

BAZY DANYCH 2020

PROPONOWANE INDEKSOWANIE BAZY
LIBRARY

31 maja 2020

Elżbieta Jowik

Propozycja 1.

Koncepcja i motywacja

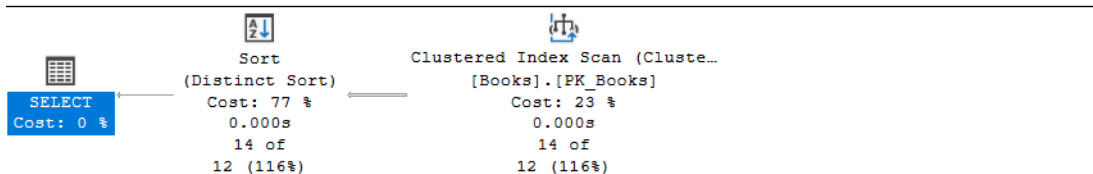
W pierwszym zaproponowanym scenariuszu, indeks został założony na tabelę agregującą informacje o księgozbiorze rozważanej biblioteki (tabela *Books*). Zastosowanie na niej indeksu ma sens, ponieważ jest to duża encja, której wymiarowość, jako funkcja czasu, ma tendencję rosnącą, a jednocześnie tabela ta nie jest poddawana częstym modyfikacjom. Koncepcja zastosowanego indeksowania, w którym kluczem indeksującym jest kolumna *Title*, wynika wprost z charakteru działalności biblioteki, ponieważ z punktu widzenia praktycznych zastosowań, porządkowanie książek i wyszukiwanie ich względem tytułów jest często wykonywaną operacją. Nałożony indeks nie został zdefiniowany jako unikalny, ponieważ założeniem towarzyszącym jest dopuszczenie możliwości zaistnienia w bazie kilku różnych książek o jednakowych tytułach. Wybór typu niezgrupowanego (nieklastrowanego) jest efektem tego, że zmienna tytułu nie jest polem klucza oraz wprost z powtarzalności wartości w kolumnie.

Plan wykonania

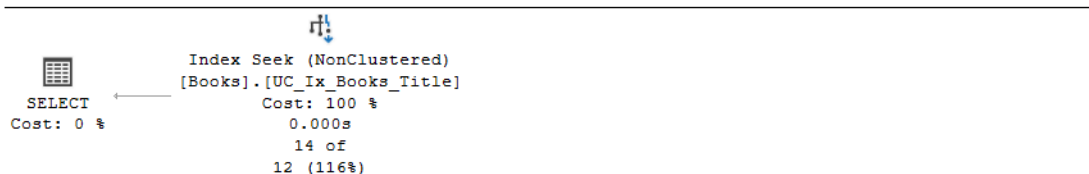
Poniższy plan wykonania został wygenerowany dla wywołań zapytania:

```
SELECT DISTINCT Title
FROM Books
WHERE Title LIKE 'The%'
ORDER BY Title;
```

Query 1: Query cost (relative to the batch): 34%
SELECT DISTINCT Title FROM Books WHERE Title LIKE 'The%' ORDER BY Title



Query 3: Query cost (relative to the batch): 8%
SELECT DISTINCT Title FROM Books WHERE Title LIKE 'The%' ORDER BY Title



Rysunek 1: Plany wykonania o najniższym koszcie jednakowego zapytania odpowiednio przed (*Query1*) i po (*Query2*) zastosowaniu indeksowania na tabeli *Books*.

Rezultat

Optymalizator zapytań zanim podda wykonaniu daną kwerendę, sprawdza jej możliwe realizacje i przekazuje do kompilacji tę, która jest najmniej obciążająca. Zatem już samo wykonanie planu, w którym uwzględniony został nowo utworzony indeks sugeruje, że podjęte działanie przyniosło oczekiwane rezultaty. Warto tutaj zwrócić uwagę, że względny koszt zapytania *Query cost*, który w wyniku zastosowanej modyfikacji spadł z 34 do 8 procent. Jak wynika z dokładniejszych statystyk (rysunek 2) różnica ta spowodowana jest uproszczeniem wyszukiwania obserwacji spełniających warunek umieszczony w klauzuli *WHERE* oraz ich porządkowania.

Sort		Clustered Index Scan (Clustered)		Index Seek (NonClustered)	
Sort the input.		Scanning a clustered index, entirely or only a range.		Scan a particular range of rows from a nonclustered index.	
Physical Operation	Sort	Physical Operation	Clustered Index Scan	Physical Operation	Index Seek
Logical Operation	Distinct Sort	Logical Operation	Clustered Index Scan	Logical Operation	Index Seek
Actual Execution Mode	Row	Actual Execution Mode	Row	Actual Execution Mode	Row
Estimated Execution Mode	Row	Estimated Execution Mode	Row	Estimated Execution Mode	Row
Actual Number of Rows for All Executions	14	Storage	RowStore	Storage	RowStore
Actual Number of Batches	0	Number of Rows Read	50	Number of Rows Read	14
Estimated Operator Cost	0,0114763 (77%)	Actual Number of Rows for All Executions	14	Actual Number of Rows for All Executions	14
Estimated I/O Cost	0,0112613	Actual Number of Batches	0	Actual Number of Batches	0
Estimated CPU Cost	0,0001711	Estimated I/O Cost	0,003125	Estimated Operator Cost	0,0032958 (100%)
Estimated Subtree Cost	0,0148133	Estimated Operator Cost	0,003337 (23%)	Estimated I/O Cost	0,003125
Number of Executions	1	Estimated CPU Cost	0,000212	Estimated Subtree Cost	0,0032958
Estimated Number of Executions	1	Estimated Subtree Cost	0,003337	Estimated CPU Cost	0,0001708
Estimated Number of Rows Per Execution	12,5	Number of Executions	1	Estimated Number of Executions	1
Estimated Row Size	30 B	Estimated Number of Executions	1	Number of Executions	1
Actual Rebinds	1	Estimated Number of Rows Per Execution	12,5	Estimated Number of Rows Per Execution	12,5
Actual Rewinds	0	Estimated Number of Rows to be Read	50	Estimated Number of Rows to be Read	12,5
Node ID	0	Estimated Row Size	30 B	Estimated Row Size	36 B
Output List		Actual Rebinds	0	Actual Rebinds	0
[Library].[dbo].[Books].Title		Actual Rewinds	0	Actual Rewinds	0
Order By		Ordered	False	Ordered	True
[Library].[dbo].[Books].Title Ascending		Node ID	1	Node ID	0
		Predicate		Predicate	
		[Library].[dbo].[Books].[Title] like 'The%'		[Library].[dbo].[Books].[Title] like 'The%'	
		Object		Object	
		[Library].[dbo].[Books].[PK_Books]		[Library].[dbo].[Books].[UC_Ix_Books_Title]	
		Output List		Output List	
		[Library].[dbo].[Books].Title		[Library].[dbo].[Books].Title	
				Seek Predicates	
				Seek Keys[1]: Start: [Library].[dbo].[Books].Title >= Scalar Operator ('The'); End: [Library].[dbo].[Books].Title < Scalar Operator('ThF')	

Rysunek 2: Tabele odpowiadające planom wykonania z rysunku 1

Polecenie zakładające powyżej zaproponowany indeks

```
-- #1. Indeks niezgrupowany
CREATE NONCLUSTERED INDEX UC_Ix_Books_Title ON [dbo].[Books]
(
    [Title] ASC
)
```

Propozycja 2.

Koncepcja i motywacja

Druga propozycja indeksowania polega na stworzeniu dwóch indeksów w tabeli przechowującej informacje dotyczące czytelników. Wymiarowość tej tabeli, poprzednio jak indeksowanej poprzednio, jest niemalejącą funkcją czasu, a modyfikacje nie są w niej dokonywane z dużą regularnością.

Koncepcja wynika wprost z charakteru działalności biblioteki, z punktu widzenia której numer karty czytelnika jest znacznie częściej wyszukiwanym polem niż jego jawny identyfikator. Pomysł polega utrzymaniu porządku rekordów w tabeli względem numeru karty bibliotecznej oraz umożliwieniu jego efektywnego wyszukiwania na podstawie podstawowych danych czytelnika, takich jak imię i nazwisko. W tym celu należy stworzyć dwa indeksy:

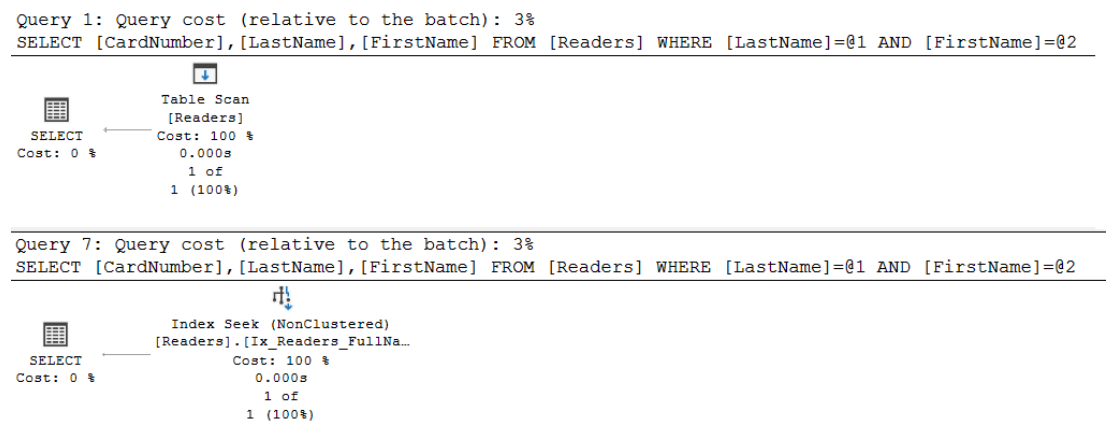
1. **zgrupowany**, dla którego kluczem indeksującym jest pole *CardNumber*,
2. **niezgrupowany**, którego klucz składa się z dwóch kolumn: *FirstName* i *LastName*

Pierwszy z rozważanych kluczy ma charakter unikalny, ponieważ założeniem towarzyszącym jest, że dany numer karty odpowiada dokładnie jednemu czytelnikowi niezależnie od jego aktywności, drugi natomiast nie jest opatrzony tym obostrzeniem, ponieważ koncepcyjnie dopuszczamy możliwość wystąpienia zbieżności imion i nazwisk.

Plan wykonania

Poniższy plan wykonania został wygenerowany dla wywołań zapytania:

```
SELECT CardNumber, LastName, FirstName
FROM Readers
WHERE LastName = 'Collins' AND FirstName = 'Adam'
```



Rysunek 3: Plany wykonania wygenerowane dla realizacji jednakowego zapytania przed (*Query1*) i po (*Query7*) zastawieniu indeksowania na tabeli *Readers*. Jest to fragment planu wykonania pełnej sekwencji kwerend, w której poza powyższymi, obecne były te, odpowiadające za tworzenie indeksów.

Rezultat

Dzięki zastosowaniu indeksowania, w tabeli czytelników porządek rekordów utrzymany jest względem numeru karty. Uwzględnienie w drugim planie wykonania sugeruje, że dzięki nowo utworzonym indeksom zoptymalizowane zostały polecenia, dla których klauzule *WHERE*, czy *ORDER BY* wykonywane są względem kolumn *FirstName*, *LastName*.

Table Scan		Index Seek (NonClustered)	
Scan rows from a table.		Scan a particular range of rows from a nonclustered index.	
Physical Operation	Table Scan	Physical Operation	Index Seek
Logical Operation	Table Scan	Logical Operation	Index Seek
Actual Execution Mode	Row	Actual Execution Mode	Row
Estimated Execution Mode	Row	Estimated Execution Mode	Row
Storage	RowStore	Storage	RowStore
Number of Rows Read	20	Number of Rows Read	1
Actual Number of Rows for All Executions	1	Actual Number of Rows for All Executions	1
Actual Number of Batches	0	Actual Number of Batches	0
Estimated I/O Cost	0,003125	Estimated I/O Cost	0,003125
Estimated Operator Cost	0,003304 (100%)	Estimated Operator Cost	0,0032831 (100%)
Estimated CPU Cost	0,000179	Estimated CPU Cost	0,0001581
Estimated Subtree Cost	0,003304	Estimated Subtree Cost	0,0032831
Number of Executions	1	Estimated Number of Executions	1
Estimated Number of Executions	1	Number of Executions	1
Estimated Number of Rows Per Execution	1	Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	20	Estimated Number of Rows to be Read	1
Estimated Row Size	27 B	Estimated Row Size	27 B
Actual Rebinds	0	Actual Rebinds	0
Actual Rewinds	0	Actual Rewinds	0
Ordered	False	Ordered	True
Node ID	0	Node ID	0
Predicate	[Library].[dbo].[Readers].[LastName]=[@1] AND [Library].[dbo].[Readers].[FirstName]=[@2]	Object	[Library].[dbo].[Readers].[Ix_Readers_FullName]
Object	[Library].[dbo].[Readers]	Output List	[Library].[dbo].[Readers].[CardNumber]; [Library].[dbo].[Readers].[FirstName]; [Library].[dbo].[Readers].[LastName]
Output List	[Library].[dbo].[Readers].[CardNumber]; [Library].[dbo].[Readers].[FirstName]; [Library].[dbo].[Readers].[LastName]	Seek Predicates	Seek Keys[1]: Prefix: [Library].[dbo].[Readers].[LastName]; [Library].[dbo].[Readers].[FirstName] = Scalar Operator([@1]); Scalar Operator([@2])

Rysunek 4: Tabele odpowiadające planom wykonania z rysunku 3

Polecenia zakładające powyżej zaproponowane indeksowanie

```
-- #1. Indeks zgrupowany
CREATE UNIQUE CLUSTERED INDEX Ix_Readers_CardNumber ON [dbo].[Readers]
(
    CardNumber
);

-- #2. Indeks niezgrupowany
CREATE NONCLUSTERED INDEX Ix_Readers_FullName ON [dbo].[Readers]
(
    LastName,
    FirstName
);
```