Paweł Rajba
pawel@cs.uni.wroc.pl
http://pawel.ii.uni.wroc.pl/

SQL Server Wyzwalacze

Agenda

- Wyzwalacze
  - Wprowadzenie
  - Tworzenie wyzwalacza
  - Wyzwalacze typu ,,po"
  - Wyzwalacze typu ,,zamiast"
  - Zarządzanie wyzwalaczami
  - Zależności referencyjne
  - Jeszcze raz rekurencja

Wprowadzenie

- Po co stosujemy wyzwalacze
  - obsługa bardziej złożonych zależności referencyjnych (np. klauzule check nie wystarczają)
  - bardziej złożone usuwanie kaskadowe
  - śledzenie zmian, rejestrowanie zachodzących akcji
  - wywoływanie akcji wewnętrznych np. wysłanie maila do osoby odpowiedzialnej za zamówienia w przypadku niskiego stanu towaru w magazynie

Wprowadzenie

- SQLServer ma dwa rodzaje wyzwalaczy
  - wyzwalacze typu ,,po"
  - wyzwalacze typu ,,zamiast"
- Wyzwalacze mogą być uruchamiane w związku z pojawieniem się każdej z poniższych instrukcji
  - INSERT
  - UPDATE
  - DELETE

Tworzenie wyzwalacza

- Przykładowa składnia
  - CREATE TRIGGER trigger_name
    ON { table | view }
    { { FOR | AFTER | INSTEAD OF }
      { [INSERT] [,] UPDATE] [,] DELETE] }
    AS BEGIN
       treść wyzwalacza
    END
- Włączanie i wyłączanie wyzwalaczy
  - ALTER TABLE nazwa { ENABLE | DISABLE } TRIGGER { ALL | trigger_name [,...n] }

Wyzwalacze typu „po”

- Są to domyślnie tworzone wyzwalacze
- Poprzez zmienną @@rowcount mamy informację o liczbie zmodyfikowanych wierszy
- Dostęp do obrazu tabeli przed i po wykonaniu instrukcji jest poprzez wirtualne tabele
  - inserted
  - deleted
- Powyższych tabeli nie można modyfikować

Wyzwalacze typu „po”

- Można utworzyć kilka wyzwalaczy dla jednej akcji, np. po update'cie na tabeli *osoba* mogą się wykonywać dwa wyzwalacze
- Do ustalenia kolejności odpalania wyzwalaczy jest procedura sp_settriggerorder
  - sp_settriggerorder[@triggername =] 'triggername'
    , [@order =] 'value'
    , [@stmttype =] 'statement_type'

Wyzwalacze typu „po”

- Uwagi do opcji procedury sp_settriggerorder
  - Zmienna @order może przyjmować wartości
    - First – będzie odpalany jako pierwszy
    - Last – będzie odpalany jako ostatni
    - None – będzie odpalany w nieokreślonej kolejności (przypadkowej)
- Dla tabeli oraz dla akcji (insert, update, delete) można tylko raz ustawić, który wyzwalacz jest pierwszy, a który ostatni
  - jeśli chcemy zmienić, to najpierw ten, który jest pierwszy lub ostatni trzeba ustawić na NONE

Wyzwalacze typu „po”

- Uwagi do opcji procedury sp_settriggerorder
  - Zmienna @stmttype może przyjmować wartości
    - INSERT, UPDATE, DELETE
  - Wartość zmiennej @stmttype musi być zgodna z akcją, po której wywoływany jest wyzwalacz

Wyzwalacze typu „po”

- Wycofywanie wyzwalaczy
  - Do przerwania wykonywania wyzwalacza służy polecenie ROLLBACK
  - Istotne uwagi
    - Wykonanie ROLLBACK przerywa wykonanie całego wsadu, w którym wyzwalacz został uruchomiony
    - W szczególności instrukcja, która uruchomiła wyzwalacz zostanie wycofana

Wyzwalacze typu „po”

- Kilka uwag
  - zarówno instrukcje print jak select mogą być używane w treści wyzwalacza, jednak nie jest to zalecane
  - powiedzmy, że mamy wyzwalacz dla insert w tabeli osoba, który również wykonuje insert do tej samej tabeli; to, czy zostanie wykonany nowy wyzwalacz zależy do ustawienia opcji recursive triggers
    - jest to opcja serwera
    - obowiązuje maksymalny stopień zagnieżdżenia, który wynosi 32

Wyzwalacze typu „po”

- Kilka uwag
  - druga sytuacja jest taka, że jeden wyzwalacz modyfikuje dane w innej tabeli; to, czy zostanie uruchomiany następny wyzwalacz zależy od opcji nested triggers
    - to jest opcja bazy danych
    - tutaj również obowiązuje maksymalny stopień zagnieżdżenia, który wynosi 32
  - Obie opcje ustawiamy następująco
    - sp_configure 'nested triggers', { 0 | 1 }
    - ALTER DATABASE nazwa
      SET RECURSIVE_TRIGGERS ON

DEMO

- 01-wyzwalacze-po

Wyzwalacze typu „zamiast”

- Są to wyzwalacze, które będą wykonane zamiast modyfikacji danych
- Dla każdej akcji może istnieć tylko jeden wyzwalacz typu ,,zamiast"
- Niezależnie od ustawień, wyzwalacze te nie są rekurencyjne
- Nie można utworzyć wyzwalacza ,,zamiast" dla
  - akcji delete, jeśli w tabeli jest klucz obcy z opcją on delete cascade
  - akcji update, jeśli w tabeli jest klucz obcy z opcją on update cascade

Wyzwalacze typu „zamiast”

- Modyfikacje widoków
  - jeśli widok jest oparty na więcej niż jednej tabeli, nie można wykonywać na nim operacji modyfikujących
    - jednak można utworzyć wyzwalacz typu ,,zamiast" i wtedy te operacje stają się dostępne
  - szczególnie przydatne dla widoków opartych na złączeniach (JOIN) i sumach (UNION)

DEMO

- 02-wyzwalacze-zamiast

Zarządzanie wyzwalaczami

- Aby zobaczyć treść wyzwalacza można skorzystać z polecenia
  - sp_helptext nazwa_wyzwalacza
- Podobnie jak w przypadku procedur, można w celu ukrycia szyfrować treść wyzwalacza
  - służy do tego opcja WITH ENCRYPTION
- Zmienić definicję wyzwalacza można przez procedurę ALTER TRIGGER
- Dodatkowe informacje można też uzyskać poprzez procedurę sp_helptrigger

Zależności referencyjne

- Zależności referencyjne można utrzymywać
  - poprzez zdefiniowanie ograniczenia FOREIGN KEY
  - poprzez zdefiniowanie odpowiednich wyzwalaczy
    - wtedy nie definiujemy FOREIGN KEY
    - definiujemy FOREIGN KEY, ale wyłączamy (zalecane)

DEMO

- 03-kaskada
- 04-praktycznie