

## Temat 7 – indeksy, transakcje

### Indeksy

1. Aby przygotować tabelę z dużą ilością danych, uruchom skrypt:

```
CREATE TABLE test (Id INT IDENTITY, Zawartosc INT, Zawartosc2 INT)
GO

DECLARE @a INT
SET @a = 1
WHILE @a < 100000 BEGIN
    INSERT INTO test (Zawartosc, Zawartosc2)
    VALUES (CONVERT(INT,RAND() * 100000), CONVERT(INT,RAND() * 100000))
    SET @a = @a + 1
END
GO
```

2. Włącz podgląd planu wykonania zapytania i uruchom zapytanie typu:

```
SELECT * FROM test WHERE Zawartosc = 12345
```

Obejrzyj plan. Zaobserwujesz zapewne operację *Table scan*. Zapisz łączny koszt obserwując wartość *Estimated subtree cost* ostatniej (najbardziej lewej) operacji w planie.

3. Załóż indeks nie pogrupowany na kolumnę *Zawartosc*. Uruchom zapytanie jeszcze raz. Porównaj plan wykonania zapytania oraz koszt. Serwer powinien użyć założonego indeksu (*Index seek*), a koszt powinien być znacznie niższy.
4. Strategia „tylko indeks”. Usuń indeks założony w punkcie 4. Wykonaj zapytanie i obejrzyj plan jego wykonania:

```
SELECT Zawartosc, Zawartosc2 FROM test WHERE Zawartosc = 12345
```

Załącz indeks złożony na kolumny *Zawartosc* oraz *Zawartosc2* (koniecznie w takiej kolejności). Zaobserwuj różnicę w planie wykonania zapytania oraz łącznym koszcie (*Estimated subtree cost* ostatniej operacji). Zauważysz zapewne, że plan składa się jedynie z dwóch operacji. Pierwsza to *Index seek*, druga to *Select*. Wynika z tego, że serwer nie musiał odczytać stron z danymi.

5. Wyszukiwanie zakresowe. Usuń wszystkie indeksy z tabeli *test* i wykonaj z podglądem planu zapytanie:

```
SELECT * FROM test WHERE Zawartosc BETWEEN 10000 and 20000
```

Zaobserwujesz zapewne operację *Table scan*. Zapisz łączny koszt zapytania (*Estimated subtree cost* najbardziej lewej operacji). Załącz indeks nie pogrupowany na kolumnie *Zawartosc*. Wykonaj zapytanie jeszcze raz. Zapewne serwer nie skorzysta z indeksu (indeksy nie pogrupowane słabo wspierają wyszukiwanie zakresowe). Usuń indeks nie pogrupowany i załącz pogrupowany. Wykonaj zapytanie jeszcze raz. Obejrzyj plan i podejrzaj koszty.

6. Sortowanie przy pomocy indeksów. Usuń wszystkie indeksy z tabeli *test* i wykonaj z podglądem planu zapytanie:

```
SELECT * FROM test ORDER BY Zawartosc
```

Zauważysz zapewne dwie kosztowne operacje: *Table scan* i sortowanie. Załóż indeks nie pogrupowany na kolumnę *Zawartosc* i wykonaj zapytanie jeszcze raz. Porównaj plan i koszty. Czy serwer skorzystał z indeksu? Usuń indeks nie pogrupowany i załóż pogrupowany. Czy tym razem serwer skorzystał z indeksu?

## Transakcje

7. Wykonuj kolejno instrukcje:

```
SET IMPLICIT_TRANSACTIONS ON -- wyłączenie automatycznego zatwierdzania
CREATE TABLE Osoba (Id INT, Nazwisko VARCHAR(50))
COMMIT
INSERT INTO Osoba VALUES (1, 'Lenkiewicz')
INSERT INTO Osoba VALUES (2, 'Kowalski')
SELECT * FROM Osoba -- powinniśmy zobaczyć dwie osoby
ROLLBACK
SELECT * FROM Osoba -- powinniśmy zobaczyć pustą tabelę
INSERT INTO Osoba VALUES (3, 'Iksiński')
COMMIT
SELECT * FROM Osoba -- powinniśmy zobaczyć jedną osobę
```

8. Spróbuj wykonać zadanie 7 ponownie, ale z wyłączoną opcją:

```
SET IMPLICIT_TRANSACTIONS OFF
```

Przed wykonaniem kolejnego zadania ustaw tę opcję z powrotem na ON.

W trakcie wykonywania instrukcji COMMIT i ROLLBACK pojawiają się błędy mówiące o tym, że nie można zatwierdzić lub wycofać transakcji. Jest to normalne, ponieważ przy wyłączonej opcji IMPLICIT\_TRANSACTIONS, każda transakcja jest zatwierdzana automatycznie. Kolejne zadania wykonuj z włączoną tą opcją.

9. Otwórz drugą zakładkę w *Management Studio*. Wykonaj:

```
SET IMPLICIT_TRANSACTIONS ON
SELECT * FROM Osoba
```

W oknie 1 zaktualizuj rekord:

```
SET IMPLICIT_TRANSACTIONS ON
UPDATE Osoba SET Nazwisko = 'Igrekowski' WHERE Id = 3
```

W oknie 2 spróbuj wykonać jeszcze raz:

```
SELECT * FROM Osoba
```

Zapytanie będzie czekać na zwolnienie blokady. W oknie 1 wykonaj:

```
COMMIT
```

Zobacz, czy wykonało się zapytanie z okna 2.

UWAGA: Przy tym i kolejnych ćwiczeniach wykonywanych przy `IMPLICIT_TRANSACTIONS` ustawionym na `ON` mogą przeszkadzać nam poprzednie, niezakończone transakcje. Dlatego warto przed każdym ćwiczeniem wykonać `COMMIT` w każdym oknie.

10. W drugim oknie MS ustaw poziom izolacji na najniższy:

`SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED`

Wykonaj ponownie zadanie 9 (lub podobne). W pierwszym oknie wprowadzaj zmiany, a w drugim zobacz, czy jesteś w stanie odczytać niezatwierdzone dane.

UWAGA: Ustawienie poziomu izolacji działa dopiero dla kolejnej transakcji i nie dotyczy transakcji, która już się rozpoczęła. Dlatego po instrukcji `SET TRANSACTION ISOLATION LEVEL ...` warto dla pewności wykonać `COMMIT` lub `ROLLBACK`.

11. Sprawdź czy istnieje tabela *Test* i czy zawiera jakieś rekordy. Jeżeli nie, utwórz ją przy pomocy skryptu z punktu 1. Nie zapomnij wykonać `COMMIT` po zakończeniu. Następnie ustaw poziom izolacji `SERIALIZABLE` w obu zakładkach. Dla pewności wyłączamy też automatyczne zatwierdzanie oraz robimy `COMMIT` (może zwrócić błąd, jeśli nie było poprzedniej transakcji).

`SET TRANSACTION ISOLATION LEVEL SERIALIZABLE`  
`SET IMPLICIT_TRANSACTIONS ON`  
`COMMIT`

Następnie wykonaj w obu oknach:

`SELECT * FROM Test`

A następnie w pierwszym oknie dowolny `INSERT`:

`INSERT INTO Test (Zawartosc, Zawartosc2) VALUES (12345, 56789)`

Zauważ, że tym razem, już wykonanie instrukcji `SELECT` spowodowało założenie blokady na całą tabelę „Osoba” i nie jesteśmy w stanie wykonać `INSERT` do momentu, gdy w oknie 2 wykonamy `COMMIT`. W przypadku `READ COMMITTED` instrukcja `INSERT` wykonałaby się od razu (fantom).