**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**ТЕФ**

**Кафедра автоматизації проектування  
енергетичних процесів і систем**

**Лабораторна робота №4**

на тему:

“Мережа автосалонів”

Виконали:

студенти 3 курсу ТЕФ

Групи ТВ-61

Гарник О. І., Артамонов О. Ю.

Перевірив:

Варава І. A.

м. Київ – 2019 р.

Проектування розподілених запитів

Після того як підключення до віддаленого джерела даних встановлено, SQL Server може звертатися до зовнішніх даних за допомогою запитів. Представлені чотири основних доступних синтаксичних методу, які відрізняються методами обробки і Розподілені запити і Management Studio

Утиліта Management Studio не підтримує графічні методи ініціалізації розподілених запитів. Не існує методів перетягування підключеного сервера або видаленої таблиці в конструктор запитів. Однак розподілений запит можна ввести вручну на панелі SQL а потім виконати його.

розподілені уявлення

Уявлення - це збережені інструкції SQL. Хоча я не рекомендую засновувати додатки клієнт / сервер на уявленнях, вони можуть виявитися корисними для поточних запитів. Так як більшість користувачів (в тому числі і розробники) не знайома з різними методами виконання розподілених запитів, приміщення такого в уявлення може виявитися вдалим рішенням проблеми.

Локальні розподілені запити

Хоча термін локальний розподілений запит звучить дивно, все не так вже й складно. Це запит, який збирає зовнішні дані в SQL Server, а потім виконує запит на локальному сервері. Так як обробка таких запитів виконується на локальному сервері, в них використовується синтаксис Т-SQL, і тому їх іноді називають локальними запитами T-SQL.

Використання чотирьохкомпонентного імені

Якщо дані знаходяться на іншому екземплярі SQL Server, то повний синтаксис чотирьохкомпонентного імені наступний:

Сервер. База даних. Схема. Ім’я\_об'єкта

Чотирьохкомпонентна ім'я може використовуватися в будь-яких запитах вилучення або модифікації даних. На моєму комп'ютері існує другий примірник SQL Server. Ім'я власника об'єкта є обов'язковим, якщо звернення здійснюється до зовнішнього сервера.

При виконанні інструкцій INSERT, UPDATE і DELETE як розподілених запитів для імені таблиці можна використовувати або чотирьохкомпонентну форму, яку функцію розподіленого запиту. У цьому прикладі в якості джерела даних для інструкції INSERT використано Чотирьохкомпонентна ім'я таблиці. Цей запит витягує назви базових таборів з електронної таблиці Excel і вставляє їх в SQL Server

SELECT \* FROM [DESKTOP-IJL2GO6\MSSQLSERVER2].[User].dbo.Student UNION SELECT Student.id, Student.name, Student.surname, Student.age FROM [User].dbo.Student;

Використання функції OpenDataSource ()

Використання функції OpenDataSource () нічим не відрізняється від використання чотирьохкомпонентного імені для доступу до пов'язаного сервера, за винятком того, що функція визначає підключення в своєму тілі, а не використовує зумовлений пов'язаний сервер. У той час як визначення зв'язку в програмі пропускає визначення самого сервера, якщо місцезнаходження зв'язку змінюється, це торкнеться всіх запитів користувачів послуги OpenDataSource (). До того ж функція OpenDataSource () не приймає змінні в якості аргументів.

Функція OpenDataSource () замінює ім'я сервера в Чотирьохкомпонентна імені і може використовуватися в будь-якій інструкції DML.

Синтаксис функції OpenDataSource () виглядає досить простим:

OpenDataSource {провайдер, строка\_ініціалізаціі)

Однак перше враження оманливе. Строка\_ініціалізаціі є символьний рядок з декількома параметрами, розділеними крапками з комами (точний список параметрів залежить від конкретного джерела даних). Потенційно в рядку ініціалізації вказується джерело даних, його місце розташування, додаткові параметри, час очікування підключення, ідентифікатор користувача, його пароль і каталог. У рядку ініціалізації повинні бути вказані всі необхідні параметри підключення до джерела даних, в тому числі контекст безпеки. Окремі параметри в рядку ініціалізації не потрібно брати в лапки. Найпоширенішою помилкою, поміченою при реалізації функції OpenDataSource (), є плутанина між запитом і крапкою з комою.

Якщо функція OpenDataSource () підключається до іншого сервера за допомогою Windows, то необхідна аутентифікація з підтримкою Kerberos.

Розподілений запит SQL Server для доступу до механізму Jet, який відкриває робочий лист Excel, буде використовувати постачальника OLE DB. Функції OpenDataSource () ми передаємо тільки ім'я сервера в чотирикомпонентної формі; при цьому, як і в випадку з Access, ім'я бази даних і власника опускаємо.

План виконання розподіленого запиту UPDATE. Він починається праворуч з блоку Remote Scan, який повертає всі рядки іменованого діапазону Excel. Після цього дані обробляються в SQL Server. Логічна операція Remote Update зводиться до того, що розподілений запит поновлення насправді змінює значення всього в двох рядках.

SELECT \* FROM OPENDATASOURCE('SQLOLEDB',

'Data Source=DESKTOP-IJL2GO6\MSSQLSERVER2; User ID=con01; Password=1111').[User].dbo.Student UNION SELECT Student.id, Student.name, Student.surname, Student.age FROM [User].dbo.Student;

Наскрізні розподілені запити

Наскрізні запити виконуються на зовнішньому джерелі даних, після чого результат передається на SQL Server. Основна причина використання таких запитів полягає в скороченні обсягу даних, переданих з сервера (зовнішнього джерела даних) і клієнта (SQL Server). Замість передачі мільйона рядків на SQL Server, щоб використовувати тільки двадцять п'ять з них, може виявитися вигідніше отримати тільки ці рядки із зовнішнього джерела даних.

У той же час наскрізні запити повинні використовувати синтаксис зовнішнього сервера. Якщо джерелом даних є база даних Oracle, то в наскрізному запиті повинен використовуватися мова PL / SQL; якщо база даних Access - то Access SQL.

У разі наскрізного запиту, що модифікує дані, сам тип віддалених даних визначає, чи буде оновлення виконуватися локально або віддалено.

Якщо оновлюються дані на іншому екземплярі SQL Server, то операція буде виконуватися саме на ньому.

Якщо дані оновлюються на сервері, відмінному від SQL Server, то постачальник визначає, де саме будуть оброблятися дані. Часто передаються запити просто відбирають коректні рядка. Ці рядки передаються на SQL Server, оновлюються на ньому, а потім повертаються віддаленого джерела даних для оновлення.

Існують дві форми локальних розподілених запитів: одна для пов'язаних серверів і одна для зовнішніх джерел даних, які визначаються в запиті. Також існують дві форми наскрізних розподілених запитів. В одному випадку функція OpenQuery () використовує вже підключений сервер; у другому - функція OpenRowSet () визначає зв'язок безпосередньо в запиті.

Використання четирьохкомпонентного імені

Якщо розподілений запит здійснює доступ до іншого екземпляра SQL Server, Чотирьохкомпонентна ім'я називають гібридним методом розподілених запитів. Залежно від пропозицій FROM і WHERE, SQL Server буде намагатися передати якомога більшу частину запиту віддаленого сервера, щоб збільшити продуктивність.

При створенні складних розподілених запитів, що використовують Чотирьохкомпонентна ім'я, досить складно передбачити заздалегідь, яка частина запиту буде передана. Нам зустрічалися сервери, які, в залежності від пропозиції WHERE, або передавали весь запит, або кожна таблиця передавала свій запит, або запит передавала тільки одна таблиця.

З чотирьох представлених методів розподілених запитів кращими є використання четирьохкомпонентного імені та функції OpenQuery () відповідно. Обидва методи використовують адміністративну настройку з'єднання з віддаленим сервером, що робить запит більш надійним, якщо конфігурація сервера змінюється.

Вибір між цими двома методами залежить від загального обсягу даних, їх витягає кількості і продуктивності сервера. Рекомендую вам протестувати обидва методи і порівняти їх плани виконання, після чого вибрати той, який краще проявляє себе у вашій ситуації. Якщо обидва методи показують рівну продуктивність, то використовуйте метод четирьохкомпонентного імені, щоб сервер мав можливість автоматично оптимізувати розподілений запит.

Функція OpenQuery ()

Серед переданих запитів функція OpenQuery () задіє ресурси вже підключеного сервера, тому з нею простіше звертатися. Вона також сприймає зміни в конфігурації серверів без зміни коду.

Рухаючись запит, який використовує функцію OpenQuery (), практично не вимагає обробки сервером - йому повертаються рівно два рядки. Пропозиція WHERE обробляється механізмом Jet при отриманні даних з робочого аркуша Excel.

У наступному випадку функція OpenQuery () дає вказівку механізму Jet, щоб той витягнув тільки два рядки, що вимагають оновлення. Реальна інструкція UPDATE виконується на сервері, а результат повертається зовнішнього джерела даних. В результаті переданий запит виконує в інструкції UPDATE тільки частина функції SELECT.

Ця функція є двійником функції OpenDataSet (). Обидві вимагають, щоб в розподіленому запиті був повністю визначений віддалений джерело даних.

SELECT \* FROM OPENQUERY([DESKTOP-IJL2GO6\MSSQLSERVER2], 'SELECT \* FROM [User].dbo.Student') UNION SELECT Student.id, Student.name, Student.surname, Student.age FROM [User].dbo.Student;

Функція OpenRowSet () має додатковий аргумент, який визначає переданий запит. Розподілений запит, який використовує функцію OpenQuery (), повертає тільки два рядки, відібрані пропозицією WHERE

Щоб виконати оновлення за допомогою функції OpenRowSet (), вставте її на місце модифікується таблиці. Пропозиція WHERE інструкції UPDATE обробляється переданої частою функції OpenRowSet ()

SELECT \* FROM OPENROWSET('SQLOLEDB',

'Server=DESKTOP-IJL2GO6\MSSQLSERVER2;UID=con01;PWD=1111;', 'SELECT \* FROM [User].dbo.Student') UNION SELECT Student.id, Student.name, Student.surname, Student.age FROM [User].dbo.Student;