**Лекція 5. Розподілені запити і розподілені транзакції MS SQL SERVER**

**1 Пов'язані сервери**

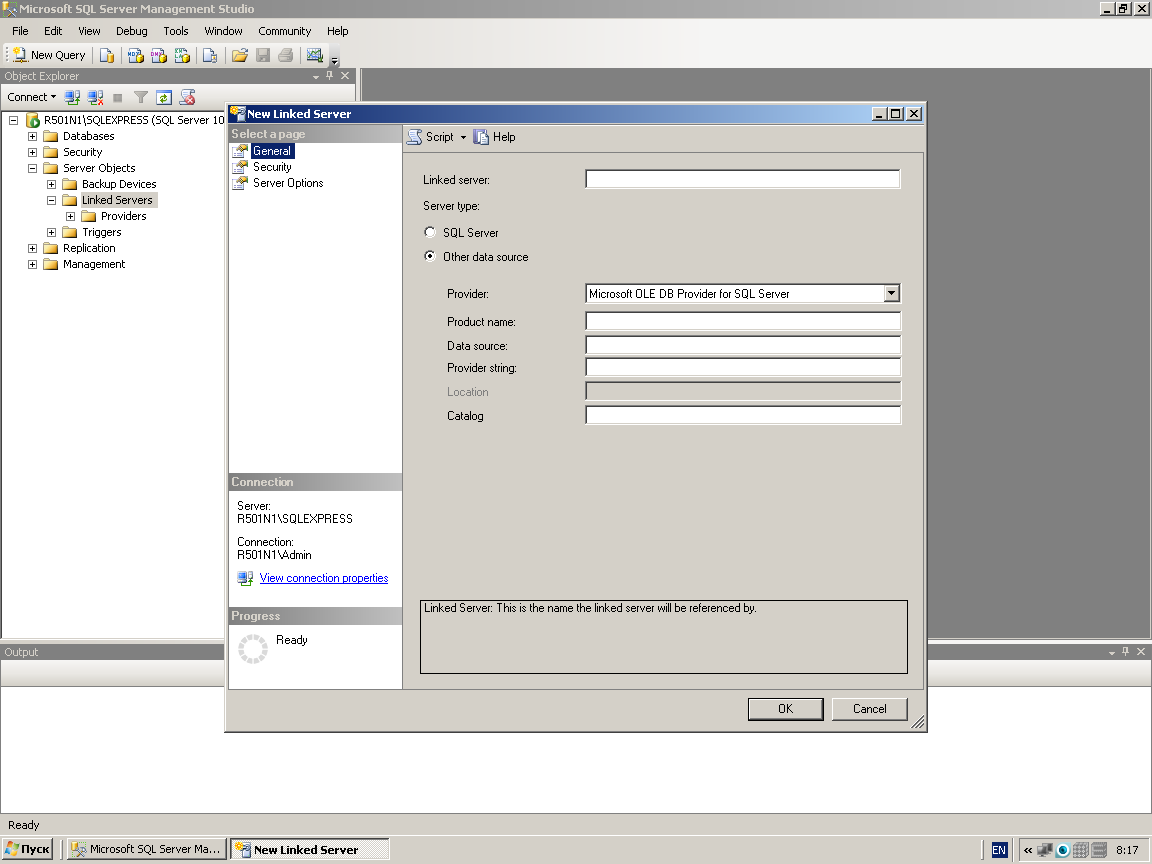
Для локального сервера можна визначити кілька пов'язаних серверів. Пов'язані сервери дозволяють виконувати розподілені різнорідні запити до джерел даних OLE DB. Системна процедура sp\_addlinkedserver створює пов'язаний сервер (на них можуть бути використані різні СУБД). Після створення пов'язаного сервера можна виконувати розподілені запити на цьому сервері. Якщо пов'язаний сервер визначено як примірника SQL Server, на ньому можуть виконуватися віддалені збережені процедури.

Створювати пов'язані сервера можна двома способами:

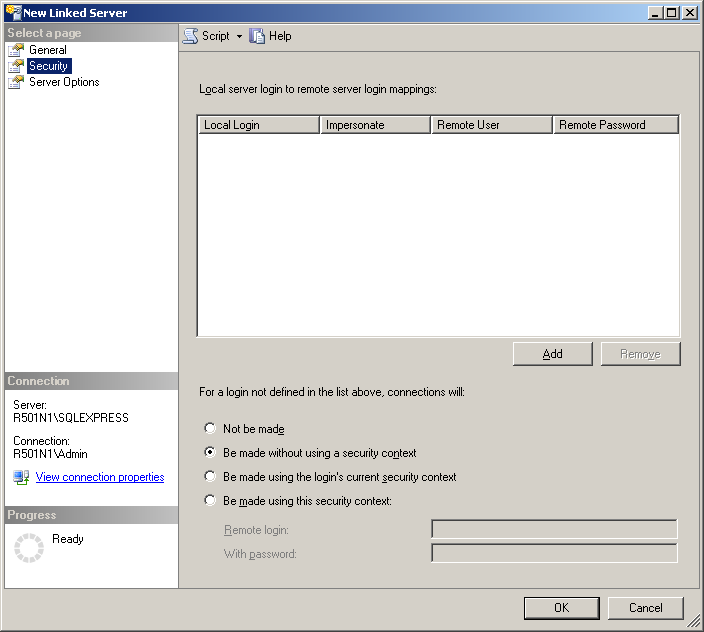
1. Використання утиліти SERVER MANAGEMENT STUDIO

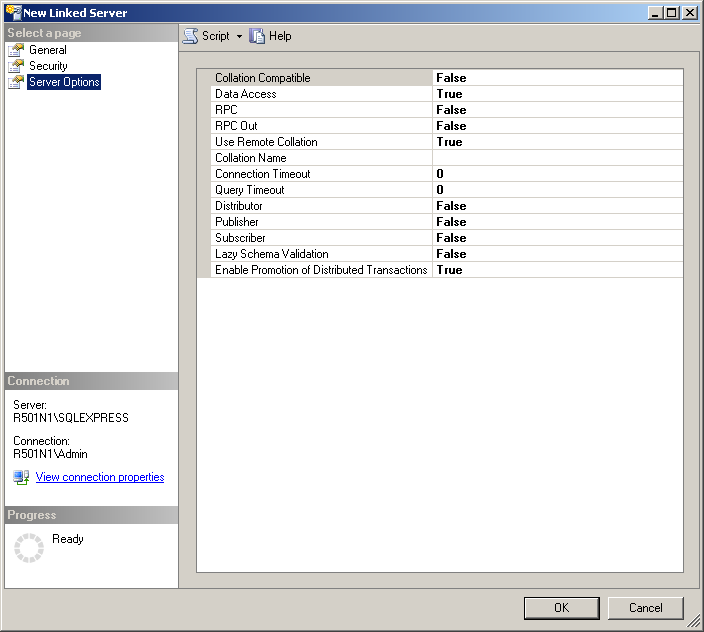
Пункт меню ServerObject -> Linked server відкриває вікно підключення пов'язаних серверів зі вкладками:

**• General** для вказівки загальних даних таких як ім'я сервера в форматі сервер / екземпляр БД, тип сервера і параметри провайдера



**• Security** для вказівки акаунтів підключення (логіни та паролі)



* **Server** options для надання додаткових конфігураційних параметрів пов'язаного сервера 

**2. Використання в T SQL – скрипту користувача системної процедури sp\_addlinkedserver.**

Синтаксис звернення: sp\_addlinkedserver [@ server =] 'server' [, [@ srvproduct =] 'product\_name'] [, [@ provider =] 'provider\_name'] [, [@ datasrc =] 'data\_source'] [, [location =] 'location'] [, [@ provstr =] 'provider\_string'] [, [@ catalog =] 'catalog']

Аргументи [@ server =] 'server' - ім'я створюваного пов'язаного сервера. Аргумент server має тип sysname і не має значення за замовчуванням.

[@ Srvproduct =] 'product\_name' - призначена для користувача назва продукту для джерела даних OLE DB, який додається в якості пов'язаного сервера, має тип nvarchar (128) і значення за замовчуванням NULL.

[@ Provider =] 'provider\_name' - унікальний програмний ідентифікатор (PROGID) постачальника OLE DB, що відповідає цьому джерелу даних, має тип nvarchar (128) і значення за замовчуванням NULL; однак якщо аргумент provider\_name опущений, використовується SQLNCLI - про свого клієнта SQL Server

[@ Datasrc =] 'data\_source' - ім'я джерела даних, як воно інтерпретується постачальником OLE DB. data\_source, має тип nvarchar (4000).

[@ Location =] 'location' - місцезнаходження бази даних, ясна постачальнику OLE DB, має тип nvarchar (4000) і значення за замовчуванням NULL.

[@ Provstr =] 'provider\_string' - рядок підключення для конкретного постачальника OLE DB, яка вказує унікальне джерело даних, має тип nvarchar (4000) і значення за замовчуванням NULL. Коли пов'язаний сервер свого клієнта SQL Server, вказати екземпляр можна за допомогою ключового слова SERVER (SERVER = servername \ instancename, servername - це ім'я комп'ютера, а instancename - ім'я конкретного екземпляра SQL Server, до якого буде підключений користувач.)

[@ Catalog =] 'catalog' - каталог, який повинен використовуватися при підключенні до постачальника OLE DB. Аргумент catalog має тип sysname і значення за замовчуванням NULL. Значення catalog передається як властивість DBPROP\_INIT\_CATALOG для ініціалізації постачальника OLE DB. Якщо пов'язаний сервер визначено для екземпляра SQL Server, то каталог посилається на базу даних за замовчуванням, з якої підтверджено пов'язаний сервер.

Код возврата- 0 (успішне завершення) або 1 (неуспішне завершення)

Примітка. Крім того можна використовувати додаткові системні процедури

sp-addlinkedserver login для створення облікових записів управління доступом

sp-helpaddlinkedserverlogin для переглянути свої облікові записи управління доступом

sp-droplinkedserverlogin для видалення облікових записів управління доступом

sp-helpserver для перегляду інформації про пов'язаних серверах

У цій табличці вказано способи настройки пов'язаного сервера для джерел даних, доступних через постачальник OLE DB. Пов'язаний сервер може бути налаштований кількома способами для конкретного джерела даних; для одного типу джерела даних можливі кілька рядків. Також в цій табличці вказано значення аргументів процедури sp\_addlinkedserver, використовувані для настройки пов'язаного сервера.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Источник OLE DB** | **Поставщик OLE DB** | **product\_name** | **provider\_name** | **data\_source** | **location** | **provider\_string** | **catalog** |
| SQL Server | OLE DB для власного клієнта MS SQL Server | SQL Server1 (за зам) |  |  |  |  |  |
| SQL Server | OLE DB для власного клієнта MSSQL Server |  | **SQLNCLI** | Мережеве ім’я SQL Server |  |  | Ім’я бази даних |
| SQL Server | OLE DB для власного клієнта MSSQL Server |  | **SQLNCLI** | servername\instancename |  |  | Ім’я бази даних |
| Oracle, версія 8 або більше | Постачальник  Oracle для OLE DB | Любий | **OraOLEDB.Oracle** | Псевдонім для бази даних Oracle |  |  |  |
| Access/Jet | OLE DB для Jet (Майкрософт) | Любий | **Microsoft.Jet.OLEDB.4.0** | Повний шлях до файлу бази |  |  |  |
| Джерело ODBC | Microsoft OLE DB для ODBC | Любий | **MSDASQL** | Системний DSN джерела ODBC |  |  |  |
| Джерело ODBC | Microsoft OLE DB для ODBC | Любий | **MSDASQL** |  |  | Рядок підключення ODBC |  |
| Файлова система | Microsoft OLE DB для служби індексування | Любий | **MSIDXS** | Ім’я каталогу служби індексування |  |  |  |
| Таблиця  Microsoft Excel | Постачальник Microsoft OLE DB для Jet | Любий | **Microsoft.Jet.OLEDB.4.0** | Повний шлях до файлу Excel |  | Excel 5.0 |  |
| База даних IBM DB2 | Постачальник Microsoft OLE DB для DB2 | Любий | **DB2OLEDB** |  |  | . | Ім’я каталогу бази даних DB2 |

1 При такому способі настройки пов'язаного сервера ім'я пов'язаного сервера збігається з мережевим ім'ям віддаленого примірника SQL Server. Використовуйте аргумент data\_source, щоб вказати сервер.

2 «Будь» вказує, що назва продукту може бути будь-яким.

Постачальник OLE DB для свого клієнта MicrosoftSQL Server використовується разом з SQL Server в разі, якщо ім'я постачальника не зазначено або SQL Server визначено як назва продукту. Навіть якщо вказано ім'я попереднього постачальника (SQLOLEDB), воно все одно буде змінено на SQLNCLI при збереженні в каталог.

Аргументи data\_source, location, provider\_string і catalog ідентифікують базу даних або бази даних, на які вказує пов'язаний сервер. Якщо значення одного з цих аргументів одно NULL, то відповідне властивість ініціалізації постачальника OLE DB не встановлено.

У кластерізованний середовищі при вказівці імен файлів для вказівки джерел даних OLE DB використовуйте формат UNC або загальні диски для вказівки розташування.

Процедура sp\_addlinkedserver не може бути виконана всередині користувальницької транзакції.

**Примітка з безпеки:**

При створенні пов'язаного сервера за допомогою процедури sp\_addlinkedserver для всіх локальних імен входу за замовчуванням додається зіставлення самим собі. Постачальники, відмінні від SQL Server, для яких виконана перевірка достовірності SQL Server, можуть отримати доступ до постачальника під обліковим записом служби SQL Server. Адміністраторам потрібно розглянути застосування процедури sp\_droplinkedsrvlogin <linkedserver\_name>, NULL для видалення глобального зіставлення.

Необхідно дозвіл ALTER ANY LINKED SERVER.

 Використання постачальника OLE DB для свого клієнта Microsoft SQL Server

У наступному прикладі показано створення пов'язаного сервера з ім'ям SEATTLESales. Назва продукту - SQL Server, ім'я постачальника не використовується.

USE master;

GO

EXEC sp\_addlinkedserver

N'SEATTLESales ',

   N'SQL Server ';

GO

Цей приклад демонструє, як створити пов'язаний сервер S1\_instance1 на екземплярі SQL Server за допомогою постачальника OLE DB для свого клієнта SQL Server.

EXEC sp\_addlinkedserver

@ Server = N'S1\_instance1 ', @ srvproduct = N' ', @ provider = N'SQLNCLI',

@ Datasrc = N'S1 \ instance1 ';

Використання постачальника Microsoft OLE DB для Microsoft Access

Постачальник Microsoft.Jet.OLEDB.4.0 з'єднується з базами даних Microsoft Access в форматі 2002-2003. У наступному прикладі показано створення пов'язаного сервера з ім'ям SEATTLE Mktg.

**Примітка:**

У цьому прикладі передбачається, що встановлена ​​база даних Microsoft Access і зразок бази даних Northwind, а база даних Northwind знаходиться в каталозі «C: \ Msoffice \ Access \ Samples».

EXEC sp\_addlinkedserverserver = N'SEATTLE Mktg ',provider = N'Microsoft.Jet.OLEDB.4.0',srvproduct = N'OLE DB Provider for Jet ',

datasrc = N'C: \ MSOffice \ Access \ Samples \ Northwind.mdb ';

GO

Постачальник Microsoft.ACE.OLEDB.12.0 з'єднується з базами даних Microsoft Access в форматі 2007. У наступному прикладі показано створення пов'язаного сервера з ім'ям SEATTLE Mktg.

**Примітка**

У цьому прикладі передбачається, що встановлена ​​база даних Microsoft Access і зразок бази даних Борей, а база даних Борей знаходиться в каталозі «C: \ Msoffice \ Access \ Samples».

 EXEC sp\_addlinkedserverserver = N'SEATTLE Mktg ',

provider = N'Microsoft.ACE.OLEDB.12.0 ',srvproduct = N'OLE DB Provider for ACE',

datasrc = N'C: \ MSOffice \ Access \ Samples \ Northwind.accdb ';

GO

**Використання постачальника Microsoft OLE DB для ODBC з аргументом джерела даних**

В даному прикладі показано, як створити пов'язаний сервер з ім'ям SEATTLE Payroll, який використовує постачальник Microsoft OLE DB для ODBC (MSDASQL) і параметр data\_source.

Примітка Зазначений джерело даних ODBC повинен бути визначений як системний DSN на сервері до того, як буде використовуватися пов'язаний сервер.

EXEC sp\_addlinkedserver

server = N'SEATTLE Payroll ',srvproduct = N' ',provider = N'MSDASQL',

datasrc = N'LocalServer ';

GO

**2. Розподілені транзакції**

Розподілені транзакції, що запускаються в Transact-SQL, мають відносно просту схему виконання:

1. Додаток або сценарій Transact-SQL виконує інструкцію Transact-SQL, яка ініціює розподілену транзакцію.

2. Примірник SQL Server Database Engine, що виконує інструкцію, стає керуючим сервером в транзакції.

3. Сценарій або додаток потім виконує або розподілені запити до пов'язаних серверів, або віддалені збережені процедури на віддалених серверах.

4. Після того як розподілені запити і виклики віддалених процедур виконані, керуючий сервер автоматично викликає координатор розподілених транзакцій (Майкрософт) (MS DTC), щоб прикріпити в розподілену транзакцію пов'язані і віддалені сервери.

5. Коли сценарій або додаток виконує інструкцію COMMIT або ROLLBACK, керуючий екземпляр SQL Server викликає MS DTC, який керує процесом двухфазной фіксації або повідомляє пов'язані і віддалені сервери, щоб вони виконали відкат своїх транзакцій.

За вказівками Transact-SQL

. Нижче представлені інструкції Transact-SQL, які необхідно виконати в сценарії або додатку Transact-SQL:

1. Запуск розподіленої транзакції.

Розподілену транзакцію в Transact-SQL можна запустити наступними способами:

− Явно почати розподілену транзакцію, використовуючи інструкцію BEGIN DISTRIBUTED TRANSACTION.

− Перебуваючи в локальній транзакції, виконати розподілений запит.

− Якщо була виконана інструкція SET REMOTE\_PROC\_TRANSACTIONS ON і локальна транзакція викликає віддалену збережену процедуру на іншому екземплярі компонента Database Engine, локальна транзакція стає розподіленим. Параметр REMOTE\_PROC\_TRANSACTIONS - це параметр сумісності, який відноситься тільки до викликів віддалених збережених процедур на віддалених серверах, визначених процедурою sp\_addserver. Параметр не застосовується до розподілених запитам, які виконують процедури, на пов'язаних серверах, визначених при використанні процедури sp\_addlinkedserver.

**2. Виконання розподілених запитів до пов'язаних серверів або виконання віддаленого** **виклику процедур на віддалених серверах.**

3. Виклик стандартних інструкцій Transact-SQL COMMIT TRANSACTION, COMMIT WORK, ROLLBACK TRANSACTION або ROLLBACK WORK для завершення транзакції.

Для будь-якої розподіленої транзакції Transact-SQL екземпляр Database Engine, що обробляє з'єднання або сценарій Transact-SQL, автоматично викликає MS DTC для координації її фіксації або відкоту.

**4. Приклади**

**Приклад розподіленого запиту**

Нехай є пов'язаний сервер SEATLESALES в якому обслуговується база даних AdventureWorks2008R2 в якій створені дві схеми

схема Production з таблицею Product і схема Sales з таблицею SalesOrderDetail

Тоді розподілений запит може мати вигляд:

SELECT p.Name, sod.SalesOrderID

FROM

   SEATLESALES. AdventureWorks2008R2. Production.Product p INNER JOIN

  SEATLESALES. AdventureWorks2008R2. Sales. SalesOrderDetail sod

ON p.ProductID = sod.ProductID ORDER BY p.Name

**Приклад розподіленої транзакції видалення**

У прикладі видаляється кандидат з бази даних База даних AdventureWorks2008R2 як на локальному екземплярі компонента Database Engine, так і на віддаленому сервері. І локальна, і віддалена база даних зафіксує транзакцію або виконає її відкат.

**Примітка.** Цей приклад наводить до повідомлення про помилку, якщо координатор MS DTC не встановлено на комп'ютері, де виконується екземпляр компонента Database Engine.

USE AdventureWorks2008R2;

GO

BEGIN DISTRIBUTED TRANSACTION;

- Delete candidate from local instance.

DELETE AdventureWorks2008R2.HumanResources.JobCandidate WHERE JobCandidateID = 13;

- Delete candidate from remote instance.

DELETE RemoteServer.AdventureWorks2008R2.HumanResources.JobCandidate

WHERE JobCandidateID = 13;

COMMIT TRANSACTION;

GO

Приклад розподіленої транзакції на оновлення

BEGIN DISTRIBUTED TRANSACTION;

- Update in local instance.

UPDATE Person

    SET LastName = 'JONSON'

WHERE PersonID = 10;

- Update inremote instance.

UPDATE [NOLI / SQL]. Family.dbo.Person SET LastName = 'JONSON'

WHERE PersonID = 10;

COMMIT TRANSACTION;

GO

**5. Моніторинг виконання розподіленої транзакції**

Роботу координатора розподілених транзакцій можна переглянути засобами операційної системи.

Ланцюжок дій Start-> Control Panel-> Administrative Tools-> Component services-> Distributed Transaction Controller

Виводить дерево функцій з можливістю перегляду журналу розподілених транзакцій і відповідної статистики