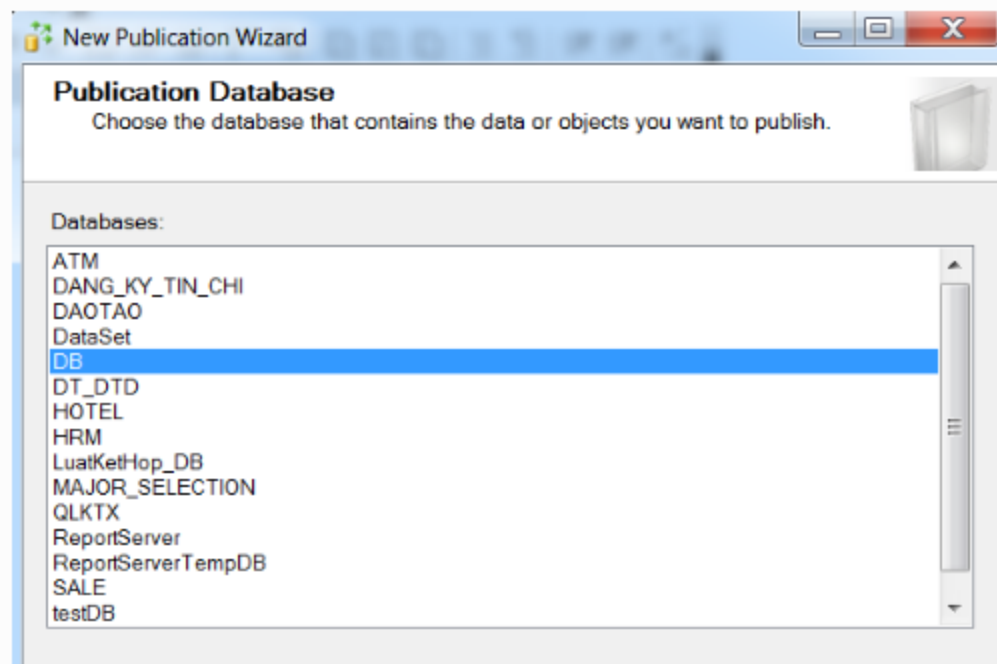
Bước 2. Đăng nhập vào SQL Server1, vào mục Replication, chọn New Publication



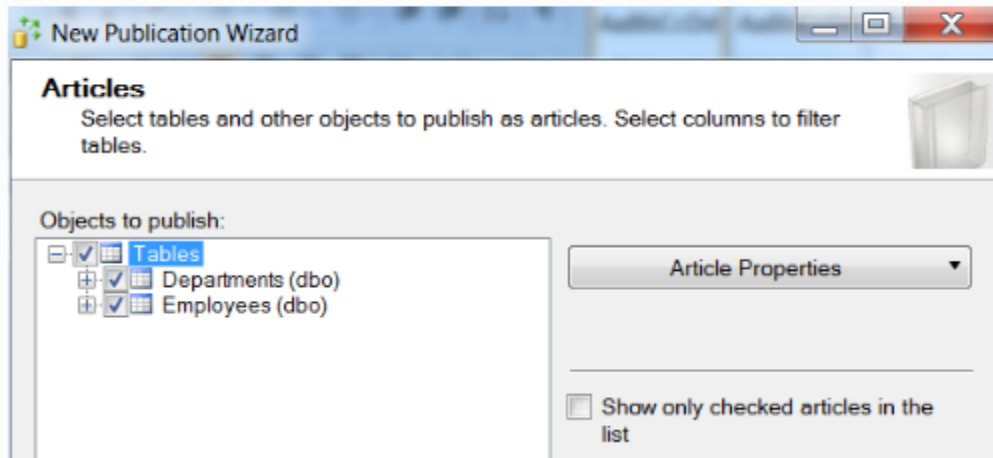
Chọn Next, chọn loại bản sao để thực hiện (Snapshot publication bản sao 1 chiều, merge publication bản sao 2 chiều)



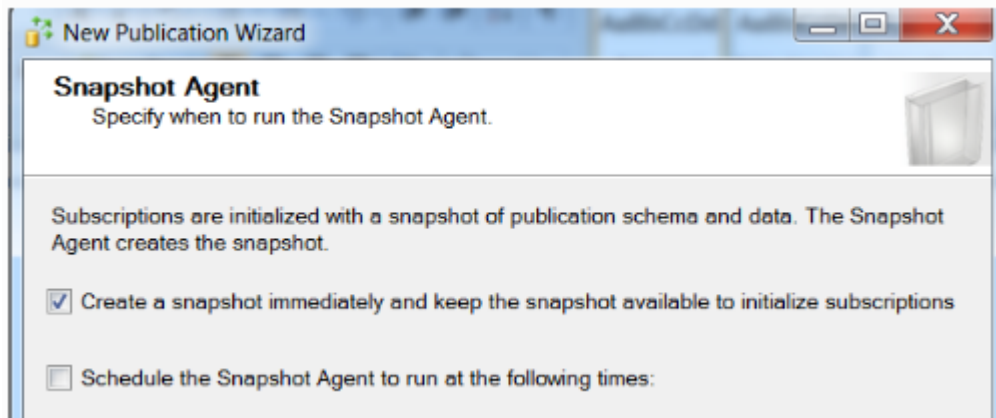
Chọn Next và chọn chọn Database muốn tạo bản sao



Chọn Next và chọn các bảng dữ liệu cần tạo bản sao



Bấm Next 2 lần, muốn tạo bản sao ngay thì chọn Create a snapshot immediately, muốn lập lịch để chạy đồng bộ thì chọn Schedule... (trong ví dụ này chọn thực hiện đồng bộ ngay)



Bấm Next, chọn Security Settings để thiết lập xác thực cho phép thực thi bản sao

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:

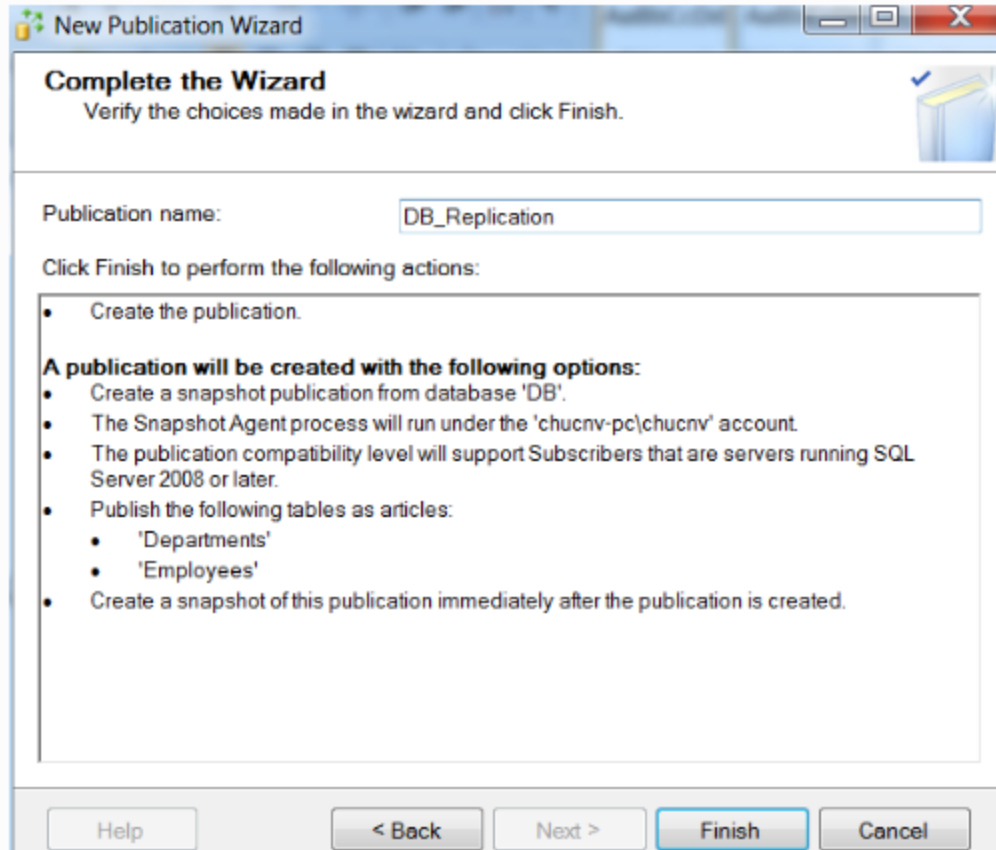
Login:

Password:

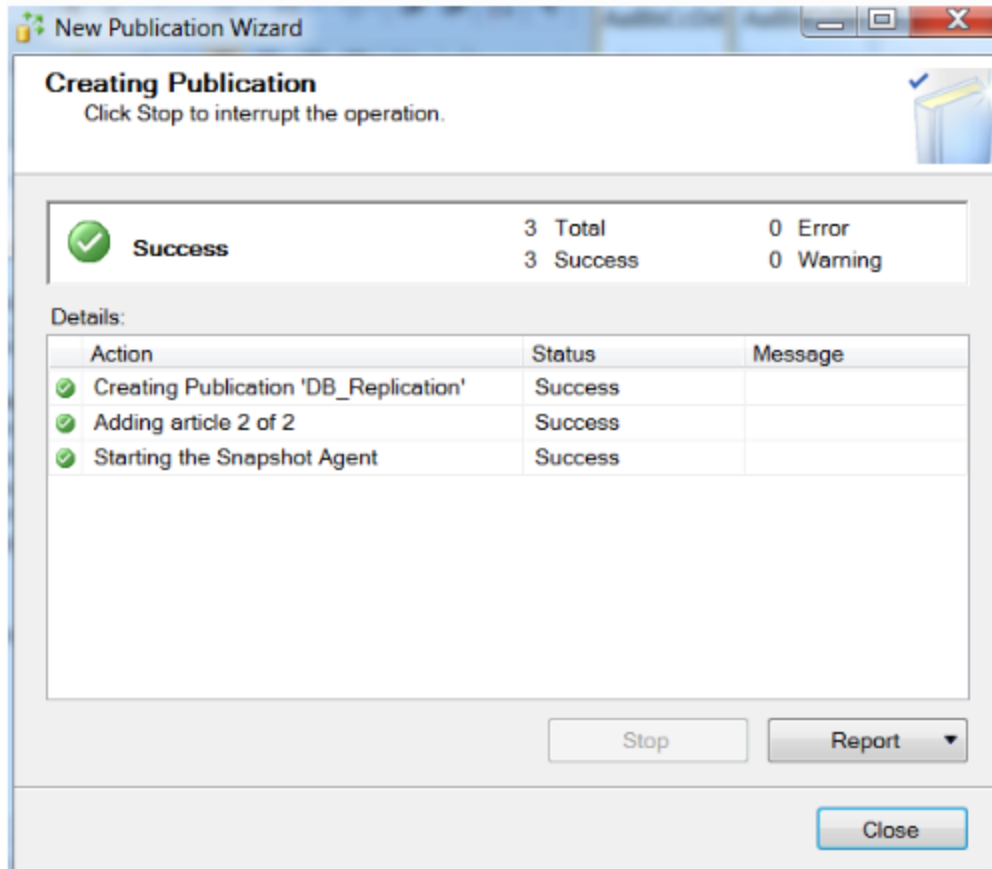
Confirm Password:

OK Cancel Help

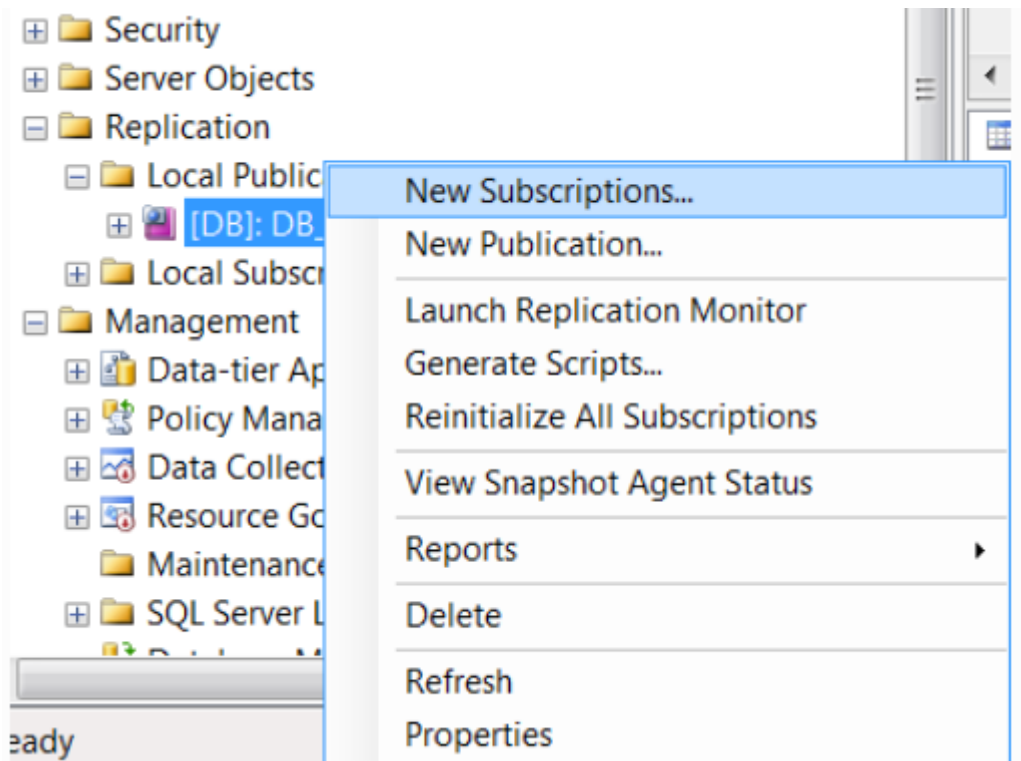
Bấm OK, Bấm Next 2 lần và đặt tên cho publication



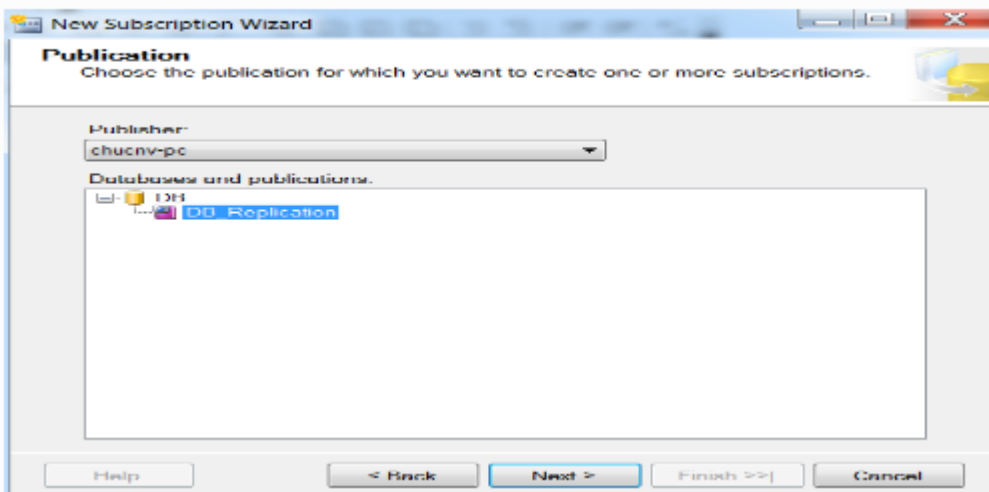
Bấm Finish để hoàn tất việc tạo Publication



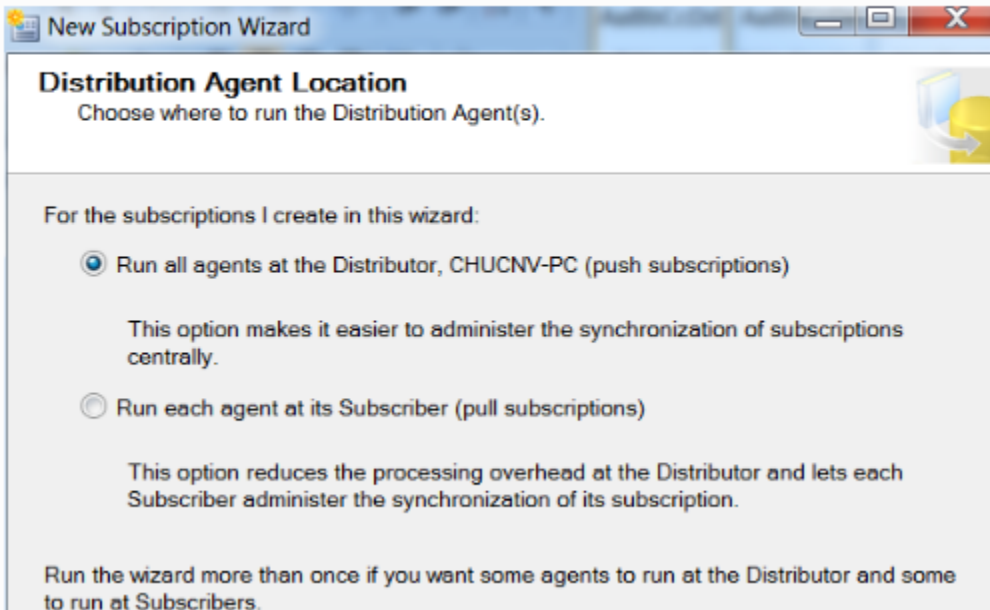
Bước 3. Tạo Subscription: Bấm phải chuột vào Publication vừa tạo, chọn New Subscription



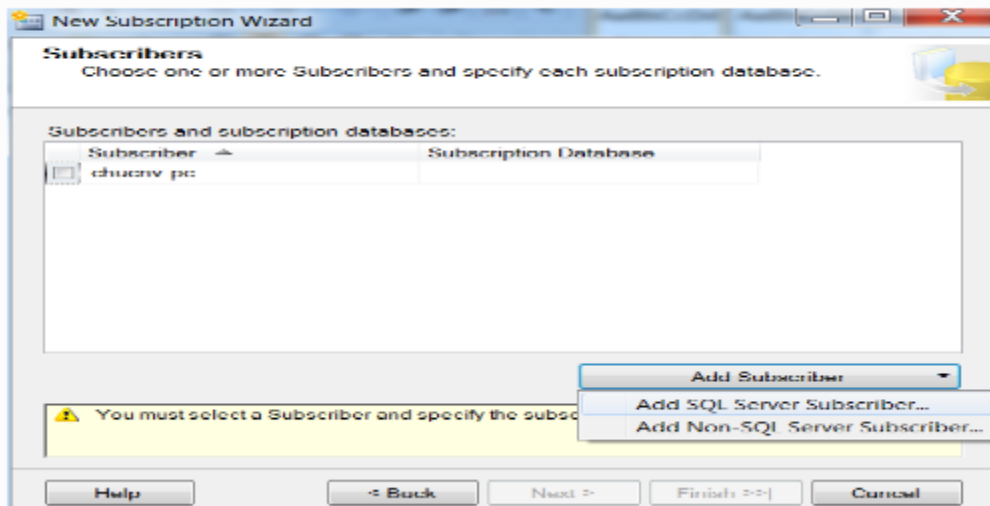
Bấm Next, chọn Publication đã tạo



Bấm Next, chọn Run all agents

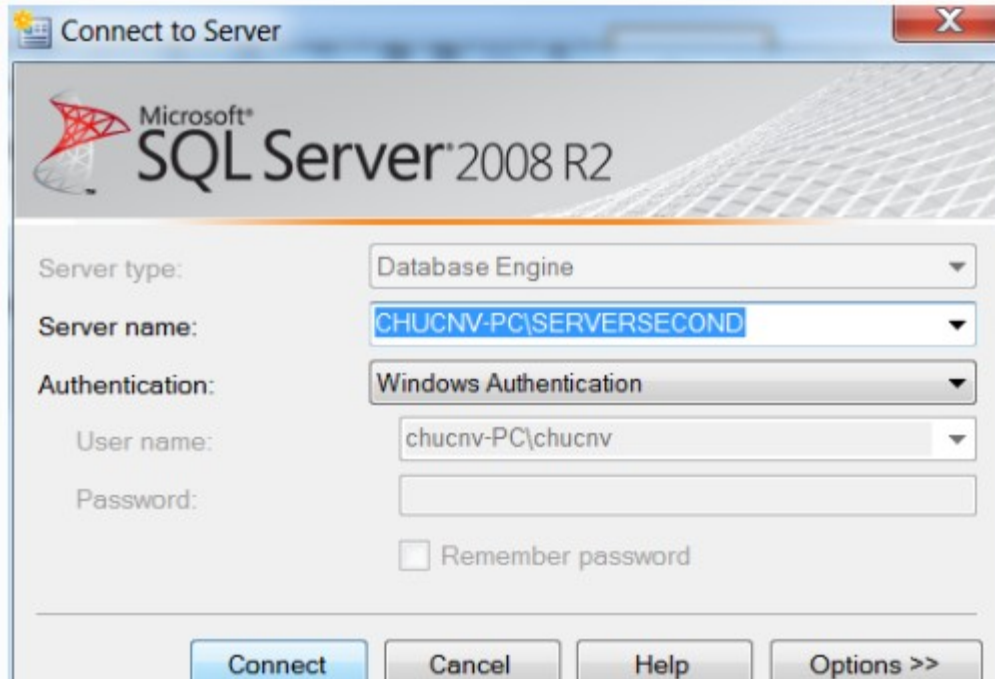


Bấm Next, chọn Add SQL Server Subscriber

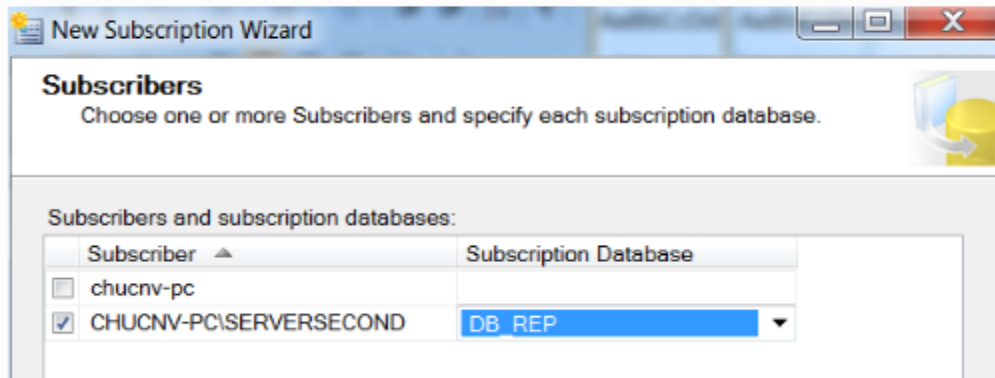


Bấm Next, đăng nhập vào Server 2

Bấm Next, đăng nhập vào Server 2



Chọn Database đồng bộ trên Server2 (trong ví dụ này là DB_REP)



Bấm Next, Xác thực tài khoản cho Distribution Agent Security

Distribution Agent Security

Specify the domain or machine account under which the Distribution Agent process will run when

☒ 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.)

Bấm OK, bấm Next, chọn chế độ thực hiện bản sao (liên tục, theo lịch...)

New Subscription Wizard

Synchronization Schedule
Specify the synchronization schedule for each agent.

Agent schedule:

Subscriber	Agent Location	Agent Schedule
CHUCNV-PC\...	Distributor	Run continuously

Chọn Next, Chọn khởi tạo Subscription ngay

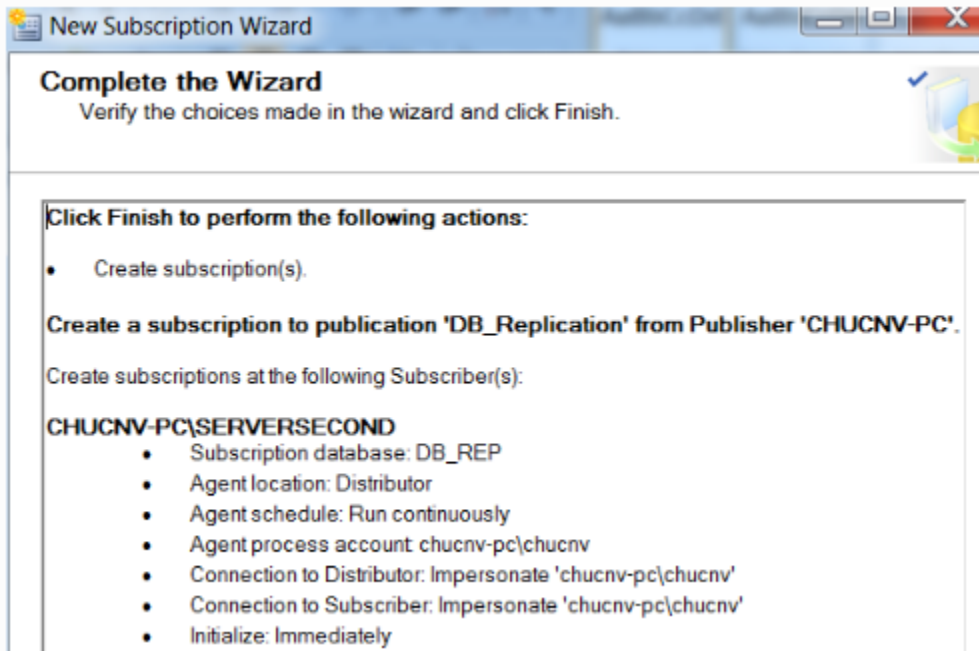
New Subscription Wizard

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

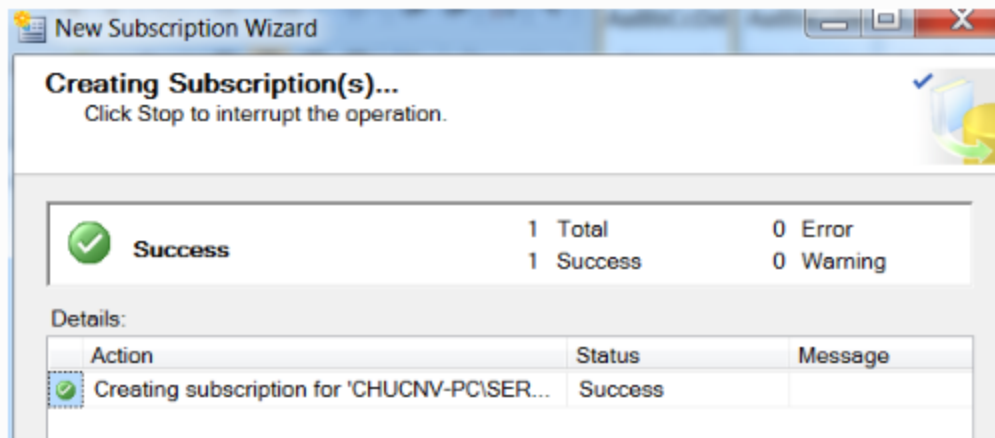
Subscription properties:

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

Bấm Next 2 lần và bấm Finish để hoàn thành việc tạo Subscription



Khởi tạo Subscription thành công



Bước 4. Kiểm tra kết quả thực hiện bản sao dữ liệu

Đăng nhập vào Server2, sẽ thấy các bảng dữ liệu trong Database DB trên Server1 đã được tạo bản sao trong Database DB_REP trên Server2



Đăng nhập vào Database Server1, mở bảng Employees, nhập thêm nhân viên mới