

Phần 1: Giới thiệu mở đầu

Hệ quản trị cơ sở dữ liệu SQL Server là 1 phần mềm giúp cho chúng ta có thể lưu giữ database, quản lý users, phân quyền users chặt chẽ. Trên phần mềm này đã cung cấp cho ta một số dịch vụ để thực hiện các yêu cầu nâng cao, chẳng hạn như dịch vụ SQL Server Agent cho phép ta nhân bản dữ liệu, hoặc tạo các job thực thi tự động 1 công việc nào đó; dịch vụ (Distributed Transactional Coperator- MSDTC) cho phép điều khiển việc thực thi 1 Stored Procedure trên môi trường cơ sở dữ liệu phân tán.

Ngoài ra, SQL Server còn cung cấp cho ta dịch vụ SQL Broker đóng vai trò lắng nghe xem có sự thay đổi gì trên cơ sở dữ liệu hay không, và nếu có thì ta có thể thực hiện 1 công việc thích hợp tương ứng.

Để kết hợp với dịch vụ SQL Broker trong việc theo dõi dữ liệu thay đổi trên cơ sở dữ liệu, phần mềm .NET có cung cấp class SQL Dependency để tự động nhận tín hiệu từ Broker chuyển tới.

Mục tiêu của chuyên đề:

- a. Tìm hiểu dịch vụ SQL Broker trong Hệ quản trị SQL Server , class SQL Dependency trên .NET
- b. Nội dung:
 - Tính năng đồng bộ dữ liệu của dịch vụ SQL Dependency trong SQL Server từ cơ sở dữ liệu tới các máy client đang kết nối.
 - Điều kiện để dịch vụ này hoạt động

Phần 2: Nội dung chuyên đề

1. Quản lý Id của Dịch vụ Broker

Mỗi cơ sở dữ liệu chứa một định danh duy nhất được sử dụng để định tuyến các thông điệp dịch vụ Broker cho cơ sở dữ liệu đó.

Mỗi cơ sở dữ liệu chứa một định danh dịch vụ Broker để phân biệt nó với tất cả các cơ sở dữ liệu khác trong mạng máy tính. Cột của **service_broker_guid** của sys.databases thể hiện Id của dịch vụ Broker cho mỗi cơ sở dữ liệu trong Server SQL. Hệ thống dịch vụ Broker có thể được thiết kế để chạy nhiều bản sao của một dịch vụ. Mỗi bản sao của dịch vụ chạy trong một cơ sở dữ liệu riêng biệt. Trong một hệ thống có nhiều bản sao của một dịch vụ, sử dụng option **BROKER_INSTANCE** của lệnh **CREATE ROUTE** tạo một đường đến bản cụ thể của dịch vụ.

Dịch vụ Broker định tuyến sử dụng định danh để đảm bảo rằng tất cả các thông điệp cho một cuộc trò chuyện được chuyển giao cho các cơ sở dữ liệu tương tự.

Theo mặc định, Dịch vụ Broker là active trong một cơ sở dữ liệu khi cơ sở dữ liệu được tạo. Để xác định xem Dịch vụ Broker là đang active trong một cơ sở dữ liệu hay không, ta kiểm tra cột **is_broker_enabled** của bảng sys.databases

Ví dụ:

Thay đổi thuộc tính **ENABLE_BROKER** của cơ sở dữ liệu :

USE master ;

GO

ALTER DATABASE QLVN SET ENABLE_BROKER with rollback immediate

Xem dịch vụ Broker đã hoạt động chưa và Id của nó:

```
SELECT NAME , is_broker_enabled, service_broker_guid  
FROM SYS.DATABASES
```

2. Class SQL Dependency:

Đối tượng **SqlDependency** đại diện cho một mối liên quan phụ thuộc giữa một ứng dụng truy vấn và một instance của SQL Server. Một ứng dụng có thể tạo ra một đối tượng **SqlDependency** và đăng ký nhận thông báo qua xử lý sự kiện **OnChangeEventHandler**.

3. Phát hiện sự thay đổi dữ liệu qua **SqlDependency**:

Một đối tượng **SqlDependency** có thể được liên kết với một **SqlCommand** để phát hiện khi kết quả truy vấn khác với ban đầu lấy ra. Trong một ứng dụng Windows Form, ta có thể tạo ra một đối tượng **SqlDependency** và duy trì một tham chiếu đến nó. Ngoài ra, ta cũng có thể gắn một liên kết đến sự kiện **OnChange**, sự kiện này sẽ nảy sinh khi có dữ liệu thay đổi trên một lệnh có liên quan.

Ta phải liên kết **SqlDependency** với một lệnh trước khi ta thực hiện lệnh. Thuộc tính **HasChanges** của **SqlDependency** cũng có thể được sử dụng để xác định xem các kết quả truy vấn đã thay đổi kể từ khi dữ liệu lần đầu tiên được lấy ra.

Lưu ý: Cơ sở hạ tầng dependency dựa trên một **SqlConnection** đã được mở ra khi phương thức **Start** được gọi để nhận được thông báo rằng các dữ liệu cơ bản đã thay đổi trên một lệnh.

4. Các bước cài đặt sự thay đổi dữ liệu:

Các bước sau đây minh họa cách khai báo một SqlDependency, thực thi một lệnh, và nhận được một thông báo khi có những thay đổi trên dữ liệu kết quả đã truy vấn qua câu lệnh :

1. Khởi tạo một SqlDependency kết nối đến Server .
2. Tạo đối tượng SqlConnection kết nối đến Server và đối tượng SqlCommand chứa lệnh Transact - SQL .
3. Tạo một đối tượng SqlDependency mới , hoặc sử dụng đối tượng đang tồn tại, và kết nó vào đối tượng SqlCommand đã tạo ở bước 2. Điều này sẽ tạo ra một đối tượng SqlNotificationRequest và kết nó vào đối tượng lệnh cần thiết.
4. Khai báo xử lý cho sự kiện OnChange của đối tượng SqlDependency .
5. Thực hiện lệnh Execute của đối tượng SqlCommand . Bởi vì đã lệnh được ràng buộc với đối tượng thông báo , Server nhận ra rằng nó phải tạo ra một thông báo, và các thông tin hàng đợi sẽ trở đến hàng đợi dependencies.
6. Ngừng kết nối SqlDependency đến Server.

Đoạn code mẫu sau đây sẽ hướng dẫn ta tạo 1 ứng dụng mẫu theo dõi dữ liệu thay đổi trên SQL Server:

```
void Initialization()
{
    // Create a dependency connection.
    SqlDependency.Start(connectionString);
}

void SomeMethod()
{
    // Mở kết nối SqlConnection.
    connection = New SqlConnection(GetConnectionString())
    connection.Open()
    // Tạo 1 đối tượng SqlCommand.
    SqlCommand command=new SqlCommand(
        " select manv as [Mã NV],Ho as [      Họ],Ten as
[Tên],phai as [Phái],diachi as [      Địa chỉ] from
dbo.Nhanvien ",    connection);

    // Tạo 1 dependency và liên kết nó với đối tượng SqlCommand.
    SqlDependency dependency=new SqlDependency(command);

    // Subscribe đến sự kiện SqlDependency.
    dependency.OnChange+=new OnChangeEventHandler(OnDependencyChange);

    // Thực thi lệnh trên đối tượng command.
    command.ExecuteReader();
}

// Handler method
void OnDependencyChange(object sender, SqlNotificationsEventArgs e )
{
}
```

```
// Handle the event (for example, invalidate this cache entry).  
}  
  
void Termination()  
{  
    // Dừng theo dõi dữ liệu trên Server  
    SqlDependency.Stop(connectionString, queueName);  
}
```