

Paweł Rajba

pawel@cs.uni.wroc.pl

<http://www.itcourses.eu/>

SQL Server

Kursory

Agenda

- Wprowadzenie
- Używanie kursorów
- Rodzaje kursorów
- Praca z kursorami

Wprowadzenie

- Co to jest kursor?
- Cykl życia kursora
 - utworzenie
 - pobieranie wartości
 - ewentualnie update lub delete
 - zamknięcie
 - zwolnienie zasobów

Wprowadzenie

- Rodzaje kursorów w SQL Server
 - Transact-SQL cursors
 - Używane skryptach, procedurach, wyzwalaczach
 - Implementowane na serwerze
 - Zarządzane przez T-SQL wysyłany z klienta do serwera
 - API server cursors
 - Wsparcie dla kursorów w OLE DB i ODBC
 - Zestaw procedur, implementacja na serwerze
 - Sterownik wysyła żądania do serwera do API kursora
 - Client cursors
 - Implementacja w sterownikach native client ODBC, ADO DLL
 - Dane buforowane na kliencie
 - Każde żądanie operacji na kursorze odbywa się na zbuforowanych danych

Wprowadzenie

- Podstawowe operacje
 - utworzenie
 - DECLARE nazwa CURSOR FOR zapytanie
 - OPEN nazwa
 - pobieranie wartości
 - FETCH NEXT FROM nazwa
 - zamykanie
 - CLOSE nazwa
 - zwolnienie zasobów
 - DEALLOCATE nazwa

DEMO

- 01-podstawowe-uzycie
- 02-uproszczone-uzycie

Zastosowanie kursorów

- Kiedy używamy kursorów?
 - Najbardziej właściwa odpowiedź brzmi:
jeśli już koniecznie musimy
 - Jeśli wystarczy, korzystamy z odpowiednich zapytań i złączeń

Zastosowanie kursorów

- Przykłady
 - mamy tabelkę
 - Pracownik(pesel, imie, nazwisko, adres, pensja)
i dla pewnej grupy pracowników B2B chcemy uruchomić procedurę oblicz podatek i ZUS
 - chcemy zrobić kopie zapasową wszystkich baz na serwerze
 - <https://www.mssqltips.com/sqlservertip/1599/sql-server-cursor-example/>

Praca z kursorami

- Zasięg kursorów
 - kursory mogą być globalne lub lokalne
 - globalne są widoczne dla wszystkich wsadów
 - lokalne tylko dla bieżącego wsadu
 - domyślnie kursory są globalne
 - można to zmienić poprzez polecenie `ALTER DATABASE alter database nazwa_bazy set cursor_default local`

Rodzaje kursorów

- Kursory statyczne
 - dla danych wynikowych kursora tworzona jest tymczasowa tabela w bazie tempdb i na tej tabeli wykonywane są operacje
- Kursory kluczowe
 - tylko wartości kluczy są kopiowane do tymczasowej tabeli w tempdb
- Kursory dynamiczne
 - pracujemy na żywej tabeli
 - nie działa fetch absolute, ale działa fetch relative

Rodzaje kursorów

- Kursory jednokierunkowe
 - Każdy z głównych typów może być jednokierunkowy
 - Przetwarzanie takiego kursora polega zwykle na przeglądzie rekordów od początku do końca
 - instrukcja SELECT jest nadal szybsza od takiego kursora
 - Nawet jeśli zastosujemy opcję FAST_FORWARD

Deklaracja kursora

- Mamy do dyspozycji dwie składnie:
 - ISO
 - DECLARE cursor_name [INSENSITIVE] [SCROLL] CURSOR
FOR select_statement
[FOR { READ ONLY | UPDATE [OF column_name [,...n]] }]
 - Rozszerzona
 - DECLARE cursor_name CURSOR
[LOCAL | GLOBAL]
[FORWARD_ONLY | SCROLL]
[STATIC | KEYSET | DYNAMIC | FAST_FORWARD]
[READ_ONLY | SCROLL_LOCKS | OPTIMISTIC]
[TYPE_WARNING]
FOR select_statement
[FOR UPDATE [OF column_name [,...n]]]

Deklaracja kursora

- Dalej będziemy się zajmować składnią rozszerzoną

Praca z kursorami

- Otwieranie kursora
 - służy do tego polecenie OPEN
 - open nazwa_kursora
 - jego zadaniem jest udostępnienie danych i w razie potrzeby wypełnienie tabel tymczasowych
- Po wykonaniu OPEN można wykonać funkcję @@CURSOR_ROWS, która zwykle zwraca
 - -1 – kursor dynamiczny, nie można użyć tej funkcji
 - n – liczba wierszy w kursorze
 - 0 – kursor nie jest otwarty lub nic nie zawiera

Praca z kursorami

- Pobieranie wiersza z kursora
 - służy do tego polecenie FETCH
 - FETCH selektor FROM nazwa [INTO @z1, @z2, ...]
 - dostępne selektory: NEXT, PRIOR, FIRST, LAST, ABSOLUTE n, RELATIVE n
 - w przypadku kursora dynamicznego nie jest dostępne ABSOLUTE n
 - w ABSOLUTE i RELATIVE n może być ujemne
 - w ABSOLUTE liczymy od końca
 - w RELATIVE liczymy wstecz od bieżącej pozycji
 - RELATIVE 0 – oznacza pobranie bieżącego wiersza

Praca z kursorami

- Pobieranie wiersza z kursora
 - Znaczenie opcji FETCH zaraz po otwarciu kursora
 - FETCH NEXT – to samo co FETCH FIRST
 - FETCH PRIOR – nic nie zwróci
 - FETCH RELATIVE n
 - jeśli $n > 0$, to samo co FETCH ABSOLUTE n,
 - jeśli $n \leq 0$, nic nie zwróci
 - Zmienna @@FETCH_STATUS
 - jeśli 0, wiersz został pobrany
 - jeśli -1, przejście poza zakres
 - jeśli -2, brak wiersza w kursorze kluczowym

Praca z kursorami

- Pobieranie wiersza z kursora
 - Klauzula INTO
 - jeśli ją podamy, wartości z wiersza zostaną przekazane pod wskazane zmienne
 - jeśli jej nie podamy, dla każdego wiersza będą generowane osobne zestawy wynikowe
 - Ta druga opcja jest oczywiście bardzo nieefektywna
 - Liczba zmiennych, typy i rozmiary muszą się zgadzać z pobranym wynikiem, inaczej błąd

Praca z kursorami

- Modyfikacja danych kursora
 - Służy do tego polecenia UPDATE
 - UPDATE tabela SET przypisania
WHERE CURRENT OF nazwa_kursora
 - Tak naprawdę wykonywane są dwie operacje:
usunięcia i wstawienia wiersza
- Usunięcie wiersza z kursora
 - Służy do tego polecenie DELETE
 - DELETE FROM tabela WHERE CURRENT OF kursor

Praca z kursorami

- Zamykanie kursora
 - Służy do tego polecenie CLOSE
 - CLOSE nazwa_kursora
 - Zamknięcie kursora nie usuwa deklaracji – można bez problemu kursor ponownie otworzyć
 - Natomiast przy zamykaniu usuwaną są tabele tymczasowe, czyli przy ponownym otwarciu są odtwarzane

Praca z kursorami

- Niszczanie kursora
 - Służy do tego polecenie DEALLOCATE
 - DEALLOCATE nazwa_kursora
 - Po tym poleceniu konieczna jest ponowna deklaracja
 - Polecenia tego nie ma w standardzie, ale dzięki niemu możemy lepiej zarządzać zasobami

DEMO

- 03-rodzaje-kursorów
- 04-nawigacja
- 05-update-delete

Do przeczytania

- [https://msdn.microsoft.com/pl-pl/library/ms191179\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms191179(v=sql.110).aspx)
- [https://msdn.microsoft.com/pl-pl/library/ms181441\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms181441(v=sql.110).aspx)
- [https://msdn.microsoft.com/pl-pl/library/ms180169\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms180169(v=sql.110).aspx)