**Лекция. Распределенные запросы и распределенные транзакции MS SQL SERVER**

**1 Связанные серверы**

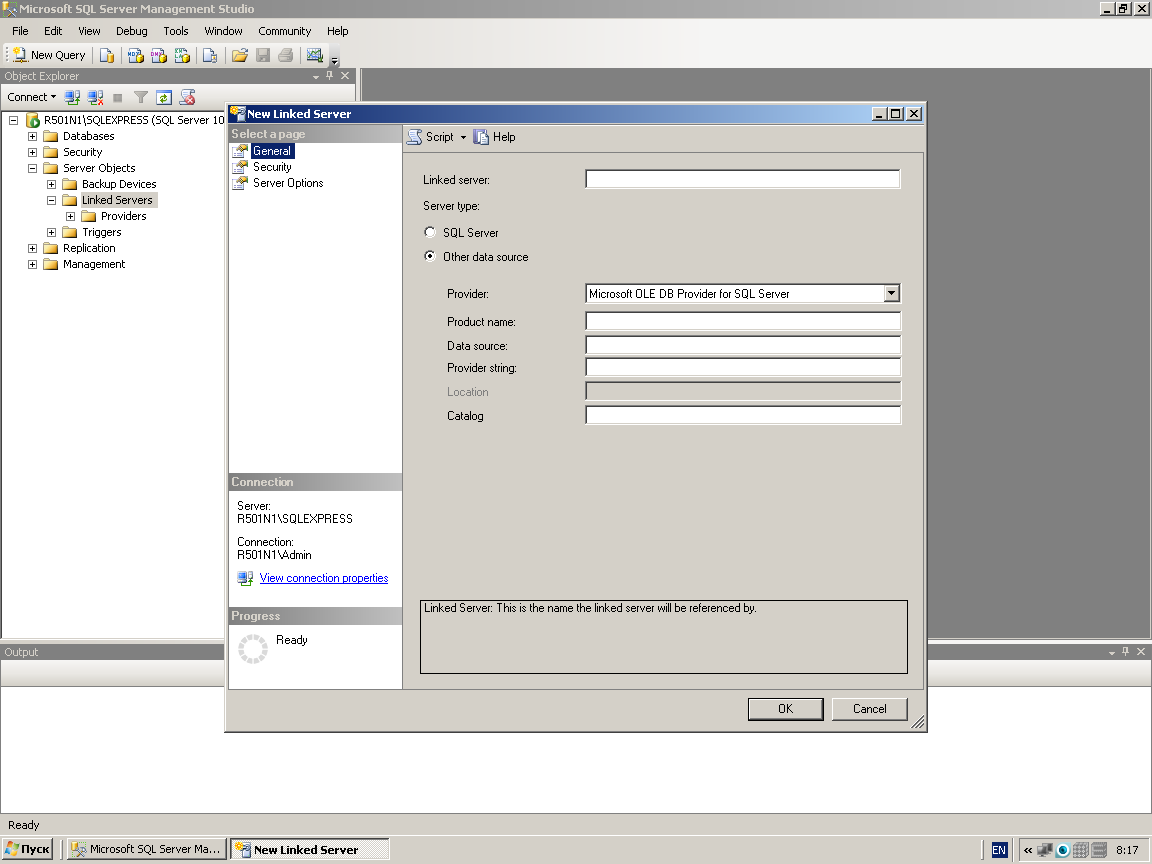
Для локального сервера можно определить несколько связанных серверов. Связанные серверы позволяют выполнять распределенные разнородные запросы к источникам данных OLE DB. Системная процедура **sp\_addlinkedserver** создает связанный сервер (на них могут быть использованы различные СУБД). После создания связанного сервера  можно выполнять распределенные запросы на этом сервере. Если связанный сервер определен в качестве экземпляра SQL Server, на нем могут выполняться удаленные хранимые процедуры.

Создавать связанные сервера можно двумя способами:

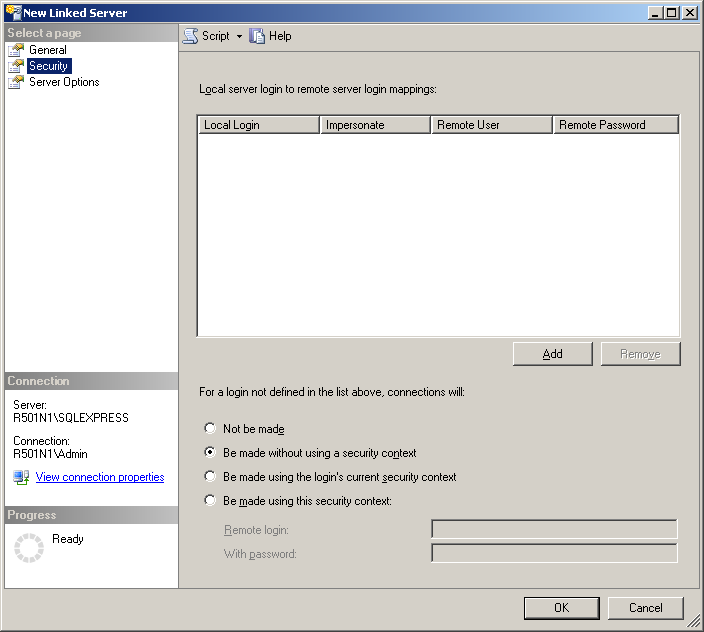
1. Использование утилиты SERVER MANAGEMENT STUDIO

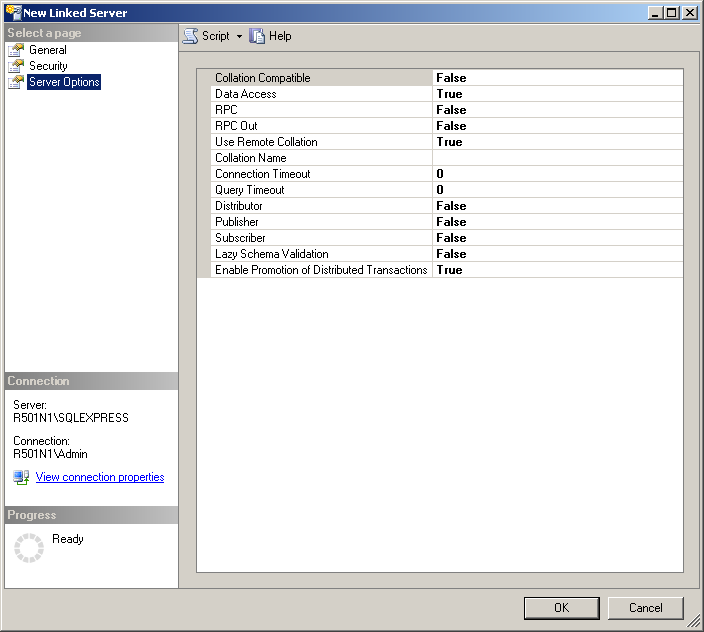
Пункт меню **ServerObject** ->**Linked server** открывает окно подключения связанных серверов со вкладками:

* **General** для указания общих данных таких как имя сервера в формате сервер/екземплярБД, тип сервера и параметры провайдера



* **Security** для указания аккаунтов подключения(логины и пароли)



* **Server options** для указания дополнительных конфигурационных параметров связанного сервера

2 Использование в T\_SQL – скрипте системной процедуры **sp\_addlinkedserver.**

Синтаксис обращения:

[Копировать](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_a1506c3c-467d-42e6-a999-73a516937334');" \o "Копировать в буфер обмена.) sp\_addlinkedserver [ @server= ] 'server' [ , [ @srvproduct= ] 'product\_name' ]

     [ , [ @provider= ] 'provider\_name' ]      [ , [ @datasrc= ] 'data\_source' ]

     [ , [ @location= ] 'location' ]      [ , [ @provstr= ] 'provider\_string' ]      [ , [ @catalog= ] 'catalog' ]

[**Аргументы**](javascript:void(0))

[ **@server=** ] **'**server**'** - имя создаваемого связанного сервера. Аргумент server имеет тип sysname и не имеет значения по умолчанию.

[ **@srvproduct=** ] **'**product\_name**'** – пользовательское название продукта для источника данных OLE DB, который добавляется в качестве связанного сервера, имеет тип nvarchar(128) и значение по умолчанию NULL.

[ **@provider=** ] **'**provider\_name**'** - уникальный программный идентификатор (PROGID) поставщика OLE DB, соответствующий этому источнику данных, имеет тип nvarchar(128) и значение по умолчанию NULL; однако если аргумент provider\_name опущен, используется SQLNCLI - собственного клиента SQL Server

[ **@datasrc=** ] **'**data\_source**'** - имя источника данных, как оно интерпретируется поставщиком OLE DB. data\_source, имеет тип nvarchar(4000).

[ **@location=** ] **'**location**'** - местонахождение базы данных, понятное поставщику OLE DB, имеет тип nvarchar(4000) и значение по умолчанию NULL.

[ **@provstr=** ] **'**provider\_string**'** - строка подключения для конкретного поставщика OLE DB, указывающая уникальный источник данных, имеет тип nvarchar(4000) и значение по умолчанию NULL. Когда связанный сервер собственного клиента SQL Server, указать экземпляр можно с помощью ключевого слова SERVER (SERVER=servername\instancename, servername — это имя компьютера , а instancename — имя конкретного экземпляра SQL Server, к которому будет подключен пользователь.)

[ **@catalog=** ] **'**catalog**'** - каталог, который должен использоваться при подключении к поставщику OLE DB. Аргумент catalog имеет тип sysname и значение по умолчанию NULL. Значение catalog передается как свойство DBPROP\_INIT\_CATALOG для инициализации поставщика OLE DB. Если связанный сервер определен для экземпляра SQL Server, то каталог ссылается на базу данных по умолчанию, с которой сопоставлен связанный сервер.

Код возврата- 0 (успешное завершение) или 1 (неуспешное завершение)

**Примечание**. Кроме того можно использовать дополнительные системные процедуры

sp-addlinkedserverlogin для создание учетных записей управления доступом

sp-helpaddlinkedserverlogin для просмотра учетных записей управления доступом

sp-droplinkedserverlogin для удаления учетных записей управления доступом

sp-helpserver для просмотра информации о связанных серверах

В следующей таблице показаны способы настройки связанного сервера для источников данных, доступных через поставщик OLE DB. Связанный сервер может быть настроен несколькими способами для конкретного источника данных; для одного типа источника данных возможны несколько строк. Также в таблице показаны значения аргументов процедуры **sp\_addlinkedserver**, используемые для настройки связанного сервера.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Источник OLE DB** | **Поставщик OLE DB** | **product\_name** | **provider\_name** | **data\_source** | **location** | **provider\_string** | **catalog** |
| SQL Server | OLE DB для собственного клиента MS SQL Server | SQL Server1 (по умолчанию) |  |  |  |  |  |
| SQL Server | OLE DB для собственного клиента MSSQL Server |  | **SQLNCLI** | Сетевое имя SQL Server |  |  | Имя базы данных (**необязательно)** |
| SQL Server | OLE DB для собственного клиента MSSQL Server |  | **SQLNCLI** | servername\instancename |  |  | Имя базы данных (необязательно) |
| Oracle, версия 8 или более | Поставщик Oracle для OLE DB | Любой | **OraOLEDB.Oracle** | Псевдоним для базы данных Oracle |  |  |  |
| Access/Jet | OLE DB для Jet (Майкрософт) | Любой | **Microsoft.Jet.OLEDB.4.0** | Полный путь к файлу базы |  |  |  |
| Источник ODBC | Microsoft OLE DB для ODBC | Любой | **MSDASQL** | Системный DSN источника ODBC |  |  |  |
| Источник данных ODBC | Microsoft OLE DB для ODBC | Любой | **MSDASQL** |  |  | Строка подключения ODBC |  |
| Файловая система | Microsoft OLE DB для службы индексирования | Любой | **MSIDXS** | Имя каталога службы индексирования |  |  |  |
| таблица Microsoft Excel | Поставщик Microsoft OLE DB для Jet | Любой | **Microsoft.Jet.OLEDB.4.0** | Полный путь к файлу Excel |  | Excel 5.0 |  |
| База данных IBM DB2 | Поставщик Microsoft OLE DB для DB2 | Любой | **DB2OLEDB** |  |  | Документацию по DB2 см. у поставщика Microsoft OLE DB. | Имя каталога базы данных DB2 |

1 При таком способе настройки связанного сервера имя связанного сервера совпадает с сетевым именем удаленного экземпляра SQL Server. Используйте аргумент data\_source, чтобы указать сервер.

2 «Любой» указывает, что название продукта может быть любым.

Поставщик OLE DB для собственного клиента MicrosoftSQL Server используется вместе с SQL Server в случае, если имя поставщика не указано или SQL Server определен как название продукта. Даже если указано имя предыдущего поставщика (SQLOLEDB), оно все равно будет изменено на SQLNCLI при сохранении в каталог.

Аргументы data\_source, location, provider\_string и catalog идентифицируют базу данных или базы данных, на которые указывает связанный сервер. Если значение одного из этих аргументов равно NULL, то соответствующее свойство инициализации поставщика OLE DB не установлено.

В кластеризованной среде при указании имен файлов для указания источников данных OLE DB используйте формат UNC или общие диски для указания расположения.

Хранимая процедура **sp\_addlinkedserver** не может быть выполнена внутри пользовательской транзакции.

|  |
| --- |
| **Примечание по безопасностиПримечание по безопасности** |
| При создании связанного сервера с помощью процедуры **sp\_addlinkedserver** для всех локальных имен входа по умолчанию добавляется сопоставление самим себе. Поставщики, отличные от SQL Server, для которых выполнена проверка подлинности SQL Server, могут получить доступ к поставщику под учетной записью службы SQL Server. Администраторам нужно рассмотреть применение процедуры sp\_droplinkedsrvlogin <linkedserver\_name>, NULL для удаления глобального сопоставления. |

Необходимо разрешение ALTER ANY LINKED SERVER.

**Использование поставщика OLE DB для собственного клиента Microsoft SQL Server**

В следующем примере показано создание связанного сервера с именем SEATTLESales. Название продукта — SQL Server, имя поставщика не используется.

USE master;

GO

EXEC sp\_addlinkedserver

N'SEATTLESales',

N'SQL Server';

GO

В этом примере показано, как создать связанный сервер S1\_instance1 на экземпляре SQL Server с помощью поставщика OLE DB для собственного клиента SQL Server.

EXEC sp\_addlinkedserver

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

@datasrc=N'S1\instance1';

**Использование поставщика Microsoft OLE DB для Microsoft Access**

Поставщик Microsoft.Jet.OLEDB.4.0 соединяется с базами данных Microsoft Access в формате 2002–2003. В следующем примере показано создание связанного сервера с именем SEATTLE Mktg.

|  |
| --- |
| **ПримечаниеПримечание** |
| В этом примере предполагается, что установлена база данных Microsoft Access и образец базы данных **Northwind**, а база данных **Northwind** находится в каталоге «C:\Msoffice\Access\Samples». |

EXEC sp\_addlinkedserver @server = 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

Поставщик Microsoft.ACE.OLEDB.12.0 соединяется с базами данных Microsoft Access в формате 2007. В следующем примере показано создание связанного сервера с именем SEATTLE Mktg.

|  |
| --- |
| **ПримечаниеПримечание** |
| В этом примере предполагается, что установлена база данных Microsoft Access и образец базы данных **Борей**, а база данных **Борей** находится в каталоге «C:\Msoffice\Access\Samples». |

[Копировать](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_da916b23-22d5-4874-a8cd-aefb89b8c08e');)

EXEC sp\_addlinkedserver @server = 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

**В.Использование поставщика Microsoft OLE DB для ODBC с аргументом источника данных**

В данном примере показано, как создать связанный сервер с именем SEATTLE Payroll, который использует поставщик Microsoft OLE DB для ODBC (MSDASQL) и параметр data\_source.

**ПримечаниеПримечание** Указанный источник данных ODBC должен быть определен как системный DSN на сервере до того, как будет использоваться связанный сервер.

EXEC sp\_addlinkedserver

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

@datasrc = N'LocalServer';

GO

**2. Распределенные транзакции**

Распределенные транзакции, запускаемые в Transact-SQL, имеют относительно простую схему выполнения:

1. Приложение или сценарий Transact-SQL выполняет инструкцию Transact-SQL, которая инициирует распределенную транзакцию.
2. Экземпляр SQL Server Database Engine, выполняющий инструкцию, становится управляющим сервером в транзакции.
3. Сценарий или приложение затем выполняет либо распределенные запросы к связанным серверам, либо удаленные хранимые процедуры на удаленных серверах.
4. После того как распределенные запросы и вызовы удаленных процедур выполнены, управляющий сервер автоматически вызывает координатор распределенных транзакций (Майкрософт) (MS DTC), чтобы прикрепить в распределенную транзакцию связанные и удаленные серверы.
5. Когда сценарий или приложение выполняет инструкцию COMMIT или ROLLBACK, управляющий экземпляр SQL Server вызывает MS DTC, который управляет процессом двухфазной фиксации или уведомляет связанные и удаленные серверы, чтобы они выполнили откат своих транзакций.

[Необходимые инструкции Transact-SQL](javascript:void(0))

. Ниже представлены инструкции Transact-SQL, которые необходимо выполнить в сценарии или приложении Transact-SQL:

1. Запуск распределенной транзакции.

Распределенную транзакцию в Transact-SQL можно запустить следующими способами:

* Явно начать распределенную транзакцию, используя инструкцию BEGIN DISTRIBUTED TRANSACTION.
* Находясь в локальной транзакции, выполнить распределенный запрос.
* Если была выполнена инструкция SET REMOTE\_PROC\_TRANSACTIONS ON и локальная транзакция вызывает удаленную хранимую процедуру на другом экземпляре компонента Database Engine, локальная транзакция становится распределенной. Параметр REMOTE\_PROC\_TRANSACTIONS — это параметр совместимости, который относится только к вызовам удаленных хранимых процедур на удаленных серверах, определенных процедурой **sp\_addserver**. Параметр не применяется к распределенным запросам, которые выполняют хранимые процедуры на связанных серверах, определенных при использовании процедуры **sp\_addlinkedserver**.

1. Выполнение распределенных запросов к связанным серверам или выполнение удаленного вызова процедур на удаленных серверах.
2. Вызов стандартных инструкций Transact-SQL COMMIT TRANSACTION, COMMIT WORK, ROLLBACK TRANSACTION или ROLLBACK WORK для завершения транзакции.

Для любой распределенной транзакции Transact-SQL экземпляр Database Engine, обрабатывающий соединение или сценарий Transact-SQL, автоматически вызывает MS DTC для координации ее фиксации или отката.

1. **Примеры**

**Пример распределенного запроса**

Пусть имеется связанный сервер SEATLESALES в котором обслуживается база данных AdventureWorks2008R2 в которой созданы две схемы

схема Production с таблицей Product и схема Sales с таблицей SalesOrderDetail

Тогда распределенный запрос может иметь вид:

**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

**Пример распределенной транзакции удаления**

В примере удаляется кандидат из базы данных База данных AdventureWorks2008R2 как на локальном экземпляре компонента Database Engine, так и на удаленном сервере. И локальная, и удаленная база данных зафиксирует транзакцию или выполнит ее откат.

**ПримечаниеПримечание.** Этот пример приводит к сообщению об ошибке, если координатор MS DTC не установлен на компьютере, где выполняется экземпляр 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

**Пример распределенной транзакции на обновление**

**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

1. **Мониторинг выполнения распределенной транзакции**

Работу координатора распределенных транзакций можно просмотреть средствами операционной системы.

Цепочка действий Start->Control Panel->Administrative Tools-> Component services->Distributed Transaction Controller

Выводит дерево функций с возможностью просмотра журнала распределенных транзакций и соответствующей статистики

