

Zadanie 1

Porównanie planu wykonania zapytania:

```
SELECT SalesOrderID, SalesOrderDetailID
FROM Sales.SalesOrderDetail
WHERE SalesOrderID = 43683
AND SalesOrderDetailID = 240
```

Wyniki testów

	Z ISTNIEJĄCYM KLUCZEM	PO USUNIĘCIU KLUCZA	ZAMIANA AND NA OR
PHYSICAL OPERATION	Clustered Index Seek	Table Scan	Table Scan
LOGICAL OPERATION	Clustered Index Seek	Table Scan	Table Scan
ESTIMATE IO	0.003125	1.82905	1.82905
ESTIMATE CPU	0.0001581	0.133606	0.133606
COMPILE MEMORY [BYTES]	232	336	304
COMPILE CPU [MS]	3	6	5
EXECUTION TIME [MS]	3	7	5
ESTIMATED ROWS READ	1	121317	121317

Z istniejącym kluczem

W przypadku, gdy istnieje klucz główny, używane jest wyszukiwanie skupione (Clustered Index Seek), co oznacza efektywne wykorzystanie indeksu klucza głównego. Szacowany czas operacji I/O oraz CPU jest niski, a ilość wczytywanych wierszy to tylko 1, co świadczy o wysokiej selektywności zapytania.

Po usunięciu klucza

Po usunięciu klucza głównego, plan wykonania zmienia się na skanowanie tabeli (Table Scan), co oznacza konieczność przeskanowania całej tabeli. Szacowany czas operacji I/O i CPU wzrasta, tak jak czas kompilacji i wykonania oraz zużycie CPU i pamięci, co może być związane z bardziej złożonym planem wykonania. Ilość wczytywanych wierszy wzrasta do 121317, co świadczy o braku selektywności.

Różnice wynikają z faktu, iż klucz główny umożliwia efektywne wyszukiwanie jednego konkretnego wiersza, podczas gdy brak klucza głównego wymusza skanowanie całej tabeli.

Zamiana słowa AND na OR

Zmiana słowa kluczowego z AND na OR nie wpływa znacząco na rodzaj operacji – skrócony zostaje czas, wykonania i kompilacji or pamięć – ponieważ i tak zachodzi skanowanie całej tabeli. W tym przypadku zmiana ta nie ma istotnego wpływu na plan wykonania.

Podsumowanie

Podsumowując, obecność klucza głównego istotnie wpływa na efektywność wykonania zapytania, umożliwiając korzystanie z indeksów i skrócenie czasu operacji. Usunięcie klucza głównego sprawia, że konieczne jest skanowanie całej tabeli, co prowadzi do zwiększenia kosztów operacji I/O i CPU oraz czasu wykonania. Zmiana AND na OR nie ma istotnego wpływu w tym konkretnym przypadku.

Zadanie 2

```
SELECT * FROM Person.Address
```

```
WHERE ModifiedDate = '2011-08-01 00:00:00.000';
```

Wyniki testów

	PRZED DODANIEM INDEKSU	PO DODANIEM INDEKSU
PHYSICAL OPERATION	Clustered Index Scan	Nested Loops
LOGICAL OPERATION	Clustered Index Scan	Inner Join
ESTIMATE IO	0.255718	0
ESTIMATE CPU	0.0217324	0.00030514
COMPILE MEMORY [BYTES]	232	280
COMPILE CPU [MS]	64	7
EXECUTION TIME [MS]	73	7
ESTIMATED ROWS READ	19614	-

Plan przed dodaniem indeksu

Fizyczna operacja (PhysicalOp) to "Clustered Index Scan", co sugeruje skanowanie indeksu klastra. Całkowite koszty subtree (EstimatedTotalSubtreeCost) wynoszą 0.27745. Skanuje cały klastę, co jest zrozumiałe, gdy brak indeksu.

Plan po dodaniu indeksu

Fizyczna operacja to "Nested Loops" z operacją "Inner Join". Całkowite koszty subtree wynoszą 0.21837, co sugeruje mniejsze koszty niż wcześniej. Wprowadzenie indeksu umożliwiło optymalizację operacji, wykorzystując "Nested Loops" zamiast skanowania całego klastra.

Wnioski

Dodanie indeksu pozwoliło na zmniejszenie kosztów wykonania zapytania, co wskazuje na poprawę wydajności. Plan wykonania po dodaniu indeksu wykorzystuje operację "Nested Loops" zamiast skanowania indeksu klastra, co może oznaczać bardziej efektywne przeszukiwanie danych.